БПОУ ВО «Грязовецкий политехнический техникум»

Рассмотрен

цикловой комиссией по общепрофессиональным дисциплинам и профессиональным модулям отделения «Электрификация и автоматизация сельского хозяйства»

Согласовано

зам. директора по ОМР
_____ Е. А. Ткаченко
«30» августа 2017 г.

Протокол № 1 от «30» августа 2017 г.

Председатель комиссии:
_____ Т. В. Невзорова

ПРАКТИЧЕСКИЕ (ЛАБОРАТОРНЫЕ РАБОТЫ)

ОП.04. Операционные системы

Специальность 09.02.02. Компьютерные сети

Грязовец
2017 г.

Лабораторная работа №1

Тема: MS-DOS.

Цель работы: Приобретение основных навыков работы в среде операционной системы MS-DOS.

1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. Назначение операционной системы

MS-DOS – это аббревиатура от **M**icro**S**oft **D**isk **O**perating **S**ystem, т.е. дисковая операционная система американской фирмы Microsoft. Термин «дисковая» означает, что ОС располагается на диске: либо на винчестере, либо на дискете (размеры MS-DOS вполне допускают это), либо даже на компакт-диске.

Алгоритм выбора того или иного устройства устанавливается в BIOS. Например, если там в разделе «BIOS FEATURES SETUP (Показать установки BIOS)» в пункте «Boot Sequence (Последовательность загрузки)» перечислены диски A:, C:, это означает следующее: если в компьютер установлена дискета (диск A:), будет предпринята попытка загрузить MS-DOS с неё, если дискеты нет, то загрузка MS-DOS пойдет с винчестера (диск C:).

1.2. Состав операционной системы

Операционная система MS-DOS состоит из следующих частей:

- **BIOS** (от **B**asic **I**nput-**O**utput **S**ystem – Базовая система ввода-вывода). Эта часть ОС жёстко «вшита» в ПЗУ или ROM (от **R**ead **O**nly **M**emory – Память только для чтения). BIOS содержит тесты проверки узлов компьютера (автоматически запускаются при включении компьютера), а также Начальный загрузчик – небольшую программу для вызова более сложного Загрузчика MS-DOS (фрагмент файла io.sys). Кроме этого, BIOS выполняет низкоуровневые операции ввода-вывода (на уровне регистров) с монитором, клавиатурой, дисками и принтером.
- **Дисковые файлы io.sys и msdos.sys.** Файл io.sys (он располагается в начальном секторе системного диска) содержит, как было сказано, программу Загрузчик MS-DOS. Кроме того, io.sys дополняет BIOS другими низкоуровневыми операциями ввода-вывода. С помощью файла msdos.sys реализуются основные высокоуровневые услуги MS-DOS.
- **Файл command.com** – Командный процессор или интерпретатор команд MS-DOS. Содержит основные команды MS-DOS: CLS, DIR, COPY и другие, называемыми внутренними. Кроме того, он работает и с внешними командами MS-DOS, такими, как FORMAT, DELTREE и другими, находящимися, как правило, в каталоге DOS и представляющими собой файлы с расширением .com или .exe. Командный процессор command.com организует выполнение любых команд пользователя, вводимым с клавиатуры или с помощью командных файлов (файлы с расширением .bat).

Перечисленные выше компоненты MS-DOS составляют минимальный состав этой операционной системы. В дополнении к нему можно указать и такие элементы MS-DOS, как:

- **Внешние команды MS-DOS.** Представляют собой, как было сказано, файлы с расширением .com или .exe и находящимися, как правило, в каталоге DOS.
- **Драйверы.** Это специальные программы (файлы с расширениями .com, .exe или .sys), обеспечивающие обслуживание новых или нестандартное использование имеющихся внешних устройств. Активизация нужных драйверов осуществляется с помощью файлов config.sys и autoexec.bat.
- **Файл config.sys.** Представляет собой обычный текстовый файл, содержащий специальные команды для активизации различных драйверов. Кроме того, с его помощью можно изменять режимы работы или конфигурацию MS-DOS.
- **Файл autoexec.bat.** Это командный файл, автоматически запускаемый операционной системой на исполнение при включении или перезапуске компьютера.

1.3. Некоторые команды

Сервисные команды
1.3.1. Запросить справку о версии MS-DOS. Формат команды: VER
1.3.2. Очистить экран. Формат команды: CLS
1.3.3. Запрос текущей даты. Формат команды: DATE
1.3.4. Запрос текущего времени. Формат команды: TIME

Команды навигации по дискам и каталогам
1.3.5. Сменить текущий диск. Формат команды: ИмяДиска:
Пример: C:\>D:
D:\>
1.3.6. Сменить текущий каталог. Формат команды: CD МаршрутКаталога
Пример: C:\>CD DOS
C:\DOS>
1.3.7. Вернуться в родительский каталог. Формат команды: CD..
1.3.8. Перейти в корневой каталог текущего диска. Формат команды: CD\

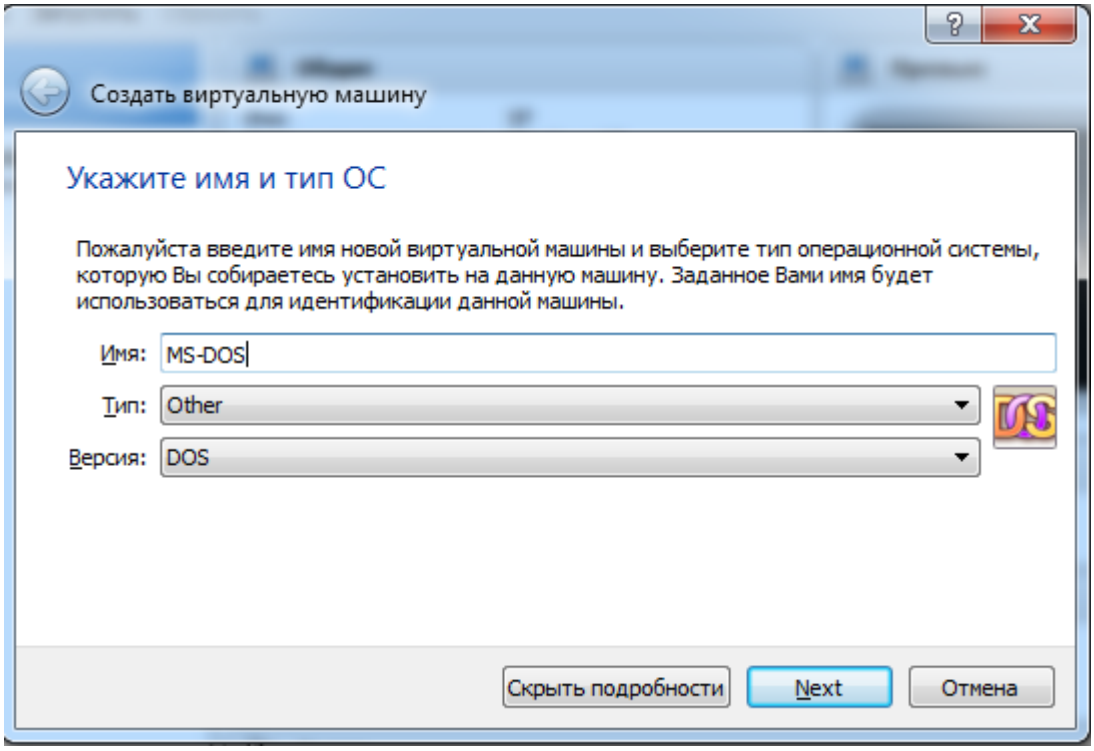
Команды для работы с файлами
1.3.9. Переименовать файл. Формат команды: REN СтароеИмя НовоеИмя
1.3.10. Просмотреть файл. Формат команды: TYPE ИмяФайла
1.3.11. Редактировать файл. Формат команды: EDIT ИмяФайла
1.3.12. Копировать файл. Формат команды: COPY ИмяФайлаОригинала ИмяФайлаКопии
1.3.13. Вывод файла на печать. Формат команды: COPY ИмяФайла PRN
1.3.14. Переместить файл. Формат команды: MOVE ИмяФайла МаршрутКаталога
1.3.15. Удалить файл. Формат команды: DEL ИмяФайла

Команды для работы с каталогами
1.3.16. Просмотреть содержимое каталога. Формат команды: DIR [МаршрутКаталога] [/P]
Где: /P – включить режим постраничного вывода информации на экран монитора (при большом объеме информации).
Примечание. Фрагменты командной строки, заключенные в квадратные скобки являются необязательными. При их
использовании квадратные скобки опускаются.
1.3.17. Создать новый каталог. Формат команды: MD ИмяКаталога
1.3.18. Переименовать каталог. Формат команды: MOVE СтароеИмя НовоеИмя
1.3.19. Удалить пустой каталог. Формат команды: RD ИмяКаталога

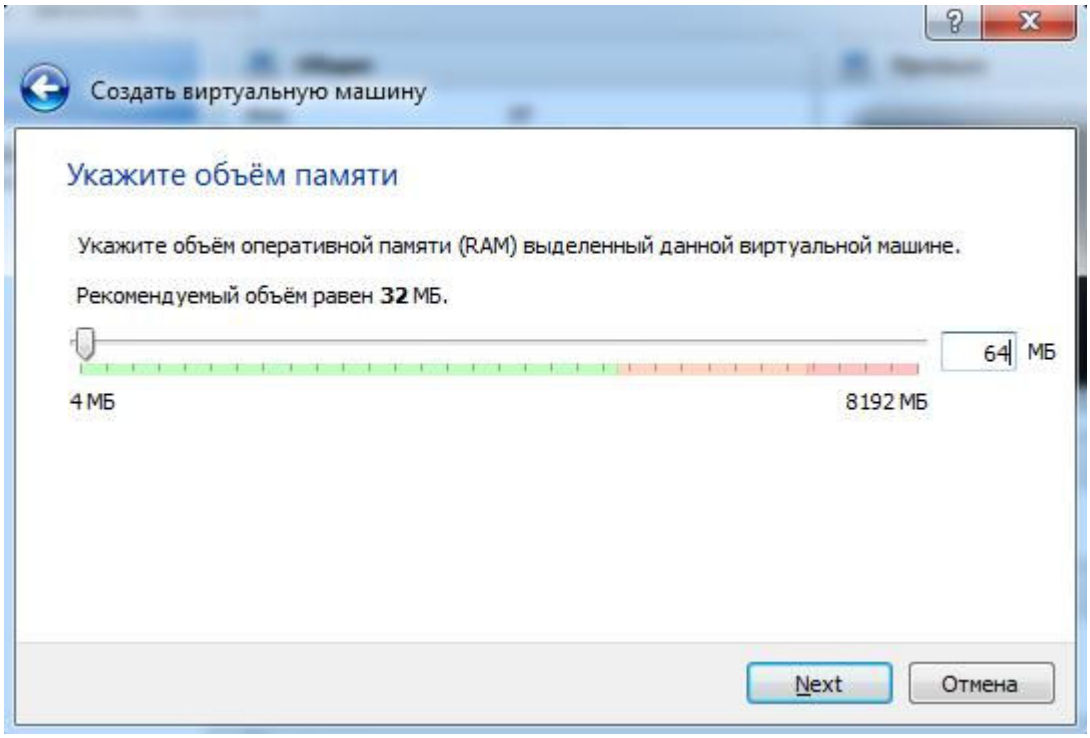
Примечание. Нельзя удалять текущий каталог, нужно перейти вначале в родительский.

2. 1 ПРАКТИЧЕСКАЯ ЧАСТЬ (часть 1)

1. Создать виртуальную машину в программе VirtualBox (имя машины должно иметь следующий формат: MS-DOS Фамилия



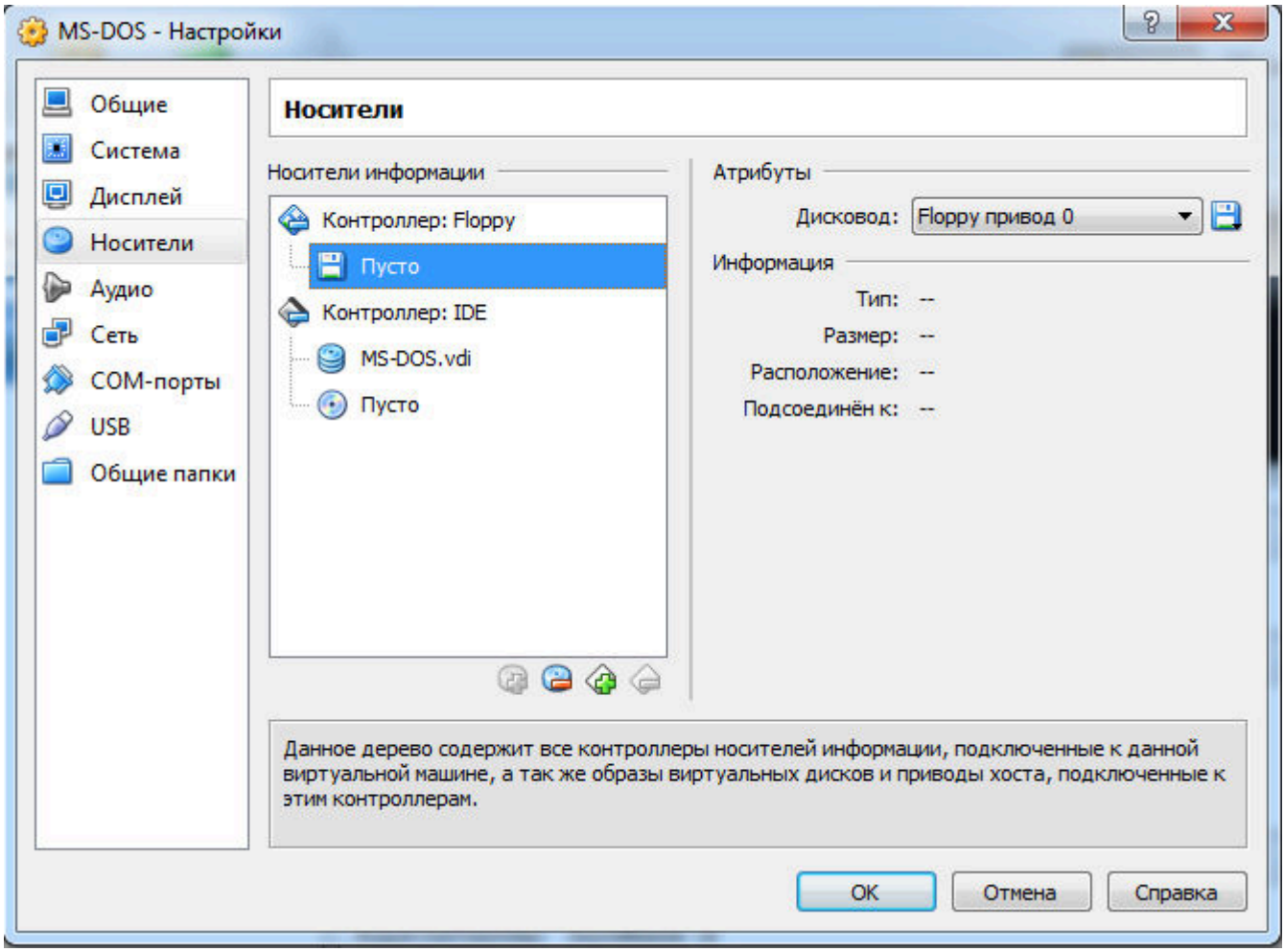
2. Укажите объем оперативной памяти 64Мб.



3. Создать новый виртуальный жесткий диск фиксированного размера 10 Мб

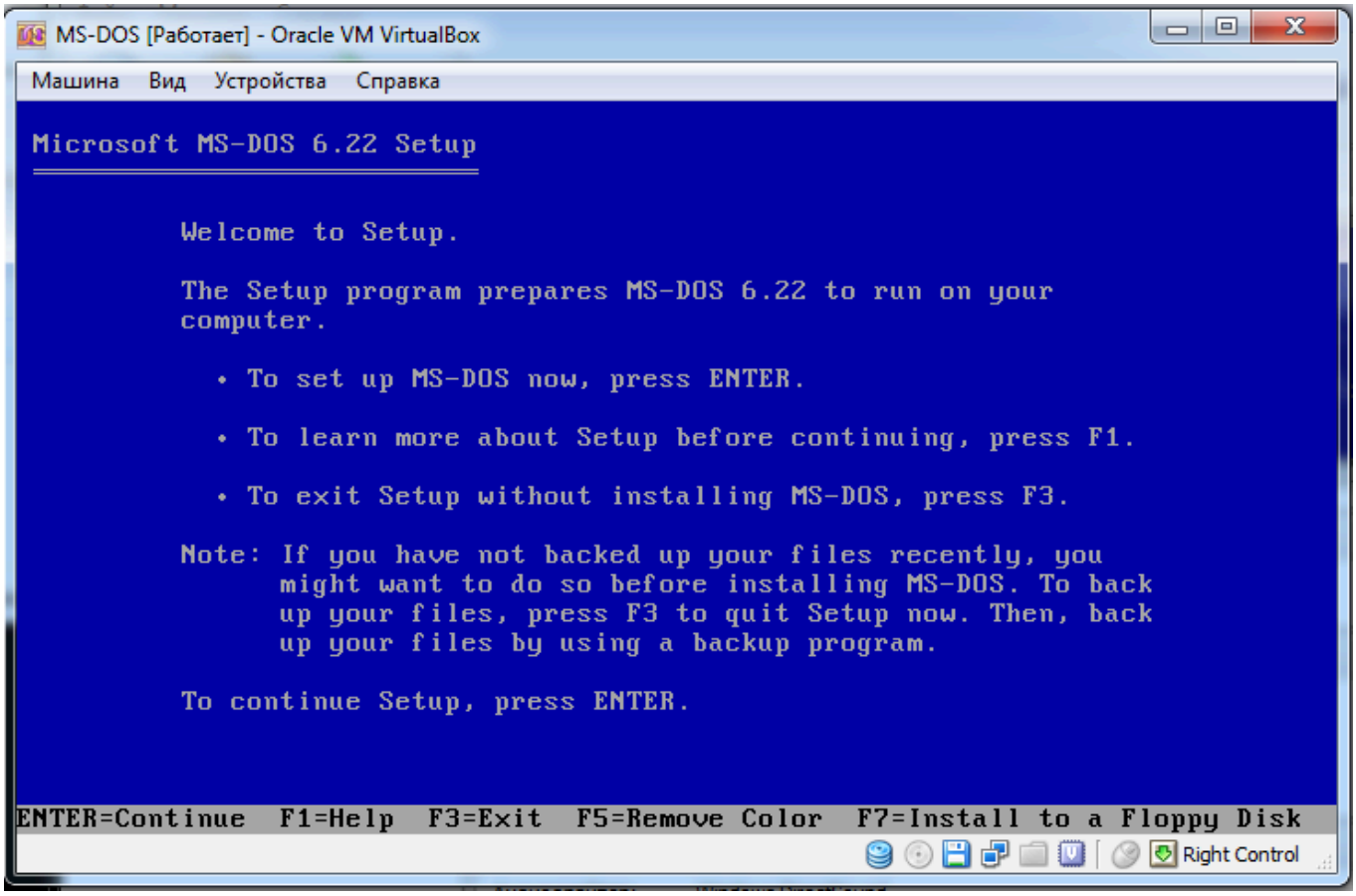
Для установки был выбран MS-DOS 6.22, так как это последняя «коробочная» версия этой ОС (остальные версии являлись частью Windows 9x). Устанавливается эта ОС на нескольких дискетах, поэтому здесь у нас есть два варианта — либо иметь полноценный флоппи-дисковод и дискеты с дистрибутивом MS-DOS, либо образы этих самых дискет. Дисковода у нас нет, но есть три установочные дискеты.

4. Войти в настройки созданной виртуальной машины, выбрать в качестве контроллера Flopy образ первой дискеты.

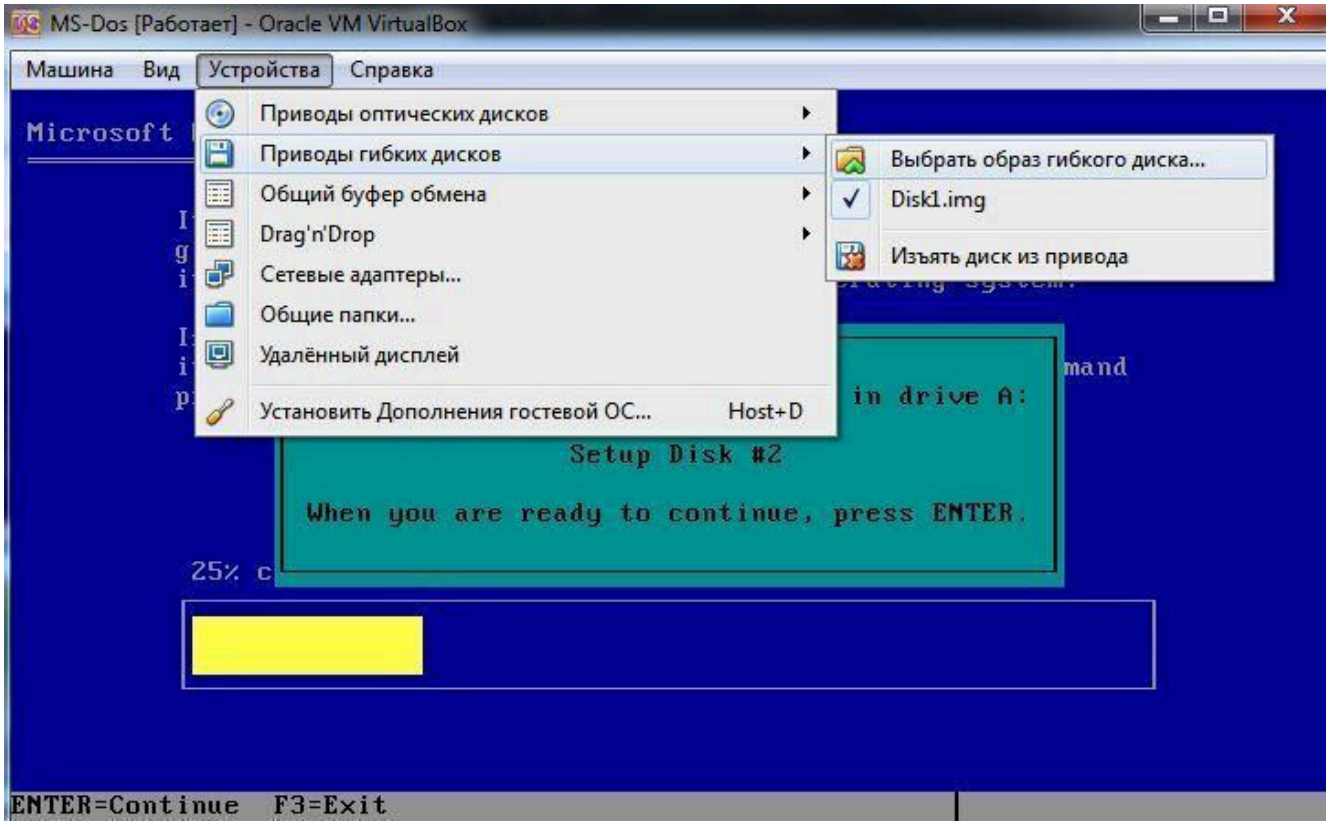


5. Загляните в раздел **Система** и убедитесь, что на вкладке **Материнская плата** в порядке загрузки первой стоит дискета (если это не так вы с легкостью можете изменить порядок при помощи кнопок со стрелочками).

6. Запустить виртуальную машину.



7. Нажать кнопку ENTER для продолжения. Далее начать установку продолжая нажимать кнопку Enter до требования поменять флоппи-диск на диск №2.



8. Далее сменить диск на диск (образ) №3.

9. По окончании установки извлеките диск из привода и нажмите ENTER.

10. После перезапуска компьютера у вас загрузится ОС MS-DOS.

11. Покажите результат выполнения первой части работы преподавателю.

2. 2 ПРАКТИЧЕСКАЯ ЧАСТЬ (часть 2)

1. Запустить виртуальную машину с установленной ОС MS-DOS
2. Очистить экран монитора
3. Запросить справку о версии MS-DOS
4. Создать новый каталог OS
5. Перейти в новый созданный каталог.
6. Создать в нем (с помощью команды EDIT) файл 2.txt.
7. Файл 2.txt должен содержать все известные вам команды ms-dos (по одной на каждой строке). Сохранить созданный файл.
8. Просмотреть созданный файл.
9. Сделать копию файла 2.txt, но с именем 2-copy.txt
10. Перейти в корневой каталог.
11. Просмотреть каталог OS.
12. Удалить файл 2-copy.txt
13. Показать результат выполнения работы преподавателю.

3. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назовите минимальный состав MS-DOS.
2. Как узнать номер установленной на компьютере версии MS-DOS?
3. Как, находясь в MS-DOS, сменить текущий диск D: на C:?
4. Как, находясь в MS-DOS, перейти из каталога C:\DOS в каталог D:\LERNEN\BAT?
5. Как, находясь в MS-DOS, перейти из каталога C:\DOS\BAT в родительский каталог C:\DOS?
6. Как, находясь в MS-DOS, перейти из каталога C:\DOS\BAT в корневой каталог диска C:?
7. Как, находясь в MS-DOS, просмотреть содержимое каталога C:\DOS из текущего каталога D:\LERNEN?
8. Как, находясь в MS-DOS, просмотреть содержимое файла autoexec.bat на дискете?
9. Как, находясь в MS-DOS, переименовать в текущем каталоге файл a.txt в b.txt?
10. Как, находясь в MS-DOS, переместить файл c.txt из каталога D:\LERNEN в корневой каталог диска D:?
11. Как, находясь в MS-DOS, сделать копию файла d.txt в этом же каталоге?
12. Как, находясь в MS-DOS, узнать сегодняшнее число?
13. Как, находясь в MS-DOS, узнать, который сейчас час?
14. Как, находясь в MS-DOS, создать каталог D:\TEMP?
15. Как, находясь в MS-DOS, переименовать каталог D:\TEMP в D:\VREM?
16. Как, находясь в MS-DOS, удалить каталог D:\VREM?

Лабораторная работа № 2

Тема: Norton Commander

NORTON COMMANDER

Работая с Windows, мы можем запускать программы, предназначенные для операционной системы MS-DOS (их называют DOS-приложениями). Я предлагаю познакомиться с самой популярной программой оболочкой для работы с файлами в DOS-режиме NORTON COMMANDER. Существуют версии этой программы для Windows.Умение работать с Нортоном пригодится и при необходимости быстро и эффективно поработать в DOS-режиме, даже если судьба занесет нас на старый компьютер, где нет Windows 95/98. Чем только не приходится человеку заниматься!

Многие операции в файловых менеджерах можно делать как мышью, так и с клавиатуры. Но изначально Нортон сориентирован на работу именно с клавишами – раньше ведь у персональных компьютеров не было мышей. А клавиши былвсегда.

Запускается Нортон командой nc, лежит обычно в папке с таким же именем (c:ncnc).

1. Работа с файлами

Файл – это однородная посвоему назначению совокупность информации, хранящаяся на диске и имеющая имя. Проще, файл-это однапрограмма. Один файл, одна картинка, один видеофильм, один мультик, одна музыкальная композиция.

Про каждый файл Windowsзнает дату и время его создания,последнего его изменения и последнего обращения к нему (просмотра), а также физические размеры. Размеры файлаизмеряются:

-<span Times New Roman"">

в байтах (1 байт (Б) – это длина одного символа, восемь двоичных символов, 1Б =8 б);

-<span Times New Roman"">

в килобайтах (тысячах байтов, записывается Кбайт или КБ);

-<span Times New Roman"">

в мегабайтах (миллионных байтов, пишется Мбайт или МБ);

-<span Times New Roman"">

в гигабайтах (миллиардах байтах, ГБ).

Имя файла в Windows 95/98/NT состоит обычно их двух частей –собственно имени и расширения (чаще всего длинной в 3 символа). Имя и расширение отделяются друг от друга точкой.Например:

Vova.doc index.html courier .ttf text-04.bac doc01_90.zip

На имена файлов накладываются некотороеограничения. Имя вместе с расширением (и точкой!) не может быть длиннее 255 символов. Кроме того, в нем еще недолжна использоваться косая черта (слеш) обеих разновидностей (и /). Кроме того, запрещено использовать знак вопроса (?), звездочку (*),знаки больше и меньше (> и <),двоеточие (:), кавычку («) и вертикальную черту (|). У всех этих символов тожеесть свое особое назначение.

Имена папок с файлами (директорий, каталогов)подчиняются тем же законам, что и имена самих файлов. С одним только отличием: папки редко дают расширения, поскольку в этом нет особого смысла.

Будьте осторожны, давая имена файлам и папкам с использованием русских букв. Надо понимать, когда это стоит делать, а когда нет, иначе могут возникнуть проблемы.

Во-первых, немалонайдется нерусифицированных программ, которые не поймут имен с использованиемрусских букв или же не совсем правильнобудут с ними работать.

А во — вторых, большие проблемы будут прочитать файлы с русскоязычными именами на нерусифицированном компьютере. До сих пор в ходу немало машин, где стоит нерусифицированная (или не полностью русифицированная) версия MS-DOS или Windows. Когда мыперенесем на компьютер свой замечательный файл, система может просто не понять его имени, посчитать, что в нем использованы какие-то неразрешенные символы, что этовообще не имя, а не понятно что.

В одной папке имена файлов должны быть уникальны и неповторимы. Зато несколько файлов могут иметь одно и тоже имя, если при этом у нихразные расширения. Например, в одной папке рядышком могут лежать файлы readme.txt, readme.doc и просто readme. Файлы, имеющие совершенно одинаковыеимя и расширения, могут лежать в разных папках.

1.1 Копирование файлов

Всякий человек, работающий с компьютером, время от времени берет дискету,вставляет ее в дисковод и записывает на нее какие-то файлы. Чтоб отнести на работу доклад, статью, план мероприятий на квартал; в учебное заведение – реферат, курсовой проект; отнести приятелю – маленькую, но симпатичную игрушку, картинку собственной работы или какую -нибудьполезную программку; доставить в редакцию газеты статью или репортаж с презентации новогоальбома супезнаменитой певицы Люси.

Скопируем для примера на трехдюймовую дискету файл, который называетсяРеферат.doc и лежит на жестком диске компьютера в папке Мои документы.

А теперь подробно.

1.<span Times New Roman"">

Вставляем дискету в дисковод.

2.<span Times New Roman"">

Дважды щелкаем по значку Мой компьютер на рабочем столе. Открывается окно. Значок трехдюймового дисковода называется Диск 3,5 (A:), а в англоязычных виндах – Floppy3.5 (A:).

3.<span Times New Roman"">

Двойной щелчок по значкуДиск 3,5 (A:) – и мы уже на дискете. Теперь мы видим содержимое дискеты. Перетаскиваем туда реферат. Итак, мы нашли, кудабудем копировать файл. Теперь надо найти сам файл.

4.<span Times New Roman"">

В Windows98 значок папки Мои документы, откуда мы собирались взять реферат, лежит прямо на рабочем столе. Можно дважды щелкнуть по нему, открыть эту папку и в ней отыскатьнужный файл.

5.<span Times New Roman"">

Теперь надо сообразить, на каком диске лежит файл с рефератом. Еслижесткий диск у вас один и не поделен на части, то можно сказать, что он будет носить буквенное имя C:.Двойной щелчок по его значку, и мы перейдем на диск.

6.<span Times New Roman"">

Отыщем на диске нужнуюпапку, двойным щелчком войдем в нее. Если нам нужна одна из вложенных папок, дважды щелкнем и по ней, чтоб войти. И так далее. В итоге найдете нужный файл.

7.<span Times New Roman"">

Витоге, оттащив файл на дискету, можете ее вынимать из дисковода и нести туда,куда собрались.

1.2. Копирование нескольких файлов.

Довольночасто возникает необходимость скопировать или переместить не один файл, а несколько. Но если вы подумали, что вас заставят копировать каждый файл в отдельности, то это вы напрасно… Проявлено некотороечеловеколюбие и продемонстрирована «дружественность интерфейса (это всясистема общения программы с пользователем: панели инструментов, меню, горячиеклавиши, команды, диалоговые окна …)».

Чтобы скопировать все файлы из папки, выделите ихкомбинацией Ctrl-A и сделайте то же самое.

Прикопировании большого объема данных, которое может заниматьнемало времени, никто не помешает вам перейти в другую программу и работать в ней в свое удовольствие. Правда,копирование на дискеты и с дискетнесколько подпортит вам удовольствие от производственной деятельности или раскладывания пасьянса, потому что отнимает у процессора на удивление много сил, и все другие программыначинают «тормозить».

1.3. Перемещение файлов

Обратим внимание на плюсик, появляющийся рядом с курсором приперетаскивании файла. Он означает, что будет выполняться именнокопирование («плюс один файл»). Если же мы станем тащить файл снажатой клавишей Shift, то плюсика нам непокажут. Будет выполняться другая операция – перемещение. При этом файлзапишется на новое место, а на старом будет стерт.

Перемещением часттопользуются, когда нужно снять файлы с дискеты: объеем ее не так уж велик, изамусорить ее настолько, что на нейне останется свободного места, довольно легко. А так: записал – снял, и место снова свободно.

Перетаскивая файлы из папки в папку в пределах одногодиска, вы всегда их перемещаете. Сделано это для того, по-видимому, чтобы файлы ваши не размножались, как кроликитеплым летом и при хорошей кормежке.

1.4. Переименование файлов.

Встаете на файл, нажимаете клавишу F6 или кнопку 6 RenMove(переименовать/переместить). Появляется окошко, только вместо Copy будет Move. Вводите новое имя и жмите Enter.

Если также поступить в окне копирования, то у нас получится копия файла с новым именем втой же папке.

Чтобы сменить только расширение, а имя не трогать, можете для быстроты написать: .doc. Инаоборот, чтобы поменять только имя, напишите имя.*

1.5. Стирание файлов

Текущий файл или группа выделение файловстирается клавишей F8 или кнопкой 8 Delete. Нортон запроситподтверждение и сотрет.

Неопытные пользователи иногда забывают про то, что у них выделены какие-тофайлы. Если директория большая и все файлы на экране разом не помещаются,то выделенных файлов может быть просто не видно.

1.6. Создание директорий

Клавиша F7 или кнопка 7MkDir (создать папку – makedirectory) вызывает окошко, вкотором вам останется только ввести имя и нажать Enter.

1.7. Просмотрфайлов

Клавиша F3 или кнопка 3 View выведет содержимое файла в окне собственного нортоновского просмотровщика. В отличие от виндоузовскихколлег нортоновский просмотровщик грузится мгновенно.

Высможете просматривать в «поиске» любые файлы, включая тексты и досовскойи виндоузовской кодировке. Тексты в формате разных текстовых редакторов.

По клавише F8 (8 Viewer)вы сможете сменить формат просмотра. Кстати, в числе прочего Нортон сможет вам показать даже оформленные файлы в формате RTF, чего не умеют виндоузовскиевьюеры.

1.8.Редактирование

у Нортонаесть простенький редактор, позволяющий быстро отредактировать текстовый файл небольшогоразмера. Перейти к редактированию текущего файла можно по клавише F4 или по кнопке 4 Edit.

Выделениефрагмента возможно только целыми строками: нажмите F3, а выделив нужное – еще раз F3, чтобы закончитьвыделение. Потом можно выделенное:

-<span Times New Roman"">

F8)

-<span Times New Roman"">

F5)

-<span Times New Roman"">

F6) туда, куда вы поставите курсор.

Отмены команды здесь нет, такчто надо быть с редактором очень осторожными. Единственный способ отменитьизменения – выйти из файла, не сохраняя результаты, или же сохранить его с другимименем.

2. Работа сдисками и дискетами

Начнем мы с программ для обслуживания жесткихдисков.

Жесткий диск, винчестер – основное запоминающее устройствокомпьютера, главное хранилище информации. И хранилище это хрупкое. Его следует беречь, холить и лелеять, если, конечно, дороги данные,которые в нем хранятся. Можете также не обращать внимания на сбои файловой системы, пинать винчестер ногами, поливать горячим чаем, а если позволит достаток, то и мятным ликером…

На страничкеСервис свойств диска представлены три главные программы из комплекта Windows. Это Проверка диска (ScanDisk), Дефрагментация диска (Defrag) и архивация данных (MicrosoftBackup).

Дважды щелкнув по значку Мой компьютер, который лежит обычно вверхнем левом углу рабочего стола, увидим окно, гдеглавное для нас сейчас – диски. Тут окажутся и значки винчестеров – всех, какие есть. Дисководы 3,5, CD-ROM'ы идругое.

Если жесткийдиск один и он не поделен на части, то увидимвсего один значок жесткого диска – с именем C:. Настоящее же имя диска в системе – это буква и двоеточие:A:, C:, D:, E: и т.д.

О том, какназывается место, в которое мы забрались, сообщает нам строка заголовка. Например: папка Мой компьютер, диск D:.

Интересно, что для Windows адреса D:GamesQuake II, d:gamesquake iiи D:GAMESQUAKEII совершенно одинаковы. Эта операционнаясистема не различает больших и маленьких букв в именах файлов и папок.

2.1. Проверкадиска

Структура данных на диске может быть нарушена по разнымпричинам. В основном это происходит из-за сбоев и зависания программво время записи на диск. Такой слегка поврежденный диск, в общем, не теряет работоспособности, но если неисправности на нем накапливаются, может и потерять…

Кроме того, на поверхности диска могут возникнуть дефекты – какой участок перестает читаться, и все на нем записанное теряется. Особенно часто такая беда случается с дискетами.

УтилитаПроверка диска (Scandisk) предназначена для того, чтоб следить и принимать меры в этом случае. Она проверяет структуру данных, папок, таблиц размещения файлов, ищет потерянные цепочки данных (кластеры) и проч. И по возможности устраняет обнаруженные ошибки. Кроме того, она может проверить поверхность диска на наличие сбойных участков ипопытаться возможные данные с этихучастков перенести на исправные. Сами же сбойные участки помечает специальным указателем (bad block – плохойблок), чтобы на это нехорошее место никакие программы уже ничего не пытались писать.

Проверять диск время от времени необходимо. Приинтенсивной работе с компьютером – не реже, чем раз в неделю, а после сбоев каждый раз. Впрочем,совершенно не обязательно делать ее именно Scandisc'ом. Есть и другие утилиты аналогичного назначения, например, NDD (NortonDiskDoctor) из пакета Norton Utilities для Windows или DiskMinderиз пакета Nuts&Bolts. Обе названные программы работают заметно быстрее майкрософтской, устраняя при этом большеразличных неисправностей дисковой системы.

2.2.Дефрагментация диска

ПрограммаДефрагментация диска (Defrag)предназначена для того, чтобы собирать файлы из фрагментов в одно целое в физическом смысле,ускоряя тем самым обращение к диску.

Начинаете дефрагментацию и можете идти работать в другую программу – этодело долгое и кропотливое даже на быстрых винчестерах. Но,опять-таки, стоит вам хоть один байт записать на оптимизируемый диск, как программа начнет

оптимизациюсначала. Так что на время дефрагментации лучше оставить его в покое. Можно работать на другом логическом диске, можно слушать музыку. Что еще?

КнопкаСведения позволит посмотреть карту оптимизируемого диска.

Приинтенсивной работе дефрагментацию надо проводить примерно раз в месяц, а то и чаще.

2.3.Копирование дискет (CopyDisk)

В контекстном меню дискеты есть уникальная команда, которая не появляется в контекстном меню ни одного другого объекта. Она называется Копироватьдиск (CopyDisk).

Этастрока вызывает маленькую программку, предназначенную для создания на одной дискете точной копиидругой.

Нажмитекнопку Начать, и программа примется считывать информацию с дискеты.Потом вас попросят вытащить дискету, вставить другую и нажать ОК. И программа пойдетзаписывать на новую дискету всю считанную раннее информацию.

2.4. Очисткадиска (DiskCleanup)

По мере замусоривания диска свободное место на нем тает. Ставим какие-то программы иигрушки и забываем про них, рисуете или сканируете какие-то немереных размеров картинки, гуляем, ни о чем не думаю, по Интернету.А потом вдруг выясняется, что на диске осталось слишком мало места для жизнедеятельности системы.

Тут-то Windows 98 нам и сообщает: пора произвести уборку. И делает этоза нас утилита Очистка диска (Disk Cleanup), которая позволит автоматизировать процедурууборки мусора с ваших жестких дисков.

Существуютспециальные программы-подметалльщики вроде Cleansweep, которыемогут находить гораздо больше всякого мусора:

-<span Times New Roman"">

windowstemp;

-<span Times New Roman"">

-<span Times New Roman"">

имена, а могут и совсем разные;

-<span Times New Roman"">

-<span Times New Roman"">

2.5.Архивация дисков (MicrosoftBackup)

Важнейшим моментом компьютерной безопасностиявляется создание резервных копий файлов (Backup).

Надо создавать и постоянно обновлять копии самых важных для вас файлов.

Если вашифайлы маленькие и их не много, копии можно держать на дискетах, периодическипроверяя Scandisck'омна наличие сбоев. А для суперважнейших файлов надоиметь две копии.

Если жефайлы ваши «велики и числом многи», так чтопользоваться дискетами неудобно и даже вовсе невозможно, надо создавать копиина внешних носителях большой емкости. Те, кто работает в локальной сети, могутсохранять резервные копии также на сетевых дисках (т.е. на других компьютерах).

Должныпомнить, что:

-<span Times New Roman"">

создавая копию на том же диске, страхуйтесьтолько от случайного стирания. Ни от вируса, ни от физической неисправностидиска этот способ не поможет;

-<span Times New Roman"">

создавая копию на другом физическом диске, страхуйтесьтакже от отказа диска. Но вирус до второго диска достанет и нагадит;

-<span Times New Roman"">

создавая копию на внешнем носителе, который поокончании копирования вынимается из компьютера и кладется на палочку, выстрахуетесь также от вирусной атаки, которая может привести к уничтожениюданных на всех подключенных дисках.

2.6.Форматирование дискет

Оболочка у дискеты квадратная, но сама дискета круглая. Информация на ней располагается вдоль концентрических дорожек, каждая дорожка имеет свой номер. Дорожки, в свою очередь, делятся на сектора.

Данные пишутся и на верхней, и на нижней поверхности магнитного диска, для каждой поверхности используется отдельная считывающая и записывающая головка. Собственно, это же касается и жесткого диска, только у него не один магнитный диск, а несколько и, соответственно, много головок для чтения и записи.

Укаждого диска и дискеты есть таблица размещенияфайлов и корневая директория, в который могут создаваться все остальные папки и размещаться файлы. Вся эта структура создается при разметке диска – форматирование.

Для форматирования дискет и жестких дисков используются также командой format. Главным образом - в режиме MS-DOS, но можно запустить ее и из виндоузовской командой строки.

3. Архивныефайлы

Что такое архиваторы?

Архиваторы – это программы, позволяющие уменьшитьразмер файла для экономии места надиске. Работают они по-разному, но сутьих деятельности в следующем: в файлахбывают какие-то повторяющиеся фрагменты, и нет смысла держать их все надиске целиком. Задача архиватора – найти эти повторяющиеся фрагменты и записатьвместо них какую-то другую информацию, по которой можно было бы потом вспомнить, кто за кем стоял.

Эффективность архиватора будет разной для разных файлов. Скажем, тексты сжимаются в среднем вдва-три раза. Файлы для черно-белых картинок без полутонов могут сжиматься и в два, и в четыре, и даже вдесять раз – в зависимости от насыщенности деталями.

3.1. Созданиеархива и добавление в него файлов

Если хочется сархивировать файлы директории DOC и у нас есть архиватор ARJ, то на «раз» входим в директорию (например, в Нортоне), на «два» набираем в командной строке: arj a document, а затем жмем на энтер.Программа начнет паковать файлы.

Здесь:

arj– имя программы-архиватора,

a – (add, «добавить») – команда создания нового архива или добавления файлов в уже в существующийархив,

document – имя создаваемогоархива (расширение arj архиватордобавляет сам).

Когдаэкономят место, часто делают не добавление, а перемещение файлов в архив (move).

В списке Actions (действия) можно выбрать одну из операций архивирования:

-<span Times New Roman"">

Add(andreplace)files –добавлять файлы в архив, а если там такие уже были, заменять;

-<span Times New Roman"">

Freshenexistingfiles– обновить («освежить») только существующие файлы;

-<span Times New Roman"">

Update (andadd) files – обновить существующие и добавить новые;

-<span Times New Roman"">

Movefiles– переместить файлы в архив.

3.2.Как достать файлы из архива

Чтобы распаковать архив – достать из него файлы, — надо вместо команды a (add) выполнит команду e (extract – «извлечь») или x (тоже, extract), для чего ввести:

arjedocument.arj– извлечь файлы из архива;

arjxdocument.arj– извлечь файлы из архива и разложить по вложенным папкам.Если таких папок на диске пока нет, архиватор их создаст.

3.3. Удалениефайлов из архива

Поройтребуется почистить непомерно разросшийся архива. С помощью WinZip'а удаляются файлы проще простого.Выделить и утащить в корзину или нажать клавишу Del.

3.3. Работа сархивами в NortonCommander

Работа с архивами в Нортонеорганизованна довольно удобно. Во всяком случае не приходится писать никакихДОСовских команд.

Чтобыпосмотреть содержание архива, достаточно вставить курсором на архивный файл инажать Enter.Появляется окно.

В заголовкеокна будет стоять имя файла как напоминание о том, что мы влезли не в обычнуюпапку, а все же в архив, например: Arj:arch011.arj

Копирование файла в эту как бы папку на самом делеявляется архивированием (добавлением файла в архив), а из нее – разархированием (извлечением из архива). Вообще с файлами вархиве можно сделать все что угодно – заглянуть в него (F3), выделить, а потом удалить (переименовать, скопировать – F8,F6, F6). Нельзя только редактировать.

При наличиию Вас в компьютере других досовских архиваторов – ARJ, Arc,LHARS, PKZIP, Zoo и др., можно воспользоваться также ими.

Можнопользоваться также комбинацией Alt-F6 для разархивирования файлов, но нагляднее иудобнее – зайти в архив.

П Л А Н:

1.5.<span Times New Roman"">

1.6.<span Times New Roman"">

1.7.<span Times New Roman"">

1.8.<span Times New Roman"">

2.<span Times New Roman"">

2.1.<span Times New Roman"">

2.2. ДЕФРАГМЕНТАЦИЯ ДИСКА

2.3. КОПИРОВАНИЕ ДИСКЕТ

2.4.<span Times New Roman"">

2.5.<span Times New Roman"">

ДИСКОВ

2.6.<span Times New Roman"">

3.<span Times New Roman"">

Лабораторная работа № 3

Тема: Файл AUTOEXEC.BAT

Цель работы: ознакомление с назначением файла AUTOEXEC.BAT и возможности его использования.

Краткие теоретические сведения.

Файл AUTOEXEC.BAT представляет собой командный файл, обращение к которому осуществляется автоматически при начальной загрузке операционной системы. При этом сам файл должен находится в корневом каталоге системного диска. Обработка его происходит вслед за обработкой содержимого файла CONFIG.SYS.

Основное назначение файла AUTOEXEC.BAT состоит в формировании удобной индивидуальной операционной обстановки. Для этого могут быть использованы команды установки пути поиска (PATH), приглашения MS-DOS (PROMPT), загружены определенные системные программы и т.д. Вообще в файле AUTOEXEC.BAT могут быть использованы любые команды MS-DOS и вызваны любые программы пользователя.

Как и любой пакетный файл, AUTOEXEC.BAT обрабатывается командным процессором последовательно, команда за командой. Выполнение его можно прервать при помощи комбинации клавиш CTRL/C. После выполнения очередной команды на экране появится надпись «Terminatebatchjob? (Y/N)», и после нажатия на клавишу Y выполнение файла прекращается.

Рассмотрим, как может выглядеть файл AUTOEXEC.BAT.

```
echo off

dosedit

path b:; c:\; c:\work

prompt $p$g

ver

echo СИСТЕМА ЗАГРУЖЕНА

pause

date

time

cls
```

Первая строка определяет, что текст всех команд командного файла не будет выводится на дисплей.

Вторая строка загружает программу, запоминающую и позволяющую вызвать и редактировать все команды пользователя на протяжении сеанса работы.

Третья строка указывает, что файлы будут отыскиваться после поиска в текущем каталоге в установленном каталоге диска B:, корневом каталоге диска C: и подкаталоге WORK диска C:.

Четвертая строка определяет, что приглашение MS-DOS будет содержать имя текущего каталога.

Пятая строка вызовет указание номера версии MS-DOS.

Шестая строка выводит сообщение «СИСТЕМА ЗАГРУЖЕНА».

Седьмая строка вызывает появление на экране сообщения «Strikeanykeywhenready.. .», в результате чего обработка следующей команды будет производиться лишь после нажатия любой клавиши.

Команды date и time вызывает инициализацию задания пользователем даты и времени. Команда CLS в последней строке командного файла очищает экран дисплея.

Выбор команд, включаемых в файл AUTOEXEC.BAT, в общем случае определяется преподавателем.

Подготовка к работе.

Изучить теоретические сведения и составить план выполнения лабораторной работы.

Подготовить собственный вариант файла AUTOEXEC.BAT.

Порядок выполнения

Одним из известных Вам способов сделать Ваш рабочий диск системным.

Загрузить операционную систему с Вашего рабочего диска.

Изучить операционную обстановку.

С помощью известного Вам редактора либо с помощью команды COPY набрать подготовленную версию файла AUTOEXEC.BAT.

Перезагрузить систему.

Изучить новую операционную обстановку.

Сформулировать желаемые отличия операционной обстановки от су-щес-тву-ю-щих и составить новую версию файла AUTOEXEC.BAT.

Набрать ее и убедиться в достижении желаемой операционной обстановки.

Содержание отчета

Цель работы.

Краткое описание основных операторов, используемых в пакетных фай-лах.

В рабочем отчете зафиксировать все действия пользователя, сделать выводы по основным пунктам лабораторной работы, привести тексты созданных пользователем версий файлаAUTOEXEC.BAT.

Лабораторная работа № 4

Тема: LINUX, работа в экранном редакторе.

Цель работы: Ознакомление с основными командами Linux, получение навыков по их использованию. Работа с текстовым редакторомVi.

Общие сведения о Linux

Что такое Linux?

Итак, что же это все-таки такое - Linux? Linux создавался как операционная система для IBM совместимых компьютеров (Теперь Linux перенесен на многие другие платформы, в том числе на 680x0, Альфу и многопроцессорные рабочие станции). Linux создан и продолжает создаваться благодаря усилиям программистов, разбросанных вокруг всего мира. Целью его разработки было создать операционную систему, принадлежащую к клону UNIX, но свободную от каких-либо коммерческих авторских прав, которую могли бы использовать программисты всего мира. Вообще-то Linux стартовал как хобби одного человека - финского студента Линуса Торвальда. Тогда его целью было создание замены MINIX. Это была похожая на UNIX учебная операционная система для компьютеров с процессорами фирмы Intel.

Что такое UNIX?

Ну, хорошо, теперь мы знаем, что такое Linux. А что такое Unix? Unix это тоже операционная система. Но эта операционная система не привязана к определенному типу компьютера. Один из самых интересных фактов из биографии UNIX это то, что ее первый прототип был написан в 1969 для машины DEC PDP-7 на ассемблере. В 1973 году она была переписана на Си. Благодаря этому она получила легкую переносимость на другие типы машин.

В те времена UNIX распространялся не очень быстро - эта ОС требовала больших ресурсов. Но благодаря легкости переноса на другие машины и удачной концепции UNIX распространялся все шире. Суть концепции UNIX в том, что задачи решаются не благодаря большим и мощным программам, а благодаря взаимодействию небольших программ. Сегодня практически на всех машинах, которые используются для научных или других применений, где требуется многозадачность и многопользовательский режим стоят или UNIX или UNIX подобные ОС. Широко разрекламированная Windows NT это тоже UNIX подобная система в исполнении фирмы Microsoft.

Чем Linux отличается от UNIX?

Главные отличия LINUX-а от UNIX-а:

Цена.Коммерческие UNIX системы стоят 1000 - 3000 USD. Linux распространяется бесплатно или для коммерческих дистрибутивов по сравнительно низкой цене.

Лицензионная политика. Linux распространяется вместе с исходными текстами и под лицензией, которая не разрешает использовать Linux не распространяя исходных текстов. Эта политика постоянно поддерживает цену коммерческих дистрибутивов Linux на низком уровне. Эта политика делает также невозможным использования тактики имени Microsoft - использование недокументированных возможностей системы.

ПортативностьLinux с самого начала был предназначен для работы на IBM совместимых компьютерах. Отсюда его невысокие требования к ресурсам.

Краткое описание команд Linux

Подключение к серверу

Для установления соединения с сервером LINUX необходимо запустить программу **telnet.exe**. Для этого в меню**"Пуск" (Windows95/98)**выберите пункт**"выполнить"**, затем введите**telnet**и нажмите**Enter**. Программа**telnet** использует сетевой протокол TCP/IP.

В строке главного меню "**Подключить**" программы**telne**t выберите пункт "**Удаленная система**", введите имя узла – 62.76.207.201, и нажмите кнопку**подключить**.

Вход в систему

Ввод команд в Linux выглядит примерно так же, как в DOS и других операционных системах, ориентированных на ввод в командной строке. Linux, как и UNIX, чувствительна к регистру, поэтому если система не воспринимает какую-либо команду, проверьте, в правильном ли регистре вы ввели ее. Как правило, команда выполняется после нажатия клавиши <Enter>.

Вызов истории команд

В Linux есть средство повторного обращения к уже выполненным командам, которое не прерывается даже при выключении компьютера. Предыдущая команда вызывается после нажатия клавиши <Up>, а для ее выполнения надо нажать <Enter>. Для вывода всего списка примененных команд воспользуйтесь командой history:

[tackett@web~] $ history

1 clear

2 adduser

3 history

Чтобы выполнить команду из хронологического списка, вызывайте с помощью клавиши <Up> предыдущую команду до тех пор, пока в командной строке не появится нужная, или же нажмите <!> и введите номер нужной команды. Например, чтобы повторно выполнить команду adduser из представленного выше списка, введите

[linux_lab@ais] $!2

Максимальное число команд в хронологическом списке задается в пользовательском конфигурационном файле .profile.

Основные команды Linux

Команда справки *man*

Для получения справки по той или иной команде Linux воспользуйтесь командой man. В ответ Linux открывает на нескольких, сменяющих друг друга экранах описание нужной команды. Если не помните точно имени нужной команды, введите команду man с параметром -k, затем ключевое слово для поиска нужной команды. Система выполнит поиск в своих файлах справки, содержащей это ключевое слово. Для этой команды имеется также псевдоним apropos.

Например, если ввести команду man ls, Linux выведет на экран справку о команде Is, в том числе обо всех ее параметрах. По команде man -k cls выводится список всех команд, в справке, о которых есть слово cls. Команда apropos cls аналогична команде man -k cls.

Команды для работы с каталогами

В Linux есть много команд для работы с каталогами. Как и в других операционных системах, в которых вам, возможно, приходилось работать, каталоги в Linux можно удалять, создавать, перемещать, а также выводить информацию об их состоянии.

Смена текущего каталога с помощью команды *cd*.

В Linux, как и в DOS, файлы хранятся в каталогах, организованных в древовидные структуры. Файл можно указывать в виде пути из корневого каталога, обозначаемого символом /, до файла. Таким образом, полное имя конфигурационного файла .emacs, принадлежащего пользователю jack, может иметь вид /home/ jack/.emacs.

Тем, кто привык работать с файлами DOS, длина имен которых не превышает восьми символов, а расширений — трех, приятно будет узнать, что в Linux подобных ограничений нет.

В Linux есть понятие рабочего каталога пользователя. Рабочий каталог обычно обозначается символом ~ (тильда). Например, команда копирования файла из текущего каталога в рабочий может иметь вид cp .emacs ~

Для перемещения по дереву каталогов Linux применяется команда cd. Для перехода в рабочий каталог эта команда вводится без параметров. Для перехода из одного каталога в другой формат команды тот же, что и DOS: cd new-directory, где new-directory — новый каталог, в который следует перейти. Кроме того, в Linux текущий каталог представляется одной точкой (.), каталог-родитель — двумя (. .) — и, конечно же, в этом DOS наследует UNIX и Linux, а не наоборот.

Будьте внимательны с символом разделителя каталогов. В DOS для этого применяется обратная косая черта (\), которая в Linux служит указателем продолжения команды с новой строки. В Linux каталоги разделяются прямой косой чертой (/). Кроме того, в DOS не имеет значения, отделены ли параметры (.) и (. .) пробелами от имени команды, в то время как в Linux это важно, Linux не поймет команды cd. ., правильный формат которой — cd ... В Linux между командой и параметром обязательно должен быть пробел.

Вывод информации о файлах и каталогах с помощью команды *ls*

ls — сокращение от list (список). В Linux по этой команде на экран выводится список файлов. Это аналог команды dir из DOS (которую можно применять и в Linux) для вывода списка файлов в каталоге.

Чтобы указать, как именно выводить список файлов, каких файлов и с какой информацией о файлах, придется ввести команду ls с параметрами. Чаще всего применяется параметр -1a, по которому выводится полная информация о каждом файле каталога. По команде is -la выводится подробная информация о файлах текущего каталога. По команде Is emacs выводится только имя этого файла, по команде же Is -la emacs — полная информация о нем.

Создание каталога с помощью команды *mkdir*

Поскольку структура каталогов составляет основу файловой системы, в Linux имеется также команда создания каталога mkdir. В отличие от DOS, где можно воспользоваться псевдонимом данной команды MD, в Linux надо вводить ее полное имя. В качестве параметра указывается имя создаваемого каталога, как в следующем примере: **mkdir backup**

Удаление каталогов с помощью команды *rmdir*

Каталоги в Linux удаляются с помощью команды rmdir, в качестве параметра которой указывается удаляемый каталог. Linux может удалить только пустой каталог. Например, если в каталоге /backup есть два подкаталога, команда rmdir /backup выполнена не будет. Чтобы удалить один из подкаталогов — /jack, — сначала по команде rmdir /backup/jack/* из него удаляются все файлы, затем с помощью команды rmdir /backup/jack — он сам.

ВНИМАНИЕ

С помощью команды rmdir нельзя удалить непустой каталог, но это можно сделать с помощью команды rm с параметром -r. Например, по команде rm -r * из текущего каталога будет удалено все, включая подкаталоги. Будьте внимательны, пользуясь этой командой, ибо, удалив каталог, нельзя будет восстановить ни его, ни содержавшиеся в нем файлы.

Команды работы с файлами

В Linux нет принципиального различия между файлами и каталогами, поэтому для работы с теми и другими применяются одни и те же команды.

Копирование файлов с помощью команды*cp*

Команда cp аналогична команде copy из DOS. Она применяется для копирования одного или нескольких файлов из одного каталога в другой. Синтаксис команды:

cp from-filename to-filename,

где from-filename — исходный файл; to-filename — файл, в который происходит копирование.

Чтобы команда была выполнена, надо указать оба параметра. Чтобы скопировать файл с тем же именем в качестве второго параметра, ставится точка (.). В этом отличие от DOS, где второй параметр в подобном случае просто опускается.

По команде cp fredl fredl.old создается резервная копия файла fredl с именем fredl.old. По команде же cp ~fredl.old /backup/jack файл fredl. old копируется из рабочего каталога в каталог /backup/jack. Рабочий каталог представлен символом ~.

Перемещение файлов с помощью команды *mv*

По команде mv, аналогичной команде move из DOS, файлы перемещаются из одного каталога в другой. Действие этой команды аналогично действию команды копирования с последующим удалением исходных файлов. Команда mv не создает копий файлов.

Синтаксис команды mv:

mv from-filename to-filename,

где from-filename — исходный файл; to-filename — новый файл.

По команде mv fredl |redl.old создается резервная копия файла fredl с именем fredl.old, затем удаляется исходный файл |redl. По команде же mv -fredl.old /backup/jack файл fredl.old перемещается из рабочего ката-уга в каталог /backup/jack.

Удаление файлов с помощью команды *rm*

Файлы в Linux удаляются по команде rm. Это опасная команда, потому что удаленный файл восстановить невозможно. Для безопасной работы следует пользоваться следующим форматом этой команды:

rm -i filename,

здесь filename — имя удаляемого файла; -i — параметр, указывающий на необходимость подтвердить удаление файла.

Например, по команде rm fredl файл fredl будет просто удален, по команде же rm -i il он будет удален только после подтверждения пользователем необходимости удаления.

Вывод содержимого файла с помощью команды *more*

По команде more на экран выводится содержимое текстового файла, при этом нет необходимости запускать текстовый редактор, распечатывать файл или нажимать клавишу паузы во время вывода текста на экран. Например, для вывода на экран содержимого конфигурационного файла emacs вводится команда:

more .emacs.

Недостаток этой команды в том, что невозможно пролистать информацию в обратном направлении. Но, этот недостаток устранен в других командах, которые мы рассмотрим ниже.

Команда /ess—усовершенствование more

По команде less информация выводится в окно терминала. Имя этой команде дано в противоположность команде more, поскольку в команде less пролистывание текстового файла возможно в обоих направлениях (игра слов: more — больше, less — меньше.*)*. Синтаксис команды less:

less файл

Команда поиска файлов *find*

Если вы не можете найти требуемый файл с помощью команды Is, воспользуйтесь командой find. Команда find исключительно полезная вещь, что делает ее одновременно одной из самых сложных в использовании команд. Использование команды find включает три этапа, которые в свою очередь могут состоять из одного или нескольких этапов.

• Где искать

• Что искать

• Что делать, когда файл найден

Если вы знаете имя файла, но не знаете, где он находится в структуре каталогов Linux, то самым простым способом использования команды find для поиска такого файла будет команда:

find / -name *filename*-print

Будьте осторожными при поиске от корня — в больших системах такой поиск может занять слишком много времени, так как будет просматриваться каждый каталог, каждый диск, включая подключенные сетевые диски.

Возможно, более приемлемым будет поиск по нескольким каталогам. Например, если вы знаете, что файл, вероятнее всего, находится в каталогах /usr или /usr2, воспользуйтесь следующей командой:

find /usr /usr2 -name *filename*-print

В команде find можно использовать множество различных параметров. Список параметров команды приведен в таблице.

Таблица. Параметры команды find

Команда	Описание
-name *file*	Параметр file может быть именем или шаблоном, содержащим символы подстановки. Если это шаблон, то для обработки выбирается каждый файл, чье имя удовлетворяет этому шаблону
-links *n*	Для обработки выбираются все файлы, на каждый из которых имеется n или больше ссылок

-size n [c] Для обработки выбираются все файлы, размер которых равен или больше n 512-байтных блоков. Если к размеру добавлен символ c, то выбираются файлы, которые состоит из n или больше символов

-atime *n* Для обработки выбираются все файлы, к которым осуществлялся доступ за последние n-дней. Обратите внимание, что сама команда find осуществляет доступ к файлам, поэтому изменяет время последнего доступа к файлу

-exec *cmd* Для каждого файла, удовлетворяющего критериям поиска, выполняется команда Linux, принимающая в качестве параметра имя найденного файла. Для использования команды -exec необходимо запомнить два простых правила: в команде имя найденного файла представляется {}, а команда должна заканчиваться последовательностью символов \;. Предположим, вы зарегистрировались как администратор и создали каталог, поэтому все файлы в этом каталоге принадлежат администратору. Чтобы сделать так, чтобы всеми файлами владел пользователь jack, необходимо выполнить команду:

find /home/jack -exec chown jack {} \;

~print Эта наиболее часто используемая команда просто отображает имена всех найденных файлов

Команда find позволяет выполнять множество логических операций. Например, если вы хотите выбрать все файлы, которые нельзя представить одним шаблоном, можно воспользоваться параметром *or*(-o):

find /home (-name *filel* *-o*-name*file2*) –print

Редактор vi

Редактор vi является мощным средством для создания и редактирования файлов. Он предназначен для работы на видеотерминале и использует окна, посредством которых можно просматривать текст файла. Несколько простых команд позволяют сделать изменения в тексте, которые очень быстро отображаются на экране. Редактор может отобразить одну или несколько строк текста. Он позволяет передвигать курсор в любую точку на экране или в файле, создавать, изменять или удалять текст. Можно также использовать некоторые команды построчного редактора, такие как глобальные команды, которые позволяют изменять множество появлений некоторой строки символов, используя одну команду. Чтобы двигаться по файлу, можно прокручивать текст вперед или назад.

Редактор vi имеет более 100 команд. В этом разделе описаны основные команды, которые позволят использовать vi просто, но эффективно.

1.Создание файла

Чтобы создать файл, введите:

vi filename<CR>

Когда ввести команду vi с именем файла, vi очистит экран и отобразит окно, в которое можно вводить и редактировать текст.

2. Режимы редактора

Редактор vi функционирует в двух режимах: режим ввода и командный режим. В режиме ввода можно добавлять и модифицировать текст; в командном режиме можно:

- редактировать и изменять существующий текст;
- удалять, перемещать и копировать текст;
- передвигаться по всему файлу;
- выполнять другие задачи.

2.1. Создание текста в режиме Append

Если вход в редактор vi сделан успешно, то мы сразу же попадаем в командный режим и vi ожидает ввода ваших команд. Чтобы создать текст необходимо:

ввести a, чтобы перейти к режиму ввода (не нажимая клавишу RETURN). Теперь можно добавить текст к файлу (файл не печатается на экране);

ввести нужный текст;

перейти на новую строку и нажать клавишу RETURN.

Если был указан циклический режим в файле .exrc, то будем переходить на новую строку всякий раз, когда достигнем точки автоматического возврата каретки.

2.2. Выход из режима Append

После окончания создания, нажать клавишу ESCAPE, чтобы из режима ввода перейти к командному режиму. Теперь можно редактировать текст, который был создан или записан
<a>

Create some text<CR>

in the screen editor<CR>

and return to<CR>

command mode.<ESC>

Если нажать клавишу ESCAPE и будет слишен звуковой сигнал, то это значит, что мы уже перешли в командный режим. Нажатие клавиши ESCAPE в то время, когда находимся в командном режиме, никак не влияет на текст в файле, даже если мы нажмем ее несколько раз.

3. Редактирование текста в командном режиме

В общем случае редактирование файла - это добавление, изменение, удаление текста. Чтобы выполнить эти задачи вы должны научиться переходить от одной страницы текста к другой странице, от одной строки к другой и между определенными точками на строке. В этом подразделе описаны команды добавления и удаления текста и способы перемещения курсора.

3.1. Перемещение курсора

Клавиши перемещения курсора позволяют вам перемещать курсор по файлу. Курсор можно переместить с помощью следующих клавиш: h - перемещает курсор на один символ влево; j - перемещает курсор вниз на одну строку; k - перемещает курсор вверх на одну строку; l - перемещает курсор на один символ вправо;

Команды j и k сохраняют позицию курсора на строке. Например, если курсор находился на 17-м символе слева, то когда вы введете команду j или k, то он переместится на новую строку, но будет находиться на 17-м символе. Если символов в новой строке меньше, то курсор переместится на последний символ строки.

Примечание. Некоторые терминалы имеют специальные символы управления курсором, помеченные стрелками. Эти стрелки можно использовать вместо клавиш h, j, k, l.

Если перед h, j, k, l поставить число, то курсор передвинется на количество строк/символов, равных числу. Например, если надо передвинуть курсор на 4 строки вверх, введите 4k. Если вы не можете продвинуться дальше в этом направлении, то vi подаст звуковой сигнал и курсор останется на прежнем месте.

Чтобы переместить курсор на 35 строк вниз, введите: 35j

Дополнительно к клавишам h и l можно использовать клавиши SPACEBAR и BACKSPACE для перемещения курсора вправо или влево. <SPACEBAR> - переместить курсор на один символ вправо; <nSPACEBAR> - переместить курсор на n символов вправо; <BACKSPACE> - переместить курсор на один символ влево; <nBACKSPACE> - переместить курсор на n символов влево;

3.2. Удаление текста

Если вы хотите удалить символ, переместите курсор к этому символу и нажмите x. Перед x может стоять число, которое говорит о том, сколько символов подряд нужно удалить. Редактор vi так корректирует строку, что на месте удаленного символа не остается пустого промежутка или на месте удаленной строки - пустой строки.

3.3. Добавление текста

В редакторе vi существует две команды для добавления текста: insert(i) и append(a). Чтобы добавить текст с помощью команды i, переместите курсор в нужное место с помощью клавишей h, j, k, l. Затем нажмите i и введите текст. Введенный вами текст появится на экране слева от символа, на который указывает курсор. Этот символ и все остальные справа от курсора будут передвинуты вправо, чтобы освободить место для вашего нового текста. Редактор vi будет хранить символы, вводимые вами, до тех пор, пока вы не нажмете клавишу ESCAPE. Если необходимо, передвинутые символы будут перенесены на следующую строку.

Таким же образом вы можете использовать команду a. Отличие состоит в том, что новый текст появится справа от символа, к которому подведен курсор.

4. Завершение работы с vi

Когда вы закончите работу с текстом, необходимо записать содержимое буфера в файл и возвратить управление shell. Для этого нажмите клавишу SHIFT и дважды нажмите Z(ZZ). Редактор сохранил имя того файла, который вы указали в начале сеанса редактирования, и в этот файл перепишет текст из буфера. Обратите внимание, что внизу экрана дается имя файла и количество строк и символов в файле. Затем shell выдает вам подсказку.

Пример.

<a>

This is a text file.<CR>

I am adding text to<CR>

a temporary buffer and<CR>

now it is perfect.<CR>

I want to write this file,<CR>

and return to the shell.<ESC><ZZ>

"stuff" [New file] 6 lines, 135 characters

$

Вы также можете использовать команды :w и :q редактора vi для записи файла и окончания работы с редактором. Команды редактора начинаются с двоеточия (:) и появляются в последней строке экрана. Команда :w записывает буфер в файл. Команда :q завершает работу редактора и возвращает управление shell. Вы можете ввести эти команды раздельно, или в одной команде :wq

5. Перемещение курсора по экрану

Ранее вы познакомились с тем, как перемещать курсор с помощью клавишей h, j, k, l, BACKSPACE и SPACEBAR. В этом подразделе приводятся еще несколько команд, которые помогут вам быстро перемещать курсор по экрану. Эти команды выполняются в командном режиме.

5.1. Позиционирование курсора на символ

Для позиционирования курсора на символ в строке существуют три способа:

переместить курсор вправо или влево до нужного символа;

переместить курсор в начало или конец строки;

поиск символа на строке.

Переместить курсор на первый или последний символ в строке можно следующим образом:

$ - переместить курсор на последний символ в строке; 0(ноль) - переместить курсор на первый символ в строке; ^ - переместить курсор на первый ненулевой символ в строке.

Третий способ позиционирования курсора в строке - это поиск указанного символа на текущей строке. Если символ не найден на текущей строке, то появляется звуковой сигнал и курсор остается на прежнем месте. Существуют команды, которые проводят поиск по шаблону. Они будут описаны далее. Здесь приведем 6 команд для поиска внутри строки: f, F, t, T, ;(двоеточие) и ,(запятая). После f, F, t и T вы можете указать символ:

fx - осуществляет поиск указанного символа x справа от текущей позиции курсора;

Fx - осуществляет поиск указанного символа x слева от текущей позиции курсора;

tx - осуществляет перемещение курсора вправо на символ до указанного символа x;

Tx - осуществляет перемещение курсора влево на символ до указанного символа x;

; - повторяет предыдущий поиск символа. Команда ; запоминает символ и ищет следующее появление символа в текущей строке;

, - повторяет предыдущий поиск символа, но в обратном направлении. Команда ; запоминает символ и ищет следующее появление символа в текущей строке;

Команда - (минус) перемещает курсор на одну строку вверх, позиционируя его на первом ненулевом символе. Чтобы переместить курсор более чем на один символ, укажите пред знаком минус количество строк. Например, чтобы переместить курсор вверх на 13 строк, введите: 13-

Если эти строки находятся выше отраженного экрана, то окно переместится так, чтобы открыть их. Этот способ помогает быстро перемещаться по файлу. Если выше окажется меньше 13-ти строк, то звуковой сигнал подскажет вам, что вы сделали ошибку и курсор останется в своем прежнем положении.

Команда + (плюс) или <CR> перемещает курсор вниз на одну строку и позиционирует на первый непустой символ. Перед командой + можно указать количество строк, на которое вы хотите переместить курсор. Например, чтобы переместить курсор на 9 строк вниз, введите: 9+

Если некоторые строки расположены ниже текущего экрана, то окно переместится вниз, чтобы открыть их.

5.2. Позиционирование слов

Редактор vi предполагает, что слово - это строка символов, которая может включать в себя буквы, цифры и символы подчеркивания. В редакторе vi существует 6 команд позиционирования: w, b, e, W, B и E. Команды w, b, e рассматривают любой символ, отличный от буквы, цифры или подчеркивания, как разделитель, означающий начало или конец слова. Пунктуация до или после пустого пространства является разделителем слов. Символ начала или конца строки также является разделителем.

Команды W, B, E рассматривают пунктуацию как часть слова. Слова разделяются пустым пространством, которое состоит из символов пробел, табуляция и новая строка.

Команды позиционирования:

w - перемещает курсор к началу следующего слова;

nw - перемещает курсор к началу n-го слова; конец строки не останавливает движение курсора. Он продолжает подсчет слов с начала следующей строки;

W - игнорирует все знаки пунктуации и перемещает курсор на следующее после пробела слово;

e - перемещает курсор на последний символ в следующем слове;

E - перемещает курсор на последний символ в слове. Игнорирует все символы пунктуации за исключением пробела. Слова разделяются пробелами;

b - перемещает курсор к первому символу предыдущего слова;

nb - перемещает курсор к первому символу n-го предыдущего слова. Команда b не останавливается в начале строки и продолжает движение к началу строки;

B - используется подобно команде b, за исключением того, что слова разделяются только символами пробел и новая строка. Она рассматривает все другие символы пунктуации как буквы.

5.3. Позиционирование курсора на предложение

Редактор vi также распознает предложения. В редакторе vi предложение заканчивается знаками !, . или ?. Если эти разделители появляются в середине строки, то за ними должны стоять два пробела, чтобы vi их мог распознать.

Вы можете перемещать курсор от предложения к предложению в файле с помощью команд (- открывающая скобка и) - закрывающая скобка.

(- перемещает курсор в начало текущего предложения;

n(- перемещает курсор в начало n-го предложения выше текущего предложения;

) - перемещает курсор в начало следующего предложения;

n) - перемещает курсор в начало n-го предложения, следующего за текущим предложением.

5.4. Позиционирование курсора по параграфам

Параграфы распознаются редактором vi, если они начинаются после пустой строки. Если вы хотите перемещать курсор по параграфам, то при создании текста не забудьте в конце каждого параграфа поставить пустую строку.

{ - переместить курсор в начало текущего параграфа, который ограничивается пустой строкой;

n{ - переместить курсор в начало n-го параграфа, следующего за текущим параграфом;

} - переместить курсор в начало следующего параграфа;

n} - переместить курсор в начало n-го параграфа ниже текущей строки.

5.5. Позиционирование в окне

Редактор vi предоставляет вам 3 команды, которые позволяют вам позиционировать курсор в окне. Эти команды должны быть введены на верхнем регистре:

H - переместить курсор к первой строке экрана; M - переместить курсор к средней строке экрана; L - переместить курсор к последней строке экрана.

6. Позиционирование курсора на неотображаемом тексте

В этом подразделе описывается, как переместить курсор на текст, который не отображен на текущем экране. Вы можете сделать это следующим образом:

прокрутка вперед и назад файла;

перейти к указанной строке в файле;

поиск по шаблону в файле.

Четыре команды редактора vi позволяют вам осуществлять прокрутку текста файла. Команды ^f (<CTRL>f) и ^d (<CTRL>d) проводят прокрутку экрана вперед, а команды ^b (<CTRL>b) и ^u (<CTRL>u) - прокрутку экрана назад.

Команда ^f осуществляет прокрутку вперед на одно окно текста, расположенного ниже текущего окна. Редактор vi очищает экран и выводит новое окно. Две последние строки текущего окна отображаются в верхней части нового окна. Если окно дополнено пустыми строками, то они помечаются знаком ~ (тильда)

Команда ^d осуществляет прокрутку на половину экрана, чтобы отобразить текст, расположенный ниже окна. Если курсор находится на последней строке файла и вы нажмете ^d, то vi подаст звуковой сигнал.

Команда ^b осуществляет прокрутку экрана назад на полное окно, чтобы отобразить текст, который находится выше текущего экрана. Редактор vi очищает экран и отображает новое окно. В отличие от команды ^f, команда ^b не оставляет в новом окне строки из текущего окна.

Команда ^u осуществляет прокрутку половины экрана, чтобы отобразить текст, расположенный выше окна. Если курсор достигнет начала файла, то vi подаст звуковой сигнал.

7. Переход на указанную строку

Команда G позиционирует курсор на указанную строку в окне. Если эта строка не находится в текущем экране, то G очищает экран и отображает окно, содержащее эту строку. Если не указана строка, то команда G перемещает курсор на последнюю строку файла: G - переместить курсор на последнюю строку в файле; nG - переместить курсор на строку n.

8. Номера строк

Каждая строка файла имеет номер, соответствующий ее положению в буфере. Чтобы получить номер конкретной строки, переместите курсор на эту строку и введите команду: ^g

В нижней части экрана команда ^g поместит информацию:

- имя файла;
- был ли буфер модифицирован после поледней записи его в файл;
- номер строки, на которой находится курсор;
- общее количество строк в буфере.

9. Создание текста

Три основные команды позволяют в редакторе vi создавать текст: a - добавить текст; i - вставить текст; o - открыть новую строку, на которой может быть введен текст.

После того, как вы создали текст с помощью одной из этих команд, можете возвратиться в командный режим редактора vi нажав клавишу ESCAPE. Чтобы в файл добавить текст, воспользуйтесь следующими командами: a - добавить текст после курсора; A - добавить текст в конце текущей строки.

Чтобы в файл вставить текст, воспользуйтесь следующими командами: i - вставить текст до курсора; I - вставить текст в начале текущей строки до первого ненулевого символа.

Для возврата в командный режим, нажмите клавишу ESCAPE.

Для открытия строки для текста воспользуйтесь следующими командами: o - создать текст с начала новой строки, расположенной ниже текущей строки. Вы можете ввести эту команду в любом месте текущей строки; O - создать текст с начала новой строки, расположенной выше текущей строки. Вы можете ввести эту команду в любом месте текущей строки.

Команды открытия создают строку непосредственно выше или ниже текущей строки и переводят вас в режим ввода текста. Курсор располагается в начале новой строки и ожидает от вас ввода текста.

10. Удаление текста

Вы можете удалить текст с помощью нескольких команд в командном режиме и небольшой объем в режиме ввода.

10.1. Удаление текста в режиме ввода текста

Если вы находитесь в режиме ввода текста редактора vi и хотите удалить символ, воспользуйтесь клавишей BACKSPACE: BACKSPACE - удалить текущий символ (слева от курсора).

Клавиша BACKSPACE перемещает курсор назад в режиме ввода текста и удаляет каждый символ, который курсор переходит. Однако удаленный символ не стирается с экрана пока вы не нажмете клавишу ESCAPE для возврата в командный режим.

Удалить текст вы можете и с помощью следующих ключей: ^w - отменить ввод текущего слова; @ - удалить на текущей строке весь введенный во время данного режима ввода текст; ^v - удалить специальное значение, если есть, следующих вводимых символов.

Когда вы вводите ^w, курсор возвращается на последнее введенное слово и ожидает на первом символе этого слова. Он не стирает слово с экрана до тех пор, пока вы не нажмете клавишу ESCAPE или не введете новые символы на месте этого слова.

10.2. Отменить последнюю команду

Отменить последнюю введенную команду можно с помощью следующих команд: u - отменить последнюю команду; U - восстановить текущую строку в состояние до произведенных изменений.

Если вы случайно удалили строки, то для отмены команды удаления введите команду u. Удаленные строки вновь появятся на экране. Если вы ввели неправильно команду, то введите команду u и неверная команда будет отменена. Команда U будет аннулировать все изменения в текущей строке до тех пор, пока курсор будет находиться на этой строке.

Если вы введете команду u дважды, то вторая команда отменит первую. Например, если вы удалили по ошибке строку и восстановили ее с помощью команды u, то нажав второй раз u, вы вновь удалите строку. Знание этих команд может спасти вас от многих неприятностей.

11. Команды удаления в командном режиме

Многие команды редактора vi, такие как команды удаления и изменения, предоставляют вам возможность использовать в командной строке команды перемещения курсора. Команды перемещения курсора могут указывать на объект текста: слово, строка, предложение, параграф. В этом случае общий формат команды будет такой:

[number][command]text_object

Квадратные скобки указывают, что данные компоненты не являются обязательными.

Все команды удаления, выдаваемые в командном режиме сразу же удаляют указанный текст с экрана и оставшуюся часть отображают на экране в обновленном виде.

Команда удаления имеет следующий формат:

[number]dtext_object

где d - команда удаления; text_object - объект удаления.

12. Замена текста

rx - Заменить текущий символ (символ, на который показывает курсор) на x. Эта команда не устанавливает режим ввода текста и, следовательно, нет необходимости нажимать после нее клавишу ESCAPE.

nrx - Заменить n символов на x. После этой команды также не надо нажимать клавишу ESCAPE.

R - Заменяет только те символы, которые введены в режиме ESCAPE. Если будет достигнут конец строки, то эта команда добавит ввод как новый текст.

Например, вы хотите заменить слово acts на ants в следующем предложении:

The circus has many acts.

Поместите курсор под буквой c в слове acts и введите команду:

rn

Предложение будет иметь вид:

The circus has many ants.

Чтобы заменить слово many на 7777, поместите курсор под буквой m в слове many и введите команду:

4r7

Команда r заменит слово many на четыре цифры 7:

The circus has 7777 ants.

Задание к лабораторной работе

1. Вывести оглавление домашнего (рабочего) каталога, предназначенного для пользователей (**/web_server**).
2. Создать в текущем каталоге каталог с именем **name**, где **name**- ваша фамилия маленькими латинскими буквами.
3. Перейти в созданный каталог.
4. Создать каталог **MLKX** в текущем каталоге.
5. Скопировать в каталог **MLKX** файл **xyz.text** из каталога**/works/linux_lab**.
6. Вывести файл **xyz.text** на экран.
7. Создать каталог с именем **FFW**.
8. Скопировать файл **xyz.text** в каталог **FFW**с именем**xyz1.text.**
9. Перейти в каталог **FFW** и просмотреть содержимое файла**xyz1.text**.
10. Переименовать файл **xyz1.text** в файл с именем**xyz2.text**.
11. Объединить файлы **xyz.text**и**xyz2.text** в файл с именем**xyz3.text** в каталоге**FFW**.
12. Просмотреть файл **xyz3.text**.
13. *Продемонстрировать содержимое файла преподавателю*.

14. Произвести поиск файла **xyz.text** с помощью команды **find**.
15. Удалить файлы **xyz.text, xyz2.text и xyz3.text.**
16. Удалить каталог **FFW**.
17. Удалить каталог **MLKX**.
18. Удалить каталог **name**.
19. Запустите оболочку bash. Создайте в любом каталоге текстовый файл text1. Команда **(# vi file1.txt)**Напечатайте в первой строчке ФИО, группу, в следующих строчках текст может быть произвольным. Опробуйте основные команды текстового редактора vi.

Контрольные вопросы

1. Записать команду просмотра содержания текущего каталога.
2. Записать команду для создания подкаталога XX в домашнем (рабочем) каталоге.
3. Как удалить каталог ?
4. Записать команды для удаления файла, переименования файла, копирования файла.
5. Как вывести содержимое файла на экран ?
6. Как создать файл в текстовом редакторе Vi?
7. Возможности редактора Vi в командном режиме.
8. Создание текста в режиме Append
9. Редактирование текста в командном режиме. Удаление и добавление текста.
10. Команды позиционирования
11. Как перейти на указанную строку ?
12. Номера строк. Какую информацию команда ^g поместит нижней части экрана ?
13. Завершение работы с vi
14. Как просмотреть содержание созданного файла ?
15. Какой командой осуществляется поиск файла и вывод на экран строк, содержащих заданный текст?

Лабораторная работа № 5.

тема: Работа с Midnight Commander в Linux.

Информация

Двухпанельные файловые менеджеры (ДПФМ или сленговое - двухпанельники) являются одним из основных инструментов системного администратора. Они предназначены для полного комплекса маниуляций с файлами, начиная от поиска и заканчивая редактированием. Современные ДПФМ позволяют расширять свои функциональные возможности за счет плагинов и виртуальных файловых систем, что позволяет им работать с любой информацией, файлами, программами и устройствами, которые могут быть описаны иерархическими или списковыми структурами.

Среди наиболее распространенных ДПФМ для операционной системы Linux следует выделить программы Midnight Commander, сокращенно - MC (http://www.midnight-

commander.org/) и Double Commander (http://doublecmd.sourceforge.net/). Для системных администраторов наиболее привлекательным продуктом является MC, так как, в отличие от Double Commander, является консольной программой, что позволяет его использовать при прямом и удаленном соединении на серверах без поддержки графического пользовательского интерфейса. С учетом широкой распространенности таких серверных систем, MC, фактически, является безальтернативным ДПФМ для системных и сетевых администраторов.

Задание

Сдача лабораторной работы подразумевает последовательное выполнение операций в файловом менеджере MC, решающих поставленные задачи. Контекстная справка доступна по нажатию F1. Разрешается задавать вопросы преподавателю, а также пользоваться глобальной сетью для поиска необходимой информации и консультаций со специалистами.

1. В левой панели создать цепочку каталогов /tmp/info/data/. В правой панели, в каталоге /usr/share/doc/ выделить одним действием все каталоги, начинающиеся на буквы "n" и "N".Затем отдельными действиями выделить в каталоге /usr/share/doc/ каталоги, начинающиеся на букву "z". Полученный результат скопировать в каталог /tmp/info/data/.
2. Определить размер каталога /tmp/info/data/. Выделить одним действием все каталоги в /tmp/info/data/ и определить размер каждого из них. Получить информацию о каталоге /tmp, найти в этой информации максимальный объем каталога /tmp.
3. Найти все файлы с расширением jpg в каталоге /usr/share/doc/ и ниже. Скопировать полученный набор файлов в каталог /tmp/info/data/. При копировании файлы с одинаковыми именами перезаписывать (если такие файлы будут найдены).
4. Отсортировать файлы в каталоге /tmp/info/data/ по размеру в прямом и обратном порядке, по времени правки в прямом и обратном порядке. Произвести изменения содержимого /tmp/info/data/ для демонстрации корректности каждой сортировки.
5. Управление правами доступа:
    - Запретить себе вход в каталог /tmp/info/data/, проверить, снова разрешить, проверить.
    - Запретить себе удаление файлов и каталогов из каталога /tmp/info/data/, проверить, снова разрешить, проверить.
    - Найти файл README в любом из подкаталогов каталога /tmp/info/data/, запретить его правку, проверить, снова разрешить, проверить, запретить чтение содержимого, проверить, снова разрешить, проверить.
6. В каталоге /tmp/info/ создать символическую ссылку subdata, связанного с каталогом data. Убедиться, что содержимое каталогов data и subdata идентично и при изменении содержимого одного каталога изменяется и содержимое другого. Определить, что каталог subdata является ссылкой на каталог data. Скопировать каталог data в data1. Удалить каталог data, убедиться, что каталог subdata стал недоступным. Переименовать каталог data1 в data.
7. Найти файл README в любом из подкаталогов каталога /tmp/info/data/, создать в каталоге /tmp/info/ символическую ссылку readme.symbol, связанную с найденным файлом и жесткую ссылку readme.hard, также связанную с найденным файлом. Убедиться, что правка любого из трех файлов приводит к синхронным изменениям в двух других. Убедиться, что при удалении исходного файла README ссылка readme.symbol становится недоступной, а readme.hard продолжает функционировать.

8. Создать новый текстовый файл /tmp/info/code.txt в редакторе MC и ввести фразу "студент 182 группы, н_студ_билета" в кодировке utf-8. В последующих 3-х строках написать эту-же фразу в кодировках cp-1251, cp-866 и koi8-r. Таким образом, при переключении на любую из 4-х кодировок, только одна из строк должна отображаться корректно.
   Найти один из своих файлов, подготовленный в ЛР№1 и включающий в себя рускоязычный текст (разделы с ОС Windows монтируются обычно в каталог /mnt). Если раздел с ОС Windows не подключен, то использовать файл с флеш-памяти или с ftp-сервера. Используя редактор или просмотрщик MC, определить кодировку, в которой подготовлен указанный файл.
9. Загрузить с помощью браузера файл электронной фотографии, например отсюда: http://shgpi.edu.ru/fileadmin/contests/gallery_foto_8/ и сохранить его в каталоге /tmp/info/. Используя один из графических редакторов или просмотрщиков, имеющихся в ОС, убедиться что фотография содержит в себе данные о фотокамере, времени и условиях сьемки. Используя шестнадцатиричный редактор MC, исправить год сьемки на предыдущий. Результат проверить в одном из графических редакторов или просмотрщиков, имеющихся в ОС.
10. Подключиться к ftp-серверу ftp.vc.shgpi, найти в каталоге /doc сервера все файлы с расширением jpg и скопировать их в каталог /tmp/info/data/, при копировании файлы с одинаковыми именами перезаписывать (если такие файлы будут найдены). Создать каталог /tmp/info/imgarchives/. Выделить все файлы (без каталогов) в каталоге /tmp/info/data/ и **перенести** их в в каталог /tmp/info/imgarchives/. Выполнить архивацию каталога /tmp/info/imgarchives/ в файл /tmp/info/imgarchives.tar.gz. Подключиться к ftp-серверу ftp.data.shgpi, в каталоге uploads создать подкаталог ном_студ_бил и скопировать туда файл /tmp/info/imgarchives.tar.gz.

Лабораторная работа № 6

Тема: Создание учетной записи в ос Windows XP

1.1. Целью лабораторной работы является получение навыков по созданию и управление учетными записями пользователей средствами защищенной операционной системы Windows-XP.

1.2. В результате выполнения лабораторной работы должны быть приобретены знания:

- назначения и возможностей использования учетных записей пользователей;
- принципа применения групп пользователей.

1.3. В процессе выполнения лабораторной работы необходимо овладеть навыками:

- управления учетной записью «гость»;
- создания и управления учетными записями пользователей средствами ОС Windows-XP;
- создания и управления группами пользователей.

1.4. Используемое оборудование и программное обеспечение - ПЭВМ IBM PC, операционная система Windows-XP.

2. Основные теоретические сведения

2.1. Изучите назначение и использование рабочих групп и доменов.

Windows XP Professional поддерживает два сетевых окружения, в которых пользователи могут совместно использовать общие ресурсы независимо от размера сети: рабочие группы и домены.

Рабочие группы

Рабочая группа (workgroup) Windows XP Professional — логическое объединение сетевых компьютеров, которые совместно используют такие общие ресурсы, как файлы и принтеры. Рабочую группу также называют одноранговой сетью (peer-to-peer network), потому что все компьютеры в ней могут использовать общие ресурсы на равных условиях, т. е. без выделенного сервера.
Каждый компьютер в рабочей группе обслуживает *базу данных политики безопасности локального компьютера* (local security database). Эта база данных представляет собой перечень учетных записей пользователя и информации о правах доступа к ресурсам на компьютере, где она постоянно находится. Поэтому использование базы данных политики безопасности локального компьютера децентрализует администрирование учетных записей пользователей и политики доступа к ресурсам в рабочей группе рис. 1.1.

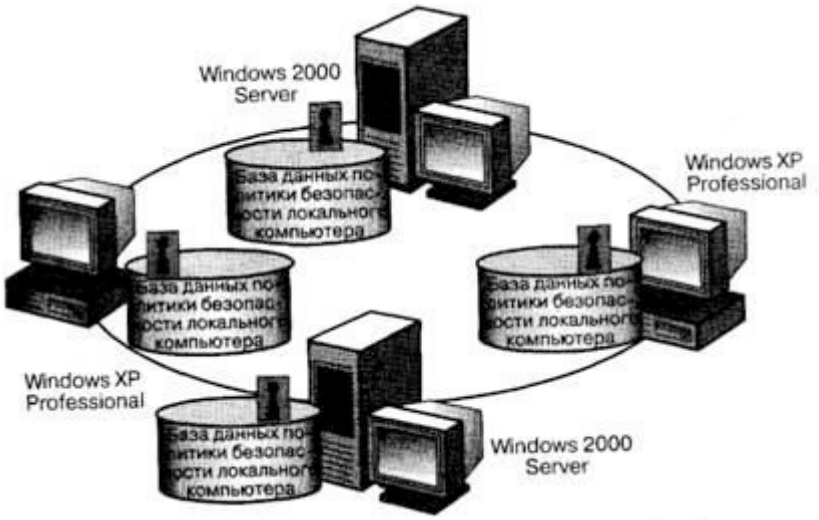


Рис. 1.1. Пример рабочей группы Windows XP Professional

Поскольку рабочие группы децентрализуют администрирование и политику доступа к ресурсам, истинны следующие утверждения:
• пользователь должен иметь свою учетную запись на каждом компьютере, к которому он хочет получить доступ;
• любое изменение учетных записей пользователя, например замена его пароля или создание новой учетной записи, необходимо выполнить на каждом компьютере рабочей группы. Если вы забудете зарегистрировать новую учетную запись на одном из компьютеров вашей рабочей группы, то новый пользователь не сможет получить доступ к этому компьютеру и его ресурсам.
Рабочая группа имеет следующие преимущества:
• она не требует включения в сеть контроллера домена для хранения централизованной информации о политиках безопасности;
• она проста в проектировании и эксплуатации. В отличие от домена, не требует крупномасштабного планирования и администрирования;

• это удобная сетевая среда для небольшого числа компьютеров, расположенных не слишком далеко друг от друга. Однако организация рабочей группы нецелесообразна в сетях, содержащих более 10 компьютеров.

Домены

Домен (domain) - это логическое объединение компьютеров в сети, которые совместно используют центральную базу данных каталога (рис 2.2) *База данных каталога* (directory database) содержит учетные записи пользователя и информацию о политиках безопасность для домена. Эту базу данных называют каталогом, и она представляет собой часть базы данных службы Active Directory — службы каталогов Windows 2000.
В домене каталог размещен на компьютерах — контроллерах домена. *Контроллер домена* (domain controller) — это сервер, который координирует все параметры безопасности при взаимодействии пользователя и домена и централизует администрирование и управление политиками безопасности.

Примечание Вы можете назначить контроллером домена только компьютер на одной из платформ Microsoft серии Windows 2000 Server. Если все компьютеры в сети работают на платформе Windows XP Professional, то единственным доступным видом сети будет рабочая группа.

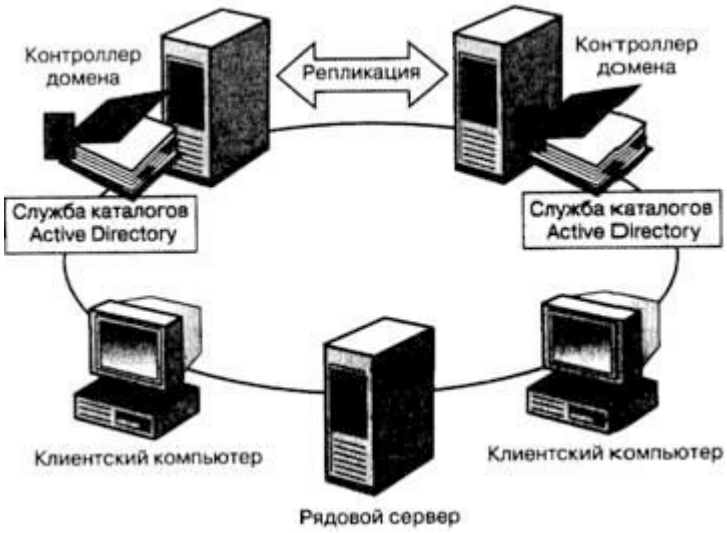


Рис. 2.2. Домен Windows 2000

Домен не имеет отношения к местоположению в сети или определенному типу сетевой конфигурации. Компьютеры в домене могут располагаться рядом в небольшой локальной сети (LAN) или находиться в различных уголках мира. Они могут связываться друг с другом по любому физическому соединению, включая телефонные линии, линии ISDN, оптоволоконные линии, линии Ethernet, кольцевые сети с маркерным доступом (token ring), подключение с ретрансляцией кадров (frame relay), спутниковую связь и выделенные линии. Достоинства домена очевидны:
• централизованное администрирование, потому что вся информация о пользователях хранится в одном месте;
• однократная регистрация пользователя для получения доступа ко всем сетевым ресурсам (файлам, принтерам и программам) при наличии требуемых прав доступа. Другими словами, вы можете зарегистрироваться на одном компьютере сети и использовать ресурсы другого компьютера при условии, что вы имеете соответствующие разрешения на

доступ;
• масштабируемость, что позволяет создавать очень большие сети. Типы компьютеров, которые включает типичный домен Windows 2000, перечислены далее.
Контроллеры домена на платформе Windows 2000 Server. Каждым контроллер домена хранит и обслуживает копию каталога. В домене вы создаете единственную учетную запись пользователя, которую Windows 2000 записывает в каталог. Когда пользователь входит в систему на компьютере домена, контроллер домена аутентифицирует пользователя, проверяя в каталоге его учетную запись, пароль и ограничения на вход в систему. Если в домене есть несколько контроллеров домена, то они периодически обмениваются данными своих копий каталога.
Рядовые серверы на платформе Windows 2000 Server. *Рядовой сервер* (member server) — это сервер, не имеющий статуса контроллера в конкретном домене. Рядовой сервер не ведет каталог и не способен аутентифицировать пользователей. Рядовые серверы обеспечивают совместный доступ к сетевым ресурсам, например к общим папкам или принтерам.
Клиентские компьютеры на платформе Windows XP Professional, Windows 200O Professional или любой другой операционной системе Microsoft. Клиентские компьютеры — это настольные системы пользователей, которые предоставляют пользователям доступ к ресурсам домена.

2.2. Изучите правила и способы создания и управления учетными записями пользователей

Учетные записи пользователей

В Windows XP Professional используются три типа учетных записей пользователей: локальные учетные записи пользователей, учетные записи пользователей домена и встроенные учетные записи пользователей.
• *Локальная учетная запись пользователя* (local user account) позволяет начать сеанс на компьютере и воспользоваться его ресурсами.
• *Учетная запись пользователя домена* (domain user account) позволяет войти в домен и воспользоваться сетевыми ресурсами.
• *Встроенная учетная запись пользователя* (built-in user account) позволяет выполнить административные задачи и воспользоваться локальными или сетевыми ресурсами.

Локальные учетные записи пользователей

Локальные учетные записи пользователей позволяют начать сеанс на компьютере, на котором они были созданы, и воспользоваться ресурсами исключительно этого компьютера. Когда вы создаете локальную учетную запись пользователя, Windows XP Professional создает учетную запись в базе данных политик безопасности лишь этого компьютера. Ее называют базой данных политик безопасности локального компьютера (рис. 2.3). Windows XP Professional использует базу данных политик безопасности локального компьютера для аутентификации локальной учетной записи пользователя, что позволяет пользователю войти в систему на этом компьютере. Windows XP Professional не копирует локальную информацию об учетных записях пользователя на какой-либо другой компьютер.
Если вы создаете локальную учетную запись пользователя в рабочей группе из пяти компьютеров на платформе Windows XP Professional, например Userl на Computerl, то, используя учетную запись Userl, можно войти в систему только на Computerl. Если вам нужен доступ под именем Userl ко всем пяти компьютерам этой рабочей группы, то следует создать локальную учетную запись Userl на всех пяти компьютерах. Кроме того, когда вы решите изменить пароль для Userl, то это придется сделать на всех пяти

компьютерах, потому что каждый компьютер имеет свою собственную базу данных локальных политик безопасности.

Локальные учетные записи пользователей
- Предоставляют доступ к ресурсам на локальном компьютере
- Создаются исключительно на не входящих в домен компьютерах
- Создаются в базе данных политик безопасности локального компьютера

Рис. 2.3. Характеристики локальных учетных записей пользователей

Примечание Домен не распознает локальные учетные записи Пользователей, поэтому не следует создавать локальные учетные записи пользователей в системах на платформе Windows XP Professional, которые являются частью домена. В противном случае вам не удастся воспользоваться ресурсами домена, а администратор домена не смогут управлять локальными учетными записями или предоставить разрешения на доступ к ресурсам домена.

Учетные записи пользователей домена позволяют войти в домен и воспользоваться сетевыми ресурсами. При входе в систему вы предоставляете такие сведения о себе, как имя пользователя и пароль. Windows 2000 Server использует эту информацию для аутентификации пользователя и предоставления маркера доступа, который содержит ваши данные и параметры безопасности. Менеджер доступа идентифицирует вас для компьютеров в домене, в котором вы хотите воспользоваться сетевыми ресурсами. Маркер доступа действителен до конца сеанса пользователя.
Вы можете создать учетную запись пользователя домена в копии каталога Active Directory на контроллере домена, как показано на рис. 2.4. Контроллер домена копирует информацию о новой учетной записи на все контроллеры домена в домене. После того как Windows 2000 Server скопирует информацию о новой учетной записи пользователя, все контроллеры домена в дереве домена смогут аутентифицировать пользователя во время входа в систему.

Учетные записи пользователей домена
- Предоставляют доступ к сетевым ресурсам
- Предоставляют маркер доступа для аутентификации
- Создаются в службах каталогов Active Directory на контроллере домена

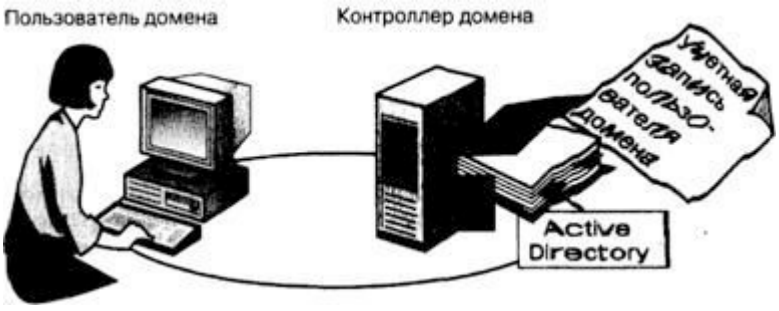


Рис. 2.4. Учетные записи пользователей домена

Встроенные учетные записи пользователей
Windows XP Professional автоматически создает *встроенные учетные записи*. Две стандартные встроенные учетные записи — администратор и гость.

Учетная запись администратора

Встроенная учетная запись администратора используется для полного управления компьютером. Она позволяет выполнять задачи по созданию и изменению учетных записей пользователей и групп, управлять политиками безопасности, создавать ресурсы печати и предоставлять пользовательским учетным записям разрешения и права на доступ к ресурсам.
Будучи администратором, вы можете создать учетную запись пользователя для выполнения неадминистративных задач и работать под учетной записью администратора только при выполнении административных задач.

Примечание Учетную запись администратора удалить нельзя. Рекомендуется переименовывать встроенную учетную запись администратора для обеспечения наибольшей безопасности. Всегда используйте имя, отличающееся от Administrator, что затруднит несанкционированный доступ в систему.

Гостевая учетная запись

Встроенную учетную запись гостя следует использовать для предоставления прав на вход в систему и доступ к ресурсам случайным пользователям. Например, сотруднику, которому нужен доступ к ресурсам на короткое время.

Примечание Гостевой доступ следует использовать только в сетях с низким уровнем безопасности. Всегда назначайте пароль для этой учетной записи. Учетную запись гостя можно переименовать, но не удалить.

Планирование новых учетных записей пользователей

Для упрощения процесса создания учетных записей пользователей планируют и организуют информацию об учетных записях пользователей, используя соглашения о назначении имен и требования к паролю.
Соглашения о назначении имен — установленный корпоративный стандарт для идентификации пользователей в домене. Последовательная политика назначения имен помогает администраторам и пользователям держать в памяти имена учетных записей. Это также упрощает администраторам систематизирование учетных записей при добавлении их в группы или администрировании. В таблице 1.1 перечислены некоторые правила назначения имен вашей организации.
Требования к паролю
В целях безопасности каждая учетная запись пользователя должна иметь пароль.
- Во избежание несанкционированного доступа всегда назначайте пароль для учетной записи администратора.
- Определитесь, кто имеет право изменять пароли: администратор или пользователи. Вы можете назначить уникальные пароли учетным записям пользователей самостоятельно и запретить пользователям их изменять или позволить пользователям задать свои собственные пароли во время первого сеанса. В большинстве случаев пользователям следует предоставить возможность изменить свой пароль.

• Используйте пароли, которые трудно угадать. Избегайте паролей содержащих, например, имена членов семьи.
• Пароли могут содержать до 128 символов; минимальная рекомендуемая длина — 8 символов.
• Используйте символы как верхнего, так и нижнего регистра (в отличие от имен пользователей, пароли чувствительны к регистру), цифры и прочие допустимые символы.

Соглашения о назначении имен Таблица 1.1

Рекомендация	Пояснение
Создайте уникальные имена учетных записей	Локальные имена учетных записей пользователей должны быть уникальны в системах, где создаются. Имена учетных записей пользователей домена должны быть уникальны в каталоге домена
Используйте не более 20 символов верхнего или нижнего регистра.	Имена учетных записей могут содержать до 20 символов. В поле разрешается ввести более 20 символов, но Windows XP Professional использует только первые 20.
Не забывайте, что имена учетных записей нечувствительны к регистру.	Для создания уникальных учетных записей пользователей можно использовать комбинацию специальных и алфавитно-цифровых символов. Имена учетных записей нечувствительны к регистру, однако Windows XP Professional сохраняет регистр в целях визуального восприятия
Не используйте недопустимые символы.	Следующие символы недопустимы: " / \ [ ]: ; | = , + *?<>
Придумайте, как можно не мешать двух пользователей со схожими именами.	Например, можно создать имя, состоящее из имени, второго инициала и других букв фамилии. Одного из двух пользователей по имени John Evans можно назвать johne, а другого johnev. Также можно задать для каждого номер, например johnel и johne2
Определите категорию служащего.	Некоторые организации предпочитают отразить статус временных служащих в их учетных записях. Например, перед именем добавить T и дефис (T-johne) или использовать скобки в конце имени — johne (Temp)
Переименуйте встроенные учетные записи Администратора и гостя.	Это следует сделать в целях безопасности

Создание, изменение и удаление учетных записей пользователей

В Windows XP Professional изменять и удалять учетные записи пользователей разрешается двумя способами: с помощью категории панели управления ***Учетные записи пользователей*** и оснастки ***Управление компьютером***.

Категория Учетные записи пользователей

Категория панели управления ***Учетные записи пользователей*** (User Accounts) (рис. 2.5) — инструментальное средство позволяющее создавать, изменять и удалять локальные учетные записи пользователей.

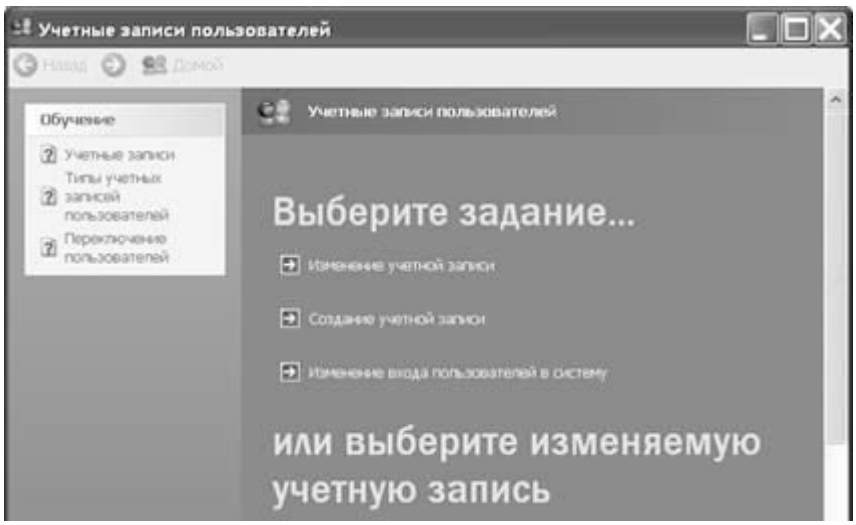


Рис. 2.5. Категория Учетные записи пользователей

Пользователь из группы администраторов может выполнять следующие задачи в разделе ***Выберите задание*** (Pick A Task) категории ***Учетные записи пользователей*** (User Accounts):

- изменить учетную запись (включая удаление учетной записи);
- создать новую учетную запись пользователя;
- изменить способ входа в систему.

Изменение учетной записи

Если вы имеете права администратора, то в разделе ***Изменение учетной записи*** можете изменить любую учетную запись пользователя на компьютере. Когда вы входите в систему под учетной записью пользователя, состав группы ***Выберите задание*** видоизменяется — в ней находятся лишь некоторые из команд, доступных администратору.

- Изменение имени. Изменяет имя учетной записи пользователя. Вы увидите эту команду, только если войдете в систему с правами администратора, потому что только администратору разрешено вы поднять эту задачу.
- Создание пароля. Создает пароль для вашей учетной записи. Вы увидите эту команду, только если ваша учетная запись пользователя не имеет пароля. Только администратору разрешено создавать пароли для других учетных записей пользователей.
- Изменение пароля. Изменяет пароль вашей учетной записи. Вы увидите эту команду вместо опции Создание пароля, только если для вашей учетной записи пользователя уже назначен пароль. Лишь администратор имеет право изменять пароли других учетных записей пользователей.
- Удаление пароля. Удаляет пароль вашей учетной записи или любой другой учетной записи на компьютере. Вы увидите эту команду, только если для вашей учетной записи пользователя уже назначен пароль. Только администратор имеет право удалять пароли других учетных записей пользователей.
- Изменение изображения. Изменяет изображение, которое появляется на экране приветствия. Только администратор имеет право изменять изображение других учетных записей пользователей.
- Изменение типа учетной записи. Изменяет тип указанной учетной записи. Только администратор имеет право изменять тип учетной записи пользователя.
- Использовать паспорт .NET. Запускает ***Мастер паспорта .NET*** (Add A .NET Passport To Your Windows XP Professional Account Wizard). Паспорт позволяет проводить интерактивные сеансы связи с семьей и друзьями, создавать собственные личные Web-

страницы и регистрироваться в любых службах и на сайтах с поддержкой .NFT Вы можете создать паспорт .NET только для собственной учетной записи.
• Удаление учетной записи. Удаляет указанную учетную запись пользователя. Вы увидите эту опцию, только если войдете в систему с правами администратора.

Примечание. При удалении учетной записи пользователя Windows XP Professional выводит окно ***Хотите сохранить файлы, принадлежавшие имя_учетной_записи***. Если вы щелкнете ***Сохранить файлы***, Windows XP Professional сохранит содержимое рабочего стола и папки ***Мои документы*** этой учетной записи на рабочий стол в новую папку с именем имя_учетной_записи. Однако, сообщения электронной почты, избранное из Интернета и другие параметры настройки при этом теряются.

Оснастка Управление компьютером.
Одно из инструментальных средств управления Windows XP Professional — ***консоль управления Microsoft*** (MMC). Консоль MMC обеспечивает стандартизированный метод создания, сохранения и открытия административных инструментальных средств. Консоль MMC не имеет самостоятельных функций администрирования, и включает так называемые оснастки (snap-in), предназначенные для выполнения ряда административных задач.
• Администрирование и устранение неполадок в локальном масштабе Вы можете выполнять большинство административных задач и решать многие проблемы при помощи консоли MMC.
• Администрирование и устранение неполадок дистанционно. Вы можете использовать большинство оснасток для удаленного администрирования и устранения неполадок. Windows XP Professional выводит диалоговое окно, если есть возможность использован оснастку дистанционно.
• Централизованное администрирование. Консоли позволяют выполнять большинство административных задач с одного компьютера. Каждая консоль имеет одну или более оснасток, в том числе и сторонние оснастки, что позволяет создать одну консоль со всеми инструментальными средствами для выполнения административных задач.
Добавляя оснастки в пустую консоль вы настраиваете ее под свои нужды. На рис. 1.6 показана одна из оснасток, которую можно добавить, — ***Управление компьютером*** (Computer Management). Оснастка ***Управление компьютером*** — еще одно инструментальное средство Windows XP Professional для создания, удаления, изменения и отключения локальных учетных записей и изменения паролей.

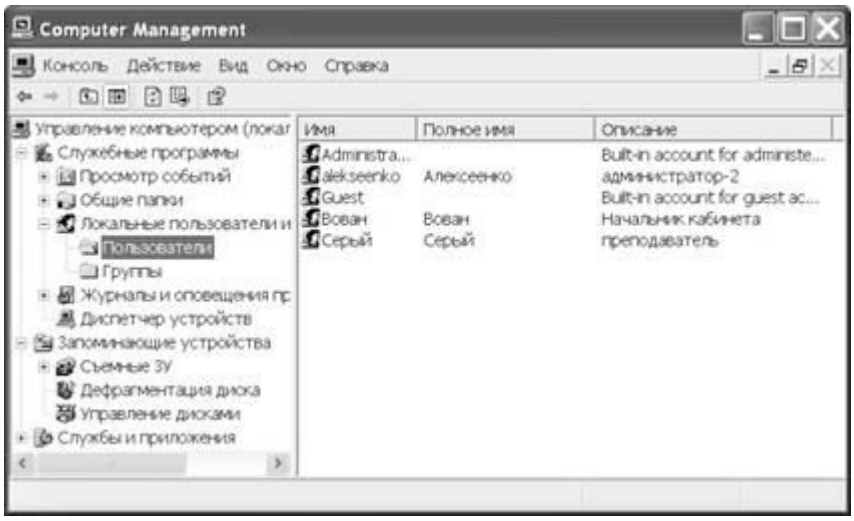


Рис. 1.6. Оснастка Управление компьютером

Изменение параметров учетных записей
Для каждой локальной учетной записи пользователя в Windows XP Professional по умолчанию задаются определенные параметры, которые после создания локальной учетной записи можно изменить средствами оснастки ***Управление компьютером***.
Параметры учетной записи сгруппированы на трех вкладках диалогового окна Свойства: Имя пользователя: Общие, Членство в группах и Профиль.
Вкладка ***Общие*** диалогового окна Свойства: Имя пользователя (рис. 1.7) позволяет задать или изменить параметры всех полей диалогового окна ***Новый пользователь*** за исключением имени пользователя, пароля и подтверждения пароля. Также на вкладке расположен флажок — ***Заблокировать учетную запись*** (Account Is Locked Out).
Если учетная запись активна и не заблокирована, то флажок ***Заблокировать учетную запись*** недоступен. Система блокирует пользователя, который превышает лимит числа неудачных попыток входа в систему. Эта функция безопасности затрудняет несанкционированный вход в систему. Если система блокирует учетную запись, то флажок ***Заблокировать учетную запись*** становится доступным, и администратор может снять флажок, чтобы разрешить доступ.
Вкладка Членство в группах (Member Of) в диалоговом окне Свойства: Имя пользователя позволяет добавить учетную запись пользователя в группу или удалить учетную запись из нее.
Вкладка Профиль (Profile) в диалоговом окне Свойства: Имя пользователя позволяет назначить путь к профилю пользователя, сценарию входа в систему и к домашней папке (рис. 1.8).

Рис. 1.7. Вкладка Общие диалогового окна Свойства: Имя пользователя

Рис. 1.8. Вкладка Профиль диалогового окна Свойства: Имя пользователя

Профиль пользователя (user profile) — это совокупность папок и данных, где хранится ваша текущая конфигурация рабочего стола, параметры настройки программ и личные данные. В ней также содержатся все установленные при входе в систему сетевые подключения, элементы меню Пуск (Start) и подключенные сетевые диски сервера. Профиль пользователя обеспечивает постоянство конфигурации рабочего стола при каждом входе в систему.
Windows XP Professional создает профиль пользователя при первом входе в систему и

сохраняет его на компьютере. Этот профиль также называют ***локальным профилем пользователя*** (local user profile).

- При входе в клиентскую систему для пользователя всегда создаются личные параметры рабочего стола и подключения независимо от того, сколько пользователей совместно используют этот клиентский компьютер.
- При первом входе в клиентскую систему Windows XP Professional создает для пользователя профиль пользователя по умолчанию. Путь к этому профилю: корень_системного_диска\Documents And Settings\имя_пользователя (обычно C:\Documents And Settings\имя_пользователя), где имя_пользователя — имя, под которым вы вошли в систему.
- Профиль пользователя содержит папку ***Мои документы***, в которой хранятся личные файлы. Мои документы — заданное по умолчанию местоположение для команд Открыть файл (File Open) и Сохранить как (Save As). Элемент Мои документы находится в меню Пуск, что облегчает доступ к личным документам.

Внимание! Пользователи могут сохранять свои документы в папке Мои документы или домашних папках, например в Основном каталоге, расположенном на сетевом сервере. Windows XP Professional автоматически назначает папку ***Мои документы*** местоположением для сохранения данных приложений Microsoft по умолчанию. Если на диске C или на диске, где установлена Windows XP Professional, достаточно места, пользователи могут сохранять свои документы в папке ***Мои документы***. Однако при этом значительно увеличивается количество минимального дискового пространства, необходимого для установки Windows XP Professional.

- Вы может изменять свой профиль пользователя, корректируя параметры рабочего стола. Например, если вы создаете сетевое подключение или добавляете файл в папку Мои документы (My Documents), Windows XP Professional включает изменения в ваш профиль пользователя при завершении сеанса. При следующем входе в систему это новое сетевое подключение или файл останутся без изменений.

Сценарий входа в систему — это файл, который можно создать для учетной записи пользователя, чтобы настроить рабочую среду пользователя. Например, вы можете использовать сценарий входа в систему, чтобы установить сетевые подключения или запустить приложения. При каждом входе пользователя в систему выполняется назначенный сценарий входа в систему.

Домашняя папка. В дополнение к папке Мои документы Windows XP Professional позволяет вам создать домашние папки, чтобы пользователи могли хранить там свои личные документы. Вы можете хранить домашние папки на клиентском компьютере, в общей папке на файловом сервере или в центральном местоположении на сетевом сервере. Хранение всех домашних папок на файловом сервере имеет следующие преимущества:

- пользователи имеют доступ к своим домашним папкам с любого клиентского компьютера в сети;
- можно централизовать резервное копирование и управление пользовательскими документами, переложив ответственность за это с пользователей на оператора архива или администратора сети.

Внимание! Храните домашние папки на томах NTFS. В этом случае вы сможете использовать разрешения файловой системы NTFS для обеспечения безопасности пользовательских документов. При хранении домашних папок на томах FAT удается ограничить доступ к домашней папке только средствами разрешений общей папки.

2.3. Изучите состав и назначение встроенных рабочих групп.

В Windows XP Professional предусмотрено две категории встроенных групп, локальные и системные, позволяющие упростить назначение прав и разрешений для часто выполняемых задач.
Группа (group) — это совокупность учетных записей пользователей. Группы упрощают администрирование, позволяя назначить разрешения и права для группы пользователей, а не индивидуально для каждой учетной записи (рис. 1.9).

- Группа — это совокупность учетных записей пользователей
- Все члены группы получают разрешения для этой группы
- Пользователи могут быть членами нескольких групп
- Группа может входить в другую группу

Рис. 1.9. Группы упрощают администрирование

Посредством *разрешений* (permissions) можно определить права пользователей при работе с ресурсами типа папки, файла или принтера. Назначив разрешения, вы предоставляете пользователям доступ к ресурсу и определяете тип доступа, который они имеют. Например, если нескольким пользователям необходимо прочесть некий файл, то можно объединить их учетные записи в группу и затем предоставить этой группе разрешение на чтение этого файла. *Права* (rights) позволяют пользователям выполнять системные задачи, например изменять время, выполнять архивацию и восстановление файлов.

Локальные группы

Локальная группа (local group) — совокупность учетных записей пользователей на компьютере. Локальные группы используются при назначении разрешений на доступ к ресурсам, постоянно хранящимся на компьютере, где создана локальная группа. Windows XP Professional создает локальные группы в базе данных локальных политик безопасности.

Подготовка к применению локальных групп

• Используйте локальные группы на компьютерах, которые не входят в домен. Вы можете использовать локальные группы только на компьютере, где они создаются. Хотя локальные группы доступны на рядовых серверах и компьютерах домена на платформе Windows 2000 Server, не используйте локальные группы на компьютерах, которые являются частью домена. Применение локальных групп на компьютерах домена лишает вас возможности централизованного администрирования группы. Локальные группы недоступны в каталоге Active Directory, и вам придется управлять ими отдельно на каждом компьютере.
• Разрешения локальных групп предоставляют доступ к ресурсам только на компьютере, где вы их создаете.

Примечание. Нельзя создать локальные группы на контроллерах домена, потому что контроллеры домена не могут иметь независимой от Active Directory базы данных политик безопасности.

Правила членства в локальных группах включают следующее:
• локальные группы могут объединять только локальные учетные записи пользователей системы, где эти группы создаются;
• локальные группы не могут принадлежать никакой другой группе.

Удаление локальных групп

Для удаления локальных групп используйте оснастку ***Управление компьютером*** (Computer Management). Каждая создаваемая вами группа имеет уникальный идентификатор, который никогда повторно не используется. Windows XP Professional применяет его для идентификации группы и ее разрешений. При удалении группы Windows XP professional снова этот идентификатор не использует, даже если вы создаете новую группу с таким же, как у удаленной группы, именем. Поэтому не удастся восстановить доступ к ресурсам, повторно создав группу.
При удалении группы вы удаляете только группу и ее разрешения и права. При этом не удаляются учетные записи пользователей, которые являются членами группы. Чтобы удалить группу, щелкните правой кнопкой мыши имя группы в оснастке ***Управление компьютером***, затем щелкните пункт меню ***Удалить***.

Встроенные локальные группы
Все изолированные серверы, рядовые серверы и компьютеры на платформе Windows XP Professional имеют ***встроенные локальные группы*** (built-in local groups). Они дают право выполнять системные задачи на отдельно взятом компьютере — резервное копирование и восстановление файлов, изменение времени и управление системными ресурсами.
Windows XP Professional хранит встроенные локальные группы в папке ***Группы*** (Groups) оснастки ***Управление компьютером***.
В таблице 2.2 перечислены встроенные локальные группы и описаны их возможности. За исключением отдельно оговоренных случаев, эти группы не имеют исходных членов.

Встроенные системные группы
Встроенные системные группы существуют на всех компьютерах под управлением Windows XP Professional. Системные группы не имеют определенных членств, которые можно изменять, но в них входят различные пользователи в разное время, в зависимости от того, каким образом пользователь получает доступ к системе или ресурсам. Системные группы не видны, когда Вы администрируете группы, но они доступны при назначении прав и разрешений для ресурсов. Windows XP Professional организует членство в системных группах в зависимости от вида доступа к компьютеру, а не от личности пользователя. В таблице 2.3 перечислены встроенные системные группы и описаны их возможности.

Возможности встроенной локальной группы Таблица 2.2

Локальная группа	Описание

Администраторы (Administrators)	Члены этой группы могут выполнять все административные задачи на компьютере. По умолчанию встроенная учетная запись администратора является членом этой группы. Когда к домену подключается рядовой сервер или компьютер на платформе Windows XP Professional, Windows 2000 Server добавляет группу ***DomainAdmins*** (Администраторы домена) в локальную группу администраторов
Операторы архива(Backup Operators)	Члены этой группы могут выполнять резервное копирование и восстановление системных компонентов с помощью служебной программы ***Архивация Windows*** (Windows Backup)
Гости (Guests)	Члены этой группы могут: • выполнять только задачи, для которых им предоставлены права; • пользоваться только теми ресурсами, на доступ к которым они имеют разрешения. Члены этой группы не могут производить постоянные изменения конфигурации рабочего стола. По умолчанию встроенная гостевая учетная запись является членом этой группы. Когда рядовой сервер или компьютер на платформе Windows XP Professional присоединяется к домену, Windows 2000 Server добавляет группу Domain Guests (Гости домена) в локальную группу Гости (Guests)
Опытные пользователи (Power Users)	Члены этой группы могут создавать и изменять локальные учетные записи пользователей на компьютере и совместно использовать ресурсы
Репликатор (Replicator)	Занимается дублированием файлов в домене
Пользователи (Users)	Члены этой группы могут: • выполнять только задачи, для которых им были предоставлены права; • пользоваться только теми ресурсами, для которых они имеют разрешения. По умолчанию Windows XP Professional добавляет в группу Пользователи (Users) все локальные учетные записи пользователей, которые администратор создает на компьютере. Когда рядовой сервер или компьютер на платформе Windows XP Professional присоединяется к домену, Windows 2000 Server добавляет группу Domain Users (Пользователи домена) в локальную группу Пользователи (Users)

Возможности встроенных системных групп Таблица 2.3

Системная группа	Описание
Все (Everyone)	Все пользователи, которые имеют доступ к компьютеру. По умолчанию при форматировании тома в NTFS, группе ***Все*** (Everyone) предоставляется разрешение полного доступа. Это вызывало проблемы в более ранних версиях Windows, в том числе и Microsoft Windows 2000. В Windows XP Professional анонимный вход в систему теперь не включен в группу Все (Everyone). При переходе с Windows 2000 Professional на Windows XP Professional ресурсы с разрешениями для группы Все (Everyone), а не

	специально для группы анонимного входа в систему, недоступны группе анонимного входа в систему
Прошедшие проверку (Authenticated Users)	Все пользователи с действительными учетными записями пользователя на компьютере. (Если ваш компьютер — часть домена, все пользователи включаются в Active Directory)
Создатель-владелец (Creator Owner)	Эта учетная запись пользователя для тех, кто создал или владеет ресурсом. Если ресурс создает член группы администраторов, то ресурсом владеет группа администраторов.
Сеть (Network)	Любой пользователь с удаленным подключением к общему ресурсу на этом компьютере.
Интерактивные (Interactive)	Учетная запись для пользователя, вошедшего в систему. Члены интерактивной группы могут пользоваться физическими ресурсами системы. При входе в систему они получают доступ к ресурсам компьютера «интерактивно»
Анонимный вход (Anonymous Logon)	Любая учетная запись, подлинность которой Windows XP Professional подтвердить не может
Удаленный доступ (Dialup)	Любой пользователь, кто в настоящее время имеет удаленный доступ к компьютеру через модем

Лабораторная работа № 7.

Тема: Работа с пользовательскими группами в ос Windows XP

1. Теоретическая часть:

1.1. Система прав доступа *Windows XP*

При включении компьютера после загрузки операционной системы на экране появляется приветствие, после чего предлагается выбрать режим работы (имя пользователя, учетная запись или аккаунт пользователя) под которым вы войдете в сеанс работы с системой. Если компьютер входит в состав домена с большим количеством пользователей, то вместо экрана «Приветствия» появится экран, где необходимо ввести названия учетной записи в системе и пароль.

Правильно составленный пароль должен иметь не менее восьми символов в длину, должен содержать строчные и прописные символы, цифры и различные метасимволы, чтобы иметь гарантию, что пароль всегда будет уникальным.

Для успешного функционирования операционной системы должна быть введена определенная система безопасности, базирующаяся на системе прав доступа. Ниже приведены стандартные права пользователей, предусмотренные в операционной системе *Windows XP*.

Чтобы каждый раз при создании нового пользователя не указывать комбинации похожих прав пользователей, в системе существует несколько готовых прав для пользователей, пригодных для реального использования системными администраторами и отдельными

пользователями, называемых группами. В *Windows XP* существует 10 основных групп пользователей:

- *Администраторы* (*Administrators*) – пользователи имеют полный и неограниченный доступ к компьютеру;

- *Операторы* сохранения и резервирования данных (*Backup Operator*) – пользователи могут заниматься сохранением информации и ее резервированием на будущее;

- *Гости* (*Guests*) – пользователи имеют доступ аналогичный группе *Пользователи*, но несколько более урезанный, по умолчанию группа отключена;

- *Операторы сетевой конфигурации* (*Network Configuration Operators*) – пользователи могут иметь некоторые административные привилегии для управления сетевыми возможностями системы;

- *Опытные пользователи* (*Power Users*) – пользователи, входящие в группу, имеют практически те же права, что и администраторы системы, но только с некоторыми ограничениями;

- *Replicator* – используется для организации работы системы в домене, а именно для репликации файлов в домене;

- *Пользователи* (*Users*) – наиболее популярная группа, в которую входят простые пользователи; все пользователи, которые работают в системе, должны входить в эту группу, т. к. они защищены от случайного уничтожения информации или изменения системы;

- *Пользователи, осуществляющие отладку* (*Debugger Users*) – пользователи имеют право на отладку программ и процессов на данной машине, локально или удаленно;

- *Помощники* (*HelpServicesGroup*) – группы пользователей для помощи и Центра Поддержки.

Если требуется, чтобы у пользователя был набор прав, состоящий из прав по нескольким группам, то можно его сделать членом всех этих групп. Тогда набор прав будет равен сумме прав, входящих в группы, членом которых пользователь является. Например, если обычному пользователю иногда нужно сохранять или резервировать информацию системы, осуществлять отладку программ, то пользователь может быть членом трех групп: *Users*, *Backup Operators* и *Debug Users*.

Для доступа к программе управления пользователями, необходимо запустить программу *Computer Management*:

Пуск - Панель управления – Администрирование - Управление компьютером - Локальные пользователи и *Группы* (рис.1.) Для выполнения административных действий: смены пароля, изменения свойств пользователя и пр. необходимы права администратора системы.

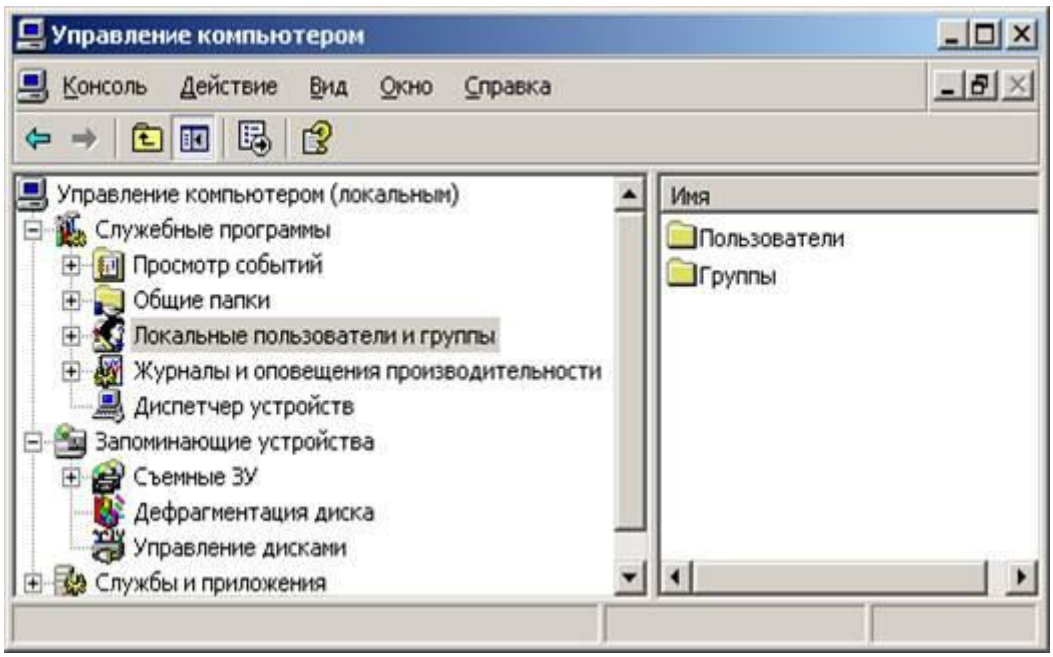


Рис.1. Окно программы Управление компьютером

(*Computer Management*)

В зависимости от выполняемой работы, нужно выбрать *Пользователи* (*Users*) или *Группы* (*Groups*).

При выборе строки *Пользователи* в окне настройки пользователей (рис.2.) после выбора двойным щелчком любого из пользователей появится окно свойств пользователя (рис.3.).

После открытия закладок *Членство в группах* и *Профиль* появятся, соответственно, окно принадлежности пользователя группам пользователей (рис.4.) и окно его настроек (рис.5.).

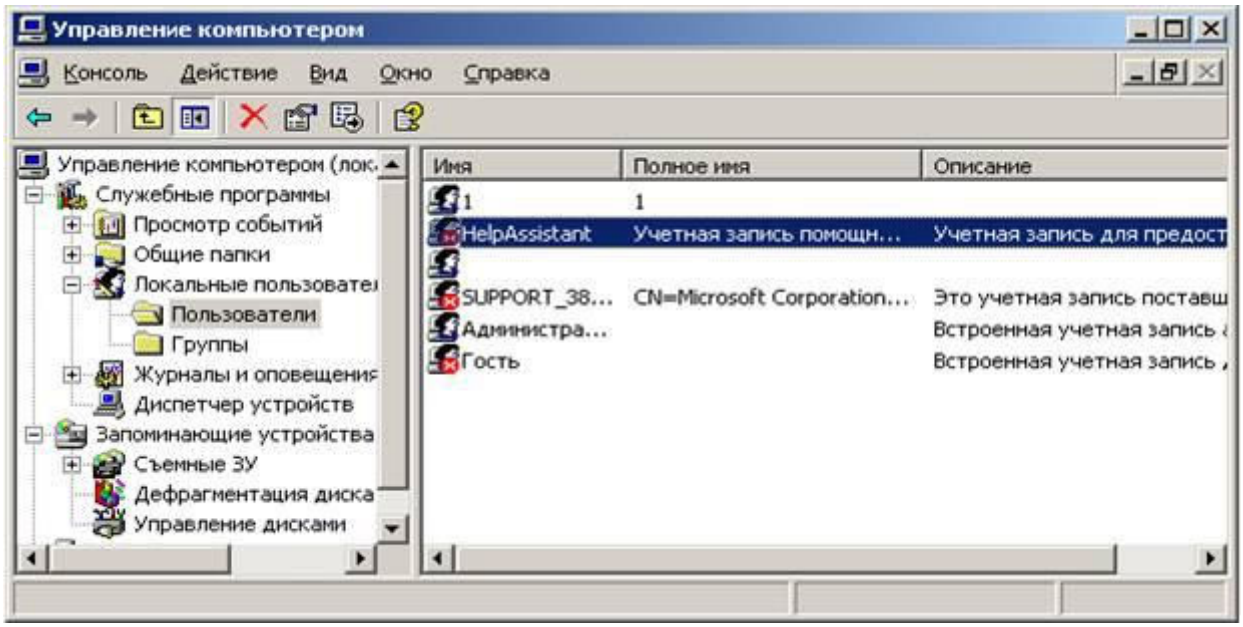


Рис.2. Окно настройки пользователей

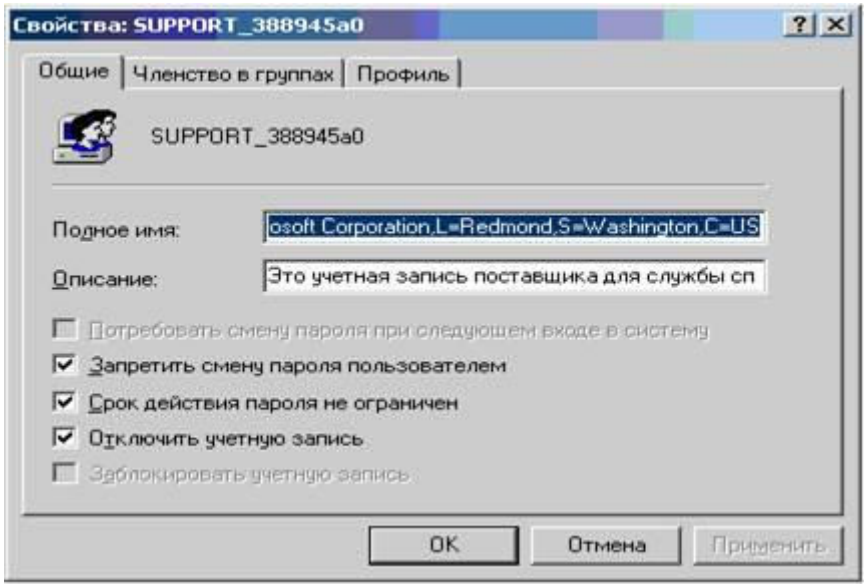


Рис.3. Окно свойств пользователей

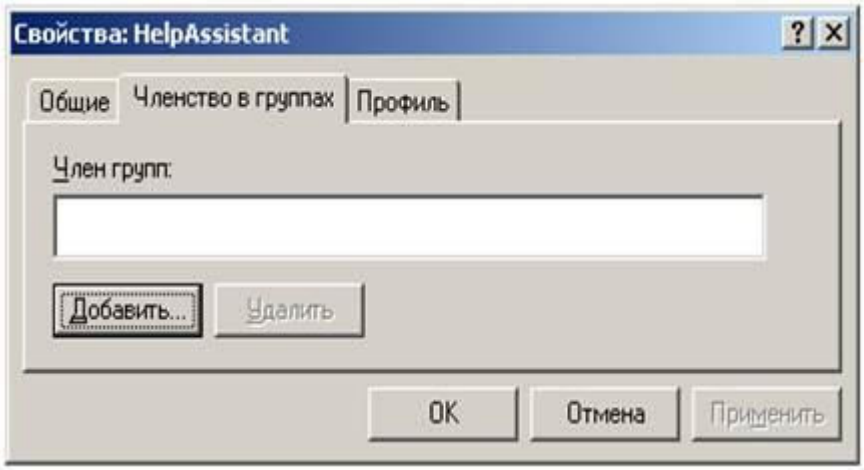


Рис.4. Окно принадлежности пользователя группам пользователей

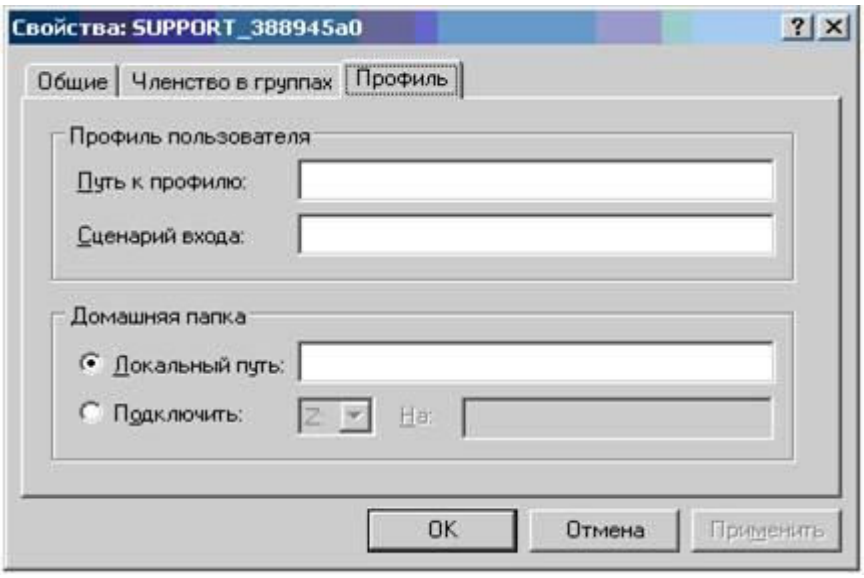


Рис.5. Окно настроек пользователей

В окне *Общие свойства пользователя* имеются следующие поля:

Потребовать смену пароля при следующем входе в систему – поле влияет на то, должен ли пользователь при своем следующем входе в систему менять свой пароль. Поле может быть полезно, когда системный администратор при создании пользователя присваивает некоторый пароль по умолчанию. При первом сеансе работы, когда пользователь будет работать со своей информацией, система предложит ему ввести свой пароль, который не будет известен системному администратору, что говорит об уровне и комфорте безопасной работы.

Запретить смену пароля пользователем – поле запрещает пользователю смену своего пароля. Данное свойство может быть полезно, например, когда в системе работает не опытный пользователь, умеющий менять пароли, но забывающий их.

- Задать вопрос

Срок действия пароля не ограничен – поле указывает на то, что срок действия пароля данного пользователя никогда не истекает.

В политике безопасности системы принято, чтобы пользователи с определенной периодичностью, определяемой системным администратором, меняли свои пароли, что необходимо для целей безопасности сложных, многопользовательских систем, в которых хранятся большие объемы важной информации: финансовой, отчетной, инженерной и прочей. Отказ от этого правила может быть лишь в следующих случаях:

- система имеет только одного пользователя, с относительно малозначимой информацией;

- система имеет одного пользователя с надежным паролем для запоминания и устойчивым к взломам кракеров;

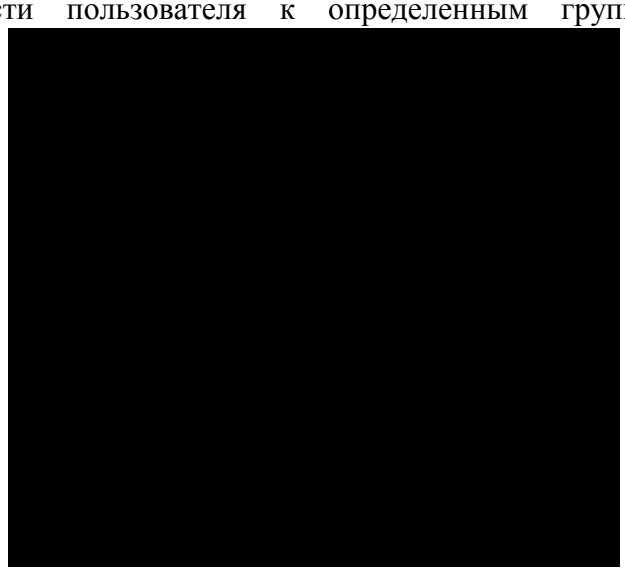
- пользователи обладают настолько малым желанием по обеспечению безопасности системы, что не желают поддержания секретности их паролей на должном уровне.

Отключить учетную запись – поле выполняет функции выключателя учетной записи пользователя. Это один из наиболее часто используемых полей в борьбе с недобросовестными пользователями, а также в целях защиты и настройки системы. Другая причина блокировки учетной записи заключается в том, что не все учетные записи принадлежат реальным пользователям. В системе существуют еще и специальные учетные записи, которые принадлежат виртуальным пользователям или некоторым системным службам, например, *аккаунт Guest*. Учетная запись имеет практически такие же права, за некоторыми исключениями, как и группа *Users*. Смысл *Гостевого входа* в том, что он используется для раздачи *Windows* сетевых ресурсов другим пользователям с удаленных систем: папок, файлов и пр. при работе системы в составе компьютерной сети. И если учетная запись *Гостя* не включена, то может оказаться невозможным вход пользователей из сети в данную систему.

Заблокировать учетную запись – поле обеспечивает работу одного механизма системы безопасности *Windows XP*. Существует ряд причин, когда система может запретить определенному пользователю входить в сеанс работы с системой. Это может быть из-за того, что система безопасности ОС настроена таким образом, что должна запрещать пользователям вход в систему, после определенного количества неправильно введенных

паролей, чтобы избежать подбора пароля учетной записи методом подбора. Только системный администратор может вновь разблокировать запись пользователя, убрав флажок из поля *Заблокировать учетную запись*.

В окне *принадлежности пользователя группам пользователей* указывается принадлежности пользователя к определенным группам (рис.4.). Под списком



пользователей находятся две кнопки: *Добавить* и *Удалить*, которые управляют добавлением и удалением новых групп пользователей. Для удаления группы следует ее выделить щелчком мыши и нажать кнопку *Удалить*. Для добавления новых групп пользователей, к которым будет принадлежать выбранный пользователь, следует нажать кнопку *Добавить*. Появится окно выбора групп пользователей (рис.6.).

Окно содержит ряд кнопок и записей. Первая запись показывает тип объектов, по умолчанию стоит запись *Группы* (*Groups*). Вторая запись показывает, в какой системе производится работа, по умолчанию вписывается имя локальной системы.

При добавлении пользователей можно пропускать первые два поля, переходя сразу к нижнему полю, в которое нужно поместить имена объектов для добавления. Для добавления объектов следует нажать кнопку *Добавить*. Появится новое диалоговое окно (рис.7.). Нажать кнопку *Поиск*. Появится окно выбора групп пользователей (рис.8.). Щелчком мыши выбираем имя группы пользователей (для выбора ряда групп пользователей используются клавиши *Shift* или *Ctrl*). Нажать кнопку *ОК* или сделать двойной щелчок по выбранной группе пользователей. Выбранные группы пользователей появятся в нижней части окна выбора групп пользователей (рис.7.). После выбора групп нажать кнопку ОК. Появится окно свойств пользователя (рис.3.) с добавленными в него группами

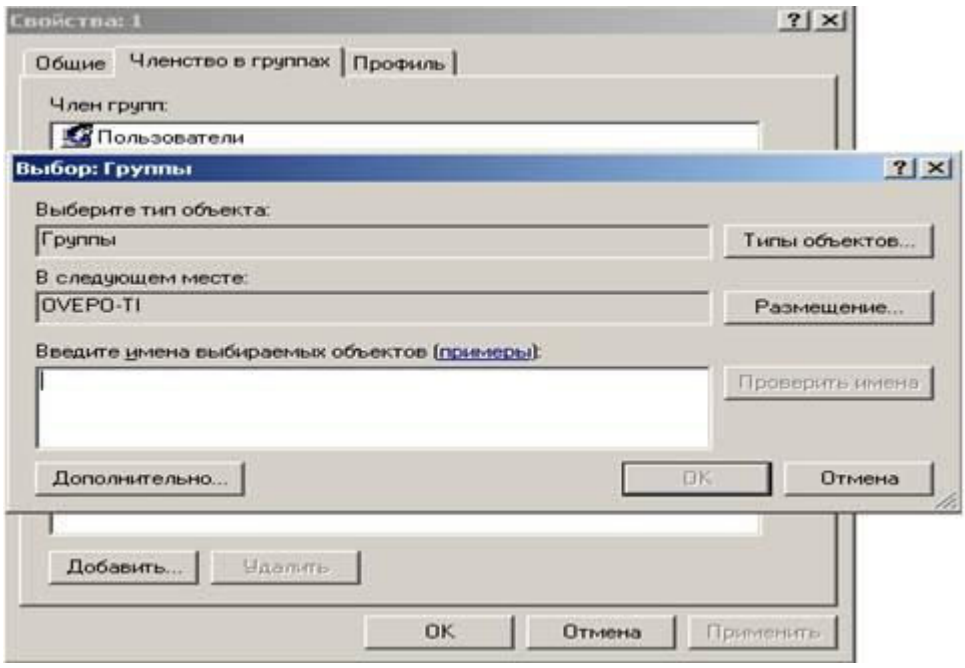


Рис.6. Окно выбора групп пользователей для добавления

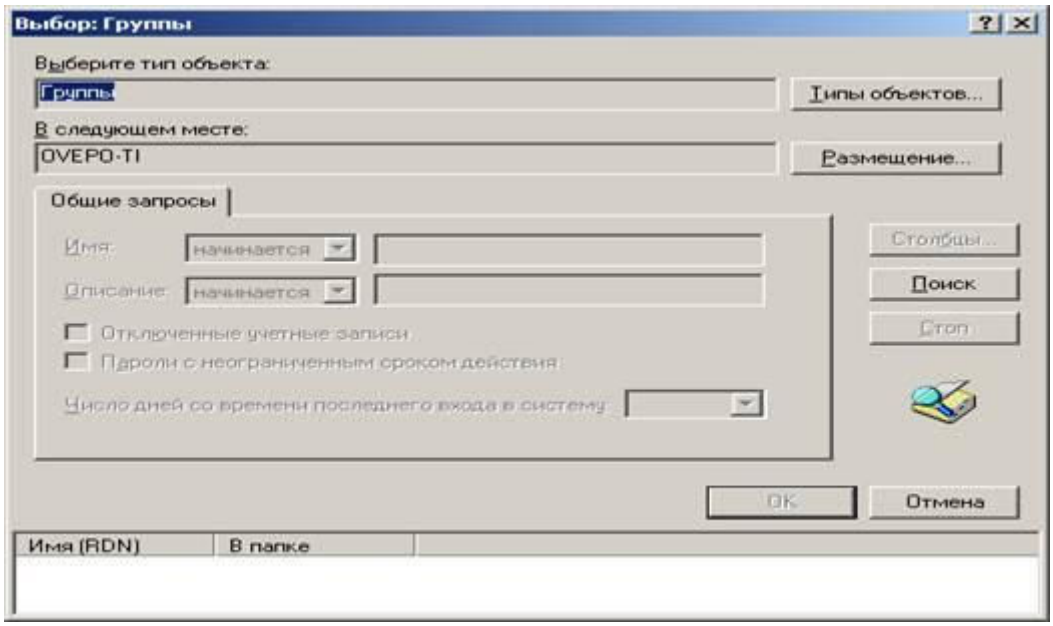


Рис.7. Окно выбора групп пользователей в расширенном варианте

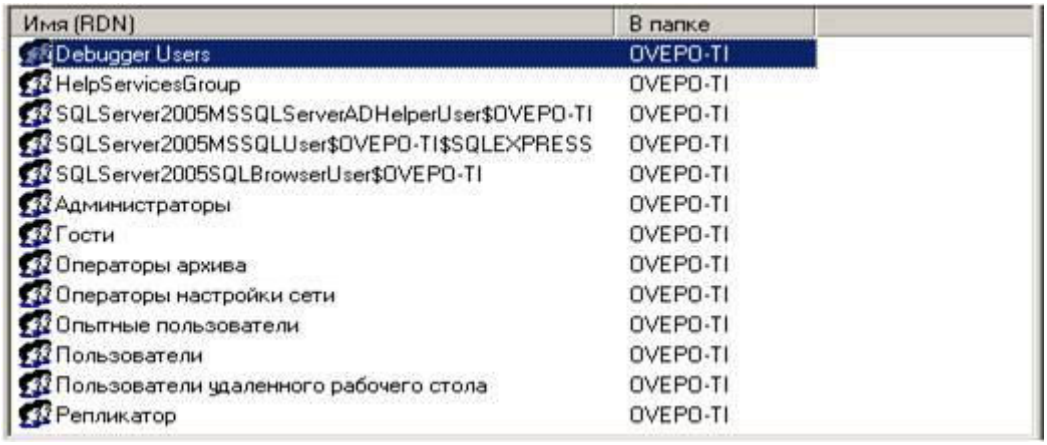


Рис.8. Вывод имен доступных групп пользователей

Далее следует произвести настройку профиля пользователя. Для этого следует выделить группу (группы пользователей) и перейти на вкладку *Профиль* (рис.5.5). Поля вкладки используются для работы *Windows XP* в больших сетях, чтобы пользователь имел доступ к другим ПК, подключенным к сети. Настройка данной вкладки гарантирует, что пользователь всегда получит доступ к личной информации, вне зависимости от его места нахождения.

1.2. Настройка пользовательской учетной записи

Для выполнения пользователем функций по настройке своей пользовательской учетной записи следует запустить программу *Учетные записи пользователей*:

Пуск - Панель управления – Учетные записи пользователей

На экране появится окно, содержащее справа сверху имя пользователя, тип его учетной записи: административная или обычная пользовательская (рис.9.).

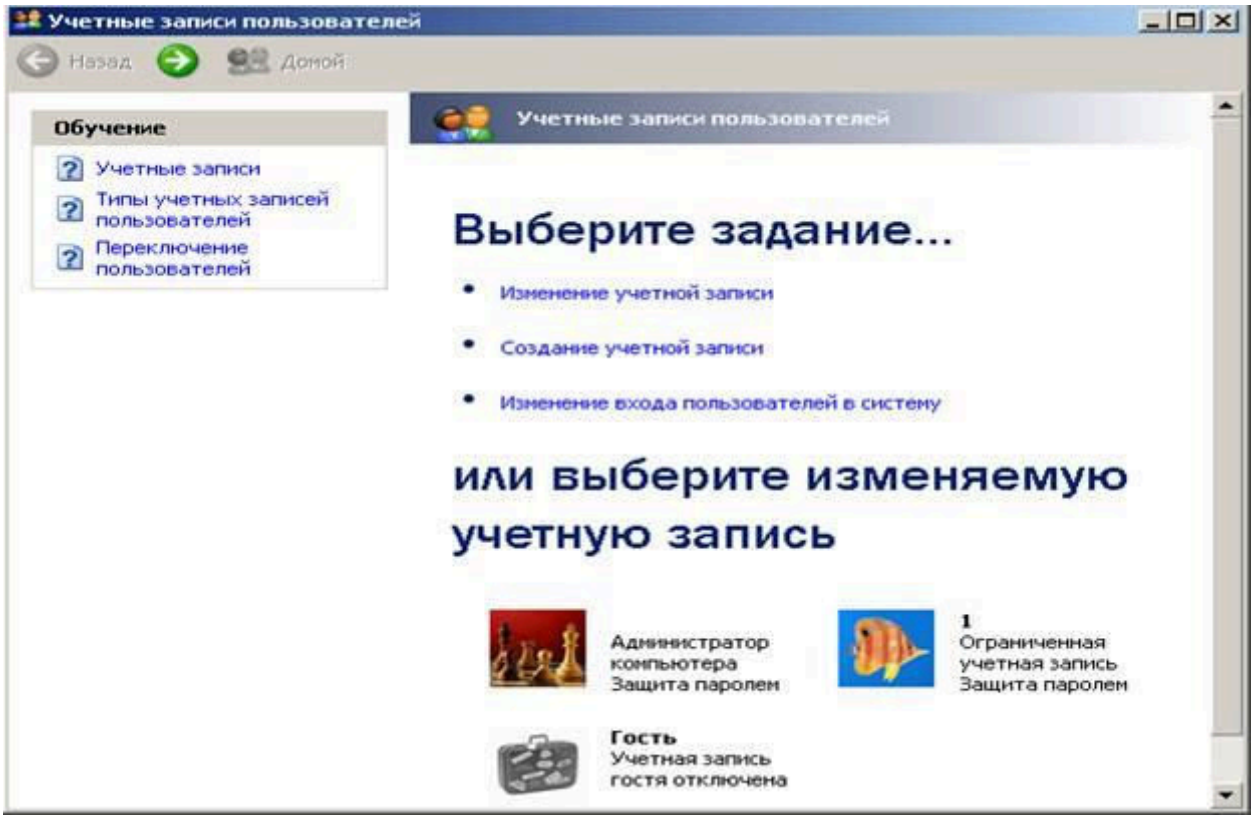


Рис.9. Окно программы *Учетные записи пользователя*

В окне отображен список учетных записей пользователя с указанием типа учетной записи: ограниченная учетная запись является обычной пользовательской; защита паролем указывает на то, что учетная запись защищена паролем. В верхней части окна приведен список действий *Выберите задание*, которые можно выполнить с конкретной (выделенной) учетной записью. Выберем учетную запись администратора. Появится новое окно (рис.10.

- Задать вопрос

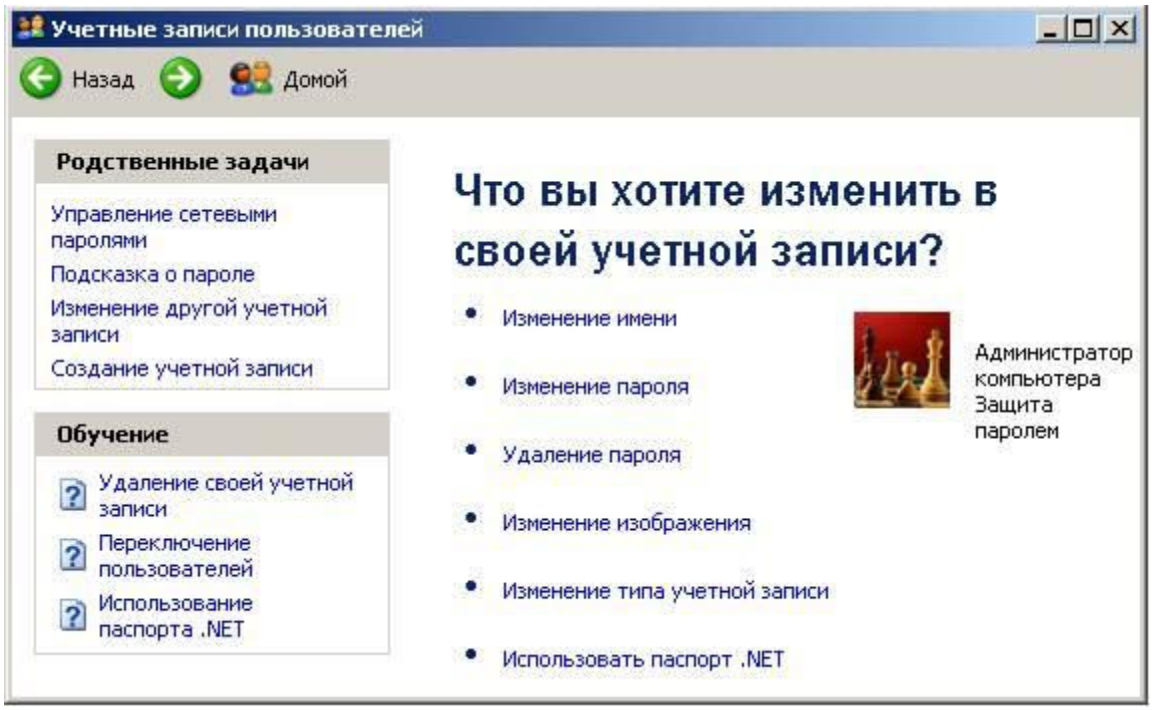


Рис.10. Окно программы *Учетные записи пользователя*

конкретного пользователя (администратора)

1.Режим *Изменение имени* позволяет изменить имя пользователя. Для смены имени пользователя следует сделать щелчок по режиму. Появится новое окно (рис.11.). На клавиатуре набрать новое имя пользователя и нажать кнопку *Сменить имя*.

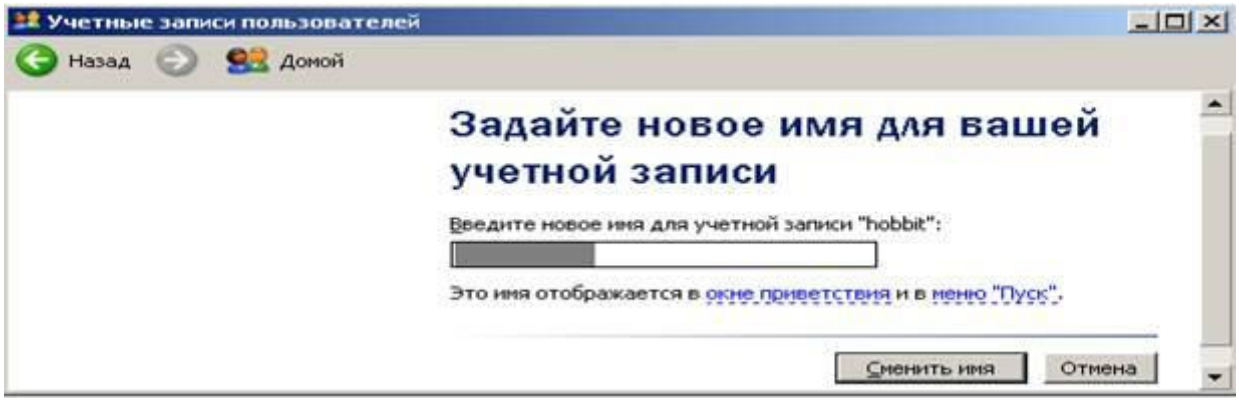


Рис.11. Окно смены имени выбранного пользователя

2.Режим *Изменение пароля* позволяет изменить пароль. После щелчка мышью по данному пункту появится окно смены пароля пользователя (рис. 11.). В первом поле вводится текущий пароль пользователя, в двух следующих – новый пароль. В последнем поле рекомендуется ввести подсказку, чтобы легче было вспомнить установленный пароль. Для подтверждения нового пароля следует нажать кнопку *Изменить пароль*.

3.Режим *Удаление пароля*. При активизации режима система поинтересуется, уверены ли вы в своем решении убрать пароль, т. к. вся информация, которая была доступна под вашим именем, станет доступна всем пользователям данной системы (рис.12.).

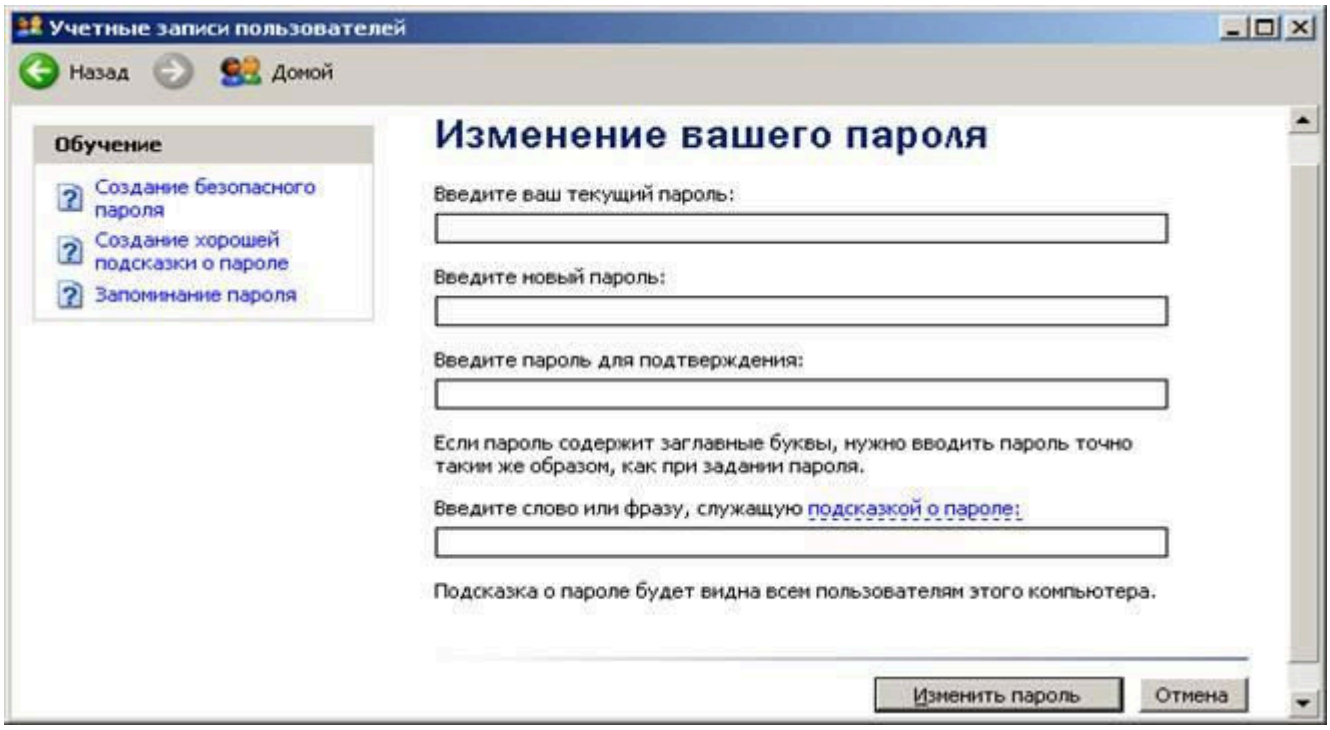


Рис.11. Окно изменения пароля выбранного пользователя

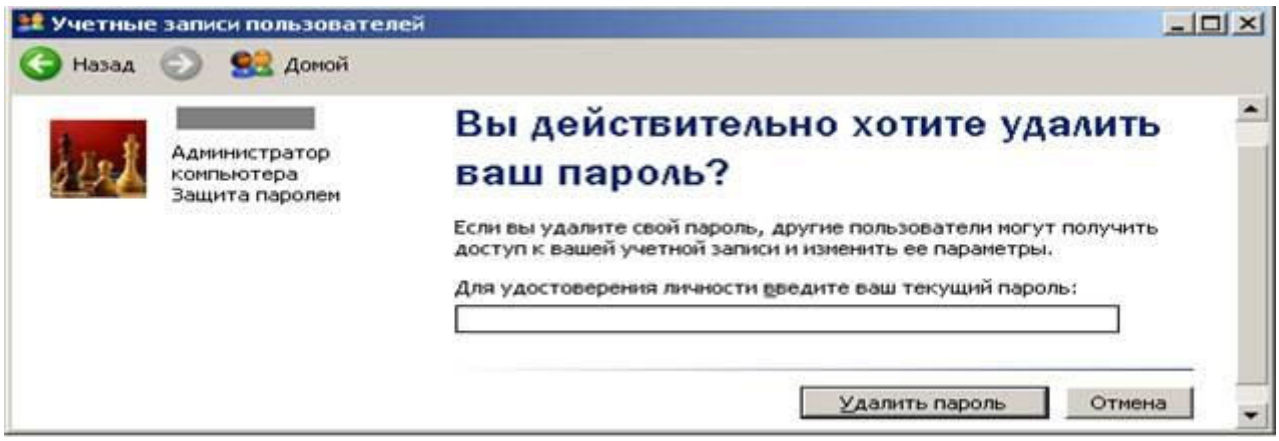


Рис.12. Окно подтверждения удаления пароля

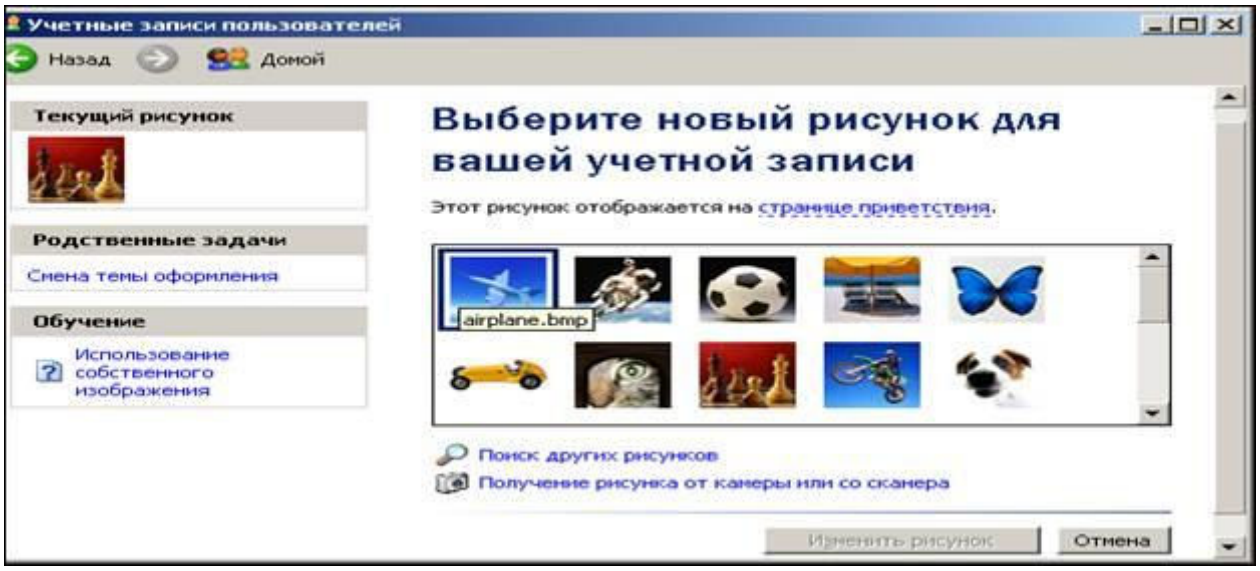


Рис.13. Окно изменения рисунка для выбранной учетной записи

4.Режим *Изменения изображения*. В *Windows XP* с именем каждого пользователя ассоциированная картинка, использование которой делает работу с пользователями более наглядной. При активизации режима появится новое окно (рис.13.) со списком рисунков для выбранной учетной записи. Для дополнения списка рисунков используется два режима:

- *Поиск других рисунков* открывает содержимое папки *Мои рисунки* и предлагает сделать выбор нового рисунка;

- *Получение рисунка от камеры или со сканера* предлагает воспользоваться фотографиями, рисунками или другими картинками отсканированными или отснятыми видеокамерой.

Ссылка *Обучение*: *Использование собственного изображения* подсказывает об использовании собственных картинок в качестве идентификатора, сопровождающего выбранную учетную запись.

5.Режим *Использовать паспорт.NET*. занимается тем, что сопоставляет данной пользовательской записи определенный уникальный сертификат – паспорт. При

активизации режима появляется окно *Мастера паспорта.Net* (рис.14.). Для продолжения работы следует воспользоваться подсказками мастера, выйти в Интернет и дальнейшая работа выполнится автоматически. Помощь при работе с данным режимом можно получить из раздела *Обучение*: *Использование паспорта .NET*.

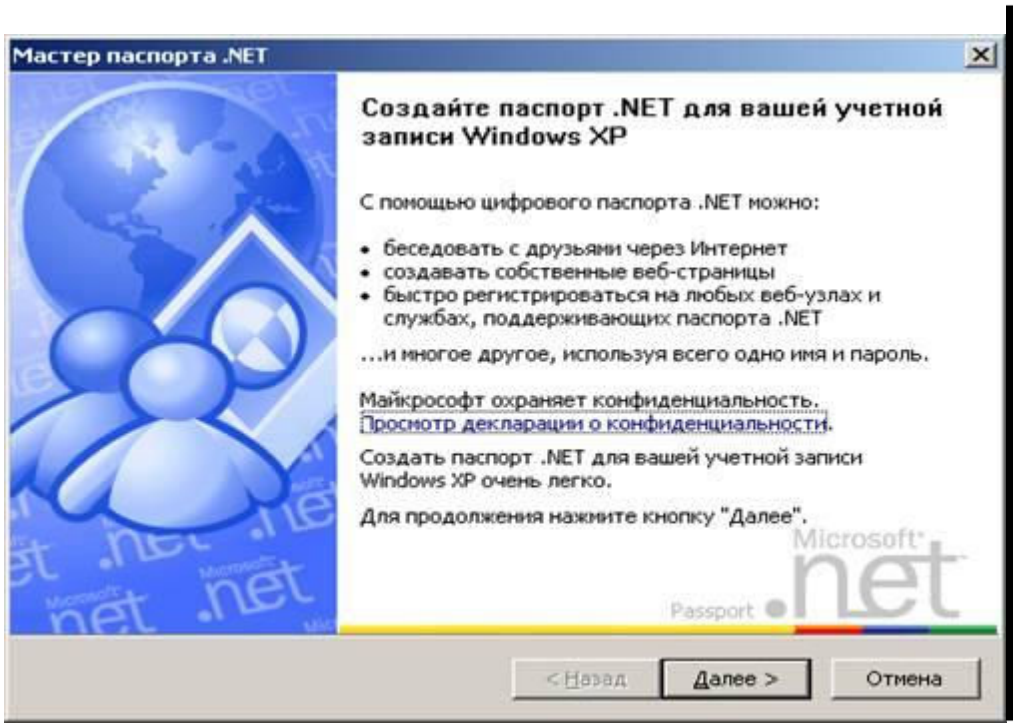


Рис.14. Окно Мастера паспорта *.Net*

В разделе *Родственные задачи* режим *Управление сетевыми паролями* активизирует окно *Сохранение имен пользователей и паролей*. В окне указываются личные данные, требуемые для подключения и регистрации в сети или на веб-узлах Интернета. Данные с помощью соответствующих кнопок можно добавлять, удалять изменять (кнопка *Свойства*).

1.3. Восстановление забытого пользователем пароля

В разделе *Родственные задачи* режим *Подсказка о пароле* активизирует окно *Мастер забытых паролей* (рис.15.). Мастер позволяет создать дискету «сброса паролей», которую можно использовать для создания нового пароля.

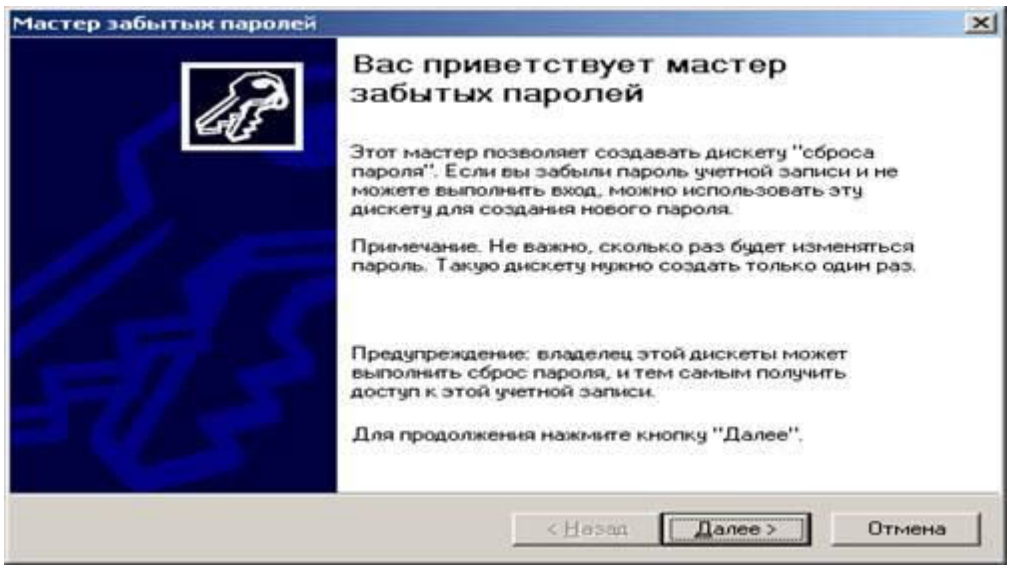


Рис.14. Окно Мастера забытых паролей

Мастер забытых паролей предлагает начать операцию сброса пароля на дискету, для которой потребуется чистый и форматированный флоппи-диск, который следует вставить в дисковод A:.

Предупреждение: *выполнение данной функции нужно делать только один раз, вне зависимости от частоты смены пароля в системе.*

В процессе выполнения мастера будет создан диск, который позволит, в случае, если пароль будет забыт, создать новый пароль для системы и успешно в нее войти. Такая возможность предоставляется пользователям *Windows XP*, благодаря использованию в ней специальных криптографических алгоритмов, использующих пару ключей. Эти алгоритмы считаются наиболее устойчивыми к взломам. На последующих рисунках (15-16) представлено использование этого мастера.

На вопрос мастера на третьем шаге нужно корректно ввести текущий пароль к системе и дождаться завершения работы мастера.

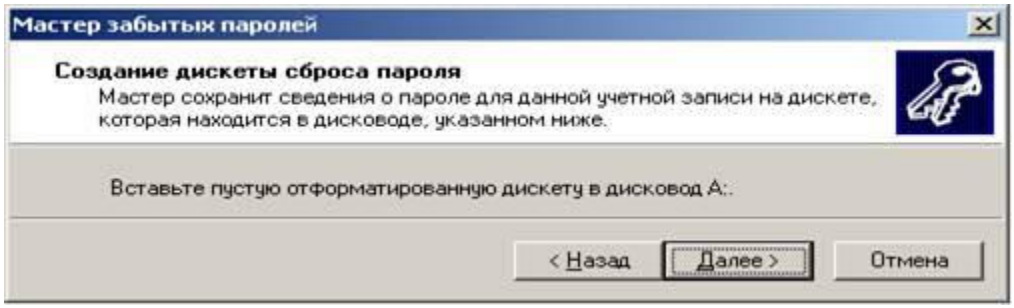


Рис.15. Второй шаг *Мастера забытых паролей*

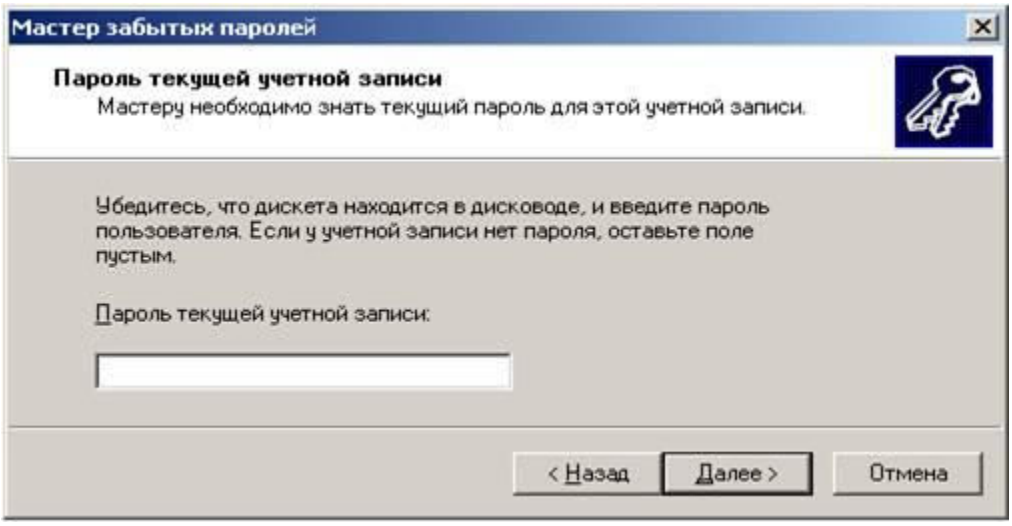


Рис.16. Третий шаг *Мастера забытых паролей*

Диск для дальнейшего использования следует спрятать в надежное и безопасное место, особенно если он сделан с правами администратора системы.

1.4. Выполнение действий над пользователями системы

Программа *Управление компьютером* (*Computer Management*) позволяет при выборе пользователя производить ряд возможных действий над ними посредством команд контекстного меню (рис.17.). Команды позволяют задавать пароль пользователю, переименовывать и удалять пользователя. При удалении пользователь из системы удаляется, но все его данные, программы, почта, временные файлы Интернета и системные области данных, включая реестр и корзину, остаются неизменными.

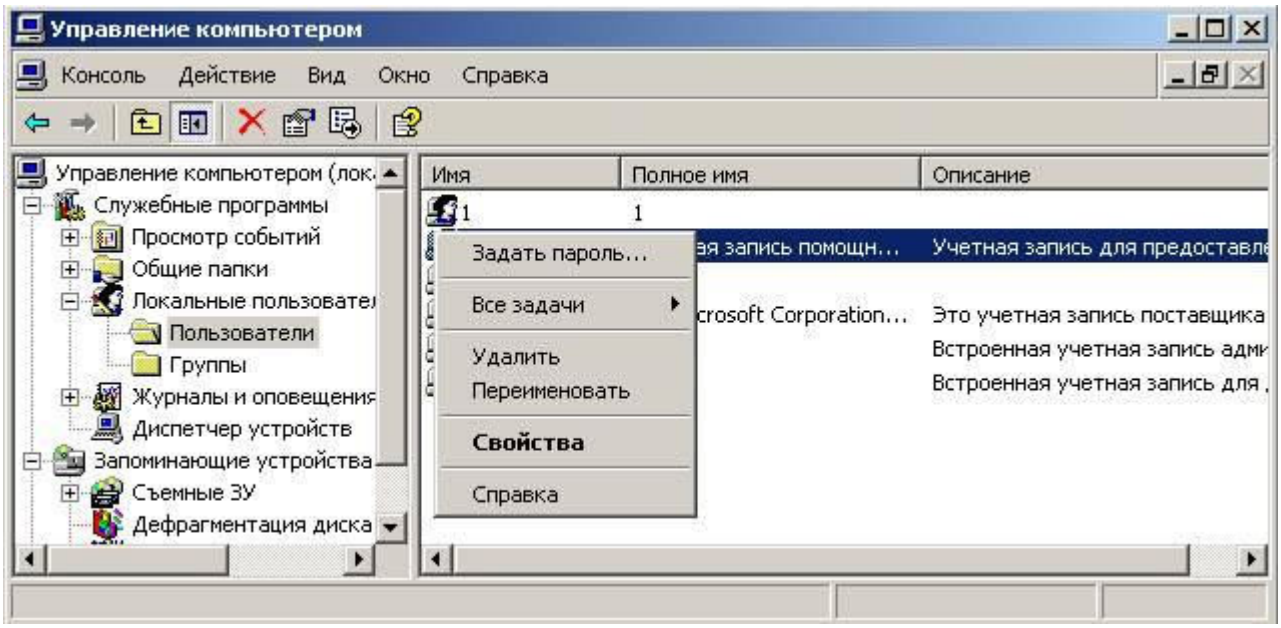


Рис.17. Окно программы *Управление компьютером* с контекстным меню

Для окончательного удаления пользователя из системы следует:

- войти в каталог: *%SystemRoot%\Documents and Settings* ;

- найти папку с именем удаленной учетной записи;

- удалить или заархивировать папку.

Аналогично следует поступить с корзиной пользователя, в качестве которой используется его идентификатор. Корзина находится в каталоге *RECYCLER*. Удаление пользователей используется по разным причинам, например, пользователь перешел в другой отдел компании или уволился.

- Задать вопрос

Переименование пользователей используется по разным причинам, например, пользователь перешел на другой компьютер или ему пришлось создать еще одну учетную запись. В этих случаях следует:

- создать новую учетную запись для нового пользователя;

- скопировать в нее все данные и файлы пользователя;

- логически удалить старую не нужную учетную запись;

- стереть все файлы старой учетной записи с жестко диска.

Существует еще одна особенность переименования учетных записей, связанная с возможностью серьезной утраты информации. Например, работающим на компьютере пользователям пришлось обменяться пользовательскими учетными записями. Наиболее логичным решением, в этом случае, является взаимное изменение имен учетных записей. Однако если удалить одного пользователя, то можно удалить с жесткого диска совпадающий по имени с учетной записью каталог, который может принадлежать другой учетной записи. Итог: оба пользователя будут удалены. Поэтому здесь следует быть особенно внимательным.

1.5. Работа с пользовательскими группами

В программе *Управление компьютером* после выбора раздела групп пользователей (рис.2.) можно выполнить ряд возможных над ними действий (рис.18.).

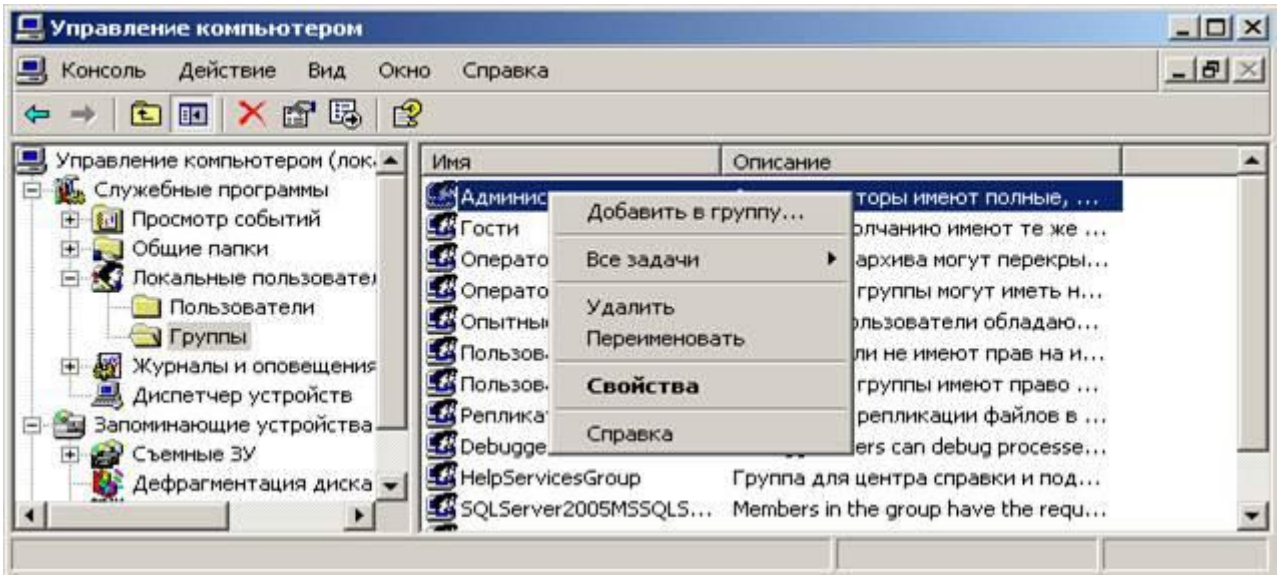


Рис.18. Окно программы *Управление компьютером*, раздел *Группы*

Команда *Добавить в группу* позволяет добавить в группу определенных пользователей или даже их группы, процесс аналогичен представленному ранее (рис.6 – 8).

Команда *Все задачи* содержит в себе тот же пункт меню: Добавить в группу.

Команды *Удалить* и *Переименовать* позволяют, соответственно, удалять и переименовывать группы пользователей, аналогично одному пользователю но с одним исключением: после удаления группы пользователей нет необходимости удаления дополнительной информации, так как группа не имеет своего представления в пользовательских папках.

Команда *Свойства* показывает членов группы, позволяя добавить к ней другие группы и пользователей, как и с помощью команды *Добавить в группу*.

При щелчке правой кнопки мыши на пустом месте появится другое контекстное меню (рис. 18).

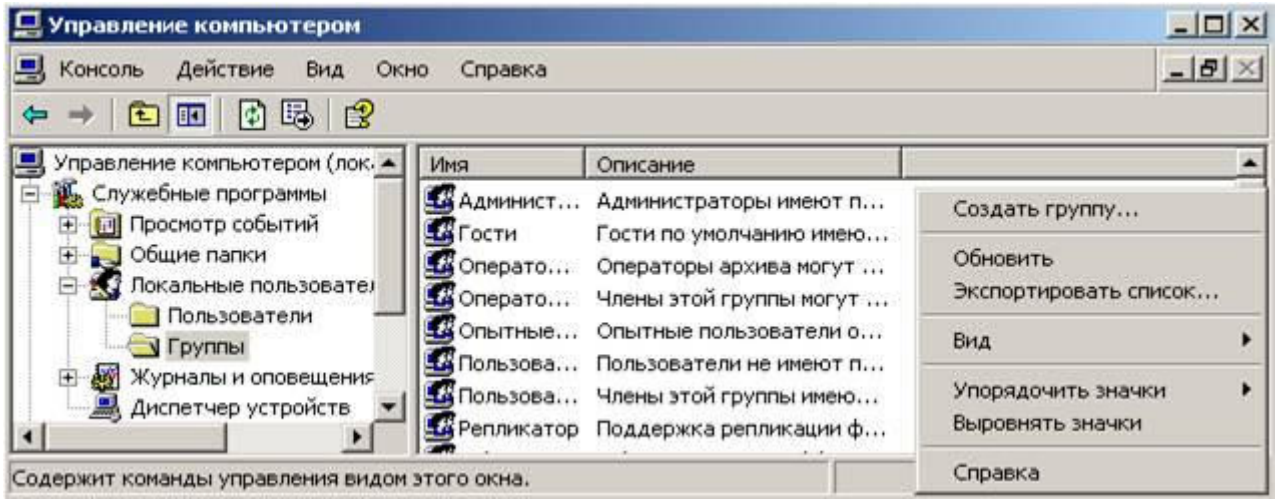


Рис.18. Окно программы *Управление компьютером*, раздел *Группы*,

с контекстным меню, вызванным в свободной области окна

Новой возможностью данного меню является команда *Создать группу*. После ее выбора на экране появится диалоговое окно создания новой группы пользователей (рис.19.).

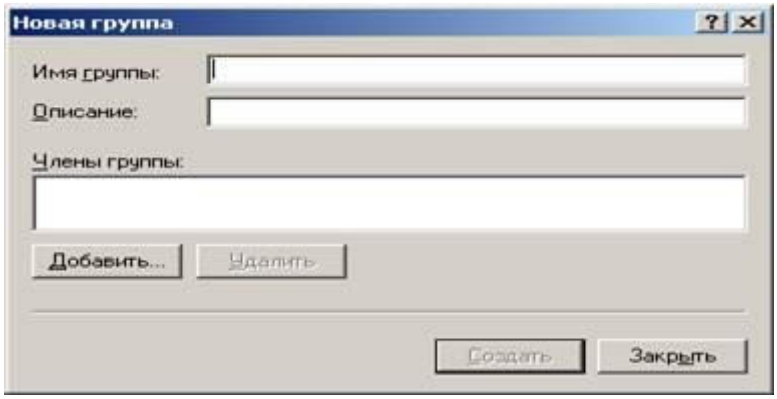


Рис.19. Диалоговое окно *Новая группа*

В первом поле диалогового окна предлагается ввести имя создаваемой группы пользователей. В следующей строке – ее краткое описание. Рекомендуется создавать простое, наглядное и емкое описание группы, что значительно упростит администрирование системы при возрастающей нагрузке и числе пользователей.

В поле *Члены группы* с помощью кнопки *Добавить* можно поместить пользователей, которые будут входить в ее состав, затем нажать кнопку *Создать*.

1.6. Управление входом пользователей в систему

Все введенные пользовательские учетные записи на экране «Приветствие» при входе в систему будут показаны. Для входа в систему под одним из них нужно его выбрать мышью и ввести пароль. После чего нажать клавишу *Enter* или стрелку рядом с паролем учетной записи и системой произведется вход в настройки выбранного пользователя и активизация его сеанса работы. Однако использование экрана «Приветствие» бывает удобно далеко не всегда, например, наличие большого количества пользователей или желание системного администратора скрыть их количество и имена, прописанных в системе.

Для того чтобы запретить экран «Приветствие», сделав вход в систему более безопасным и походящим на классический экран входа пользователей, в *Windows XP* необходимо под учетной записью, имеющей права администратора, войти в программу *Учетные записи пользователей* (рис.9).

Рядом с каждой учетной записью отображается ассоциированная с ней картинка. Для изменения учетной записи выбранного пользователя следует выбрать в верхней части окна режим *Изменение входа пользователя в систему* (рис.20.).

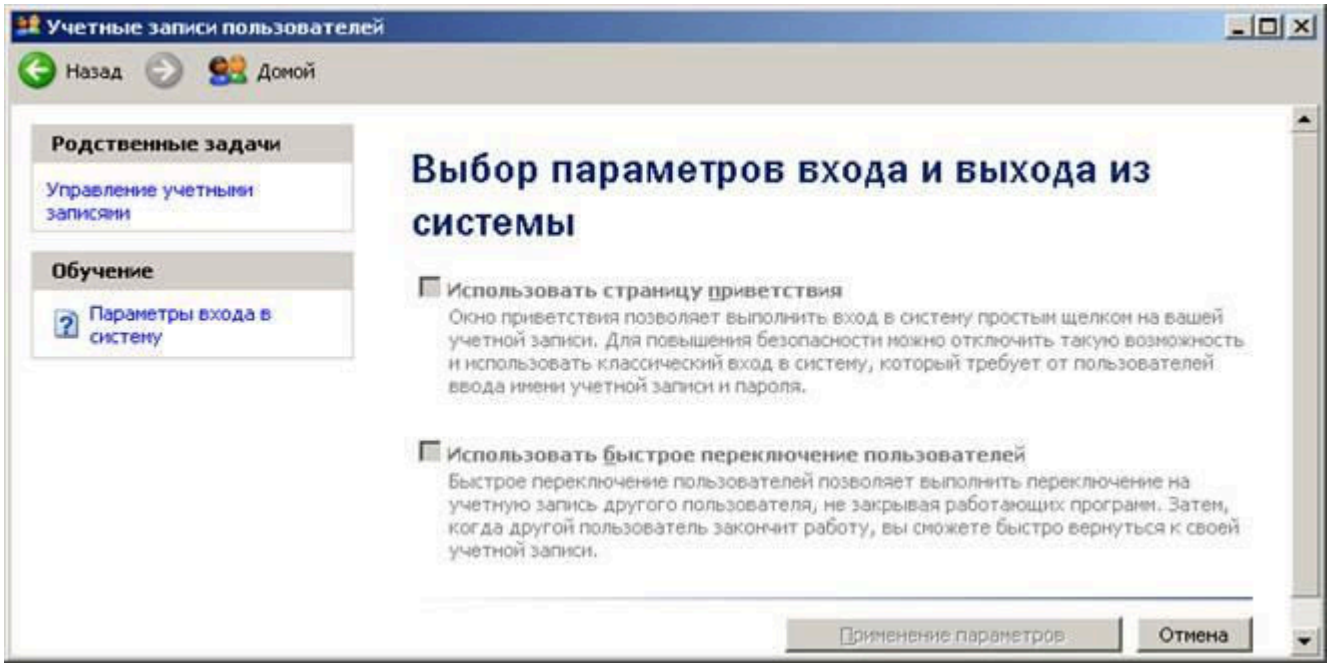


Рис.20. Окно *Выбор параметров входа и выхода из системы*

Режим *Использовать страницу приветствия* – управляет использованием экрана «Приветствие» при входе в систему. Если флажок в режиме установлен, то экран будет использован, в ином случае будет произведен классический вход в систему, как в Windows NT, которое предложит для входа в систему ввести имя учетной записи пользователя и ее пароль.

Режим *Использовать быстрое переключение пользователей* управляет переключением пользователей. Это связано с тем, что ОС *Windows NT* многозадачная и многопользовательская и в процессе ее работы можно переключаться между пользователями, которые работают в данной системе. При этом все настройки пользователей и их программы продолжат свое выполнение, даже если будет производиться переключение между учетными записями пользователей, работающих в данное время в системе. В случае если флажок не установлен, то переключение пользователей становится невозможным.

Переключение пользователей производится путем выбора: *Пуск\Выход из системы\Смена пользователя*. После выполнения команды появится экран «Приветствие» или классический вход *Windows NT* в зависимости от настройки системы.

2. Практическая часть

2.1. Вопросы по разделу

1. Что происходит при включении компьютера после загрузки операционной системы?

2. Что происходит при включении компьютера после загрузки операционной системы, если компьютер входит в состав домена с большим количеством пользователей?

3. Какой вид должен иметь правильно составленный пароль?

4. На чем базируется система безопасности в операционной системе *Windows XP*?

5. Перечислите основные группы пользователей в операционной системе *Windows XP*?

6. Что нужно сделать, если требуется, чтобы у пользователя был набор прав, состоящий из прав по нескольким группам?

7. Какую программу нужно запустить для доступа к программе управления пользователями?

8. Для выполнения административных действий: смены пароля, изменения свойств пользователя и пр. необходимы права администратора системы.

9. Какие правила используются в политике безопасности системы для целей безопасности сложных, многопользовательских систем, в которых хранятся большие объемы важной информации?

10. Перечислить случаи отказа от правил, используемых в политике безопасности системы для целей безопасности сложных, многопользовательских систем?

11. Перечислите причины отключения (блокирования) учетной записи.

12. Как определить к каким группам относится пользователь?

13. Как произвести добавление новых групп пользователю?

14. Как произвести добавление пользователей?

15. Как произвести вывод на экран доступных имен всех групп пользователей?

16. Как произвести вывод на экран списка учетных записей пользователя с указанием типа учетной записи?

17. Как произвести настройку учетной записи пользователя?

18. Как произвести изменение имени пользователя?

19. Как произвести изменение пароля пользователя?

20. Как произвести изменение рисунка учетной записи пользователя?

21. Что означает режим «Использование паспорта *.NET*?

22. Какие задачи решаются в разделе *Родственные задачи* режима *Управление сетевыми паролями*?

23. Как восстановить забытый пользователем пароль?

24. Какое предупреждение *Мастера забытых паролей* необходимо знать при восстановлении забытого пароля?

25. Какие действия производит над пользователями программа *Computer Management*?

26. Какие действия нужно произвести для окончательного удаления пользователя из системы?

27. Для каких целей используется переименование учетной записи пользователя?

28. Перечислите возможные действия при работе с пользовательскими группами в программе *Computer Management*?

29. Каким образом производится управление входом пользователя в систему?

30. Как произвести переключение пользователя во время сеанса работы?

Лабораторная работа № 8

Тема: Linux, работа с файлами и каталогами

Работа с файловой системой

Команда ls

Синтаксис:

`ls [-acltuFR] [`*`файлы_или_каталоги`*`]`

Описание:

При указании в аргументах *`файла`* выводится информация о нем согласно указанным опциям, в случае *`каталога`* — то же для файлов каталога. Если аргументов нет, — выдается листинг текущего каталога.

Опции:

`-a`	Выводить информацию обо всех файлах в каталоге (по системному соглашению файлы, имя которых начинается с точки, являются скрытыми, и командой `ls` не показываются
`-c`	Использовать время последней модификации i-node, связанного с файлом (создание файла, изменение прав, и т.д.), а не время модификации
`-i`	Выдать номер i-node, связанного с файлом
`-l`	Выдавать информацию о файлах в длинном формате: тип файла (обычный файл -, каталог *`d`*, сокет *`s`*, символьная ссылка *`l`*, символьное устройство *`c`*, блочное устройство *`b`*, FIFO *`p`*), права доступа (чтение — *`r`*, запись — *`w`*, исполнение — *`x`*; первые 3 символа относятся к владельцу, следующие 3 — к членам группы, владеющей файлом, и последние 3 — ко всем остальным), владелец, группа, размер в байтах, дата модификации, имя файла
`-t`	Отсортировать листинг по времени
`-u`	Использовать время последнего доступа к файлу, а не время модификации
`-F`	После каждого имени выдавать значок, характеризующий тип последнего: каталог (/), исполняемый файл (*), FIFO (|), символьная ссылка (@), сокеты семейства AF_UNIX (=)
`-R`	Выдавать листинг рекурсивно по подкаталогам

Примеры:

Получение полного листинга текущего каталога, включая скрытые файлы.

```
$ ls -la
итого 224
drwx------ 16 guest guest 4096 Мар 21 18:16 .
drwxr-xr-x  3 root root 4096 Мар 21 17:53 ..
-rw-------  1 guest guest   21 Мар 21 17:54 .bash_history
-rw-r--r--  1 guest guest   33 Дек 16 22:42 .bash_logout
-rw-r--r--  1 guest guest  176 Дек 16 22:42 .bash_profile
```

```
-rw-r--r--  1 guest guest  124 Map 16 22:42 .bashrc
drwxr-xr-x  2 guest test  4096 Map 21 17:53 Desktop
```

Команда pwd

Синтаксис:

```
pwd
```

Описание:

Выводит имя текущего каталога.

Примеры:

Определяем текущий каталог.

```
$ pwd
/home/guest
```

Команда cd

Синтаксис:

```
cd [каталог]
```

Описание:

Сменить текущий *каталог* на указанный в аргументе. При отсутствии аргументов происходит переход в домашний каталог пользователя `$HOME`.

Примеры:

Переход из текущего каталога в каталог `/usr/share/doc/`.

```
$ pwd
/home/guest
$cd /usr/share/doc/
$pwd
/usr/share/doc/
```

Команда cp

Синтаксис:

```
cp [-i] файл ... файл_или_каталог
```

```
cp -r [-i] каталог ... каталог
```

Описание:

Копирует *файлы* или *каталог*, указанный в первых параметрах, в файл или каталог, указанный в последнем.

Опции:

`-i`	Интерактивно — запрашивается подтверждение на перезапись существующих файлов
`-r`	Рекурсивное копирование каталога

Примеры:

Рекурсивное копирование каталога `a` в `b`:

```
$ cp -r a b
```

Интерактивное копирование файлов из каталога `a` в `b`:

```
$ cp -i a/* b
cp: overwrite b/1 (yes/no)? y
cp: overwrite b/2 (yes/no)? n
cp: overwrite b/2 (yes/no)? y
```

Команда ln

Синтаксис:

`ln [-fs]` *`файл_или_каталог ...`* `[`*`ссылка_или_каталог`*`]`

Описание:

Устанавливает ссылку в файловой системе (символьную или жесткую). Если второй операнд является уже существующим каталогом, то ссылки создаются внутри него. В случае, если второй аргумент отсутствует, ссылка создается в текущем каталоге с именем источника.

Опции:

`-f`	Устанавливать ссылку вместо существующего файла
`-s`	Символьная ссылка

Примеры:

Сделать символьную ссылку `b` на `a`:

```
$ln -s a b
$ls -l
итого 8
-rw-rw-r-- 1 guest guest 0 Map 21 18:57 a
lrwxrwxrwx 1 guest guest 1 Map 21 18:57 b -> a
```

Сделать жесткую ссылку между `a` и `c`:

```
$ln a c
$ls -l
итого 12
-rw-rw-r-- 2 guest guest 0 Map 21 18:57 a
lrwxrwxrwx 1 guest guest 1 Map 21 18:57 b -> a
```

```
-rw-rw-r-- 2 guest guest 0 Map 21 18:57 c
$ ls -il
итого 12
4247456 -rw-rw-r-- 2 guest guest 0 Map 21 18:57 a
4247457 lrwxrwxrwx 1 guest guest 1 Map 21 18:57 b -> a
4247456 -rw-rw-r-- 2 guest guest 0 Map 21 18:57 c
```

Команда mv

Синтаксис:

`mv [-i] файл_или_каталог файл_или_каталог`

Описание:

Переименовать *файл* или *каталог*, указанный в первом аргументе, в *файл* или *каталог*, указанный во втором.

Опции:

`-i`	Интерактивно — запрашивать разрешение на перезапись уже существующих объектов.

Примеры:

Переименовать `b` в `d`:

```
$mv b d
$ ls -l
итого 12
-rw-rw-r-- 2 guest guest 0 Map 21 18:57 a
-rw-rw-r-- 2 guest guest 0 Map 21 18:57 c
lrwxrwxrwx 1 guest guest 1 Map 21 18:57 d -> a
```

Попытка переименования файла самого в себя:

```
$ mv d c
mv: `d' и `c' - один и тот же файл
```

Команда rm

Синтаксис:

`rm [-f] [-i] файл...`

`rm -r [-f] [-i] каталог...`

Описание:

Удалить файлы или каталоги. Если права доступа не позволяют этого сделать немедленно, то последние выводятся в восьмеричной форме и требуется подтверждение операции.

Опции:

`-f`	Не спрашивать подтверждений, удалить все, что возможно
`-i`	Запрашивать подтверждение на каждый удаляемый файл
`-r`	Выполнить удаление рекурсивно, включая каталоги

Примеры:

Рекурсивное удаление каталога `.mozilla/firefox/gb16cbr1.default/Cache` в принудительном режиме:

```
$ rm -rf .mozilla/firefox/gb16cbr1.default/Cache
```

7.1.8. Команда mkdir

Синтаксис:

```
mkdir [-p] каталог
```

Описание:

Создать *каталог*[7].

Опции:

`-p`	Создаются также и все несуществующие к этому моменту родительские каталоги указанного места

Примеры:

Создать каталог `/tmp/a/b/c`:

```
$ mkdir -p /tmp/a/b/c
$ ls -R /tmp
/tmp:
a             gconfd-guest

/tmp/a:
b

/tmp/a/b:
c

/tmp/a/b/c:
```

7.1.9. Команда rmdir

Синтаксис:

```
rmdir [-p] каталог
```

Описание:

Удалить пустой *каталог*.

Опции:

`-p`	Удалить также и все пустые родительские каталоги

Примеры:

Удалить каталог `/tmp/a/b/c`:

```
$ ls -R /tmp
/tmp:
a             gconfd-guest

/tmp/a:
b

/tmp/a/b:
c

/tmp/a/b/c:

$ rmdir -p /tmp/a/b/c
 $ ls -R /tmp
/tmp:
 gconfd-guest
```

Лабораторная работа № 9

Тема: «Linux. управление пользователями, работа с учетными записями пользователей».

Введение

Учётная запись пользователя - это запись, которая содержит сведения, необходимые для идентификации пользователя при подключении к системе, а также информацию для авторизации и учёта. Это имя пользователя и пароль (или другое аналогичное средство аутентификации - например, биометрические характеристики). Пароль или его аналог, как правило, хранится в зашифрованном или хэшированном виде (в целях его безопасности).

Для повышения надёжности могут быть, наряду с паролем, предусмотрены альтернативные средства аутентификации - например, специальный секретный вопрос (или несколько вопросов) такого содержания, что ответ может быть известен только пользователю. Такие вопросы и ответы также хранятся в учётной записи.

Учётная запись может содержать следующие дополнительные анкетные данные о пользователе:

- имя;
- фамилию;
- отчество;
- псевдоним (ник);

- пол;
- национальность;
- расовую принадлежность;
- вероисповедание
- группу крови;
- резус-фактор;
- возраст;
- дату рождения;
- адрес электронной почты;
- домашний адрес;
- рабочий адрес;
- нетмейловый адрес;
- номер домашнего телефона;
- номер рабочего телефона;
- номер мобильного телефона;
- номер ICQ;
- идентификатор Skype, ник в IRC;
- другие контактные данные систем обмена мгновенными сообщениями;
- адрес домашней страницы и/или блога в Интернете или интранете;
- сведения о хобби;
- сведения о круге интересов;
- сведения о семье;
- сведения о перенесённых болезнях;
- сведения о политических предпочтениях;
- и многое другое

Конкретные категории данных, которые могут быть внесены в такую анкету, определяются администраторами системы.

Учётная запись может также содержать одну или несколько фотографий или аватар пользователя. Учётная запись пользователя также может учитывать различные статистические характеристики поведения пользователя в системе: давность последнего входа в систему, продолжительность последнего пребывания в системе, адрес использованного при подключении компьютера, интенсивность использования системы, суммарное и (или) удельное количество определённых операций, произведённых в системе, и так далее.

Создание учетных записей пользователей

В операционной системе **Windows 7** можно создавать несколькими способами как учетные записи пользователей для компьютеров, состоящих в рабочих группах, так и учетные записи пользователей для компьютеров, которые входят в состав домена. Домены, рабочие группы и домашние группы представляют разные методы организации компьютеров в сети. Основное их различие состоит в том, как осуществляется управление компьютерами и другими ресурсами.

Рабочая группа - это группа компьютеров, подключенных к сети, которые совместно используют ресурсы. При настройке сети операционная система Windows автоматически создает рабочую группу и присваивает ей имя по умолчанию.

Домен - это группа компьютеров одной сети, имеющих единый центр, использующий единую базу пользователей, единую групповую и локальную политики, единые

параметры безопасности, ограничение времени работы учётной записи и прочие параметры, значительно упрощающие работу системного администратора организации, если в ней эксплуатируется большое число компьютеров.

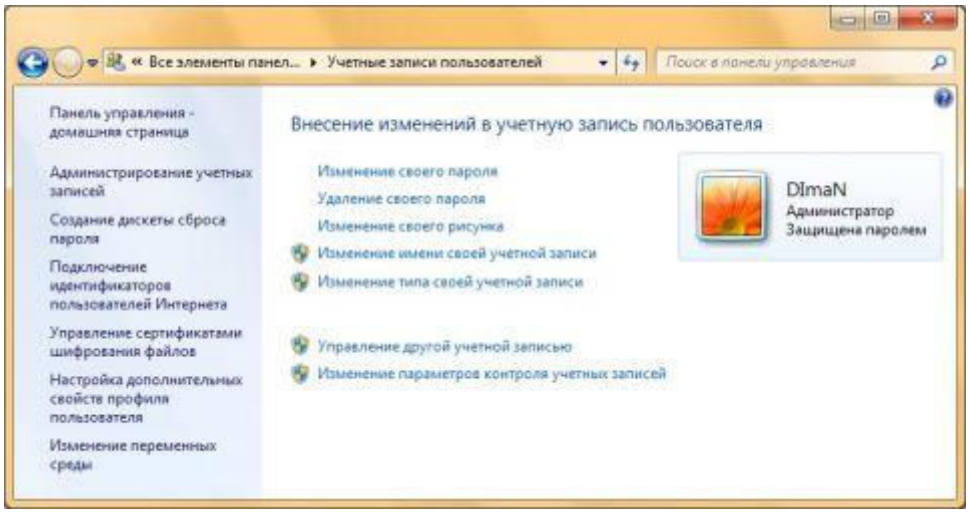
Создание учетных записей пользователей для компьютеров, состоящих в рабочей группе

В операционной системе Windows 7 для компьютеров, которые состоят в рабочей или домашней группе, учетные записи можно создавать следующими способами:

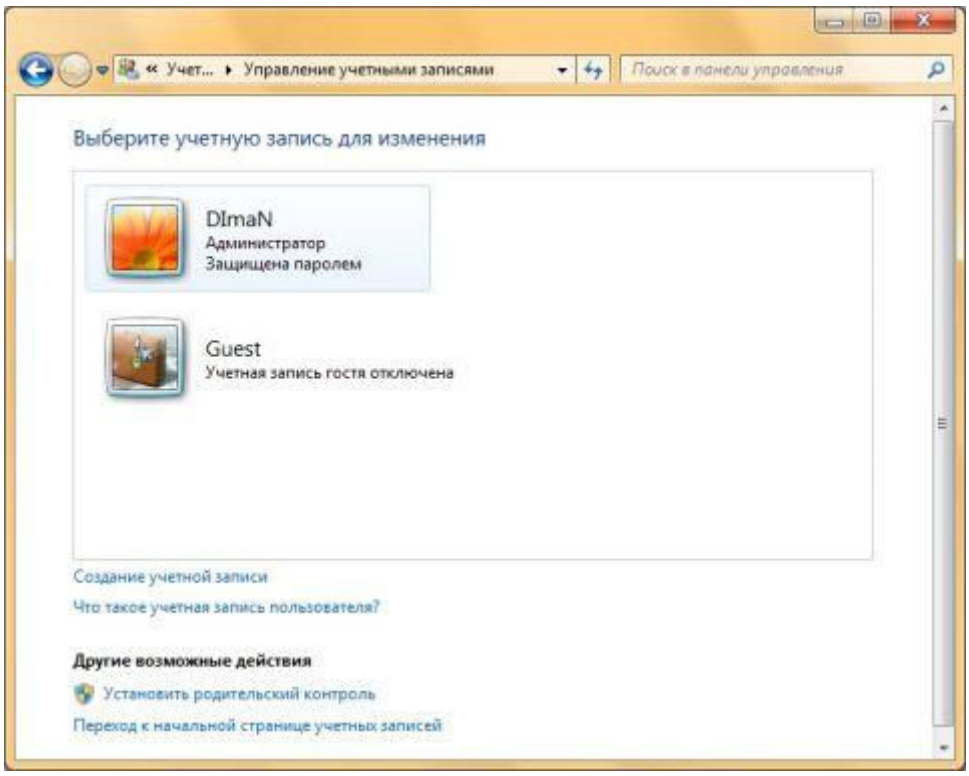
Создание учетной записи при помощи диалога "Управление учетными записями пользователей"

Для того чтобы создать учетную запись при помощи диалога **"Учетные записи пользователей"**, нужно сделать следующее:

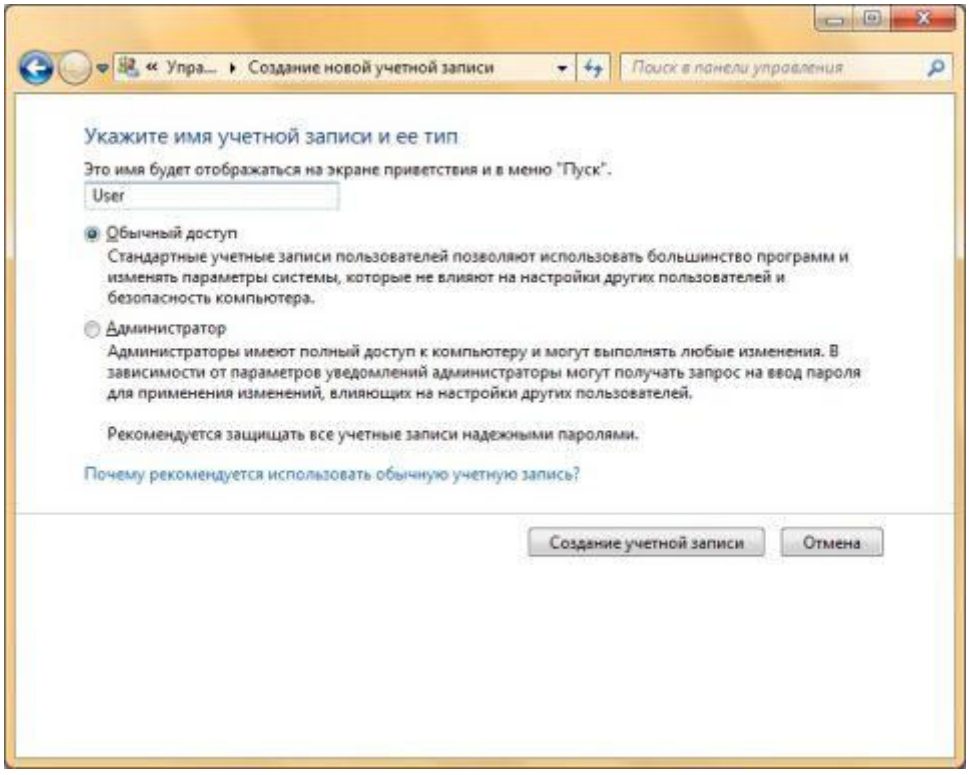
1. Нажмите на кнопку **"Пуск"** для открытия меню, откройте **"Панель управления"** и из списка компонентов панели управления выберите **"Учетные записи пользователей"**;



2. В диалоге **"Учетные записи пользователей"** перейдите по ссылке **"Управление другой учетной записью"**, а затем нажмите на **"Создание учетной записью"**;



3. Здесь нужно будет ввести имя для учетной записи, выбрать тип учетной записи и нажать на кнопку **"Создание учетной записи"**;



Имя пользователя не должно совпадать с любым другим именем пользователя или группы на данном компьютере. Оно может содержать до 20 символов верхнего или нижнего регистров, за исключением следующих: " / \ [ ] : ; / =, + * ? <> @, а также имя пользователя не может состоять только из точек и пробелов.

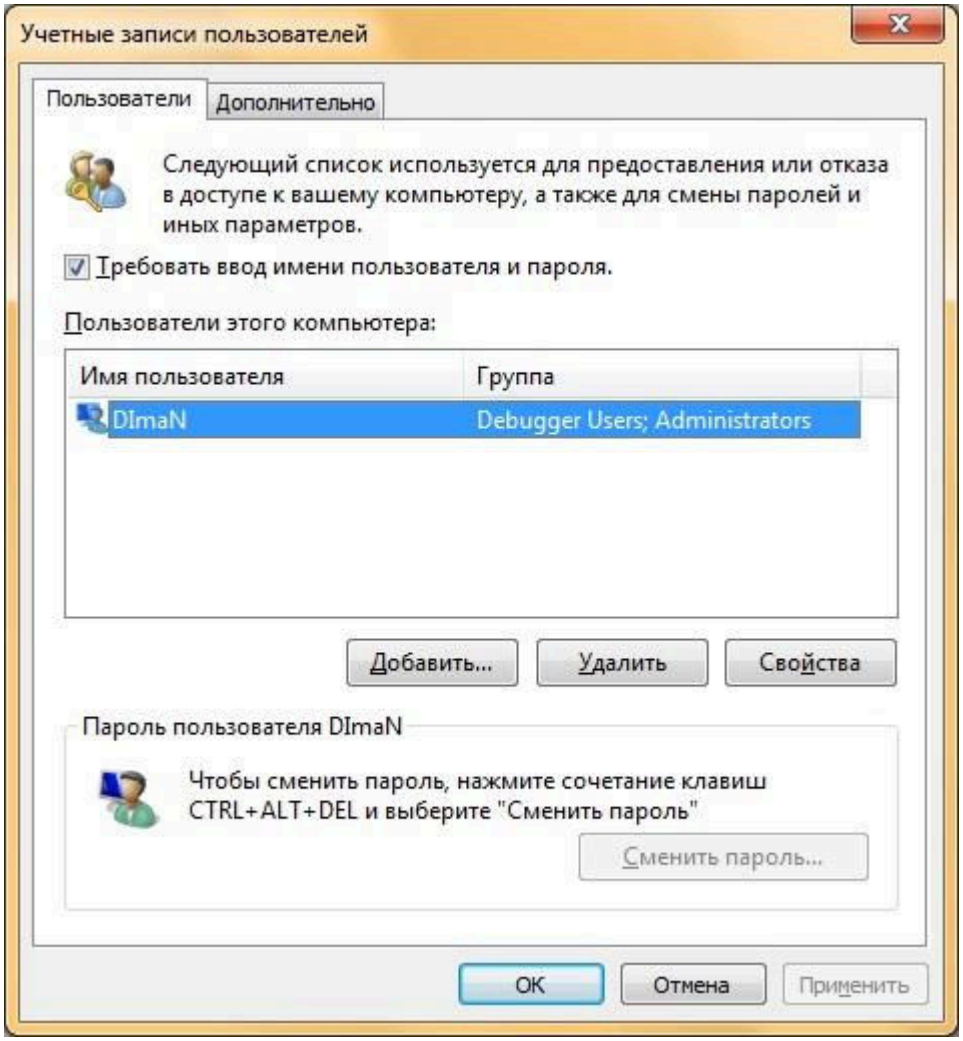
В этом диалоге, можно выбрать одну из двух типов учетных записей: **"обычные учетные записи пользователей"**, которые предназначены для повседневной работы или **"учетные записи администратора"**, которые предоставляют полный контроль над компьютером и применяются только в необходимых случаях.

Создание учетной записи при помощи диалога "Учетные записи пользователей"

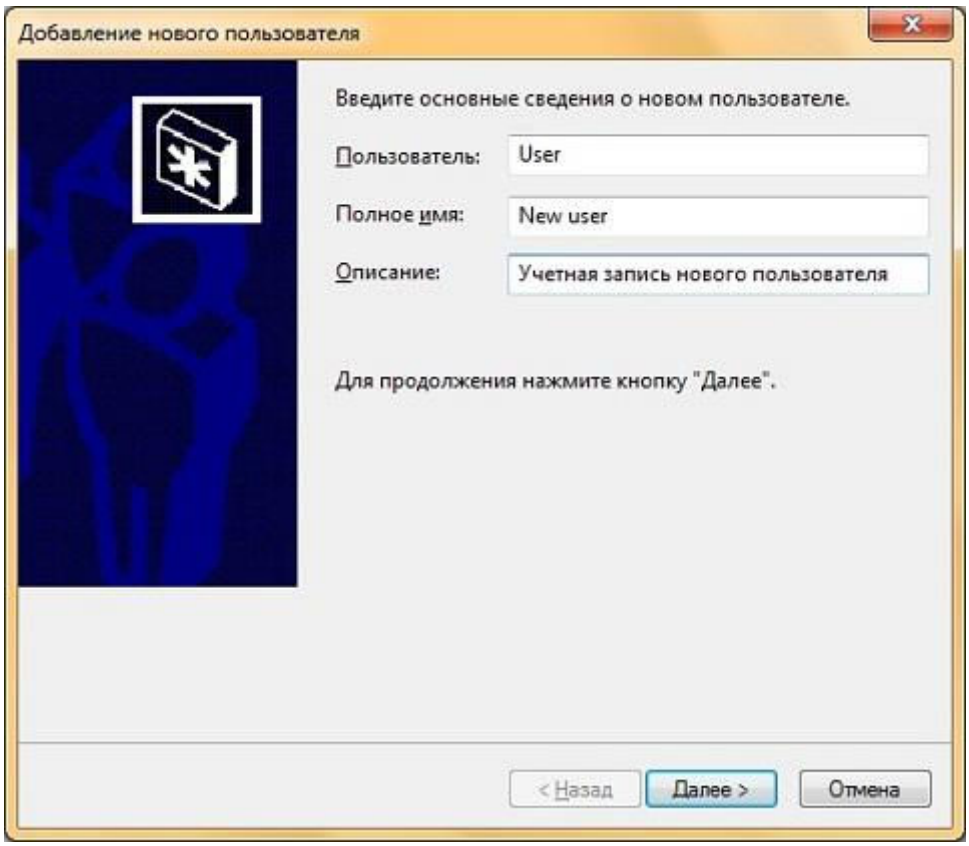
Доступный через панель управления диалог **"Управление учетными записями пользователей"** имеет очень серьезное ограничение: оно предлагает на выбор только учетные записи типа Обычный доступ или Администратор . Для того чтобы при создании нового пользователя его можно было поместить в какую-либо определенную группу, нужно сделать следующее:

1. Воспользоваться комбинацией клавиш +R для открытия диалога **"Выполнить"**;

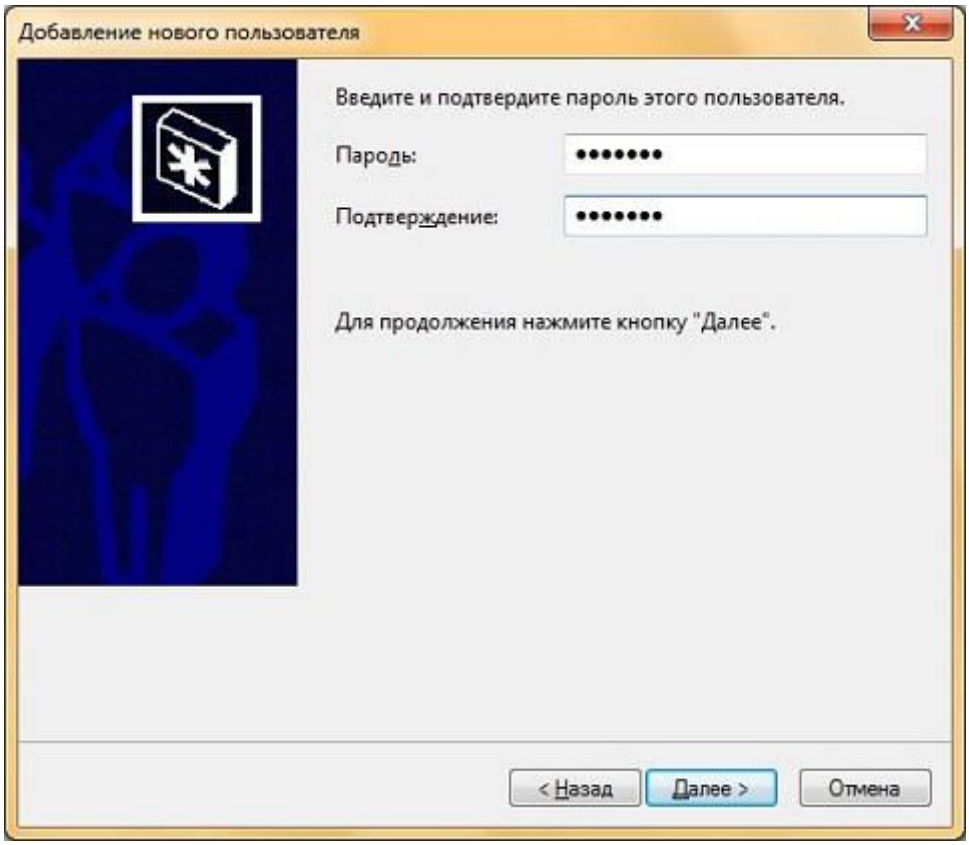
2. В диалоговом окне **"Выполнить"**, в поле **"Открыть"** введите control userpasswords2 и нажмите на кнопку **"ОК"**;



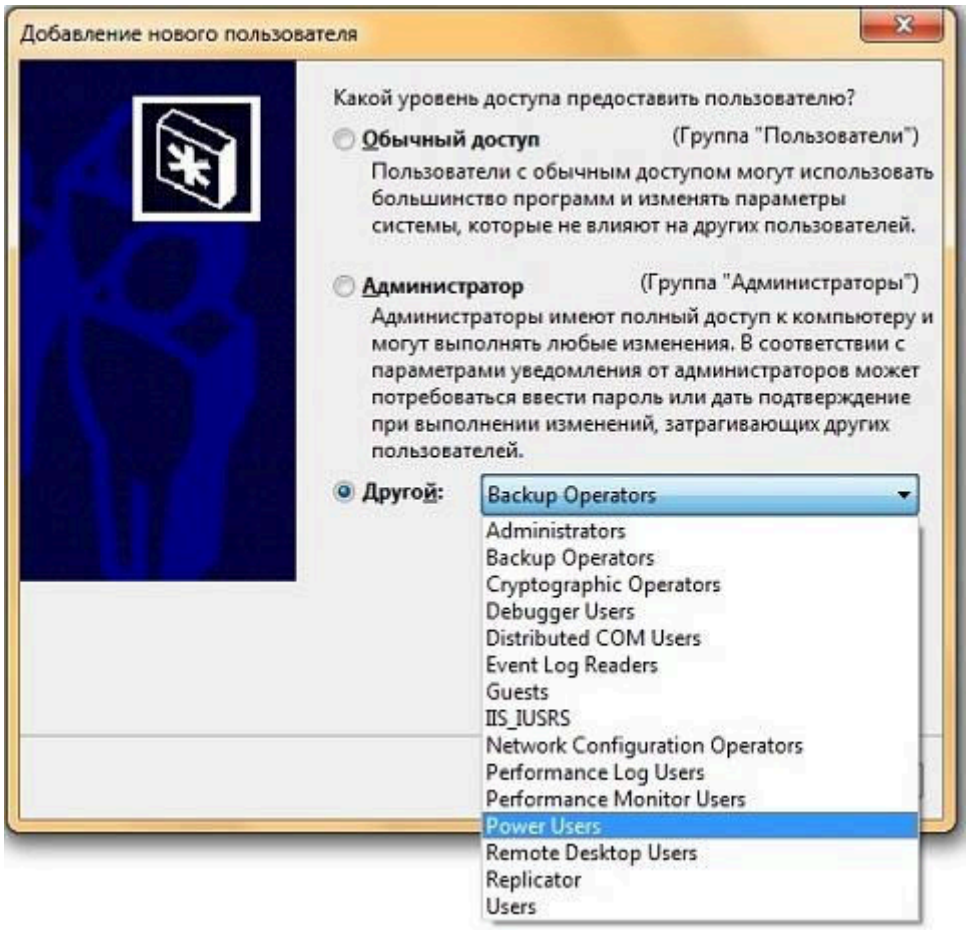
3. В диалоговом окне **"Учетные записи пользователей"** нажмите на кнопку **"Добавить"** для запуска мастера добавления нового пользователя;



4. В появившемся диалоговом окне **"Добавление нового пользователя"** введите имя пользователя. Поля **"Полное имя"** и **"Описание"** не являются обязательными, то есть их можно заполнять при желании. Нажимаем на кнопку **"Далее"**;



5. В диалоге **"Введите и подтвердите пароль этого пользователя"** введите пароль для данной учетной записи, а затем продублируйте его в поле **"Подтверждение"**, после чего нажмите на кнопку **"Далее"**;



6. Это последний диалог мастера добавления нового пользователя. Здесь необходимо установить переключатель, определяющий группу безопасности, к которой должна относиться данная учетная запись пользователя. Можно выбрать одну из следующих групп: Обычный доступ, Администратор или Другой. Последний переключатель стоит использовать в том случае, если нужно отнести пользователя к какой-то другой группе, созданной по умолчанию в операционной системе Windows 7.

В следующем списке перечислены 15 встроенных групп операционной системы Windows 7. Эти права назначаются в рамках локальных политик безопасности:

- **Administrators (Администраторы)**. Пользователи, входящие в эту группу, имеют полный доступ на управление компьютером и могут при необходимости назначать пользователям права пользователей и разрешения на управление доступом. По умолчанию членом этой группы является учетная запись администратора. Если компьютер подключен к домену, группа **"Администраторы домена"** автоматически добавляется в группу **"Администраторы"**. Эта группа имеет полный доступ к управлению компьютером, поэтому необходимо проявлять осторожность при добавлении пользователей в данную группу;
- **Backup Operators (Операторы архива)**. Пользователи, входящие в эту группу, могут архивировать и восстанавливать файлы на компьютере независимо от любых разрешений, которыми защищены эти файлы. Это обусловлено тем, что право

выполнения архивации получает приоритет над всеми разрешениями. Члены этой группы не могут изменять параметры безопасности.

- **Cryptographic Operators (Операторы криптографии)**. Членам этой группы разрешено выполнение операций криптографии.
- **Debugger Users (Группа удаленных помощников)**. Члены этой группы могут предлагать удаленную помощь пользователям данного компьютера.
- **Distributed COM Users (Пользователи DCOM).** Членам этой группы разрешено запускать, активировать и использовать объекты DCOM на компьютере.
- **Event Log Readers (Читатели журнала событий)**. Членам этой группы разрешается запускать журнал событий Windows.
- **Guests (Гости)**. Пользователи, входящие в эту группу, получают временный профиль, который создается при входе пользователя в систему и удаляется при выходе из нее. Учетная запись **"Гость"** (отключенная по умолчанию) также является членом данной встроенной группы.
- **IIS_IUSRS**. Это встроенная группа, используемая службами IIS.
- **Network Configuration Operators (Операторы настройки сети).** Пользователи, входящие в эту группу, могут изменять параметры TCP/IP, а также обновлять и освобождать адреса TCP/IP. Эта группа не имеет членов по умолчанию.
- **Performance Log Users (Пользователи журналов производительности).** Пользователи, входящие в эту группу, могут управлять счетчиками производительности, журналами и оповещениями на локальном или удаленном компьютере, не являясь при этом членами группы **"Администраторы"**.
- **Performance Monitor Users (Пользователи системного монитора)**. Пользователи, входящие в эту группу, могут наблюдать за счетчиками производительности на локальном или удаленном компьютере, не являясь при этом участниками групп **"Администраторы"** или **"Пользователи журналов производительности"**.
- **Power Users (Опытные пользователи)**. По умолчанию, члены этой группы имеют те же права пользователя и разрешения, что и учетные записи обычных пользователей. В предыдущих версиях операционной системы Windows эта группа была создана для того, чтобы назначать пользователям особые административные права и разрешения для выполнения распространенных системных задач. В этой версии операционной системы Windows учетные записи обычных пользователей предусматривают возможность выполнения большинства типовых задач настройки, таких как смена часовых поясов. Для старых приложений, требующих тех же прав опытных пользователей, которые имелись в предыдущих версиях операционной системы Windows, администраторы могут применять шаблон безопасности, который позволяет группе **"Опытные пользователи"** присваивать эти права и разрешения, как это было в предыдущих версиях операционной системы Windows.
- **Remote Desktop Users (Пользователи удаленного рабочего стола).** Пользователи, входящие в эту группу, имеют право удаленного входа на компьютер.
- **Replicator (Репликатор)**. Эта группа поддерживает функции репликации. Единственный член этой группы должен иметь учетную запись пользователя домена, которая используется для входа в систему службы репликации контроллера домена. Не добавляйте в эту группу учетные записи реальных пользователей.
- **Users (Пользователи)**. Пользователи, входящие в эту группу, могут выполнять типовые задачи, такие как запуск приложений, использование локальных и сетевых принтеров и блокировку компьютера. Члены этой группы не могут предоставлять общий доступ к папкам или создавать локальные принтеры. По умолчанию членами этой группы являются группы **"Пользователи домена"**, **"Проверенные**

пользователи" и **"Интерактивные"**. Таким образом, любая учетная запись пользователя, созданная в домене, становится членом этой группы.

Создание учетной записи при помощи оснастки "Локальные пользователи и группы"

Оснастка **"Локальные пользователи и группы"** расположена в компоненте **"Управление компьютером"**, представляющем собой набор средств администрирования, с помощью которых можно управлять одним компьютером, локальным или удаленным. Оснастка **"Локальные пользователи и группы"** служит для защиты и управления учетными записями пользователей и групп, размещенных локально на компьютере. Можно назначать разрешения и права для учетной записи локального пользователя или группы на определенном компьютере (и только на этом компьютере).

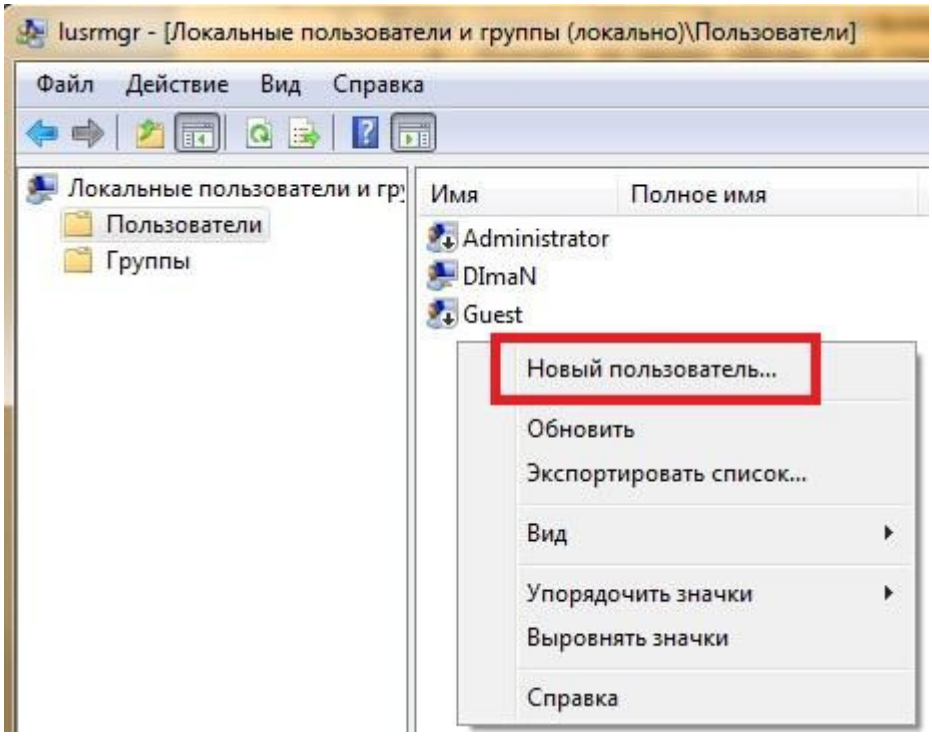
Использование оснастки **"Локальные пользователи и группы"** позволяет ограничить возможные действия пользователей и групп путем назначения им прав и разрешений. Право дает возможность пользователю выполнять на компьютере определенные действия, такие как архивирование файлов и папок или завершение работы компьютера. Разрешение представляет собой правило, связанное с объектом (обычно с файлом, папкой или принтером), которое определяет, каким пользователям, и какой доступ к объекту разрешен.

Для того чтобы создать локальную учетную запись пользователя при помощи оснастки **"Локальные пользователи и группы"**, нужно сделать следующее:

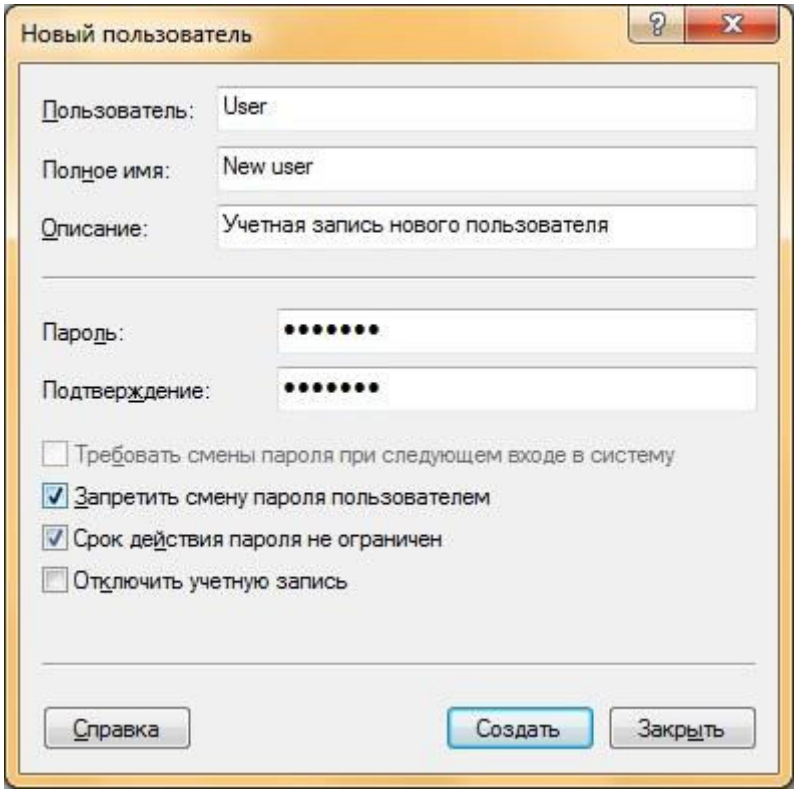
1. Откройте оснастку **"Локальные пользователи и группы"** одним из следующих способов:

- Нажмите на кнопку **"Пуск"** для открытия меню, откройте **"Панель управления"** и из списка компонентов панели управления выберите **"Администрирование"**, затем откройте компонент **"Управление компьютером"**. В **"Управлении компьютером"** откройте **"Локальные пользователи и группы"**;
- Открыть **"Консоль управления ММС"**. Для этого нажмите на кнопку **"Пуск"**, в поле поиска введите mmc, а затем нажмите на кнопку **"Enter"**. Откроется пустая консоль ММС. В меню **"Консоль"** выберите команду **"Добавить или удалить оснастку"** или воспользуйтесь комбинацией клавиш Ctrl+M. В диалоге **"Добавление и удаление оснасток"** выберите оснастку **"Локальные пользователи и группы"** и нажмите на кнопку **"Добавить"**. Затем нажмите на кнопку **"Готово"**, а после этого - кнопку **"ОК"**. В дереве консоли откройте узел **"Локальные пользователи и группы (локально)"**;
- Воспользоваться комбинацией клавиш +R для открытия диалога **"Выполнить"**. В диалоговом окне **"Выполнить"**, в поле **"Открыть"** введите lusrmgr.msc и нажмите на кнопку **"ОК"**;

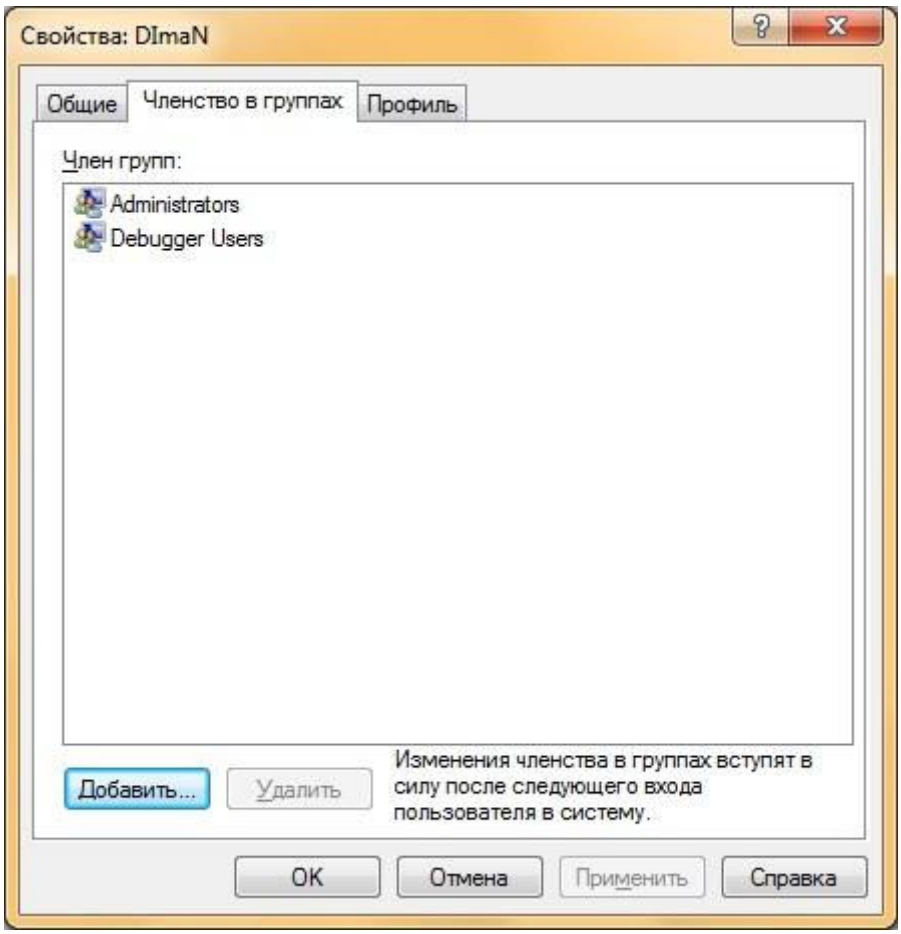
2. Откройте узел **"Пользователи"** и либо в меню **"Действие"**, либо из контекстного меню выбрать команду **"Новый пользователь"**;



3. В диалоговом окне **"Новый пользователь"** введите соответствующие сведения. Помимо указанных данных, можно воспользоваться следующими флажками: **Требовать смену пароля при следующем входе в систему**, **Запретить смену пароля пользователем**, **Срок действия пароля не ограничен**, **Отключить учетную запись** и нажать на кнопку **"Создать"**, а затем **"Закрыть"**.

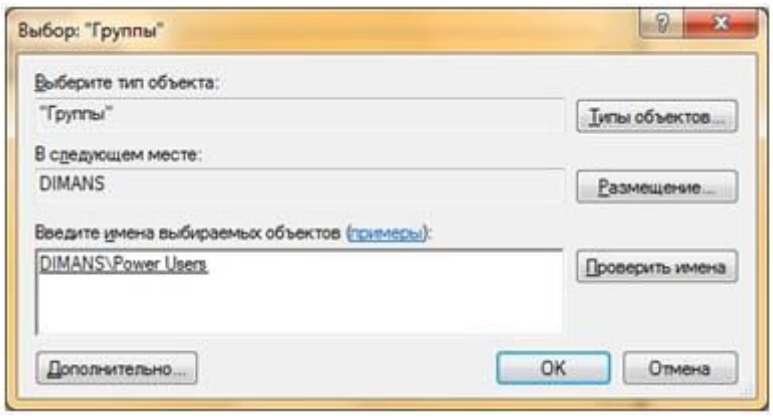


Для того чтобы добавить пользователя в группу, дважды щелкните имя пользователя для получения доступа к странице свойств пользователя. На вкладке **"Членство в группах"** нажмите на кнопку **"Добавить"**.

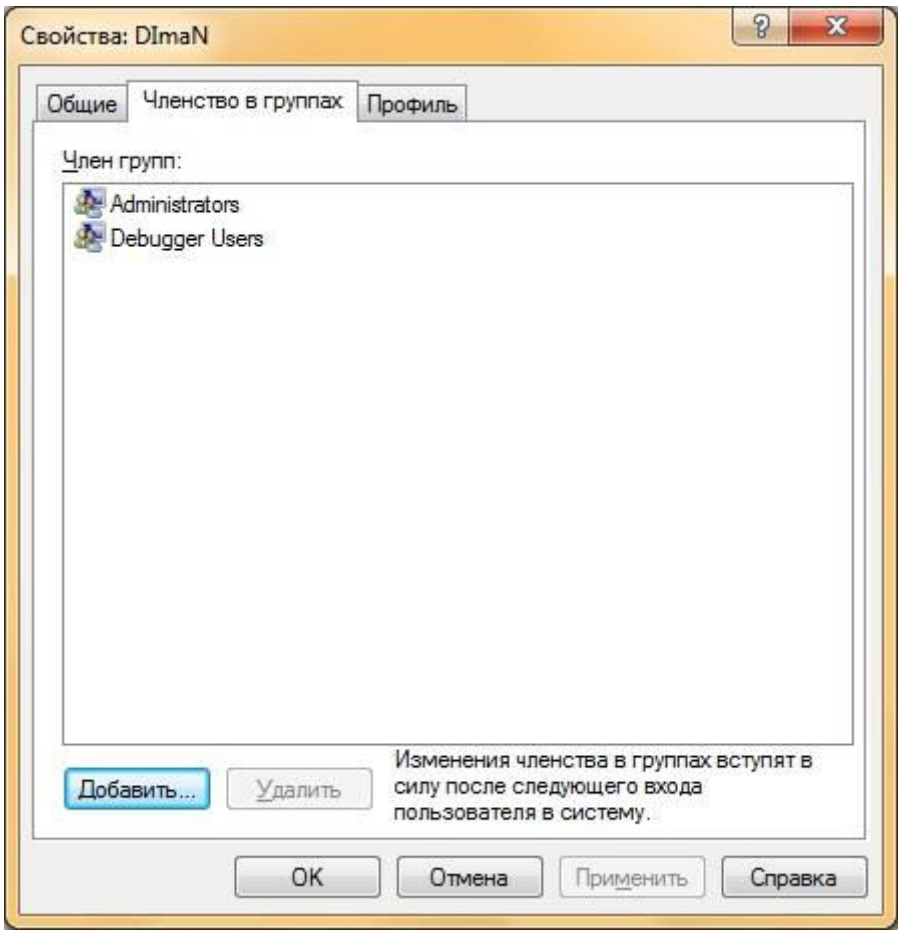


В диалоге **"Выбор группы"** можно выбрать группу для пользователя двумя способами:

1. В поле **"Введите имена выбираемых объектов"** введите имя группы и нажмите на кнопку **"Проверить имена"**, как показано на следующем скриншоте:



2. В диалоге **"Выбор группы"** нажмите на кнопку **"Дополнительно"**, чтобы открыть диалоговое окно **"Выбор группы"**. В этом окне нажмите на кнопку **"Поиск"**, чтобы отобразить список всех доступных групп, выберите подходящую группу и нажмите два раза на кнопку **"ОК"**.



Создание учетной записи при помощи командной строки

Помимо вышеперечисленных способов, учетные записи пользователей можно создавать, изменять и удалять при помощи командной строки. Для этого нужно выполнить следующие действия:

1. Запустите командную строку от имени администратора;

2. Для создания учетной записи при помощи командной строки используйте команду net user .

Команда net user используется для добавления пользователей, установки паролей, отключения учетных записей, установки параметров и удаления учетных записей. При выполнении команды без параметров командной строки отображается список учетных записей пользователей, присутствующих на компьютере. Информация об учетных записях пользователей хранится в базе данных учетных записей пользователей.

Пример команды:

net user User /add /passwordreq:yes /times:monday-friday, 9am-6pm/fullname:"New user"

Используемые параметры:

/add - этот параметр указывает, что необходимо создать новую учетную запись;

/passwordreq - этот параметр отвечает за то, чтобы при первом входе в систему пользователь сменил свой пароль;

/times - этот параметр определяет, сколько раз пользователю разрешено входить в систему. Здесь можно указывать как единичные дни, так и целые диапазоны (например Sa или M-F). Для указания времени допускается как 24-часовый формат, так и 12-часовый формат;

/fullname - этот параметр идентичен полю "Полное имя" при создании пользователя предыдущими способами.

Создание учетных записей пользователей для компьютеров, состоящих в домене

В серверной операционной системе Windows Server 2008 или Windows Server 2008 R2 в домене Active Directory учетные записи пользователей можно создавать шестью способами. Рассмотрим подробно каждый из них:

Создание пользователей при помощи оснастки "Active Directory - пользователи и компьютеры"

Для создания нового пользователя в домене при помощи оснастки **"Active Directory - пользователи и компьютеры"** нужно сделать следующее:

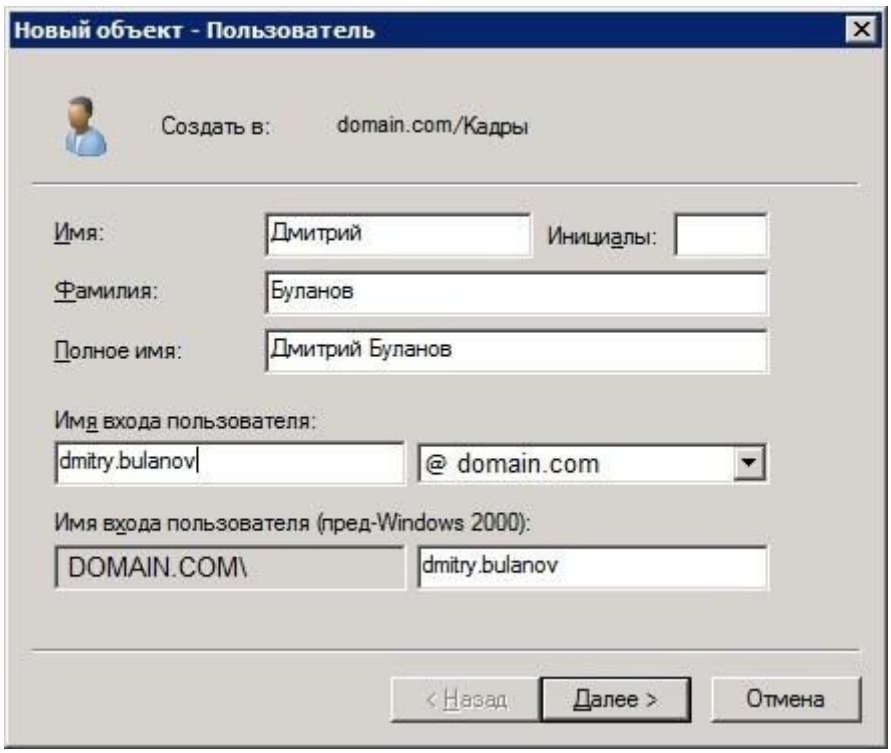
1. Открыть оснастку **"Active Directory - пользователи и компьютеры"**;

2. В дереве консоли разверните узел, предоставляющий домен и найдите контейнер, в котором нужно создать учетную запись пользователя;

3. Нажмите на подразделение или контейнер правой кнопкой мыши, выберите опцию **"Создать"** и примените команду **"Пользователь"**;

4. В диалоговом окне **"Новый объект - пользователь"** введите в поле **"Имя"** - имя пользователя, в поле **"Инициалы"** - его инициалы, в поле **"Фамилия"** - фамилию пользователя. Поле **"Полное имя"** должно заполниться автоматически, согласно CN пользователя. В поле **"Имя входа"** введите имя входа пользователя в систему и в раскрывающемся списке выберите суффикс основного имени пользователя, который будет прикреплен к имени входа с символом @. В поле **"Имя входа пользователя (пред-Windows 2000)"** введите имя входа для систем, предшествующих Windows 2000, так называемое низкоуровневым именем входа. Нажать на кнопку **"Далее"**.

5. В следующем диалоге введите пароль для данной учетной записи, а затем продублируйте его в поле **"Подтверждение"** и установите нужные флажки, после чего нажмите на кнопку **"Далее"**.



6. В последнем диалоге можно просмотреть введенные параметры и нажать на кнопку **"Готово"** для создания нового пользователя.

Создание пользователей с помощью командной строки

Для автоматизации создания любых объектов в домене Active Directory можно использовать команду DSADD USER UserDN, при помощи которой можно создавать объекты пользователей и принимать параметры, указывающие его свойства. Нового пользователя при помощи командной строки можно создать следующим образом:

dsadd user "CN=Дмитрий Буланов, OU=Кадры, DC=server, DC=com" -samid Dmitry.bulanov -pwd * -mustchpwd yes -profile \\server01\Profiles\dmitry.bulanov -fn "Дмитрий" -ln "Буланов" -display "Дмитрий Буланов" -upn Dmitry.bulanov@server.com

Определение используемых параметров:

Samid - указывает имя входа пользователя;

Pwd - этот параметр определяет пароль для учетной записи пользователя. Если указывать символ *, то будет предложено ввести пароль пользователя;

Mustchpwd - указывает, что пользователь должен изменить свой пароль при следующем входе в систему;

Profile - указывает путь к профилю учетной записи пользователя;

Fn - указывает имя пользователя;

Ln - указывает фамилию пользователя;

Display - указывает отображаемое имя пользователя;

Upn - указывает имя входа пользователя (пред-Windows 2000).

Импорт пользователей с помощью команды CSVDE

Утилита командной строки CSVDE позволяет импортировать и экспортировать объекты Active Directory в виде текстового файла с разделенными запятыми (Comma-Separated Values, *.csv). Эти файлы можно создавать и изменять при помощи таких программ, как Блокнот или, например, Microsoft Office Excel. Эта утилита - способ автоматизации создания учетных записей пользователя на основе информации пользователей из базы данных Excel и Microsoft Office Access. Команда импортирует текстовый файл, в котором строка определяет атрибуты импорта с помощью их имен LDAP. Синтаксис команды следующий:

Csvde -i -f имя_файла -k

Параметр i указывает режим импорта, а параметр k используется для игнорирования ошибок.

CSV файл должен выглядеть следующим образом:

DN, objectClass, sAMAccountName, sn, givenName, userPrincipalName

"cn=Дмитрий Буланов, ou=Пользователи, dc=server, dc=com", user, Dmitry.bulanov, Буланов, Дмитрий, dmitry.bulanov@server.com

Импортировать пароли при помощи команды CSVDE нельзя.

Импорт пользователей с помощью команды LDIFDE

При помощи команды LDIFDE также можно импортировать и экспортировать объекты Active Directory. В данном случае используется стандарт файлового формата LDIF (Lightweight Directory Access Protocol Data Interchange Format). Этот файловый формат состоит из блока строк, которые вместе образуют одну операцию. Разные операции разделяются пустой строкой. Каждая строка содержит имя атрибута, а после него должно стоять двоеточие со значением атрибута. Далее можно увидеть листинг LDIF файла:

DN: CN=Дмитрий Буланов, OU=пользователи, DC=server, DC=com

changeType: add

CN: Дмитрий Буланов

objectClass: user

sAMAccountName: Dmitry.bulanov

userPrincipalName: Dmitry.bulanov@server.com

givenName: Дмитрий

sn: Буланов

displayName: Дмитрий Буланов

Файл можно создавать в такой программе как Блокнот, но сохранять его нужно с расширением *.ldf. в командной строке введите следующее:

Ldifde -i -f имя_файла -k

Создание пользователей с помощью Windows PowerShell

При помощи Windows PowerShell для создания пользователя в Active Directory пользователя можно создать следующим образом:

1. Подключитесь к контейнеру, в котором будет создан объект;

2. Примените метод Create совместно с классом и отличительным именем RDN;

3. Заполните атрибуты при помощи метода Put;

4. Подтвердите изменения при помощи метода SetInfo.

Далее можно увидеть листинг скрипта PowerShell - *.ps1-файла:

```
$ObjOU=[ADSI]"LDAP://OU=Пользователи, DC=server, DC=com"

$ObjUser=$ObjOU.Create("user", "CN=Дмитрий Буланов")

$ObjUser.Put("sAMAccountName", "dmitry.bulanov")

$ObjUser.Put("userPrincipalName", "dmitry.bulanov @server.com")

$ObjUser.Put("displayName", "Дмитрий Буланов")

$ObjUser.Put("givenName", "Дмитрий")

$ObjUser.Put("sn", "Буланов")

$ObjUser.Put("description", "Тестировщик программного обеспечения")

$ObjUser.Put("company", "Company")

$ObjUser.Put("department", "Отдел тестирования")

$ObjUser.Put("title", "Тестировщик программного обеспечения")

$ObjUser.Put("mail", " dmitry.bulanov @server.com ")

$ObjUser.Put("c", "UA")
$ObjUser.Put("postalCode", "73003")
$ObjUser.Put("st", "Херсон")
$ObjUser.Put("l", "Херсон")

$ObjUser.Put("streetAddress", "Улица")
```

```
$ObjUser.Put("postOfficeBox", "Номер дома")

$ObjUser.SetInfo()

$ObjUser.SetPassword("P@ssword")

//$ObjUser.Put("pwdLastSet", 0) - для смены пароля при следующем старте

$ObjUser.psbase.InvokeSet("AccountDisabled", $false)

$ObjUser.SetInfo()
```

Можно вводить все строки вручную, а можно использовать *.ps1-файлы для автоматизации создания новых пользователей. Для того, чтобы разрешить Windows PowerShell открывать скрипты, введите следующую команду:

Set-ExecutionPolicy RemoteSigned

Политика выполнения указывает сценарии, которые можно запускать. После назначения политики выполнения можно запустить сценарий, но если указывать для запуска только имя сценария, то может возникнуть ошибка. Чаще всего нужно будет указывать еще и путь к самому сценарию.

Создание пользователей с помощью VBScript

В связи с тем, что VBScript также как и Windows PowerShell использует интерфейс ADSI для манипулирования объектами в Active Directory, процесс создания пользователя в VBScript идентичен созданию пользователя в Windows PowerShell. Прежде всего, сценарий подключается к контейнеру OU, в котором будет создан пользователь. После чего сценарий применит к объекту ADSI инструкцию GetObject. При присвоении объекта переменной, для создания объектной ссылки используется инструкция Set.

После этого активизируется метод Create для создания объекта конкретного класса так же, как и в PowerShell. Далее используется метод Put, но аргументы заключаются в круглые скобки. Последняя строка - идентична Windows PowerShell. Пример скрипта:

```
Set objOU=GetObject("LDAP: //OU=Пользователи, DC=server, DC=com")

Set objUser=objOU.Create("user", "CN=Дмитрий Буланов")

objUser.Put "sAMAccountName", " dmitry.bulanov"

objUser.Put "displayName", " Дмитрий Буланов"

objUser.Put "givenName", " Дмитрий"

objUser.Put "sn", " Буланов"

ObjUser.SetInfo()
```

Лабораторная работа № 10

Тема: Установка и удаление программ и оборудования в ос Windows

Цель: получение практических навыков в конфигурировании ОС Windows для ПК в соответствии с его функциональным назначением.

Теоретические сведения:

Установка и удаление программ.

С помощью данной утилиты, которая вызывается командой **Пуск – Панель управления – Установка и удаление программ**можно установить новую программу или корректно удалить ненужную программу. (Рис. 1). Обычно установочный файл имеет имя Setup.exe.

Рис. 1. Установка и удаление программ.

Установка и удаление компонентов Windows.

Для установки и удаления компонентов Windows выполните следующие действия:

1. В окне **Установка и удаление программ**щелкните **Установка компонентов Windows.**На экране появится окно с перечнем компонентов (Рис. 2). Дважды щелкните на название компонента, чтобы узнать, какие из входящих в него программ установлены, а какие нет.

2. Для установки компонента отметьте его флажком и выберите **Далее**.

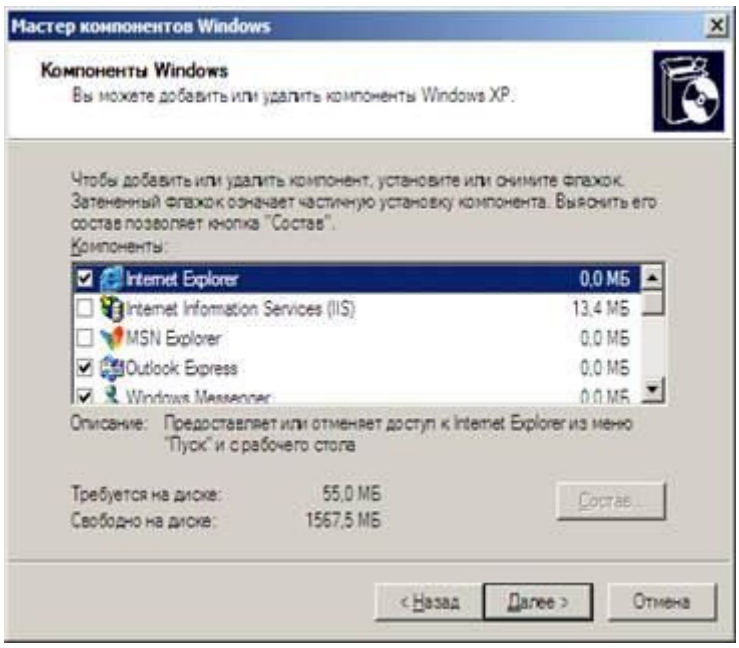


Рис. 2 Установка компонентов

В случае повреждения системного реестра операционная система Windows может не загрузиться. Единственным выходом кажется переустановка Windows, процесс долгий. Но этого легко избежать, используя универсальный способ - резервное копирование.

Чтобы не знать никаких проблем с восстановлением Windows после не удачных экспериментов с реестром, сделайте следующее.

· Скопируйте инсталянт Windows на жесткий диск и сделайте загрузочную дискету. В любом случае восстановление Windows придется вести из режима командной строки, а в таком случае доступ к CD-ROM будет невозможен, если не загрузить драйвера для него.

· Установите на жесткий диск в новую папку в корневом каталоге Norton Commander для MS DOS либо Volkov Commander либо Far manager.

· Найдите в каталоге с операционной системой файлы System.dat и User.dat. Скопируйте их, в корневой каталог или любой другой. По окончании процесса список резервов должен выглядеть так:

-папка WinXp в корневом каталоге с инсталянтом Windows;

-папка VC в корневом каталоге с Volkov Commander;

-файлы system.da_ и user.da_ в корневом каталоге;

-файлы system.da_ и user.da_ в каталоге VC.

Если ваш компьютер сконфигурирован для работы на нем нескольких пользователей, то алгоритм действий будет такой же, только к копируемым файлам еще придется прибавить свой User.dat - из каталога Profiles\Имя. Если катастрофа все же происходит, и при загрузке операционной системы вы видите сообщение: "Системный реестр поврежден",

или "Невозможно загрузить драйвер…", или что-нибудь подобное, то действия должны быть следующими:

1. Нажать кнопку ***Reset***.

2. При появлении строчки Starting Windows XP нажать кнопку ***F8***. В появившемся меню выберите пункт ***Command Prompt Only*** и дождитесь появления системного приглашения. Если у вас компьютер настроен так, что на F8 он не реагирует, то вставьте в дисковод загрузочную дискету и загрузитесь с нее.

3. С помощью команд "***cd имя_каталога***" (переход в нужный каталог) или "cd .." (переход на один уровень выше) доберитесь до директории с Norton и запустите оболочку.

4. Зайдите для начала в папку с операционной системой, сотрите там файлы ***System.dat*** и ***User.dat***, а затем переименуйте файлы ***System.da0*** и ***User.da0*** в ***System.dat*** и ***User.dat***. Для начала надо попробовать использовать автоматически зарезервированные копии - там сохранены самые последние настройки.

5. Перезагрузите компьютер кнопкой ***Reset*** (без дискеты). Если Windows загружается без проблем, то все нормально, продолжайте работать. Если же нет, то пришло время воспользоваться резервной копией реестра.

6. Повторите шаги 1-3. Но теперь уже скопируйте ваши резервные файлы реестра (в вышеприведенном примере - ***system.da_*** и ***user.da_***) из корневого каталога в папку с операционной системой, сотрите там файлы ***System.dat*** и ***User.dat***, а затем переименуйте файлы ***System.da_*** и ***User.da_*** в ***System.dat*** и ***User.dat***. Снова перезагрузитесь.

7. Если опять операционная система не загружается, попробуйте заменить, как описано в шаге 6, файлы ***System.dat*** и ***User.dat*** из каталога с операционной системой файлами из второй резервной копии, если она есть.

Лабораторная работа №11

Тема: Работа с виртуальной машиной в ос Windows XP

Цель работы: Изучение основных понятий о виртуальных машинах для их практического применения.

Теоретические сведения:

1. Виртуальные машины в целом

1.1. Определение и понятие

Чтобы построить полный взгляд на виртуальные машины, разберем для начала, а что такое виртуальная машина?

Виртуальная машина — программная или аппаратная среда, исполняющая некоторый код (например, байт-код, шитый код, p-код или машинный код реального процессора), или спецификация такой системы (например: «виртуальная машина языка программирования Си»). [Википедия]

Для сравнения приведем несколько других определенней, а именно: Виртуальная машина — это полностью изолированный программный контейнер, способный выполнять собственную операционную систему и приложения, как физический компьютер. Виртуальная машина работает абсолютно так же, как физический компьютер, и содержит собственные виртуальные (т.е. программные) ЦП, ОЗУ, жесткий диск и сетевую интерфейсную карту (NIC).

Проще говоря, виртуальная машина – это программа, которую вы запускаете из своей операционной системы. Программа эмулирует реальную машину. На виртуальные машины, как и на реальные, можно ставить операционные системы. У неё есть BIOS, отведенное место на вашем жестком диске, сетевые адаптеры для соединения с реальной машиной, сетевыми ресурсами или другими виртуальными машинами.

1.2. Преимущества и недостатки виртуальных машин

1.2.1. Преимущества виртуальных машин

Приведу вам несколько преимуществ использования виртуальных машин:

1. Приведу самый просто пример. Нынче, как мы знаем, вышли новые операционные системы. Windows Vista и Windows 7. И как многие из вас убедились, некоторые приложения, в частности игры, на них не работают. Так в чём проблема? Когда можно установить виртуальную машину с, допустим, операционной системой Windows XP. И всё прекрасно будет работать.
2. Второй пункт можно отнести к злобным хакерам или просто к компьютерным хулиганам. Имеется в виду, что на виртуальной машине вы можете спокойно написать вирус или вредоносное программное обеспечение, которое сможет повредить вам лишь гостевую операционную систему виртуальной машины.
3. Третий пункт можно было отнести ко второму. А именно то, что на виртуальную машину вы можете ставить любое ПО, не опасаясь чего-либо. Вы можете экспериментировать с различными настройками и прочее.

4. Ну и одно из самых главных это то, что вы можете легко изучать новые операционные системы, не стирая свою старую.

Это конечно далеко не все преимущества виртуальных машин. Каждый пользователь может сам придумать, для чего ему нужна виртуальная машина.
Перед возможностью установки нескольких хостовых операционных систем на один компьютер с их раздельной загрузкой, виртуальные машины имеют следующие неоспоримые преимущества:

1. Возможность работать одновременно в нескольких системах, осуществлять сетевое взаимодействие между ними.
2. Возможность сделать «снимок» текущего состояния системы и содержимого дисков одним кликом мыши, а затем в течение очень короткого промежутка времени вернуться в исходное состояние.
3. Простота создания резервной копии операционной системы (не надо создавать никаких образов диска, всего лишь требуется скопировать папку с файлами виртуальной машины).
4. Возможность иметь на одном компьютере неограниченное число виртуальных машин с совершенно разными операционными системами и их состояниями.
5. Отсутствие необходимости перезагрузки для переключения в другую операционную систему.

1.2.2. Недостатки виртуальных машин

Тем не менее, несмотря на все преимущества, виртуальные машины также имеют и свои недостатки:

1. Потребность в наличии достаточных аппаратных ресурсов для функционирования нескольких операционных систем одновременно.
2. Операционная система работает несколько медленнее в виртуальной машине, нежели на «голом железе». Однако, в последнее время показатели производительности гостевых систем значительно приблизились к показателям физических ОС (в пределах одних и тех же ресурсов), и вскоре, за счет улучшения технологий реализации виртуальных машин, производительность гостевых систем практически будет равна реальным.
3. Существуют методы определения того, что программа запущена в виртуальной машине (в большинстве случаев, производители систем виртуализации сами предоставляют такую возможность). Вирусописатели и распространители вредоносного программного обеспечения, конечно же, в курсе этих методов и в последнее время включают в свои программы функции обнаружения факта запуска в виртуальной машине, при этом никакого ущерба вредоносное ПО гостевой системе не причиняет.
4. Различные платформы виртуализации пока не поддерживают полную виртуализацию всего аппаратного обеспечения и интерфейсов. В последнее время количество поддерживаемого аппаратного обеспечения стремительно растет у всех производителей платформ виртуализации. Помимо основных устройств компьютера, уже поддерживаются сетевые адаптеры, аудиоконтроллеры, интерфейс USB 2.0, котроллеры портов COM и LPT и приводы CD-ROM. Но хуже всего обстоят дела с виртуализацией видеоадаптеров и поддержкой функций аппаратного ускорения трехмерной графики.

Все недостатки в принципе можно решить, да и преимущества виртуальных машин перевешивают их недостатки. Именно поэтому виртуализация сейчас продвигается семимильными шагами вперёд. А пользователи находят всё больше и больше причин их использовать.

1.3. Архитектура виртуальных машин

Виртуализация один из важных инструментов разработки компьютерных систем, а сами виртуальные машины используются в самых разных областях.
Виртуальные машины разрабатываются большим количеством специалистов, преследующих самые разные цели, и в этой области существует не так уж много общепринятых концепций. Поэтому лучше всего будет рассмотреть понятие виртуализации и всё разнообразие архитектур виртуальных машин в единой перспективе.

1.3.1. Абстракция и виртуализация

Компьютерные системы разрабатываются по определенной иерархии и имеют хорошо определенные интерфейсы, из-за чего они и продолжают развиваться. Использование таких интерфейсов облегчает независимую разработку аппаратных и программных подсистем силами разных групп специалистов. Абстракции скрывают детали реализации нижнего уровня, уменьшая сложность процесса проектирования.

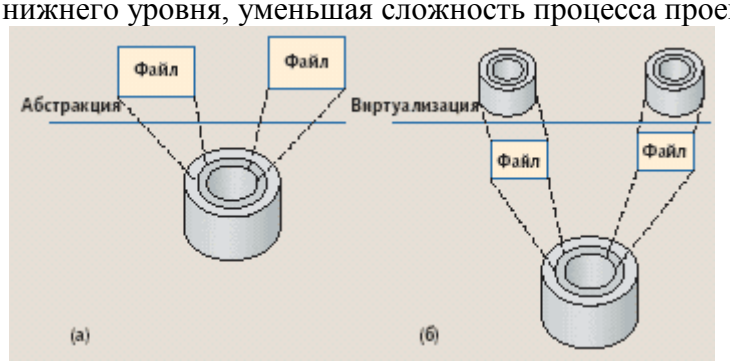


Рис. 1 Абстракция и виртуализация в применение к дисковой памяти.

На рис. 1 (а) приведен пример абстракции в применении к дисковой памяти. Операционная система абстрагируется от тонкостей адресации на жестком диске, от его секторов и дорожек, чтобы для прикладной программы диск выглядел как набор файлов переменного размера. Опираясь на эту абстракцию, «прикладные» программисты могут создавать файлы, записывать и читать данные, не зная устройства и физической организации жесткого диска.
Концепция архитектуры системы команд компьютера (instruction set architecture, ISA) наглядно иллюстрирует преимущества хорошо определенных интерфейсов. Они позволяют разрабатывать взаимодействующие компьютерные подсистемы не только в разных организациях, но и в разные периоды, иногда разделенные годами. Например, Intel и AMD создают микропроцессоры с системой команд IA-32 (x86), в то время как разработчики Microsoft пишут программное обеспечение, которое компилируется в эту систему команд. Поскольку обе стороны соблюдают спецификацию ISA, можно ожидать, что программное обеспечение будет правильно выполняться любым ПК на базе микропроцессора с архитектурой IA-32.
К сожалению, хорошо определенные интерфейсы имеют и недостатки. Подсистемы и компоненты, разработанные по спецификациям разных интерфейсов, не способны взаимодействовать друг с другом. Например, приложения, распространяемые в двоичных кодах, привязаны к определенной ISA и зависят от конкретного интерфейса к операционной системе. Несовместимость интерфейсов может стать сдерживающим

фактором, особенно в мире компьютерных сетей, в котором свободное перемещение программ столь же необходимо, как и перемещение данных.
Виртуализация позволяет обойти эту несовместимость. Виртуализация системы или компонента (например, процессора, памяти или устройства ввода/вывода) на конкретном уровне абстракции отображает его интерфейс и видимые ресурсы на интерфейс и ресурсы реальной системы. Следовательно, реальная система выступает в роли другой, виртуальной системы или даже нескольких виртуальных систем.
В отличие от абстракции, виртуализация не всегда нацелена на упрощение или сокрытие деталей. В примере на рис. 1(б) виртуализация позволяет преобразовать один большой диск в два меньших виртуальных диска, каждый из которых имеет собственные секторы и дорожки. При отображении виртуальных дисков на реальный программные средства виртуализации используют абстракцию файла как промежуточный шаг. Операция записи на виртуальный диск преобразуется в операцию записи в файл (и, следовательно, в операцию записи на реальный диск). Отметим, что в данном случае никакого абстрагирования не происходит — уровень детализации интерфейса виртуального диска (адресация секторов и дорожек) ничем не отличается от уровня детализации реального диска.

1.3.2. Процессные и системные виртуальные машины

Понятия пошли от того, что система и процесс видят машину по-разному, поэтому и виртуальные машины бывают процессные и системные.
Процессная виртуальная машина — это виртуальная платформа для выполнения отдельного процесса. Она предназначена для поддержки процесса, создаётся при его активации и «умирает» после его окончания. Системная виртуальная машина – полнофункциональная, постоянно действующая системная среда, служащая для поддержки операционной системы вместе с большим количеством её пользовательских процессов; она обеспечивает «гостевой» операционной системе доступ к виртуальным аппаратным средствам, в том числе к процессору и памяти, устройствам ввода/вывода, а иногда — и к графическому интерфейсу.
Процесс или система, которые выполняются на виртуальной машине, называются гостем, платформа, поддерживающая виртуальную машину, — хостом. Программное обеспечение, реализующее процессную виртуальную машину, называют рабочей средой, а программное обеспечение виртуализации системной виртуальной машины – монитором виртуальной машины.
Процессные виртуальные машины создают среды ABI и API для пользовательских приложений, что позволяет в многозадачном режиме осуществлять репликацию операционной среды, эмулировать систему команд, оптимизировать код или выполнять программы на языках высокого уровня.
Системная виртуальная машина обеспечивает полнофункциональную среду, в которой могут сосуществовать операционная система и несколько процессов, относящихся к разным пользователям. С помощью них одна аппаратная платформа может поддерживать несколько гостевых операционных систем одновременно.

1.3.3. Типы виртуализаций

Рассмотрим основные типы виртуализации различных компонент ИТ — инфраструктуры.

1. Виртуализация операционной системы. Является наиболее распространенной в данный момент формой виртуализации. Виртуальная операционная система (виртуальная машина) представляет собой, как правило, совмещение нескольких

операционных систем, функционирующих на одной аппаратной основе. Каждая из виртуальных машин управляется отдельно при помощи VMM (Virtual Machine Manager). Лидерами в области поставок решений для виртуализации информационных систем являются Microsoft, AMD, Intel и VMware.

2. Виртуализация серверов приложений.Под данным процессом виртуализации понимают процесс интеллектуальной балансировки нагрузки. Балансировщик нагрузки управляет несколькими веб — серверами и приложениями, как единой системой, пользователь, при этом, «видит» только один сервер, который, фактически, предоставляет функционал нескольких серверов.
3. Виртуализация приложений.Под виртуализацией приложений следует понимать использование программных решений в рамках изолированной виртуальной среды (более подробно виртуализация приложений будет рассмотрена в последующих лекциях).
4. Виртуализация сети.Представляет собой объединение аппаратных и программных ресурсов в единую виртуальную сеть. Выделяют внутреннюю виртуализацию сети — создающую виртуальную сеть между виртуальными машинами одной системы, и внешнюю — объединяющую несколько сетей в одну виртуальную.
5. Виртуализация аппаратного обеспечения.В данном случае виртуализация заключается в разбиении компонент аппаратного обеспечения на сегменты, управляемые отдельно друг от друга. В некоторых случаях, виртуализация операционных систем невозможна без виртуализации аппаратного обеспечения.
6. Виртуализация систем хранения.В свою очередь делится на два типа: виртуализацию блоков и виртуализацию файлов.Виртуализация файлов, как правило используется в системах хранения, при этом ведутся записи о том, какие файлы и каталоги находятся на определенных носителях. Виртуализация файлов отделяет статичный указатель нахождения виртуального файла (C:\, к примеру) от его физического местоположения. Т.е. при запросе пользователем файлаC:\file.doc решение виртуализации файлов отправит запрос к месту реального размещения файла.Виртуализация блоков. Используется в сетях распределенного хранения данных. Сервера — хранилища данных используют RAID-технологию. iSCSI интерфейс также использует блочную виртуализацию, позволяя операционной системе распределить виртуальное блочное устройство. Более подробную информацию о виртуализации систем хранения см. в п.№4 списка источников для самостоятельного изучения.
7. Виртуализация сервисов.По своей сути, виртуализация сервисов является объединением всех вышеуказанных типов виртуализации. Решение виртуализации сервисов позволяет работать с приложением вне зависимости от физического расположения его частей, объединяя и управляя их взаимодействием.

Приведенная выше типология рассматривает виртуализацию, в зависимости от части ИТ — инфраструктуры, в которой она применяется. Подходы к созданию интерфейсов между виртуальными машинами и системами виртуализации ресурсов также можно разделить на следующие типы:

- Полная виртуализация — технология, которая обеспечивает полную симуляцию базового оборудования, гостевая операционная система остается в нетронутом виде.
- Аппаратная виртуализация — технология, позволяющая запускать на одном компьютере (хосте) несколько экземпляров операционных систем (гостевых операционных систем). При этом гостевые ОС независимы друг от друга и от аппаратной платформы.Аппаратная виртуализация представляет собой набор инструкций, облегчающих выполнение операций на аппаратном уровне, которое до

этого могли выполняться только программно, при этом затрачиваются дополнительные программные ресурсы.
- Паравиртуализация — техника виртуализации, при которой гостевые операционные системы подготавливаются для исполнения в виртуализированной среде, для этих целей в ядро ОС вносят незначительные изменения. Для взаимодействия с гостевой операционной системой используется API — интерфейс.

2. Различные виртуальные машины

Все отличия существующих виртуальных машин, по сути, сводятся лишь к перечню поддерживаемых ими **операционных систем**, а так же **стоимости**. Наиболее распространены сегодня системы VirtualBox, Windows Virtual PC и VMWare. Чем же они отличаются?

2.1. ORACLE VirtualBox — универсальная бесплатная виртуальная машина

VirtualBox — очень простой, мощный и бесплатный инструмент для виртуализации, развивающийся благодаря поддержжке знаменитой корпорации ORACLE. Он распространяется бесплатно, с открытым исходным кодом. VirtualBox

позволяет устанавливать в качестве «гостевой» практически любую современную операционную систему, будь то Windows, MacOS или любой из многочисленных представителей семейства Linux.

Преимуществом VirtualBox является простой и понятный пользовательский интерфейс. Хорошо сделан перевод на русский язык. Все основные функции вынесены в виде кнопок под меню. Создание виртуальных машин выполняется с помощью пошагового мастера.

VirtualBox поддерживает работу с сетями, поэтому ваша виртуальная ОС сможет легко выйти в Интернет. Очень полезной является функция «снимков» операционной системы. Виртуальная машина записывает на винчестер «точки восстановления», к которым вы в любой момент можете откатить гостевую систему в случае возникновения ошибок или сбоев.

2.2 Windows Visual PC — виртуальная машина от Microsoft

Windows Virtual PC — виртуальная машина для работы только и исключительно с Windows. Установить на Visual PC операционную систему Linux или MacOS просто невозможно.

Visual PC позволяет запускать несколько разных копий Windows на одном компьютере. Поддерживается работа с операционными системами Microsoft разных поколений, в том числе с 64-битными.

Плюсом Visual PC является возможность задать, какая из запущенных виртуальных машин будет более приоритетной по сравнению с другими. При этом «хостовый» компьютер сможет в автоматическом режиме выделять под ее нужды большее количество ресурсов за счёт других виртуальных систем, если «гостевой» системе это необходимо.

Моноплатформенность виртуальной машины Visual PC является её главным недостатком, впрочем, если требуется тестировать только разные версии Windows, это не актуально.

Некоторым недостатком можно считать менее функциональный и менее удобный чем в VirtualBox интерфейс. В остальном Visual PC вполне надёжный инструмент, позволяющий тестировать операционные системы Microsoft.

2.3 VMware Workstation — для серьёзных задач

VMware Workstation – мощная, платная, максимально-надёжная программа для виртуализации, которая поддерживает работу с Windows и Linux. Для виртуализации MacOS, данная машина не предназначена.

Благодаря высокой надёжности и широчайшей функциональности VMware Workstation часто используется не просто для тестирования, а даже для постоянной работы виртуальных машин в качестве серверов даже для бизнес-приложений, будь то фаервол, отеляющий сеть организации от Интернет или даже сервер какой-либо базы данных.

VMware Workstation можно очень гибко настраивать, включая множество параметров сетевых подключений для работы с интернетом. Система имеет собственный виртуальный 3D-ускоритель, который позволяет получить высокое качество графики.

Интерфейс VMware Workstation достаточно грамотно организован, поэтому освоиться со всем её богатым функционалом довольно легко. В программе полностью поддерживается русский язык.
Необходимо отметить, что у VMware Workstation есть бесплатный «младший брат» — VMWare Player. В отличие от версии Workstation, плеер не умеет создавать виртуальные машины, но позволяет запускать ранее созданные. Эта программа будет полезна в случаях тестирования, когда, к примеру, разработчик какой-либо автоматизированной системы передат её на ознакомление именно в виде образа виртуальной машины. Эта практика получает всё большее распространение, поскольку избавляет пользователя от необходимости разворачивать незнакомую программу самостоятельно.

Практическая часть

1. Установить **ORACLE VirtualBox.**
2. Запустить программу на исполнение.
3. Создать виртуальную машину для установки ОС Windiws XP.
4. Укажите объем оперативной памяти 343МБ.
5. Создайте новый виртуальный жесткий диск (тип VDI).
6. Укажите формат хранения «Фиксированный виртуальный жесткий диск».
7. Размер жесткого диска должен быть 11ГБ.
8. Покажите результат преподавателю.

Вопросы для самоконтроля

1. Что называется виртуальной машиной?
2. Какие преимущества у виртуальной машины? Какие недостатки?
3. Чем отличается системная виртуальная машина от процессорной?
4. Перечислите основные типы виртуализаций.
5. Какие существуют подходы к созданию интерфейсов между виртуальными машинами и системами виртуализации ресурсов?
6. Какие существуют виртуальные машины? В чем их отличие друг от друга?

Лабораторная работа № 12

Тема: Linux. монтируемые файловые системы.

Монтирование файловых систем

С одной стороны, файловая система Linux – это одно большое дерево с корневой директорией /, с другой стороны, мы говорим о файловых системах на различных устройствах и разделах. Как разрешить это кажущееся противоречие? Монтирование корневой файловой системы (/) является частью процесса инициализации. Все остальные созданные файловые системы невозможно использовать в Linux до тех пор, пока они не будут *смонтированы* в определенных *точках монтирования*.

В текущем наборе смонтированных файловых систем точка монтирования является обычной директорией дерева каталогов, в которую файловая система устройства добавляется при монтировании. Монтирование – это процесс, благодаря которому файловая система устройства становится доступной для использования. Например, можно смонтировать файловые системы разделов жесткого диска в точках монтирования /boot, /tmp или /home, файловую систему дискеты в точке монтирования /mnt/floppy, а файловую систему компакт-диска в точке монтирования /media/cdrom1. Как видите, точки монтирования могут располагаться как в корневой директории, так и в поддиректориях дерева каталогов любого уровня вложенности.

Помимо файловых систем на разделах, дискетах и компакт-дисках, существуют и другие типы файловых систем. Например, файловая система tmpfs является файловой системой в виртуальной памяти. Сетевые файловые системы, такие как NFS или AFS, позволяют монтировать на локальном компьютере файловые системы, расположенные на других компьютерах. Можно даже создать файл в существующей файловой системе, отформатировать его в качестве отдельной файловой системы (возможно, другого типа), после чего смонтировать эту новую файловую систему. Этот способ часто используется при работе с образами оптических носителей, когда загруженный ISO-образ CD- или DVD-диска монтируется вместо реального физического носителя для последующего копирования. Другим таким примером может являться пространство подкачки, размещенное не на выделенном разделе, а в файле.

Несмотря на то, что в процессе монтирования фактически монтируется *файловая система* какого либо устройства (или другого ресурса), принято говорить просто о "монтировании устройства", хотя на самом деле подразумевается, что речь идет о "монтировании файловой системы устройства".

Для монтирования и демонтирования файловых систем обычно требуются привилегии пользователя root. Если вы вошли в систему под учетной записью обычного пользователя, используйте для получения прав суперпользователя команду `su -` или `su`. Если в наших примерах вы видите, что приглашение командной строки начинается с символа #, как в листинге 1, значит, для их выполнения необходимо обладать привилегиями пользователя root.

Базовая форма команды `mount` принимает на вход два параметра: имя устройства (или другого ресурса), содержащего монтируемую файловую систему, и точку монтирования. В листинге 1 приведен пример, в котором мы смонтировали FAT32-раздел /dev/sda9 в точке монтирования /dos.

Листинг 1. Монтирование в точку монтирования /dos

```
[root@echidna ~]# mount /dev/sda9 /dos
```

Точка монтирования, в которую монтируется файловая система, должна существовать. В противном случае возникнет ошибка, и нужно будет либо создать указанную точку монтирования, либо использовать другую точку монтирования, как показано в листинге 2.

Листинг 2. Ошибка монтирования

```
[root@echidna ~]# mount /dev/sda9 /dos
mount: mount point /dos does not exist
[root@echidna ~]# mkdir /dos
[root@echidna ~]# mount /dev/sda9 /dos
```

Когда файловая система монтируется в существующую директорию, все файлы и поддиректории этой файловой системы становятся файлами и поддиректориями точки монтирования. Если директория точки монтирования уже содержала какие-либо файлы или поддиректории, то они не теряются, но становятся невидимыми до тех пор, пока смонтированная файловая система не будет демонтирована. Во избежание таких проблем рекомендуем использовать в качестве точек монтирования только пустые директории.

После того как файловая система смонтирована, все файлы и директории, создаваемые или копируемые в точку монтирования или ее поддиректории, будут созданы в смонтированной файловой системе. Если в нашем примере мы создадим файл /dos/sampdir/file.txt, то он будет создан в файловой системе FAT32, которая была смонтирована в точку монтирования /dos.

Обычно команда `mount` автоматически определяет тип монтируемой файловой системы. Тем не менее, иногда вам может потребоваться явно указать тип файловой системы, для чего можно использовать опцию `-t`, как показано в листинге 3.

Листинг 3. Монтирование с явным указанием типа файловой системы

```
[root@echidna ~]# mount -t vfat /dev/sda9 /dos
```

Чтобы просмотреть список смонтированных файловых систем, используйте команду `mount` без каких-либо параметров. В листинге 4 показан пример нашей тестовой системы. Заметьте, что для просмотра списка файловых систем привилегии суперпользователя не требуются.

Листинг 4. Просмотр смонтированных файловых систем

```
[ian@echidna ~]$ mount
/dev/sda6 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda2 on /grubfile type ext3 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
gvfs-fuse-daemon on /home/ian/.gvfs type fuse.gvfs-fuse-daemon
(rw,nosuid,nodev,user=ian)
dw.raleigh.ibm.com:/vol/vol1/dwcontent on /mnt/dwcontent type nfs
(rw,addr=9.42.155.6)
```

```
/dev/sdb9 on /mnt/sdb9 type ext3 (rw)
/dev/sda9 on /dos type vfat (rw)
/dev/sr0 on /media/KNOPPIX type iso9660
(ro,nosuid,nodev,uhelper=udisks,uid=1000,gid=1000
,iocharset=utf8,mode=0400,dmode=0500)
```

Эту же информацию можно получить, просмотрев содержимое файла /proc/mounts или /etc/mtab; оба этих файла содержат информацию о смонтированных файловых системах

Опции монтирования

Команда `mount` имеет ряд опций, которые изменяют ее поведение по умолчанию. Например, с помощью опции `-o ro` можно смонтировать файловую систему в режиме "только для чтения". Если файловая система уже была смонтирована, добавьте опцию `remount`, как показано в листинге 5.

Листинг 5. Повторное монтирование в режиме только для чтения

```
[root@echidna ~]# mount -o remount,ro /dos
```

Примечания.

- Если используется несколько опций (например, `remount` и `ro`), необходимо разделять их запятыми.
- Если выполняется повторное монтирование уже смонтированной файловой системы, достаточно указать либо точку монтирования, либо имя устройства. Нет необходимости указывать и то, и другое.
- Невозможно смонтировать файловую систему, предназначенную "только для чтения", в режиме чтения и записи. Носители, файловые системы которых изменить невозможно (например, CD- и DVD-диски), автоматически монтируются в режиме "только для чтения".
- Для повторного монтирования допускающего возможность записи устройства в режиме чтения и записи, используйте опцию `-o remount,rw`.

Команды повторного монтирования не будут успешно завершены, если какой-либо процесс использует файлы или директории повторно монтируемой файловой системы. Дополнительная информация содержится в разделе этой статье.

Метки, идентификаторы и ссылки

В UNIX и ранних системах Linux директория /dev обычно содержала записи всех возможных устройств, которые могли быть подключены к системе. Любое подключаемое устройство всегда располагалось в одном и том же месте дерева /dev, поэтому использование имен устройств (например, /dev/sda6) было естественным. Современные устройства, поддерживающие технологию горячей замены, такие как устройства с интерфейсом USB или Firewire (IEEE 1394), могут подключаться каждый раз к разным портам. В таких случаях может возникнуть желание всегда монтировать определенный USB-накопитель в одну и ту же директорию (например, /media/myusbstick) независимо от того, к какому порту он будет подключен. В другой статье нашей серии (тема 102) упоминалось, что вместо имен устройств для идентификации разделов можно использовать метки и универсальные идентификаторы UUID (Universally Unique ID). Если файловая система раздела поддерживает метки или UUID, то их можно использовать и для работы с командой `mount`. Чтобы узнать, какой идентификатор и метка (если она есть)

назначены устройству, используйте команду `blkid`. В листинге 6 показано, как использовать `blkid` для определения метки и UUID нашего корневого раздела, создать две дополнительные точки монтирования и смонтировать в них корневой раздел. Этот пример приведен лишь для демонстрации, и его не стоит использовать его на практике.

Листинг 6. Монтирование с использованием меток и идентификаторов UUID

```
[root@echidna ~]# blkid /dev/sda6
/dev/sda6: LABEL="Fedora-13-x86_64" UUID="082fb0d5-a5db-41d1-ae04-
6e9af3ba15f7"
 TYPE="ext4"
[root@echidna ~]# mkdir /mnt/sda6label
[root@echidna ~]# mkdir /mnt/sda6uuid
[root@echidna ~]# mount LABEL="Fedora-13-x86_64" /mnt/sda6label
[root@echidna ~]# mount UUID="082fb0d5-a5db-41d1-ae04-6e9af3ba15f7"
/mnt/sda6uui
```

Благодаря менеджеру устройств udev, в директории /dev можно найти дополнительные символические ссылки на такие устройства, как, например, жесткие диски. В листинге 7 показаны ссылки на устройство /dev/sda6 в моей операционной системе Fedora 13.

Листинг 7. Символические ссылки на устройство /dev/sda6

```
[ian@echidna ~]$ find /dev -lname "*sda6"
/dev/disk/by-label/Fedora-13-x86_64
/dev/disk/by-uuid/082fb0d5-a5db-41d1-ae04-6e9af3ba15f7
/dev/disk/by-path/pci-0000:00:1f.2-scsi-0:0:0:0-part6
/dev/disk/by-id/wwn-0x50014ee001a8d027-part6
/dev/disk/by-id/scsi-SATA_WDC_WD1001FALS-_WD-WMATV3772868-part6
/dev/disk/by-id/ata-WDC_WD1001FALS-00J7B1_WD-WMATV3772868-part6
/dev/block/8:6
```

Символические ссылки можно использовать как еще один способ указания имени устройства при его монтировании.

Процесс загрузки и файл fstab

В другой статье нашей серии упоминалось, что параметр `root=` говорит менеджерам загрузки LILO и GRUB о том, что файловая система должна быть смонтирована как корневая. После того, как файловая система смонтирована, процесс инициализации запускает команду `mount` с опцией `-a` для автоматического монтирования ряда файловых систем, перечисленных в файле /etc/fstab. В листинге 8 показано содержимое файла /etc/fstab в тестовой операционной системе Fedora 13. В этом примере все разделы диска идентифицируются с помощью UUID. В листинге 9 показан другой пример, на этот раз из операционной системы Ubuntu 9.10.

Листинг 8. Пример файла fstab в Fedora 13

```
#
# /etc/fstab
# Created by anaconda on Fri May 28 12:37:05 2010
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=082fb0d5-a5db-41d1-ae04-6e9af3ba15f7 /                       ext4    defaults
1 1
```

```
UUID=488edd62-6614-4127-812d-cbf58eca85e9 /grubfile          ext3    defaults
1 2
UUID=2d4f10a6-be57-4e1d-92ef-424355bd4b39 swap               swap    defaults
0 0
UUID=ba38c08d-a9e7-46b2-8890-0acda004c510 swap               swap    defaults
0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
```

Листинг 9. Пример файла fstab в Ubuntu 9.10

```
# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
# / was on /dev/sda7 during installation
UUID=8954fa66-e11f-42dc-91f0-b4aa480fa103 /               ext3
errors=remount-ro 0  1
# /grubfile was on /dev/sda2 during installation
UUID=3a965842-b6dd-4d52-8830-2d0fdb4284a2 /grubfile       ext3    defaults
0  2
/dev/sda5       none            swap    sw              0       0
/dev/scd0       /media/cdrom0   udf,iso9660 user,noauto,exec,utf8 0       0
/dev/fd0        /media/floppy0  auto    rw,user,noauto,exec,utf8 0       0
```

Строки, начинающиеся с # – это комментарии. Остальные строки содержат по шесть полей. Поскольку эти поля позиционные, то все они должны быть не пустыми.

file system

Это поле может содержать имя устройства, например, /dev/sda1, метку (LABEL=) или уникальный идентификатор (UUID=). Для корневой файловой системы Fedora 13 из нашего примера значением этого поля может быть /dev/sda6, LABEL="Fedora-13-x86_64" или UUID="082fb0d5-a5db-41d1-ae04-6e9af3ba15f7". Использование меток или UUID обеспечивает дополнительную гибкость в работе при добавлении или извлечении устройств.

mount point

Это поле содержит точку монтирования, о которой уже упоминалось в разделе. Для пространства подкачки это поле должно содержать значение 'none' или 'swap'. В более ранних системах значением этого поля обычно было 'none'.

type

Это поле определяет тип файловой системы. CD- и DVD-приводы часто поддерживают файловые системы ISO9660 или UDF, поэтому можно указать через запятую все возможные значения, как это сделано в листинге 9. Если вы хотите, чтобы команда `mount` автоматически определяла тип, то укажите значение `auto`, как в последней строке листинга 9 (опция для флоппи-дисковода).

option

Это поле содержит опции монтирования. Если вы хотите использовать опции по умолчанию, укажите значение `defaults`. Перечислим несколько полезных опций, о которых следует знать.

- `rw` и `ro` – определяют, должна ли файловая система быть смонтирована для чтения и записи, либо в режиме "только для чтения".
- `noauto` – запрещает автоматическое монтирование файловой системы во время загрузки или при выполнении команды `mount -a`. В нашем примере эта опция используется для сменных носителей.
- `user` – разрешает монтировать и демонтировать файловую систему обычному (не root) пользователю. Эта опция особенно полезна для использования со сменными носителями. В более ранних системах эта опция указывалась в файле /etc/fstab, а не в команде `mount`. В современных системах эта опция может быть указана в правилах udev, расположенных в файлах правил /lib/udev/rules.d или /etc/udev/rules.d. Опции для DVD-привода моего компьютера под управлением Fedora 13 указаны в правилах udev, поэтому в файле /etc/fstab запись для оптического привода отсутствует.
- `exec` и `noexec` – разрешают или запрещают выполнение файлов, расположенных в смонтированной файловой системе. По умолчанию для монтируемых пользователем файловых систем используется опция `noexec`, если только опция `exec` не указана **после** опции `user`.
- `noatime` – запрещает запись времени доступа к файлам, что позволяет повысить быстродействие.

dump

Это поле определяет, должна ли команда `dump` учитывать файловые системы ext2 или ext3 при создании резервных копий. Значение 0 указывает команде `dump` игнорировать эти файловые системы.

pass

Значение этого поля, отличное от нуля, определяет очередность проверки файловых систем во время загрузки

При монтировании файловых систем, перечисленных в файле /etc/fstab, можно указать либо имя устройства, либо точку монтирования. Нет необходимости указывать и то, и другое.

В некоторых системах, например SUSE 11.2, файл `fstab`, сгенерированный в процессе инсталляции системы, может содержать символические ссылки на устройства. Таким образом, вместо имени /dev/sda6 вы можете увидеть /dev/disk/by-id/ata-WDC_WD1001FALS-00J7B1_WD-WMATV3772868-part6.

Для получения дополнительной информации, включая не рассмотренные в этой статье опции, обратитесь к man-страницам `fstab`, `mount` и `udev`.

Демонтирование файловых систем

Обычно операционная система выполняет демонтирование всех смонтированных файловых систем автоматически при перезагрузке или выключении компьютера. При демонтировании файловой системы все ее кэшированные данные, находящиеся в оперативной памяти, сбрасываются на диск.

Файловые системы можно демонтировать и вручную. Более того, вы **должны** делать это при извлечении доступных для записи съемных устройств, таких как дискеты или USB-накопители.

Для демонтирования файловой системы выполните команду `umount`, указав в качестве аргумента *либо* имя устройства, *либо* точку монтирования. В листинге 10 показано, как демонтировать файловую систему, указав точку монтирования /dos, а затем снова смонтировать ее и вновь демонтировать, указав на этот раз имя устройства.

Листинг 10. Демонтирование файловых систем

```
[root@echidna ~]# umount /dos
[root@echidna ~]# mount /dev/sda9 /dos
[root@echidna ~]# umount /dev/sda9
```

После того, как файловая система будет демонтирована, все файлы директории, использовавшейся в качестве точки монтирования, снова окажутся видимыми.

Если вы попытаетесь демонтировать файловую систему, содержащую файлы, открытые каким-либо процессом, то вы получите сообщение об ошибке. Прежде чем начинать демонтирование файловой системы, следует проверить, что ее файлы не используются работающими процессами. Чтобы посмотреть, какие файлы открыты определенным процессом или какой процесс использует те или иные файлы, используйте команды `lsof` и `fuser`. Чтобы не выдавались предупреждения, относящиеся к Gnome Virtual File system (gvfs), может потребоваться использовать опцию `-w` команды `lsof`. О дополнительных опциях монтирования и команде `lsof` можно узнать из соответствующих man-страниц. Если вы проверяете все устройство целиком, то можно указать имя устройства или точку монтирования. Также можно проверить использование отдельного файла.

Для демонстрации работы этих команд я создал копию файла /etc/fstab в директории /dos и написал небольшой сценарий, который каждые 10 секунд считывает строки с устройства стандартного ввода и выводит их в устройство стандартного вывода. В листинге 11 показано сообщение об ошибке, выводимое командой `umount` при использовании файлов демонтируемой файловой системы, а также приведен пример использования команд `lsof` и `fuser` для проверки файлов в директории /dos, т. е. на устройстве /dev/sda9.

Листинг 11. Проверка на наличие открытых файлов

```
[root@echidna ~]# umount /dos
umount: /dos: device is busy.
        (In some cases useful info about processes that use
         the device is found by lsof(8) or fuser(1))
[root@echidna ~]# lsof -w /dos
COMMAND    PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
slowread. 2560  ian    0r   REG    8,9      899  123 /dos/fstab
sleep     2580  ian    0r   REG    8,9      899  123 /dos/fstab
[root@echidna ~]# lsof -w /dev/sda9
COMMAND    PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
slowread. 2560  ian    0r   REG    8,9      899  123 /dos/fstab
sleep     2580  ian    0r   REG    8,9      899  123 /dos/fstab
[root@echidna ~]# fuser -m /dos
/dos:                 2560  2586
[root@echidna ~]# fuser -m /dev/sda9
/dev/sda9:            2560  2588
```

На этом этапе вы можете либо подождать, пока все открытые файлы не освободятся, либо выполнить *"ленивое" демонтирование*, указав опцию `-l`. Такой способ демонтирования немедленно удаляет файловую систему из дерева каталогов, а также все ссылки на нее по мере того, как она освобождается.

Файловые системы на съемных носителях

В этой статье я упоминал о съемных носителях, таких как устройства с интерфейсом USB или Firewire (IEEE 1394). Зачастую неудобно каждый раз переключаться в режим суперпользователя для монтирования или демонтирования таких устройств. То же самое касается флоппи-дисководов, CD- и DVD-приводов, когда каждый раз для смены диска или дискеты нужно демонтировать устройство. Далее, рассматривая команду `fstab`, я упомянул об опции `user`, которая позволяет монтировать и демонтировать устройства обычным пользователям. В листинге 9 показан один из способов настройки записей в `fstab` для флоппи-дисководов, CD- и DVD-приводов.

Обратите внимание на то, что для оптических дисков указаны типы файловых систем `udf,iso9660`, а для флоппи-дисков тип файловой системы выбирается автоматически (опция `auto`). В случае оптических дисков процесс монтирования сначала проверяет наличие файловой системы udf (обычно используется на DVD-дисках), а затем iso9660 (обычно используется на CD-дисках). При использовании дискет тип файловой системы определяется в результате проверок. Вы можете создать или отредактировать файл /etc/filesystems и изменить порядок выполнения различных проверок файловых систем.

Примечание. Всегда следует выполнять демонтирование съемных устройств или носителей, прежде чем отключать или извлекать их. В противном случае вы можете потерять данные, которые еще не были записаны на устройство из кэша.

Если вы работаете с графическим рабочим столом, например, с Nautilus, то, как правило, эта среда содержит опции, позволяющие автоматически монтировать съемные устройства и носители. Например, если я вставлю DVD-диск с Knoppix в DVD-привод моего компьютера, то выдается запись о монтировании, подобную той, что показана в листинге 12. Запись 'uid=1000' означает, что диск может быть демонтирован пользователем с идентификатором 1000. Команда `id` показывает, что 1000 – это значение идентификатора uid пользователя ian, поэтому Ян может демонтировать этот диск.

Листинг 12. Монтирование DVD в режиме графического рабочего стола

```
[ian@echidna ~]$ mount | grep sr0
/dev/sr0 on /media/KNOPPIX type iso9660 (ro,nosuid,nodev,uhelper=udisks,
uid=1000,gid=1000,iocharset=utf8,mode=0400,dmode=0500)
[ian@echidna ~]$ id ian
uid=1000(ian) gid=1000(ian) groups=1000(ian)
```

Если такая возможность поддерживается дисководом (например, она поддерживается большинством приводов CD или DVD), то для извлечения съемного носителя можно использовать команду `eject`. Если перед этим устройство не было демонтировано, команда `eject` демонтирует его и извлечет носите.

Пространство подкачки

При обсуждении `fstab` вы могли обратить внимание на то, что пространство подкачки не имеет точки монтирования. В процессе монтирования обычно активируется пространство подкачки, определенное в файле /etc/fstab (если не используется опция `noauto`). Чтобы вручную управлять пространством подкачки в работающей системе (например, если добавляется новый раздел подкачки), используйте команды `swapon` и `swapoff`. Для получения подробной информации обратитесь к man-страницам.

Для просмотра пространств подкачки, доступных в текущий момент, выполните команду `cat /proc/swaps` или `swapon -s`, как показано в листинге 13.

Листинг 13. Просмотр пространства подкачки

```
[ian@echidna ~]$ swapon -s
Filename                                Type            Size    Used    Priority
/dev/sdb1                                partition      514044  0       -1
/dev/sdb5                                partition      4192928 0       -2
[ian@echidna ~]$ cat /proc/swaps
Filename                                Type            Size    Used    Priority
/dev/sdb1                                partition      514044  0       -1
/dev/sdb5                                partition      4192928 0       -2
```

Лабораторная работа № 13

Тема: Планирование заданий в ос Windows XP

В процессе эксплуатации системы часто требуется выполнять периодические регламентные задачи обслуживания в автоматическом режиме без вмешательства человека, например, проверку наличия обновлений, создание резервных копий критически важных файлов, выполнения сценариев администрирования и т.п. Для планирования и автоматического выполнения задач в среде Windows предусмотрено стандартное средство **Планировщик заданий**.

Планировщик заданий обеспечивает выполнение заранее подготовленных задач в определенное время, или при возникновении определенных событий, однократно или с периодичностью, в контексте системных или пользовательских учетных записей. Задачи могут обеспечивать исполнение сценариев командной строки, скриптов WSH, командлетов PowerShell или приложений. В отличие от предыдущих версий Windows, Windows Vista, 7, 8 содержат обширную библиотеку предварительно настроенных заданий. Эти задания выполняют широкий круг операций, обеспечивающих обслуживание системы и поддержание ее в рабочем состоянии. Кроме того, Планировщик заданий в данных операционных системах, стал необходимым для нормальной работы компонентом, который нельзя отключить стандартными средствами.

Выполнение заданий по расписанию обеспечивается службой Планировщика заданий. Эта служба выполняется под локальной системной учетной записью, но позволяет настраивать задания для выполнения в контексте учетных записей определенных пользователей, указав при создании задания имя пользователя и пароль. Задачи можно создавать не только на локальном компьютере, но и на удаленном, по отношению к которому имеются соответствующие права. Процесс создания заданий планировщика упрощается при использовании соответствующего мастера.

В Windows существуют два основных типа планируемых заданий

Стандартные задания. Задания этого типа используются для автоматизации повседневных задач обслуживания компьютерной системы. Пользователь может видеть эти задания и модифицировать их в случае необходимости.

Скрытые задания. Задания этого типа используются для автоматизации выполнения специальных системных задач. Эти задания в стандартном режиме просмотра скрыты от пользователей и их изменение нежелательно.

Каждое задание планировщика имеет определенные **свойства**:

Триггеры - задают условия начала и завершения выполнения задания. Выполнение задания можно начинать по расписанию, а также по событиям - при входе пользователя в систему, при запуске компьютера, при подключении/отключении терминальной сессии или при записи в журналы системы событий с определенным кодом. **Действия** - определяют операцию, которую должно выполнять запущенное на выполнение задание, в качестве которого может использоваться приложение или сценарий.

Условия - определяют условия, при которых активированное задание запускается или останавливается. Например, условия можно использовать, чтобы выполнение задания было возможным только в случае наличия определенного сетевого подключения. С

помощью условий можно запускать, останавливать и перезапускать задания. .

Параметры - определяют дополнительные параметры, влияющие на выполнение задания – перезапуск после сбоя, выполнение просроченного задания, поведение при наличии уже выполняющегося задания.

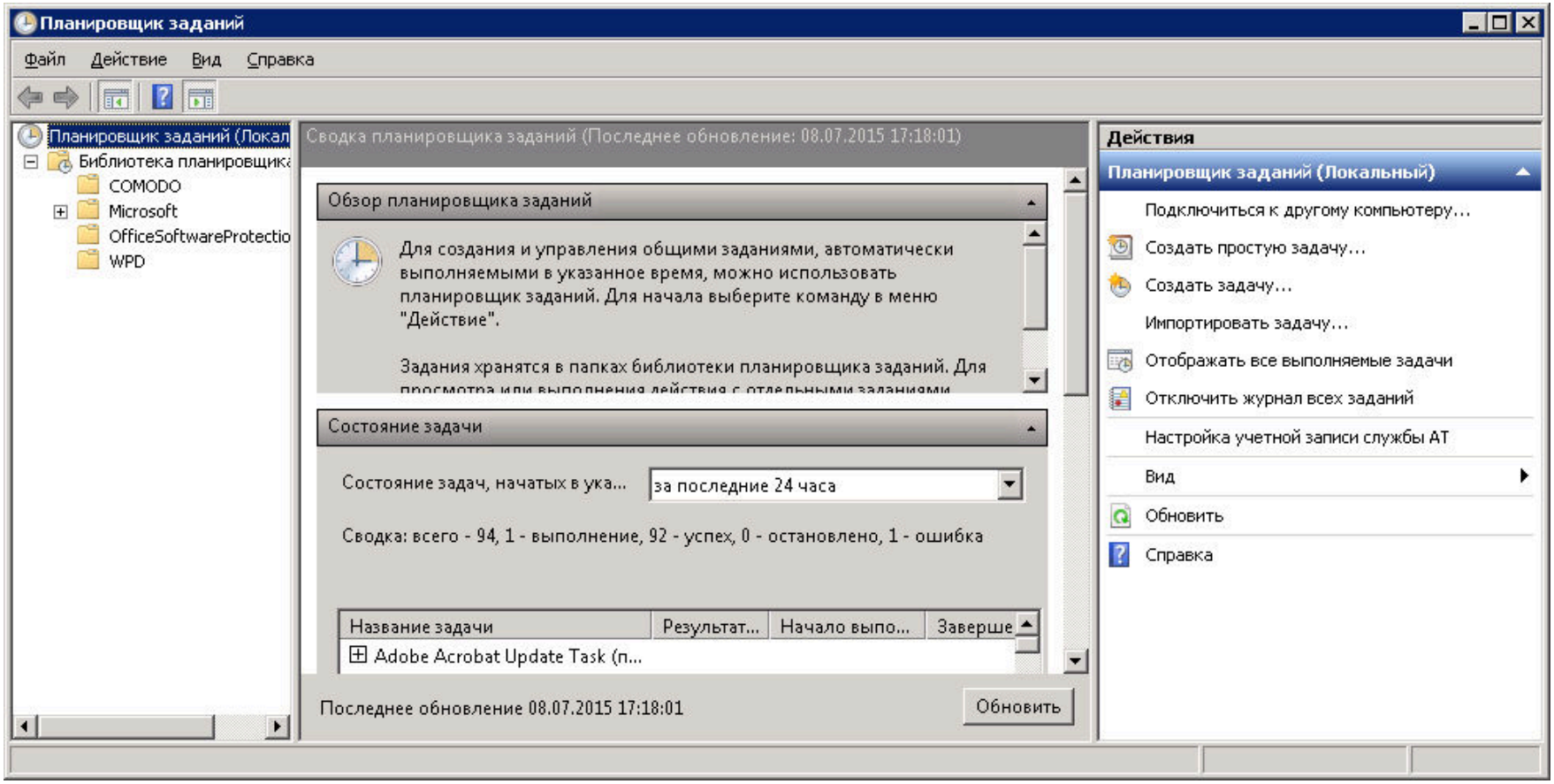
Настройки планировщика и параметры заданий хранятся в разделе реестра

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule

Результаты работы планировщика заданий могут фиксироваться в журнале, если это определено его настройками (В консоли управления планировщиком заданий - **Действие** - **Отключить (Включить) журнал всех заданий**).

Управление заданиями на локальных и удаленных системах.

Управление заданиями Windows выполняется с помощью специальной оснастки консоли управления Microsoft (mmc.exe), для запуска которой можно воспользоваться главным меню **Панель управления - Администрирование - Управление компьютером - Планировщик заданий**, или **Выполнить**(комбинация Win+R) - taskschd.msc.


Планировщик заданий
Файл Действие Вид Справка
Планировщик заданий (Локал
Библиотека планировщика
COMODO
Microsoft
OfficeSoftwareProtectio
WPD
Сводка планировщика заданий (Последнее обновление: 08.07.2015 17:18:01)
Обзор планировщика заданий
Для создания и управления общими заданиями, автоматически выполняемыми в указанное время, можно использовать планировщик заданий. Для начала выберите команду в меню "Действие".
Задания хранятся в папках библиотеки планировщика заданий. Для
Состояние задачи
Состояние задач, начатых в ука...
за последние 24 часа
Сводка: всего - 94, 1 - выполнение, 92 - успех, 0 - остановлено, 1 - ошибка
Название задачи
Результат...
Начало выпо...
Заверше
Adobe Acrobat Update Task (n...
Последнее обновление 08.07.2015 17:18:01
Обновить
Действия
Планировщик заданий (Локальный)
Подключиться к другому компьютеру...
Создать простую задачу...
Создать задачу...
Импортировать задачу...
Отображать все выполняемые задачи
Отключить журнал всех заданий
Настройка учетной записи службы AT
Вид
Обновить
Справка

В левой части основного окна отображаются списки заданий, упорядоченные в соответствии с назначением, в виде структуры папок. В средней части отображается информация о состоянии заданий и их свойствах. В правой части отображается меню действий, допустимых по отношению к заданиям. По умолчанию отображаются задания, относящиеся к локальному компьютеру. Для работы с заданиями удаленных компьютеров щелкните правой кнопкой мыши по элементу верхнего уровня "Планировщик заданий (Локальный)" и выберите пункт меню "Подключиться к другому компьютеру". В открывшемся диалоговом окне установите переключатель "Другой компьютер" и введите имя или IP-адрес требуемого компьютера. Для доступа к управлению заданиями на удаленном компьютере требуются права администратора системы, и, если текущий пользователь таковыми не обладает, то подключение нужно выполнить с использованием другой учетной записи, включив режим "Подключаться как другой пользователь". Естественно, удаленные подключения должны быть разрешены правилами брандмауэра и, кроме того, нужно учитывать, что структура данных заданий планировщика отличается в разных версиях Windows и подключение из среды, например Windows XP не выполнится при подключении к компьютеру с Windows 8.

Для работы с заданием щелкните по нему правой кнопкой мышки в основной панели и в контекстном меню выберите одну из следующих команд:

- **Удалить** (Delete) — полностью удалить задание;
- **Отключить** (Disable) — временно отключить задание;
- **Свойства** (Properties) — просмотреть и/или редактировать свойства задания.
- **Экспортировать** (Export) — экспортировать задание в файл, данные из которого можно импортировать на другом компьютере. Операционные системы Windows 8 и Windows Server 2012 имеют такую же архитектуру заданий, как и Windows 7 и Windows Server 2008 R2, но архитектура заданий более ранних версий Windows другая. При экспорте заданий можно указать операционную систему, с которой это задание допустимо использовать, с помощью параметра "Настроить для" (Configure for) на вкладке "Общие" окна свойств задания.
- **Выполнить** (Run) — запустить задачу на выполнение;
- **Завершить** (End) — если задача выполняется, остановить ее выполнение.

Задания, созданные пользователями и прикладными программами можно изменять или удалять без каких-либо особых проблем, но большинство заданий, созданных самой операционной системой, трогать не стоит, поскольку это может иметь неприятные последствия. Для отображения системных задач в меню "Вид" необходимо установить флажок "Отобразить скрытые задачи" (Show hidden tasks).

Просмотр списка заданий, исполняющихся на компьютере в настоящее время.

Для получения перечня заданий, выполняемых в данный момент времени, можно выбрать верхний уровень структуры "Планировщик заданий" и в контекстном меню, вызываемом правой кнопкой мышки, выбрать пункт "Отображать все выполняемые задачи". Можно

также использовать пункт основного меню "Действие" - "Отображать все выполняемые задачи"

Создание заданий для планировщика.

Библиотека планировщика заданий, отображаемая в левой части окна оснастки планировщика, имеет довольно непростую иерархическую структуру, поэтому, для освоения приемов работы с запланированными задачами, можно создать отдельную папку, с использованием контекстного меню, вызываемого правой кнопкой мышки и пункта **Создать папку**, ввести имя папки, и в дальнейшем, именно в ней создавать свои тестовые или рабочие задания.

В качестве задания планировщика будем использовать задачу, выполняющую командный файл, который определяет с какими привилегиями он выполняется, и выводит на экран сообщение об этом текущему пользователю. Использование такой задачи позволит наглядно продемонстрировать некоторые настройки свойств задания.

Для проверки уровня привилегий используется стандартная команда **whoami** в цепочке с командой **find**, для вывода сообщений - **msg**. В результате выполнения командного файла, на экран выводится сообщение "Задача выполняется с правами администратора" или " Задача выполняется с правами пользователя - имя ". Пусть данный командный файл имеет имя **who-admin.bat**. При желании, описание используемых команд и принцип определения привилегий, с которыми выполняется командный файл, можно найти в разделе Список команд CMD Windows

Содержимое файла с именем **who-admin.bat**:

```
@echo OFF
WHOAMI /PRIV | find /i "SeRemoteShutdownPrivilege"
if %ERRORLEVEL% == 0 goto admin
msg * Задача выполняется с правами пользователя - %USERNAME%
exit
:admin
msg * Задача выполняется с правами администратора.
```

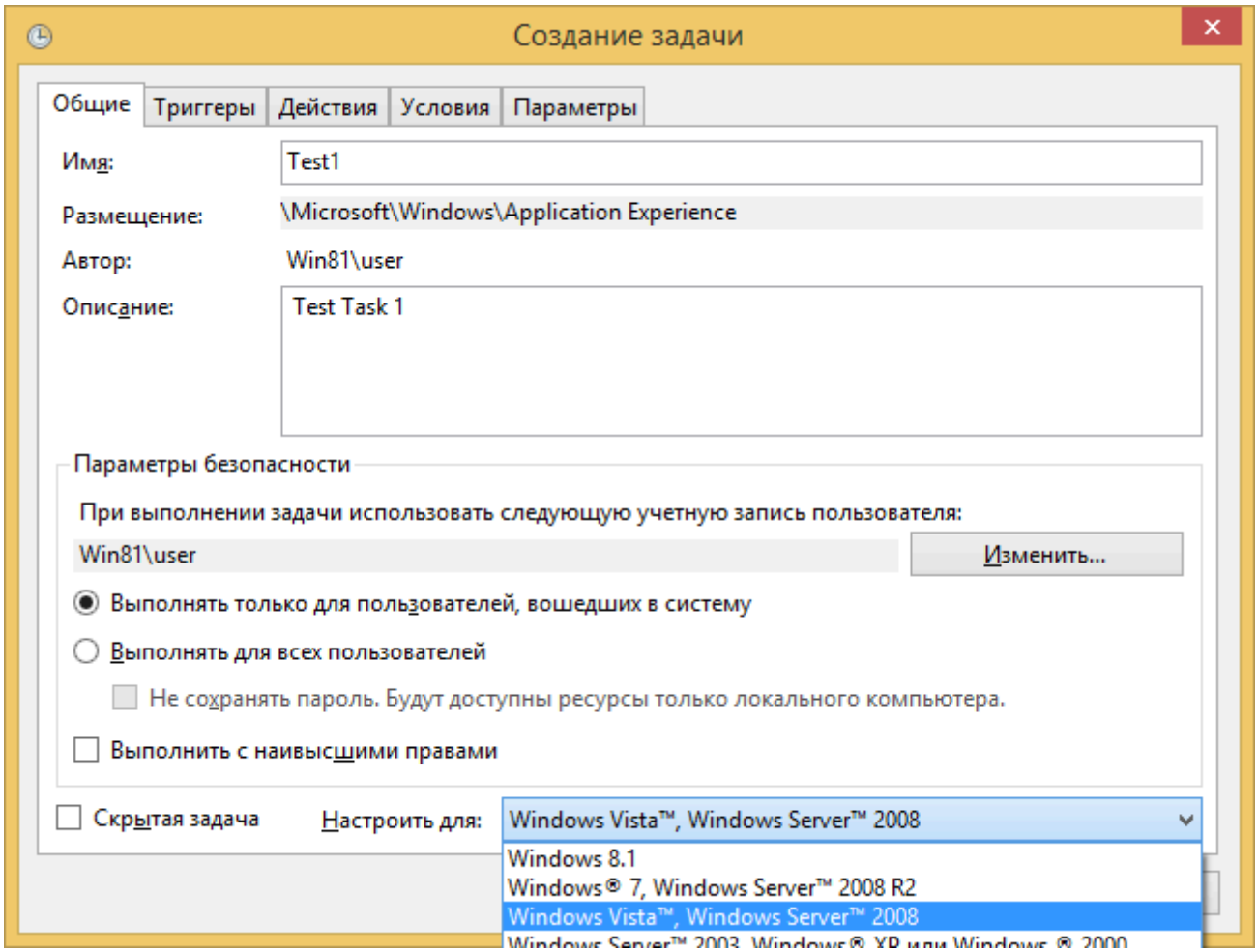
Файл можно создать в обычном текстовом редакторе, скопировав приведенный выше текст, выделенный зеленым цветом и присвоив ему имя who-admin.bat . Для правильного отображения кириллицы необходимо использовать DOS-кодировку (CP866, кодовую страницу 866) или изменить текст сообщений для вывода на английском языке. Если текстовый редактор не поддерживает кириллицу в DOS-кодировке, можно воспользоваться сторонними программами (например - **iconv**) или онлайн-сервисами наподобие Универсального декодера-конвертера кириллицы. Можете просто скачать архив who-admin.zip с командным файлом в правильной кодировке.

Для создания задач планировщика могут использоваться два мастера, вызываемые в режимах **Создать простую задачу** и **Создать задачу**. При создании простой задачи используется минимальный набор параметров, не предусматривающий наличие множественных условий выполнения и множественных действий.

Новую задачу можно создать следующим способом:

С использованием пункта меню **Действие - Создать задачу** или через контекстное меню,

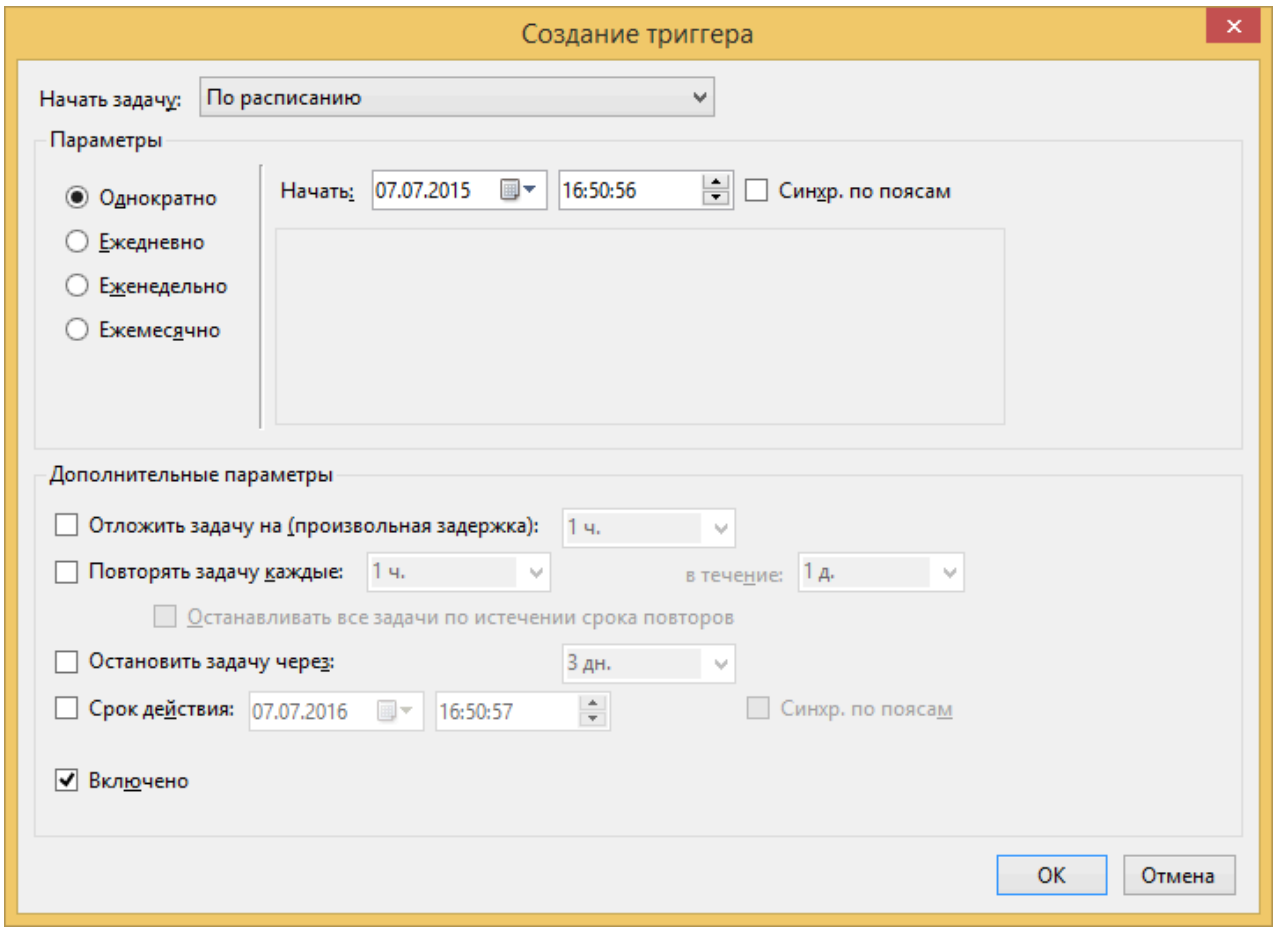
вызываемое правой кнопкой мышки на уровне "Планировщик заданий". После чего запускается мастер создания задачи:



На вкладке **Общие** окна мастера введите имя задания и его описание. Имя, для примера - **Test1**, а описание "Test Task 1". Описание может быть произвольным текстом, но желательно, чтобы оно отражало суть создаваемой задачи. Остальные настройки можно не выполнять, оставив значения по умолчанию. Эти настройки будем менять перед выполнением созданной задачи, чтобы понять их смысл и назначение.

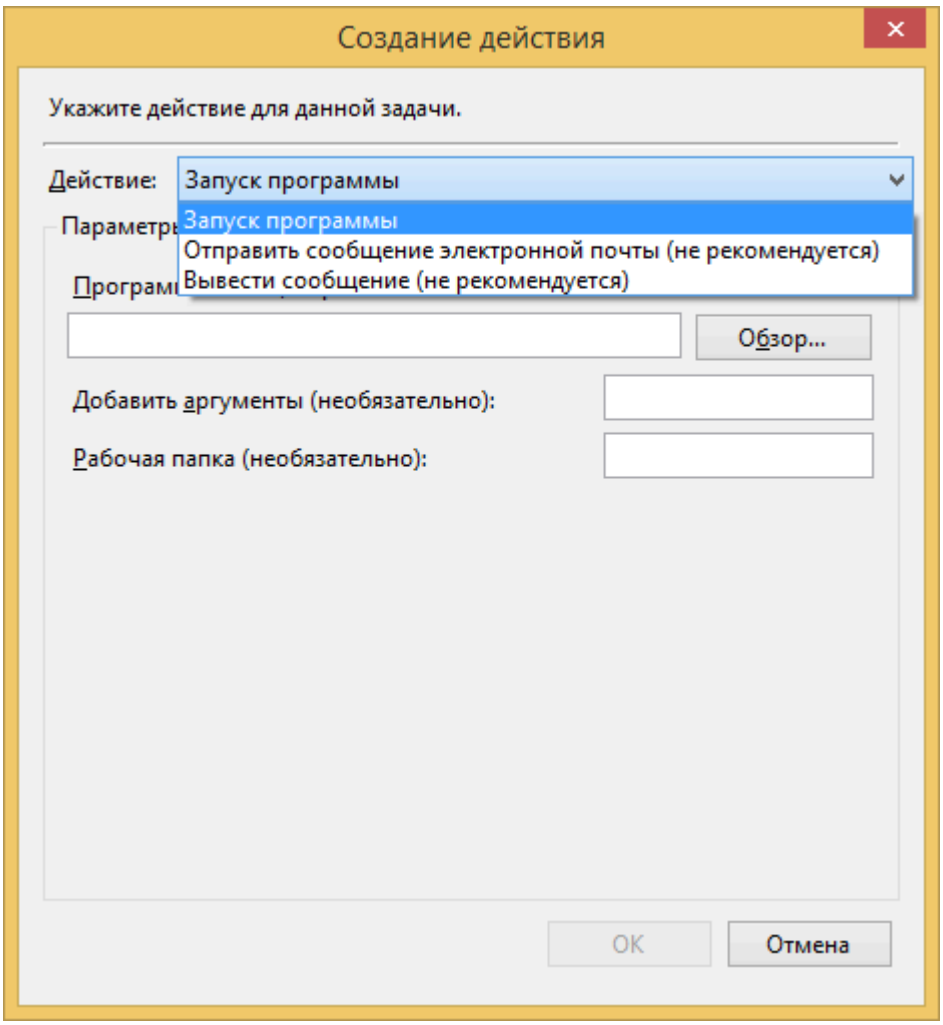
По умолчанию задания исполняются, только если пользователь выполнил вход в систему. Чтобы выполнять задачу независимо от того, работает пользователь в системе или нет, используется переключатель **Выполнять вне зависимости от регистрации пользователя** (Run whether user is logged on or not). Также можно задать выполнение задания с наивысшими полномочиями и/или настроить его для определенных версий Windows.

На вкладке **Триггеры** нажмите кнопку Создать, в открывшемся окне **Создание триггера** и задайте условие при возникновении которого будет выполняться создаваемая задача.

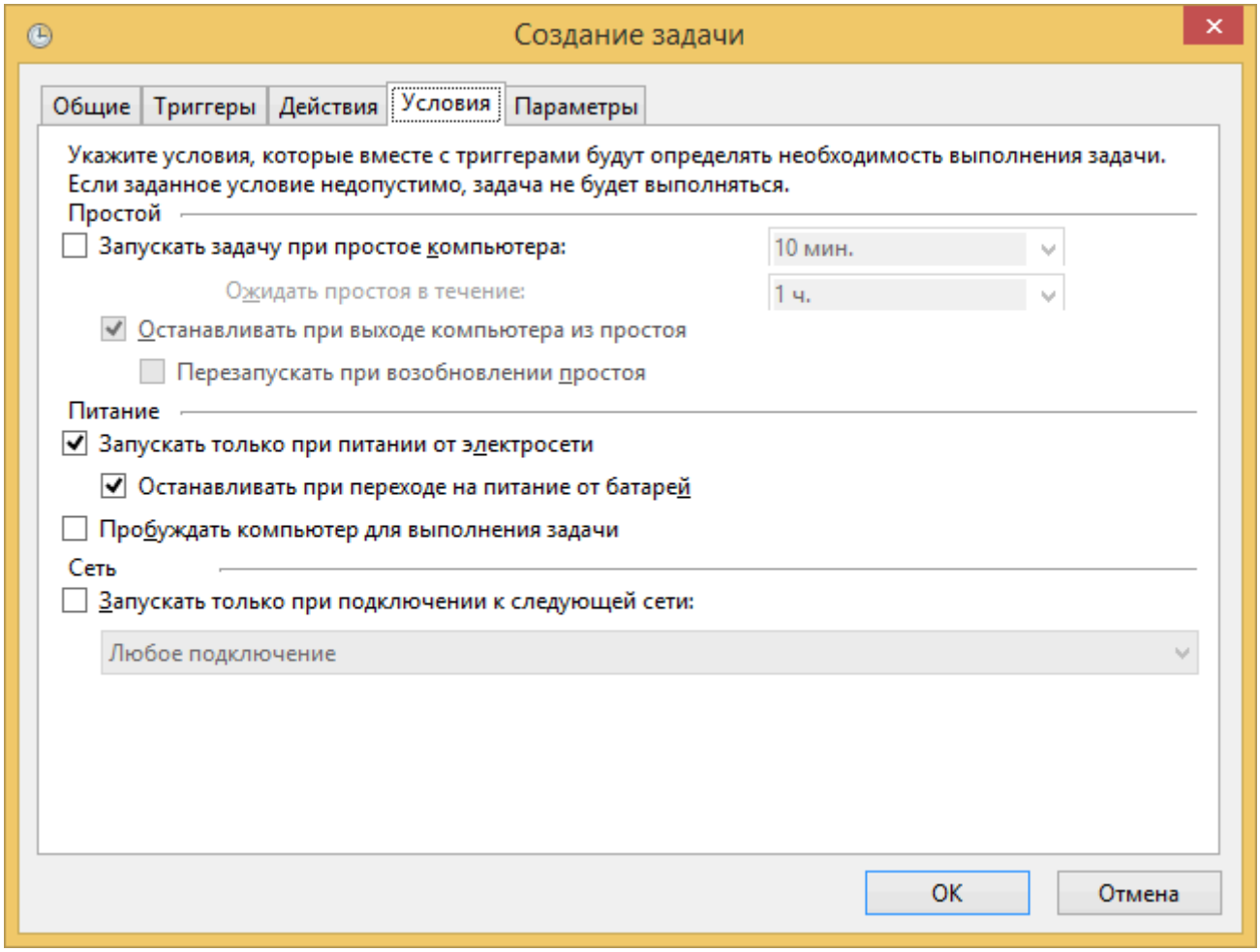


Для целей обучения, лучше определить триггер таким образом, чтобы задача не выполнялась автоматически, а только вручную, например, выбрав режим однократного запуска на уже прошедшую или будущую дату. Это позволит менять настройки задания и выполнять его в нужный момент времени вручную.

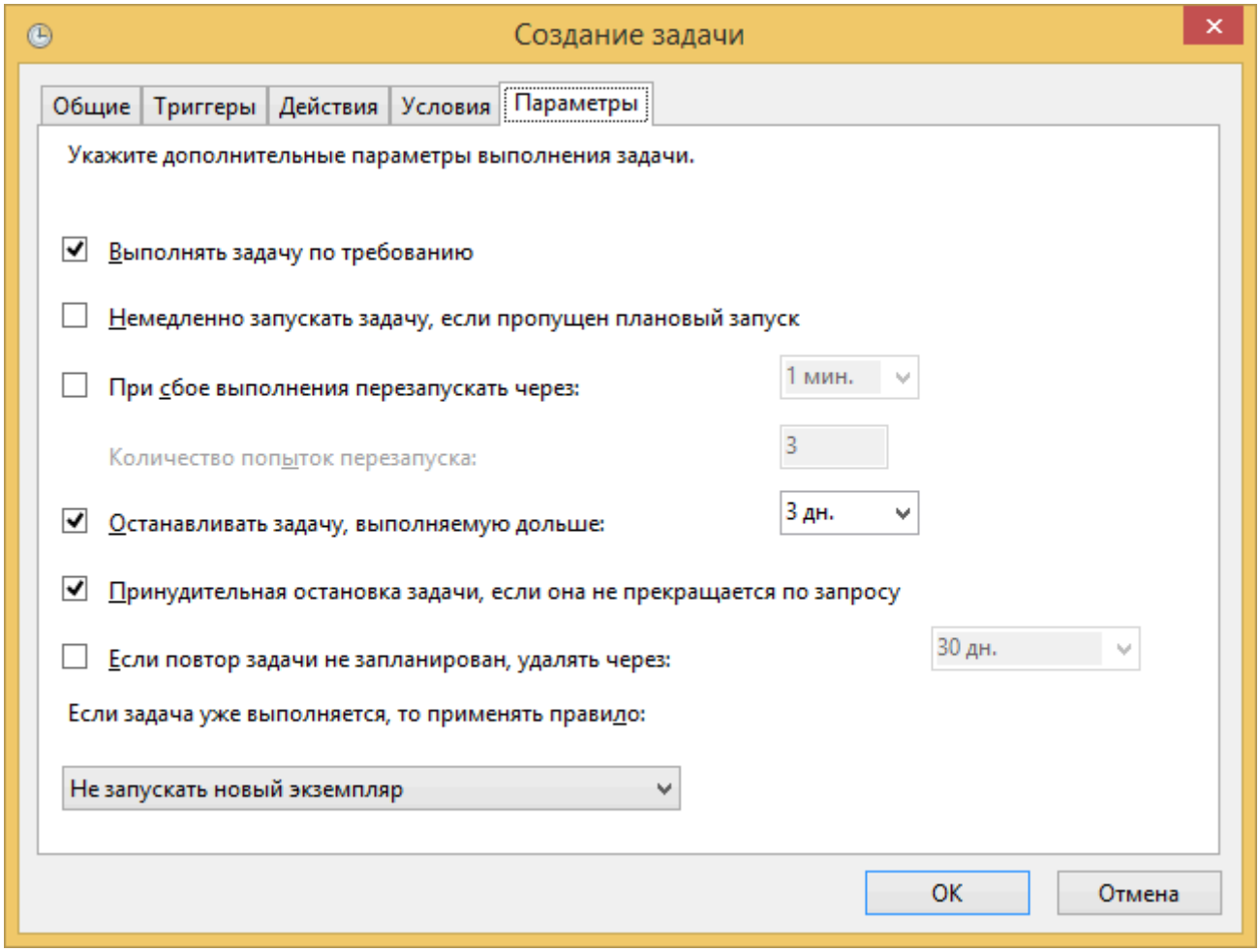
На вкладке **Действия** в качестве действия выбираем **Запуск программы** в качестве которой будет выступать созданный ранее командный файл **who-admin.bat**, который нужно выбрать с использованием кнопки **Обзор…**.



На вкладке **Условия** оставьте настройки по умолчанию. Изменения настроек можно выполнить для определения их влияния на поведение создаваемого задания позже.



На вкладке **Параметры** дополнительные параметры для задания также оставим для последующих экспериментов.



Выполнив все требуемые настройки, нажмите кнопку **OK** и задание будет создано. Практически любые настройки созданного задания можно изменять при наличии у пользователя соответствующих прав.

Диагностика состояния заданий планировщика.

В процессе настройки заданий можно столкнуться с несколькими типами проблем:

- Задание не запускается;
- Задание не завершается;
- Задание завершается с ошибкой;

Для определения статуса задания, выберите требуемое задание в Планировщике заданий и просмотрите сведения, такие как состояние, время прошлого запуска, результат последнего запуска и т. п. Состояния:

Отключено (Disabled) означает, что задание существует, но его выполнение запрещено настройками триггера.

Поставлено в очередь (Queued) означает, что задание ожидает выполнения в

запланированное время.

Готово означает, что задание готово к следующему выполнению.

Также, отображается время следующего запуска, время прошлого запуска, результат выполнения, автор задания и дата его создания.

Для длительного контроля за выполнением заданий планировщиком, можно включить режим ведения журнала, в котором будет сохраняться история выполнения заданий.

Состояние **Работает**, может в действительности означать, что задание зависло, ожидает ввод пользовательских данных, или, например, из-за ошибок программирования, ушло на бесконечный цикл. Узнать, действительно ли задание зависло, можно, проверив столбец **Время прошлого запуска**, в котором указано время запуска задания. Если задание выполняется более суток, то, за редким исключением, это не нормально. Чтобы остановить задание, щелкните на нем правой кнопкой мышки и в контекстном меню выберите команду **Завершить**. Для заданий, время выполнения которых известно, можно установить предельное время выполнения в **Свойства-Параметры-Остановить задачу, выполняемую дольше:**. В случае превышения отведенного на выполнение времени, задание будет принудительно завершено планировщиком.

Имея тестовое задание, попробуйте установить скрытый режим его выполнения ("Скрытая задача"), выполнить с правами администратора ("Выполнить с наивысшими правами"), перенести его на другой компьютер, используя режим экспорта и импорта. Полезно будет поэкспериментировать над условиями и параметрами задания, например, добавив в командный файл команду **pause** при выполнении в скрытом режиме, и тем самым вызвав имитацию зависания задания. Желательно также отработать вариант выполнения просроченного задания, когда запланированное время соответствует выключенному состоянию компьютера, и включен параметр "Немедленно запускать задачу, если пропущен плановый запуск".

Перечень некоторых стандартных заданий Windows 7,8:

Application Experience\AitAgent - Сбор и передача данных дистанционного отслеживания приложений (при явном согласии участвовать в программе улучшения качества программного обеспечения). Даже если время выполнения установлено на 2-3 часа ночи, в настройках присутствует признак выполнения просроченного задания и передача данных в Microsoft выполняется.

Application Experience\ProgramDataUpdater - Сбор телеметрических данных программы при участии в программе улучшения качества ПО

Autochk\Proxy - собирает и загружает данные SQM (при участии в программе улучшения качества программного обеспечения).

Customer Experience Improvement Program\BthSQM - Задача программы улучшения качества Bluetooth собирает статистику по Bluetooth, а также сведения о вашем компьютере, и отправляет их в корпорацию Майкрософт. Полученные сведения используются для повышения надежности, стабильности и общей функциональности Bluetooth в Windows. При отсутствии согласия пользователя на участие в программе улучшения программного обеспечения Windows эта задача не выполняет никаких действий. Задача присутствует в Windows 8.

Customer Experience Improvement Program\Consolidator При выполнении задачи программы улучшения качества ПО, выполняющейся в режиме ядра (Kernel CEIP), осуществляется сбор дополнительных данных о системе, которые затем передаются в корпорацию Майкрософт. Если пользователь не дал своего согласия на участие в данной программе, то эта задача не выполняет никаких действий.

Customer Experience Improvement Program\UsbCeip - При выполнении задачи программы улучшения качества ПО шины USB (USB CEIP) осуществляется сбор статистических данных об использовании универсальной последовательной шины USB и сведений о компьютере, которые направляются инженерной группе Майкрософт по вопросам подключения устройств в Windows. Полученные сведения используются для повышения надежности, стабильности и общей производительности шины USB в Windows. При отсутствии согласия пользователя на участие в программе улучшения программного обеспечения Windows задача не выполняет никаких действий.

RAC\RacTask - Задача средства анализа стабильности системы, предназначенная для обработки данных о надежности компьютера.

Одним из малоизвестных заданий является **Registry\RegIdleBackup** - Задание архивации реестра, выполняющееся раз в 10 дней и обеспечивающее сохранение файлов реестра в папке **C:\Windows\System32\config\RegBack**. В некоторых случаях, когда нет точек восстановления или они некондиционны, есть шанс восстановить работоспособное состояние системы, загрузившись в другой ОС и скопировав файл(ы) реестра в каталог **\Windows\System32\config**. Можно также использовать ручное выполнение данного задания для получения копии файлов реестра, нужных в данный момент времени.

В Windows 8 присутствует папка **\Microsoft\Windows\WS** с несколькими задачами использующими обмен данными с магазином Windows.

Это далеко не полный перечень системных заданий, присутствующих в библиотеке планировщика Windows 7 и Windows 8. Полезность многих из них сомнительна, и даже более того, отрицательная, поскольку может приводить к снижению времени работы ноутбуков от батарей, расходу недешевого мобильного трафика, дополнительных затрат электроэнергии, снижению реальной производительности и т.п. В заключение напомню, что отключение системных заданий требует осторожности и понимания того, что вы делаете, в противном случае, можно получить неработоспособную или частично работоспособную систему.

В библиотеке планировщика, как правило, присутствуют также задания созданные антивирусами для обновлений программ и баз данных, задания, созданные прочими, не всегда нужными, программами, проверяющими наличие новых версий (продукты Adobe, Google, бесплатное ПО и пр.) и даже обновляющихся без ведома пользователя.

Управление заданиями планировщика из командной строки CMD Windows.

Для управления заданиями из командной строки Windows используется утилита **schtasks.exe**.Примеры использования:

schtasks /Query - вывести список всех заданий;

SCHTASKS /Run /I /TN "System Backup" - выполнить на локальном компьютере задание "System Backup" немедленно, игнорируя любые ограничения.

SCHTASKS /end /TN "System Backup" - снять задачу "System Backup"

schtasks /change /tn "\Microsoft\Windows\Application Experience\ProgramDataUpdater" /disable - отключить задание \Microsoft\Windows\Application Experience\ProgramDataUpdater

В заключение, приведу текст командного файла, позволяющего отключить или включить системные задания планировщика, полезность которых сомнительна. В комментариях, которые предшествуют команде **schtasks** представлено краткое описание задания, взятое из общих свойств задания, отображаемое в консоли управления. Для выполнения данного командного файла требуется запуск от имени администратора. Действие, выполняемое по отношению к заданиям планировщика определяется значением переменной **task_action**. Команда **set task_action=disable** приведет к отключению заданий, **set task_action=enable** - к включению. При выполнении команды **schtasks**, ее вывод сохраняется в файле журнала, имя которого соответствует имени командного файла, с расширением **txt**.

```
@echo off
Rem action - установка значение disable или enable
set task_action=disable
REM Проверка наличия прав администратора
@WHOAMI /PRIV | find /i "SeRemoteShutdownPrivilege" > nul
if %ERRORLEVEL% == 0 goto admin
echo Execution aborted. Run this batch file AS Administrator  !!!
pause
exit
:admin
@echo try %task_action% scheduled tasks ...

rem Сбор и передача данных дистанционного отслеживания приложений (при явном
согласии участвовать в
rem программе улучшения качества программного обеспечения).
schtasks /change /tn "\Microsoft\Windows\Application Experience\AitAgent"
/%task_action% >> %~dn0.log

rem Сбор телеметрических данных программы при участии в программе улучшения
качества ПО
schtasks /change /tn "\Microsoft\Windows\Application
Experience\ProgramDataUpdater" /%task_action% >> %~dn0.log

rem Эта задача собирает и загружает данные SQM при участии в программе
улучшения качества
rem программного обеспечения.
schtasks /change /tn "\Microsoft\Windows\Autochk\Proxy" /%task_action% >>
%~dn0.log

rem Если пользователь изъявил желание участвовать в программе по улучшению
качества программного
rem обеспечения Windows, эта задача будет собирать и отправлять сведения о
работе программного
rem обеспечения в Майкрософт.
schtasks /change /tn "\Microsoft\Windows\Customer Experience Improvement
Program\Consolidator" /%task_action% >> %~dn0.log

rem При выполнении задачи программы улучшения качества ПО, выполняющейся в
режиме ядра (Kernel
rem CEIP), осуществляется сбор дополнительных данных о системе, которые затем
передаются в
```

```
rem корпорацию Майкрософт. Если пользователь не дал своего согласия на
участие в данной программе,
rem то эта задача не выполняет никаких действий.
schtasks /change /tn "\Microsoft\Windows\Customer Experience Improvement
Program\KernelCeipTask" /%task_action% >> %~dn0.log

rem При выполнении задачи программы улучшения качества ПО шины USB (USB CEIP)
осуществляется сбор
rem статистических данных об использовании универсальной последовательной
шины USB и сведений о
rem компьютере, которые направляются инженерной группе Майкрософт по вопросам
подключения устройств
rem в Windows. Полученные сведения используются для повышения надежности,
стабильности и общей
rem производительности шины USB в Windows. При отсутствии согласия
пользователя на участие в
rem программе улучшения программного обеспечения Windows задача не выполняет
никаких действий.
schtasks /change /tn "\Microsoft\Windows\Customer Experience Improvement
Program\UsbCeip" /%task_action% >> %~dn0.log

rem Эта задача выполняет дефрагментацию жестких дисков компьютера.
schtasks /change /tn "\Microsoft\Windows\Defrag\ScheduledDefrag"
/%task_action% >> %~dn0.log

rem Измеряет быстродействие и возможности системы
schtasks /change /tn "\Microsoft\Windows\Maintenance\WinSAT" /%task_action%
>> %~dn0.log

REM Запуск приложений, настроенных для Windows HotStart
schtasks /change /tn "\Microsoft\Windows\MobilePC\HotStart" /%task_action% >>
%~dn0.log

rem Это задание предназначено для анализа состояния системы и поиска условий,
которые могут привести
rem к повышенному энергопотреблению.
schtasks /change /tn "\Microsoft\Windows\Power Efficiency
Diagnostics\AnalyzeSystem" /%task_action% >> %~dn0.log

rem Задача средства анализа стабильности системы, предназначенная для
обработки данных о надежности
rem компьютера.
schtasks /change /tn "\Microsoft\Windows\RAC\RacTask" /%task_action% >>
%~dn0.log

REM (Windows Live Social Object Extractor Engine) After installing Windows
Live Essentials 2011,
REM it creates a task called "Extractor Definitions Update Task". The purpose
of this task is to
REM update the definition file for the Windows Live Social Object Extractor
Engine. The task
REM can be found under, Task Scheduler Library\Microsoft\Windows Live\SOXE. -
REM Read more at http://www.shouldiremoveit.com/windows-live-soxe-5304-
program.aspx
schtasks /change /tn "\Microsoft\Windows Live\SOXE\Extractor Definitions
Update Task" /%task_action% >> %~dn0.log

rem Эта запланированная задача предназначена для уведомления пользователя о
том, что архивация
rem Windows не настроена.
schtasks /change /tn "\Microsoft\Windows\WindowsBackup\ConfigNotification"
/%task_action% >> %~dn0.log
```

Учтите, что отключение системных заданий - небезопасная операция, и ее последствия целиком на вашей совести. Так, например, командный файл отключает задание на дефрагментацию файлов, которое не имеет смысла при использовании SSD-диска, но может быть полезным при использовании механических HDD. Кроме того, если вы посмотрите действия, выполняемые заданием дефрагментации, то увидите, что выполняется запуск программы **%windir%\system32\defrag.exe –c**, где **-c** - параметр, задающий выполнение дефрагментации на всех дисках. Может быть, имеет смысл создать несколько заданий с разными триггерами для разных дисков, исключив SSD. Можно также изменить парвметры существующей задачи, исключив дефрагментацию SSD - дисков:

defrag /E C: - выполнить дефрагментацию на всех локальных дисках, кроме диска **C:**

Лабораторная работа №14

Тема: Процессы в системе linux

Цель работы: изучение вопросов порождения и взаимодействия процессов в ОС LINUX.

Теоретическая часть

В ОС Linuxдля создания процессов используется системный вызов***fork():***

#include <sys/types.h>

#include <unistd.h>

pid_t fork (void);

В результате успешного вызова ***fork()***ядро создаёт новый процесс, который является почти точной копией вызывающего процесса. Другими словами, новый процесс выполняет копию той же программы, что и создавший его процесс, при этом все его объекты данных имеют те же самые значения, что и в вызывающем процессе. Созданный процесс называется***дочерним процессом***, а процесс, осуществивший вызов***fork()***, называется***родительским***. После вызова родительский процесс и его вновь созданный потомок выполняются одновременно, при этом оба процесса продолжают выполнение с оператора, который следует сразу же за вызовом***fork()***. Процессы выполняются в разных адресных пространствах, поэтому прямой доступ к переменным одного процесса из другого процесса невозможен.

Следующая короткая программа более наглядно показывает работу вызова ***fork()*** и использование процесса:

#include <stdio.h>

#include <unistd.h>

Int main ()

{

pid_t pid; /* идентификатор процесса */

printf (“Пока всего один процесс\n”);

pid = fork (); /* Создание нового процесса */

printf (“Уже два процесса\n”);

If (pid = = 0){

printf (“Это Дочерний процесс его pid=%d\n”, getpid());

printf (“А pid его Родительского процесса=%d\n”, getppid());

}

else if (pid > 0)

printf (“Это Родительский процесс pid=%d\n”, getpid());

else

printf (“Ошибка вызова fork, потомок не создан\n”);

}

Для корректного завершения дочернего процесса в родительском процессе необходимо использовать функцию ***wait()*** или ***waitpid()***:

pid_t wait(int *status); pid_t waitpid(pid_t pid, int *status, int options);

Функция ***wait***приостанавливает выполнение родительского процесса до тех пор, пока дочерний процесс не прекратит выполнение или до появления сигнала, который либо завершает текущий процесс, либо требует вызвать функцию-обработчик. Если дочерний процесс к моменту вызова функции уже завершился (так называемый***«зомби»***), то функция немедленно возвращается. Системные ресурсы, связанные с дочерним процессом, освобождаются.

Функция ***waitpid ()***приостанавливает выполнение родительского процесса до тех пор, пока дочерний процесс, указанный в параметре***pid***, не завершит выполнение, или пока не появится сигнал, который либо завершает родительский процесс, либо требует вызвать функцию-обработчик. Если указанный дочерний процесс к моменту вызова функции уже завершился (так называемый***«зомби»***), то функция немедленно возвращается. Системные ресурсы, связанные с дочерним процессом, освобождаются. Параметр***pid***может принимать несколько значений:

pid< -1 означает, что нужно ждать любого дочернего процесса, чей идентификатор группы процессов равен абсолютному значению***pid***.

pid= -1 означает ожидать любого дочернего процесса; функция**wait**ведет себя точно так же.

pid = 0означает ожидать любого дочернего процесса, чей идентификатор группы процессов равен таковому у текущего процесса.

pid> 0означает ожидать дочернего процесса, чем идентификатор равен***pid***.

Значение ***options***создается путем битовой операции***ИЛИ***над следующими константами:

WNOHANG- означает вернуть управление немедленно, если ни один дочерний процесс не завершил выполнение.

WUNTRACED - означает возвращать управление также для остановленных дочерних процессов, о чьем статусе еще не было сообщено.

Каждый дочерний процесс при завершении работы посылает своему процессу-родителю специальный сигнал ***SIGCHLD***, на который у всех процессов по умолчанию установлена реакция "игнорировать сигнал". Наличие такого сигнала совместно с системным вызовом***waitpid()***позволяет организовать асинхронный сбор информации о статусе завершившихся порожденных процессов процессом-родителем.

Для перегрузки исполняемой программы можно использовать функции семейства ***exec***. Основное отличие между разными функциями в семействе состоит в способе передачи параметров.

Int execl(char *pathname, char *arg0, arg1, ..., argn, null);

int execle(char *pathname, char *arg0, arg1, ..., argn, NULL, char **envp);

int execlp(char *pathname, char *arg0, arg1, ..., argn, NULL);

int execlpe(char *pathname, char *arg0, arg1, ..., argn, NULL, char **envp);

int execv(char *pathname, char *argv[]);

int execve(char *pathname, char *argv[],char **envp);

int execvp(char *pathname, char *argv[]);

int execvpe(char *pathname, char *argv[],char **envp);

Основное отличие между разными функциями в семействе состоит в способе передачи параметров. Как видно из рис. 1, все эти функции выполняют один системный вызов ***execve***.

execl

execle

execve

execvp

execv

execlp

Рис. 1. Дерево семейства вызовов ***exec***

Порядок выполнения работы

1. Изучить теоретическую часть лабораторной работы.
2. Написать программу, создающую два дочерних процесса с использованием двух вызовов ***fork()***. Родительский и два дочерних процесса должны выводить на экран свой***pid***и***pid***родительского процесса и текущее время в формате:***часы: минуты:***

секунды: миллисекунды. Используя вызов***system (),***выполнить команду***ps -x***в родительском процессе. Найти свои процессы в списке запущенных процессов.

Варианты индивидуальных заданий

1. Написать программу нахождения массива ***N***последовательных значений функции***y[i]=sin(2*PI*i/N) (i=0,1,2…N-1)***с использованием ряда Тейлора. Пользователь задаёт значения***N*** и количество***n***членов ряда Тейлора. Для расчета каждого членаряда Тейлоразапускается отдельный процесс. Каждый процесс выводит на экран и в файл свой***pid***и рассчитанное значениеряда. Головной процесс считывает из файла значения всех рассчитанных членов ряда Тейлора, суммирует ихи полученное значение ***y[i]*** записывает в файл. Проверить работу программы для***N=256 n=5; N=1024 n=10.***
2. Написать программу синхронизации двух каталогов, например, ***Dir1***и***Dir2***. Пользователь задаёт имена***Dir1***и***Dir2***. В результате работы программы файлы, имеющиеся в***Dir1***, но отсутствующие в***Dir2***, должны скопироваться в***Dir2***вместе с правами доступа. Процедуры копирования должны запускаться в отдельном процессе для каждого копируемого файла. Каждый процесс выводит на экран свой***pid,***полный путь к копируемому файлу и число скопированных байт. Число одновременно работающих процессов не должно превышать***N***(вводится пользователем). Скопировать несколько файлов из каталога***/etc*** в свой домашний каталог. Проверить работу программы для каталога***/etc и***домашнего каталога.
3. Написать программу поиска одинаковых по содержимому файлов в двух каталогов, например, ***Dir1***и***Dir2***. Пользователь задаёт имена***Dir1***и***Dir2***. В результате работы программы файлы, имеющиеся в***Dir1***, сравниваются с файлами в***Dir2*** по их содержимому. Процедуры сравнения должны запускаться в отдельном процессе для каждой пары сравниваемых файлов. Каждый процесс выводит на экран свой***pid***, имя файла, общее число просмотренных байт и результаты сравнения. Число одновременно работающих процессов не должно превышать***N***(вводится пользователем). Скопировать несколько файлов из каталога***/etc*** в свой домашний каталог. Проверить работу программы для каталога***/etc и***домашнего каталога.
4. Написать программу поиска заданной пользователем комбинации из ***m***байт (***m <255***) во всех файлах текущего каталога. Пользователь задаёт имя каталога. Главный процесс открывает каталог и запускает для каждого файла каталога отдельный процесс поиска заданной комбинации из***m***байт. Каждый процесс выводит на экран свой***pid,*** полный путь к файлу, общее число просмотренных байт и результаты (сколько раз найдена комбинация) поиска. Число одновременно работающих процессов не должно превышать***N***(вводится пользователем). Проверить работу программы для каталога***/etc и***строки ***ifconfig***.
5. Разработать программу «***интерпретатор команд***», которая воспринимает команды, вводимые с клавиатуры, (например,***ls -l /bin/bash***) и осуществляет их корректное выполнение. Для этого каждая вводимая команда должна выполняться в отдельном процессе с использованием вызова***exec()***. Предусмотреть возможность перенаправления (ввода\вывода ’>> ’, ’> ’, ’ <’).
6. Написать программу подсчета количества слов в файлах заданного каталога его подкаталогов. Пользователь задаёт имя каталога. Главный процесс открывает каталоги и запускает для каждого файла каталога отдельный процесс подсчета количества слов. Каждый процесс выводит на экран свой ***pid,*** полный путь к файлу, общее число просмотренных байт и количество слов. Число одновременно работающих процессов не должно превышать***N***(вводится пользователем). Проверить работу программы для каталога***/etc и***строки ***ifconfig***.

7. Написать программу подсчета частоты встречающихся символов в файлах заданного каталога его подкаталогов. Пользователь задаёт имя каталога. Главный процесс открывает каталоги и запускает для каждого файла каталога и отдельный процесс подсчета количества слов. Каждый процесс выводит на экран свой ***pid,*** полный путь к файлу, общее число просмотренных байт и количество слов. Число одновременно работающих процессов не должно превышать***N***(вводится пользователем). Проверить работу программы для каталога***/etc и***строки ***ifconfig***.

Лабораторная работа №15

Тема: Управление памятью в ос linux

Цель работы - изучение механизма взаимодействия процессов с использованием общей памяти.

Теоретическая часть

Использование общей или разделяемой памяти заключается в создании специальной области памяти, позволяющей иметь к ней доступ нескольким процессам. Системныевызовыдля работы с разделяемой памятью:

#include <sys/mman.h>

Int shm_open (const char *name, int oflag, mode_t mode);

Int shm_unlink (const char *name);

Вызов***shm_open***создает и открывает новый (или уже существующий) объект разделяемой памяти. При открытии с помощью функции***shm_open()***возвращается файловый дескриптор. Имя***name***трактуется стандартным для рассматриваемых средств межпроцессного взаимодействия образом. Посредством аргумента***oflag***могут указываться флаги***O_RDONLY, O_RDWR, O_CREAT, O_EXCL***и/или***O_TRUNC***. Если объект создается, то режим доступа к нему формируется в соответствии со значением***mode***и маской создания файлов процесса. Функция***shm_unlink***выполняет обратную операцию, удаляя объект, предварительно созданный с помощью***shm_open***. После подключения сегмента разделяемой памяти к виртуальной памяти процесса этот процесс может обращаться к соответствующим элементам памяти с использованием обычных машинных команд чтения и записи, не прибегая к использованию дополнительных системных вызовов.

Int main (void) {

int fd_shm; /* Дескриптор объекта в разделяемой памяти*/

If ((fd_shm = shm_open (“myshered.Shm”, o_rdwr | o_creat, 0777)) < 0) { perror (''error create shm''); return (1); }

Для ускорения работы следует использовать файлы, отображаемые в памяти при помощи системного вызова ***mmap()***:

#include <unistd.h> #include <sys/mman.h>

Void * mmap(void *start, size_t length, int prot , int flags, int fd, off_t offset);

Функция ***mmap***отображает***length***байтов, начиная со смещения***offset***файла, определенного файловым описателем***fd***, в память, начиная с адреса***start***. Последний параметр***offset***необязателен, и обычно равен***0***. Настоящее местоположение отраженных данных возвращается самой функцией***mmap***, и никогда не бывает равным***0***.

Аргумент***prot***описывает желаемый режим защиты памяти (он не должен конфликтовать с режимом открытия файла):

PROT_EXEC данные в памяти могут исполняться;

PROT_READ данные в памяти можно читать;

PROT_WRITE в область можно записывать информацию;

PROT_NONE доступ к этой области памяти запрещен.

Параметр ***flags***задает тип отражаемого объекта, опции отражения и указывает, принадлежат ли отраженные данные только этому процессу или их могут читать другие. Он состоит из комбинации следующих битов:

MAP_FIXED использование этой опции не рекомендуется;

MAP_SHARED разделить использование этого отражения с другими процессами, отражающими тот же объект. Запись информации в эту область памяти будет эквивалентна записи в файл. Файл может не обновляться до вызова функций***msync(2)***или***munmap(2)***;

MAP_PRIVATE создать неразделяемое отражение с механизмом***copy-on-write***. Запись в эту область памяти не влияет на файл. Не определено, являются или нет изменения в файле после вызова***mmap***видимыми в отраженном диапазоне.

Для компиляции программы необходимо подключить библиотеку ***rt.lib*** следующим способом:***gcc 1.c –o 1.exe -lrt***

Порядок выполнения работы

1. Изучить теоретическую часть лабораторной работы
2. Написать программу, создающую дочерний процесс. Родительский процесс создаёт семафор (***сем1***) и общую память. Дочерний процесс записывает в память по 70 строк всего***1000***строк вида:***номер_строкиpid_процессатекущее_время*** (мсек). Родительский процесс читает из файла по 81 строке и выводит их на экран в следующем виде:***pid строка_прочитанная_из_файла***. Семафор***сем1***используется процессами для разрешения, кому из процессов получить доступ к файлу.

1. Организовать функционирование процессов следующей структуры:

1-2-3-4(процесс 1 создаёт процесс 2, процесс 2 создаёт процесс 3, процесс 3 создаёт процесс 4). Далее организовать с использованием общей памяти передачу/приём следующей информации в следующей последовательности:***1-2, 2-3, 3-4, 4-1, 1-2, 2-3, 3-4, 4-1***и так далее. Каждый процесс выдерживает паузу***t=100***мсек между приёмом и посылкой информации и передаёт следующую информацию:

N pid ppid текущее время(мсек) ***процесс_такой-то***.

Для синхронизации работы использовать семафоры.

2. Организовать функционирование процессов следующей структуры:

1-2-3-4(процесс 1 создаёт процесс 2, процесс 2 создаёт процесс 3, процесс 3 создаёт процесс 4). Далее организовать с использованием общей памяти передачу/приём следующей информации в следующей последовательности:***1-2, 2-3, 3-4, 4-1, 1-2, 2-3, 3-4, 4-1***и так далее. Каждый процесс выдерживает паузу***t=100***мсек между приёмом и посылкой информации и передаёт следующую информацию:

N pid ppid текущее время(мсек) ***процесс_такой-то***.

Для синхронизации работы использовать сигналы.

Лабораторная работа № 16

Тема: Политика безопасности и ограничения программ в ос Windows XP

1. Права пользователей

Политика безопасности системы является одной из важнейших составляющих в обеспечении надежной и защищенной работы *Windows XP*. Настройка политики безопасности осуществляется в программе *Local Security Settings*:

Пуск\Панель управления\Администрирование\Локальная политика безопасности\Назначение прав пользователя

После запуска программы *Назначение прав пользователя* появится окно *Локальные параметры безопасности* (рис.1.1.)

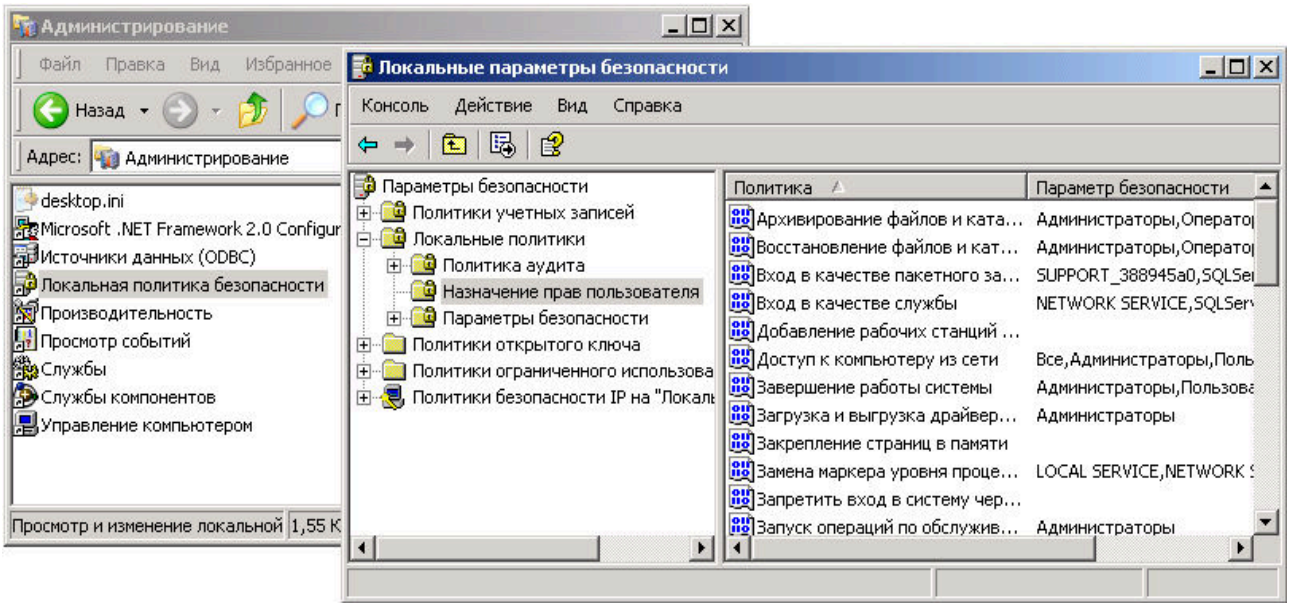


Рисунок 1.1. Окно Локальные параметры безопасности

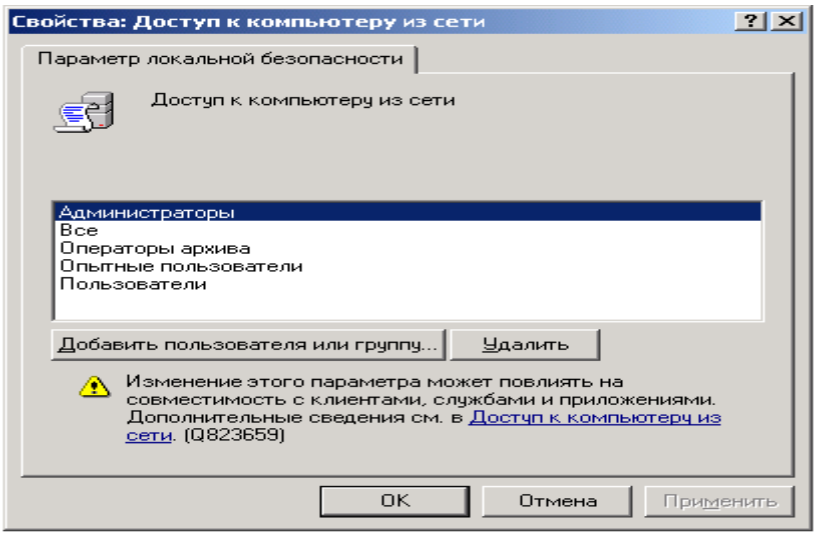


Рисунок 1.2. Окно *Параметр локальной безопасности*

Основные пункты политики безопасности.

1.Пункт *Доступ к компьютеру из сети* – определяет, какие именно пользователи и группы пользователей могут получать доступ к данному компьютеру по компьютерной сети. Если компьютер не подключен к локальной сети, рекомендуется запретить доступ пользователей извне, это позволит избежать атак взломщиков и их проникновение в систему при работе в Интернете. Для запрета доступа сетевых пользователей к компьютеру следует:

- в окне *Политика* программы *Назначение прав пользователя* щелчком мыши выбрать политику *Доступ к компьютеру из сети*;
- появится окно *Параметр локальной безопасности Доступ к компьютеру из сети* (рис.1.2.);
- выделить всех пользователей (или лишних пользователей) при помощи указателя мыши и клавиши *Shift*;
- сделать щелчок по кнопке *Удалить*;
- нажать кнопку *ОК*.

Пользователи, которым разрешен доступ к компьютеру, должны быть отображены в данном пункте политики безопасности, иначе они не смогут войти в систему. Если пользователи в списке окна отсутствуют, то их следует добавить при помощи кнопки *Добавить пользователя или группу.* Для этого следует:

- сделать щелчок по кнопке *Добавить пользователя или группу*;
- в появившемся диалоговом окне сделать щелчок по кнопке *Дополнительно*;
- в окне *Пользователи* или группы нажать кнопку *Поиск*;
- в нижней части окна появится список всех пользователей и групп;
- щелчком выбрать нужную строку нажать кнопку *ОК*;
- в появившемся диалоговом окне в поле *Введите имена выбираемых объектов* появится выбранный пользователь (группа), нажать кнопку *ОК*;
- выбранный пользователь (группа) будет отображен в окне *Доступ к компьютеру из сети*.

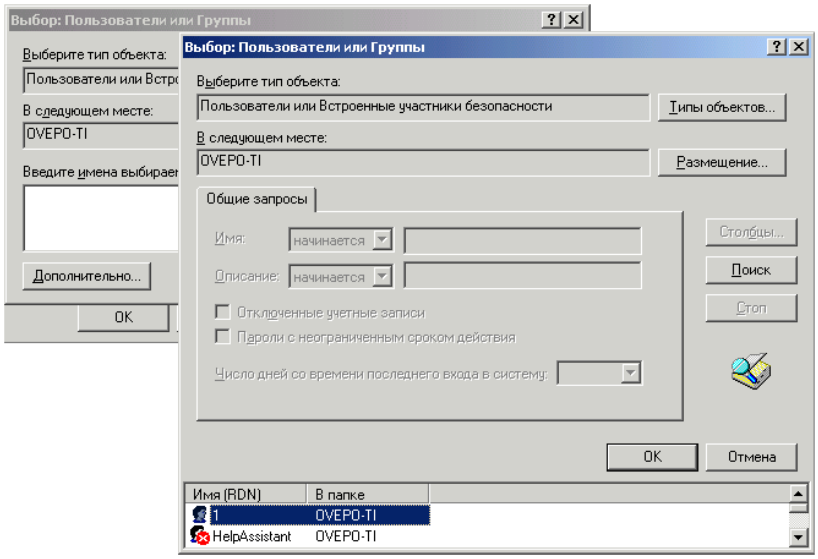


Рисунок 1.3. Добавление пользователей или групп

2.Пункт *Разрешать вход в систему через службу терминалов* является аналогичным предыдущему, но вход пользователей в систему осуществляется в качестве клиентов терминал-сервера. Если данный сервис не используется, то рекомендуется аналогичным методом запретить вход в систему всех пользователей, убрав их из значения данного пункта как клиентов терминал-сервера. В случае необходимости всегда можно добавить нужных пользователей и их группы при помощи кнопки *Добавить пользователя или группу* (рис.1.4.).

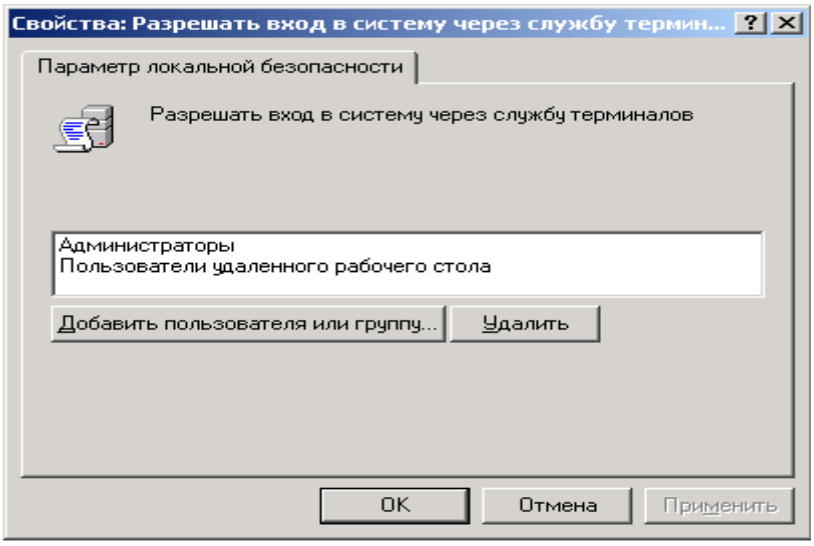


Рисунок 1.4. Окно *Разрешить вход в систему через службу терминалов*

3.Пункт *Изменение системного времени*, позволяющий пользователям, перечисленным в нем, менять системное время, а также просматривать календарь, появляющийся на экране при двойном щелчке по текущему времени на панели задач. По умолчанию данной возможностью обычные пользователи не смогут воспользоваться. Для разрешения пользователям выполнять такое действие следует их внести в список данного пункта политики безопасности при помощи кнопки *Добавить пользователя или группу* (рис.1.5.).

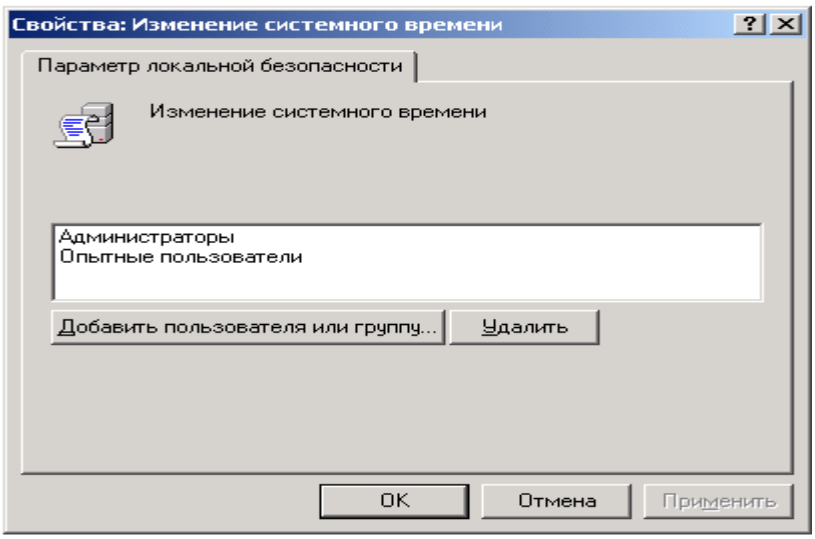


Рисунок 1.5. Окно *Изменение системного времени*

4.Пункт *Отладка программ* позволяет указать пользователей, которые смогут подсоединять свой отладчик к процессам и производить их отладку. Следует включать в этот пункт только тех пользователей, которым это действительно нужно, например, *системный администратор* и *системные программисты*. Не следует давать это право другим пользователям, так как этой возможностью могут воспользоваться вирусы для заражения системы, запущенные под одной из пользовательских записей, имеющей право

на отладку процессов.

5.Пункт *Отказ в доступе к компьютеру из сети* содержит пользователей и их группы, которым запрещен вход в систему по компьютерной сети. При необходимости можно добавить пользователей, которым запрещен доступ к компьютеру с помощью кнопки *Добавить пользователя или группу* (рис.1.6.).

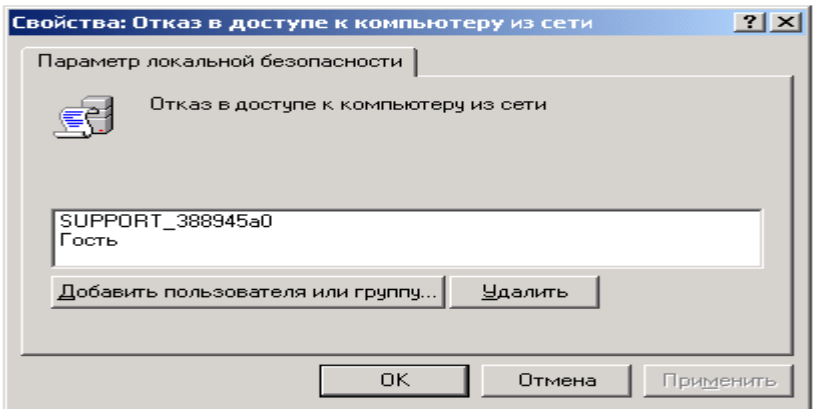


Рисунок 1.5. Окно *Отказ в доступе к компьютеру из сети*

6.Пункт *Отклонить локальный вход* содержит пользователей и их группы, которым запрещен локальный вход в систему. При необходимости можно добавить пользователей, которым запрещен доступ к компьютеру с помощью кнопки *Добавить пользователя или группу* (рис.1.7.).

7.Пункт *Запретить вход через службу терминалов* также содержит пользователей и их группы, которым запрещен вход в систему как клиентов терминал-сервера. При необходимости можно добавить пользователей, которым запрещен доступ к компьютеру с помощью кнопки *Добавить пользователя или группу* (рис.1.8.).

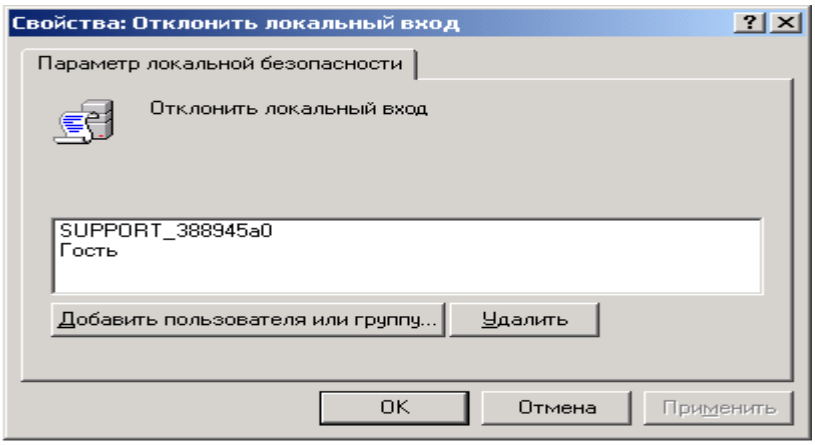


Рисунок 1.7. Окно *Отлонить локальный вход*

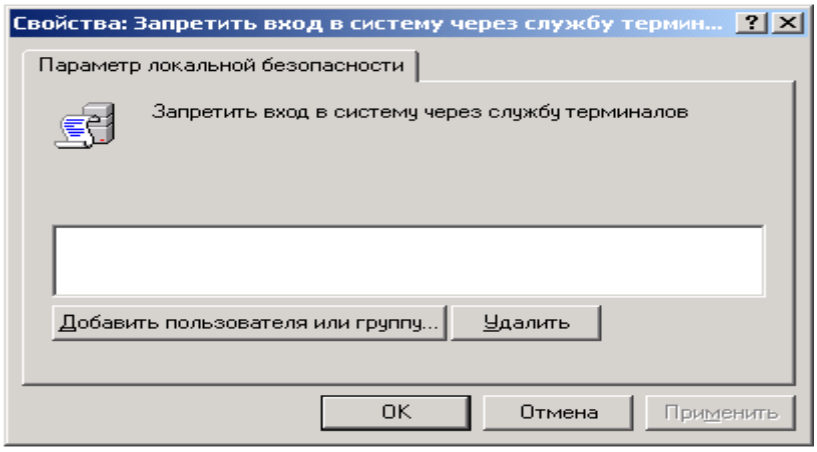


Рисунок 1.8. Окно *Запретить вход в систему через службу терминалов*

С помощью трех перечисленных выше опций локальной политики безопасности можно запретить пользователям, которые по структуре организации не должны получать доступа, вход в систему. Этим можно предотвратить внутренние коллизии организации и защитить данные от их искажения или разрушения пользователями, которые удаленно пытаются ими воспользоваться.

8.Пункт *Принудительное удаленное завершение* является очень важным в настройке локальной политики безопасности, так как если его не настроить соответствующим образом, то система может получить команду на выключение или перезагрузку от удаленно пользователя. Поэтому в данном пункте следует указывать только пользователей, которым действительно может потребоваться с машин, находящихся в локальной сети, выключить или перезапустить систему.

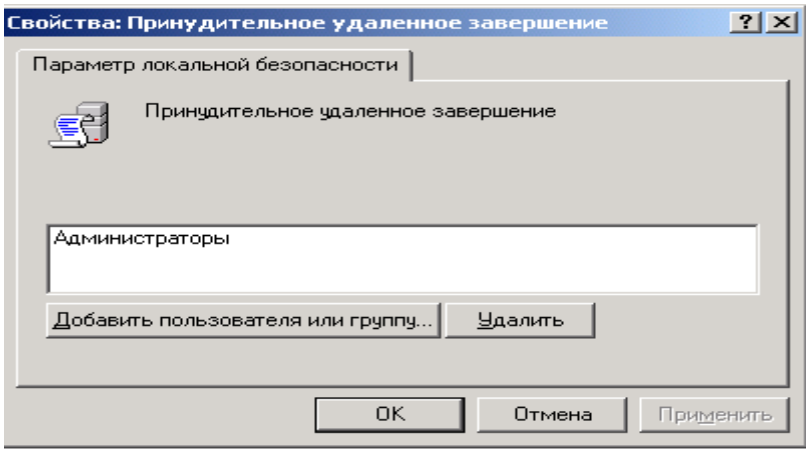


Рисунок 1.9. Окно *Принудительное удаленное завершение*

9.Пункт *Загрузка и выгрузка драйверов устройств* позволяет указать, кто из пользователей может динамически устанавливать и выгружать драйвера устройств. Это право необходимо для установки драйверов устройств, имеющих спецификацию *Plug and Play*.

10.Пункт *Локальный вход в систему* является очень важным и определяет, какие пользователи и их группы могут локально входить в систему.

11.Пункт *Управление аудитом и журналом безопасности* относится к механизму аудита системы и определяет, какие пользователи и их группы могут устанавливать аудит доступа к определенным объектам, таким как файлы, ключи реестра и пр. По умолчанию в данном пункте перечислена лишь одна группа локальных системных администраторов.

12.Пункт *Изменение параметров среды оборудования* определяет пользователей, которые будут иметь право в *Windows XP* менять значения системных переменных. По умолчанию на это имеют право только пользователи, принадлежащие локальной группе администраторов.

13.Пункт *Запуск операций по обслуживанию тома* позволяет указать пользователей и их группы, которые будут иметь право выполнять задачи по поддержанию работы накопителей, такие как очистка диска или его дефрагментация. Выполнение данных задач, по умолчанию, доверяется только пользователям из группы системных администраторов.

14.Пункт *Восстановление файлов и каталогов* позволяет указывать пользователей и их группы, которые могут выполнять операцию восстановления файлов и директорий из сохраненных копий, а также ставить им необходимые права доступа. По умолчанию в системе такими пользователями являются члены группы системных администраторов, а также операторы сохранения данных.

15.Пункт *Завершение работы системы* указывает, кто из локальных пользователей, имеющих учетные записи в системе, имеет право на ее выключение или перезагрузку. По умолчанию на это имеют право все пользователи. Однако, в ряде случаев, может потребоваться запретить выполнять данные функции некоторым пользователям. Например, если нужно, чтобы компьютеры работали в то время, когда некоторые пользователи их пытаются отключить. В этом случае нужно убрать этих пользователей из данного пункта. Особенно это может быть полезно, если определенные пользователи пытаются выключить компьютер, на котором находится информация, используемая удаленно другими пользователями.

16.Пункт *Овладение файлами или иными объектами* отвечает за возможность пользователей, перечисленных в нем, брать на себя право становиться владельцами файлов и объектов. Этими объектами могут быть структуры *Active Directory*, ключи реестра, принтеры и процессы. По умолчанию на это имеют право только пользователи группы системных администраторов. Добавление к этому пункту пользователей означает предоставление им всех прав по доступу к различным объектам.

1.2. Глобальные параметры безопасности системы

Глобальные параметры безопасности устанавливаются в разделе локальной политики безопасности *Параметры безопасности* (рис.1.10).

Пуск\Панель управления\Администрирование\Локальная политика безопасности\Параметры безопасности

Рассмотрим наиболее важные пункты.

1.Пункт *Учетные записи*: *Состояние учетной записи ‘Администратор’* предоставляет возможность выбора: будет ли учетная запись администратора системы включена или отключена, при нормальном функционировании системы. В случае использования системы в безопасном режиме запись администратора будет включена, независимо от значения данного пункта. Для изменения значения этого пункта следует его выбрать двойным щелчком мыши и в появившемся окне поставить флажок в соответствующем режиме (рис.1.11.)

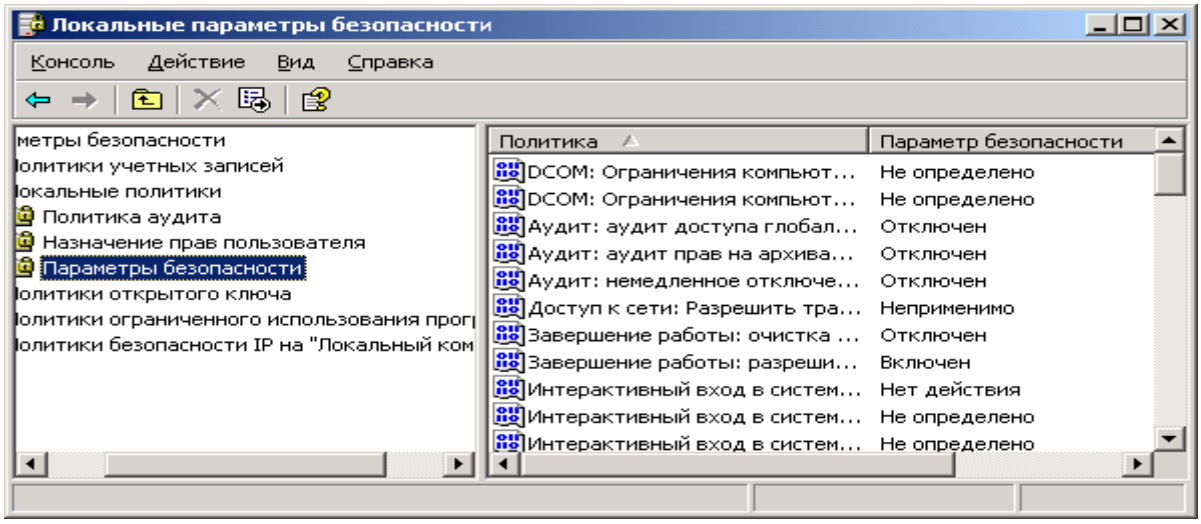


Рисунок 1.10. Окно *Параметры безопасности*

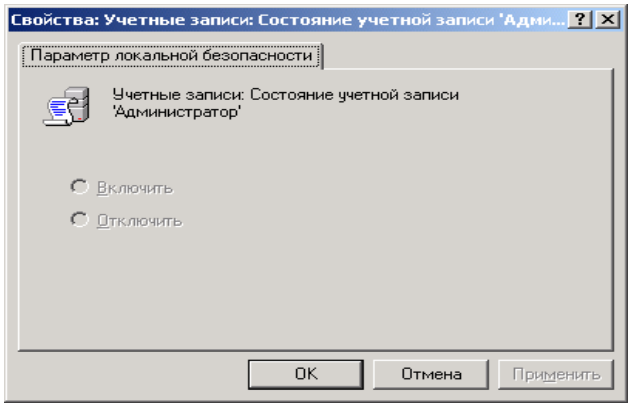


Рисунок 1.11. Окно *Состояние учетной записи ‘Администратор’*

Отключение учетной записи системного администратора может быть полезно, т.к. это дает гарантированную защиту от атак взломщиков на эту учетную запись. Если необходимо включить учетную запись системного администратора, то это можно сделать под учетной записью другого пользователя, принадлежащего к группе системных администраторов, или в защищенном режиме работы операционной системы.

Пункт *Учетные записи: Состояние учетной записи 'Гость’* позволяет отключать учетную

запись гостя, т.к. для входа под данной учетной записью не требуется пароль, что может нарушить политику прав доступа пользователями. Учетная запись *Гость* по умолчанию отключена.

Пункт *Accounts: Limit local account use of blank passwords to console logon only*, в случае включения позволяет ограничить доступ к незащищенным паролями консольным учетным записям локальных пользователей со стороны различных сетевых сервисов, например: терминал-сервера, *Telnet* и *FTP*. По умолчанию, в целях защиты системы от сетевых атак, данный пункт включен.

Пункт Accounts: Rename administrator account позволяет переименовать встроенную учетную запись администратора системы. Это делается в целях защиты от атаки методом подбора паролей. Чтобы изменить имя учетной записи администратора, нужно дважды щелкнуть мышью по имени этого пункта и в появившемся окне ввести новое имя этой учетной записи.

Пункт *Audit: Audit the use of Backup and Restore privilege* позволяет контролировать выполнение всех операций сохранения и восстановления данных. Система будет сохранять сообщения обо всех резервируемых и восстанавливаемых файлах и папках. Это очень удобно для проведения контроля за операциями резервирования и восстановления данных. Для работы данного пункта необходимо включение в политике аудита опции *Аудит использования привилегий*. По умолчанию данный пункт локальной политики безопасности отключен.

Пункт *Audit: Shut down system immediately if unable to log security audits* является очень полезным и позволяет после своего включения, в случае обнаружения операционной системой невозможности производить запись событий аудита, произвести автоматическое выключение системы. Невозможность записи аудита событий системы обычно связана с переполнением хранилища этих сообщений. Для продолжения нормальной работы системы необходимо войти в нее под учетной записью администратора и произвести в программе:

Пуск\Панель управления\Администрирование\Просмотр событий

очистку всех этих сообщений, возможно, предварительно их сохранив. Это является гарантией того, что все действия системы или пользователей будут контролироваться администратором.

Пункт *Devices*: *Prevent users from installing printer drivers* – позволяет запретить пользователям устанавливать драйвера принтеров под их учетными записями.

Пункт *Devices: Restrict CD-ROM access to locally logged-on user only* позволяет ограничить доступ сетевых пользователей к локальному *CD-ROM*-приводу системы. Это может быть полезно, когда нужно чтобы сетевые пользователи имели доступ только к тем ресурсам, к которым они должны его иметь.

Пункт *Devices: Restrict floppy access to locally logged-on user only* позволяет ограничить доступ сетевых пользователей к локальному *CD-ROM*-приводу системы. Это может быть полезно, когда вы хотите, чтобы сетевые пользователи имели доступ только к тем ресурсам, к которым они должны его иметь. Это позволит локальным пользователям приватно работать с их личными носителями.

Пункт *Devices: Unsigned driver installation behavior* позволяет указать поведение системе, при попытке пользователей установить драйвер, не прошедший процедуру сертификации *Microsoft*. Он может иметь три значения:

- *Silent succeed* – происходит инсталляция этого драйвера и никаких сообщений не выдается;
- *Warn but allow installation* – происходит предупреждение пользователя о том, что драйвер не прошел сертификацию, но инсталляция продолжается. Данный пункт используется по умолчанию;
- *Do not allow installation* – накладывается запрет на установку драйверов системы, не прошедших сертификацию.

Пункт *Interactive logon: Do not display last user name*, в случае своего включения, запрещает показ системе имени пользователя, который в ней работал последним. Это удобно в тех случаях, когда нужно избежать подбора паролей взломщиками к учетным записям пользователей системы, т.к. если у них не будет не только пароля, но и имени учетной записи пользователя, то их задача может стать в два раза сложнее. Данный пункт работает только в том случае, если отключен экран приветствия системы.

Пункт *Interactive logon: Do not require CTRL+ALT+DEL,* в случае его выключения, производит отображение на экране таблички, требующей от пользователя нажатия комбинации клавиш *CTRL+ALT+DEL* для входа в систему. В случае включения этого пункта данное сообщение системы появляться не будет. Данный пункт работает только в том случае, если отключен экран приветствия. Смысл ввода этой комбинации клавиш для входа в систему заключается в том, что она обрабатывается только системой. И это гарантирует то, что в операционную систему входит человек, а не программа по подбору паролей пользователей. Таким образом, данное сообщение может быть дополнительным барьером, охраняющим систему от взломщиков.

Пункт *Interactive logon: Prompt user to change password before expiration* устанавливает количество дней до конца срока действия пароля пользователя, когда система будет предупреждать пользователя об этом. Данный пункт имеет смысл только в том случае, если пароли пользователей имеют определенный срок действия.

Пункт *Recovery console: Allow automatic administrative logon* устанавливает автоматический вход системного администратора в консоль восстановления системы. Это удобно тем, что не требует ввода пароля администратора, но по той же причине, создает большие проблемы с безопасностью, так как консолью восстановления с администраторскими правами сможет воспользоваться любой желающий.

Пункт Recovery console: *Allow floppy copy and access to all drives and all folders*, в случае его включения, позволяет вам использовать команду *SET* консоли восстановления, которая может помочь установить следующие значения переменных:

- *AllowWildCards* – переменная включает поддержку масок у команд, например, *DEL*;

- *AllowAllPath* – переменная позволяет получить доступ ко всем файлам и папкам системы.
- *AllowRemovableMedia* – переменная позволяет копировать файлы на сменные носители, например, гибкие диски;
- *NoCopyPrompt* – переменная запрещает системе выдавать дополнительные сообщения при перезаписи существующего файла.

С помощью данного пункта можно скопировать или удалить информацию с жесткого диска системы. Поэтому не рекомендуется совмещать его использование с включенным предыдущим пунктом, позволяющим вход в консоль восстановления системы без администраторского пароля, т.к. можно лишиться всей информации.

Пункт *Shutdown: Allow system to be shut down without having to log on* позволяет, в случае его включения, производить выключение операционной системы до непосредственного входа в нее пользователями. Если вы не хотите, чтобы пользователи, не имеющие на это прав, выключали систему, установите данный пункт в положение Отключен.

Пункт *Shutdown: Clear virtual memory pagefile* является чрезвычайно важным в обеспечении безопасности вашей системы. При выключении системы в ее файле подкачки остаются данные, которые использовались в работе различными пользовательскими приложениями. Среди этих данных могут быть, частично или полностью, ваши документы, с которыми вы работали в течение сеанса работы с системой. Впоследствии, во время вашего отсутствия, эти данные могут быть кем-либо извлечены из файла подкачки. Таким образом, возможна утечка информации. И чтобы этого не случилось, системе может потребоваться очищать свой файл подкачки. Это можно сделать, включив данный пункт. Однако учтите, время очистки файла займет некоторое дополнительное время, и система будет выключаться чуть дольше.

1.3.Политика обновления

Любое программное обеспечение содержит ошибки (баги), т.к. на этапе проектирования приложений и систем невозможно все предусмотреть. Поэтому в любом приложении появляются места кода, которые работают не так, как рассчитывали разработчики, что может привести к нештатной работе программного обеспечения, а также появлению новых ошибок или уязвимостей при его работе.

Для выявления ошибок все компании-разработчики стараются тестировать свое программное обеспечение, т.е. проверять работу программного обеспечения в шоковых для него условиях, когда его ограничивают в размере доступной памяти, дискового пространства, скорости работы центрального процессора и пр. На этом этапе вылавливаются ошибки и вносятся исправления в код программного обеспечения, улучшающие его стабильность (робастность) или отказоустойчивость. Однако эти меры лишь частично позволяют избавиться от наиболее явных ошибок, которые проявили себя в тестировании. В н.в. не существует аппаратных или математических методов, позволяющих избавиться от ошибок в программном обеспечении на этапе его разработки.

После долгих поисков компании-разработчики нашли простой метод, который позволяет практически со стопроцентной вероятностью избавиться от ошибок в конечных продуктах, находящихся у пользователей. Этим методом является *периодическое обновление программных продуктов*. Компании разработчики решили, что в идеале программное обеспечение должно работать двадцать четыре часа в сутки, семь дней в неделю и в его работе не должно быть никаких нештатных ситуаций, вызванных ошибками. Это можно достигнуть с помощью грамотной *политики обновления*, которая используется практически в любом современном программном продукте, т.е. система периодически выходит в Интернет и проверяет сайт компании-разработчика на появление обновлений к программному обеспечению.

Компания-разработчик для программного продукта периодически помещает на своем сайте исправления, обновления и дополнения. Исправления – это специальные заплатки для программного обеспечения, которые исправляют существующие в нем ошибки, замеченные пользователями или специалистами компании. Обновления включают различные обновления программного продукта и заплатки от обнаруженных в нем ошибок. Дополнения добавляют программному продукту определенную функциональность.

Обновления для пользователей *Windows XP* позволяют не только избежать ошибок ОС, проявляющихся при ее использовании, но и практически гарантированно защитить ее от взломщиков и вирусов, т.к. исправляются все замеченные ошибки в системе безопасности. Алгоритм работы системы обновления *Windows XP* настроен на периодическую проверку сайта компании *Microsoft* на наличие различных обновлений и скачивание или предупреждение пользователя, в зависимости от его настроек. Все настройки политики безопасности Windows XP находятся в программе *Система*:

Пуск\Настройка\Панель управления\Система

Все операции настройки политики обновления ОС, а также выполнения процедуры обновления и получения от нее различных сообщений возможны только под учетной записью администратора системы. После запуска прогарммы появится окно, в котором нужно выбрать закладку *Автоматическое обновление* (*Automatic Updates*) (рис.1.1). На закладке можно установить один из четырех параметров, которые будут определять частоту обновления системы:

1.*Автоматически (рекомендуется)*, *Automatic (recommended)* – параметр устанавливается операционной системой *Windows XP* по умолчанию и означает регулярное обновление системы, заданное в двух нижерасположенных параметрах: частоты обновления и времени обновления. По умолчанию *Windows XP* будет обновляться каждый день в три часа (рис.1.1.). Скачивание обновлений операционной системы может происходить параллельно с работой в Интернете, т.к. операционной системой резервируется двадцать процентов пропускной способности канала связи с Интернетом, что позволяет быстро и незаметно скачивать системные обновления с сайта *Microsoft*.

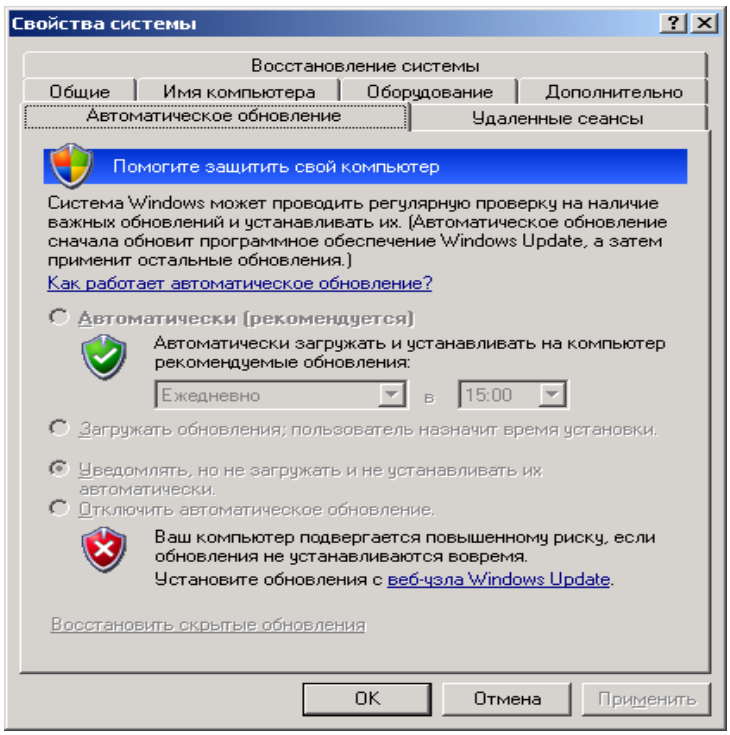


Рисунок 1.1. Закладка *Автоматическое обновление*

программы *Система*

Если пользователь работает на домашнем компьютере, достаточно производить обновления системы раз в неделю. Если система является корпоративной или часто находится в Интернете, рекомендуется проводить каждодневное обновление, чтобы надежно защититься от сетевых взломщиков. Если система скачает обновления, и они будут готовы к инсталляции, система сообщит об этом. Если же за время выхода в Интернет система не успеет скачать все обновления, они будут скачаны в следующий раз. Все скачанные и установленные обновления можно удалить, воспользовавшись для этого программой *Установка и удаление программ*: *Пуск\Панель управлениях Установка и удаление программ*.

2.*Загружать обновления, пользователь назначит время установки*, *Download updates for me, but let me choose when to install them* опция позволяет операционной системе самостоятельно скачивать обновления, но для их установки она должна спросить разрешения у пользователя.

3.*Уведомлять, но не загружать и не устанавливать их автоматически, Notify me but don't automatically download or install them* опция
позволяет операционной системе проверять наличие обновлений,
но запрещает их непосредственное скачивание или установку. Эта опция полезна, если пользователь сам устанавливаете обновления, например с компакт-диска, также она позволяет сэкономить на интернет-трафике.

4.*Отключит автоматическое обновление*, *Turn off Automatic Updates* опция запрещает работу системы обновления *Windows XP*.

2. Практическая часть

2.1. Вопросы по разделу

1. Определите назначение *политики безопасности* системы.
2. Где производится настройка *политики безопасности* системы?
3. Как запретить доступ сетевых пользователей к компьютеру?
4. Как разрешить доступ сетевым пользователям, которым разрешено работать в системе к компьютеру?
5. Определите назначения пункта политики безопасности *Разрешать вход в систему через службу терминалов*.
6. Как предоставить определенной группе пользователей вносить изменения в системное время?
7. Определите назначение пункта политики безопасности *Отладка программ*.
8. Каким образом запретить вход определенной группе пользователей в систему по локальной сети?
9. Определите назначение пункта политики безопасности *Принудительное удаленное завершение*.
10. Как установить пользователей и их группы, которые могут локально входить в систему?
11. Как запретить определенной группе пользователей завершать работу системы, и в каких случаях это актуально?
12. В каком разделе производится настройка глобальных параметров безопасности?
13. Определите назначение *политики обновления*.
14. Как произвести настройку *политики обновления*?

2.2. Упражнение 1

1.Произвести настройку *Политики безопасности* на своем ПК.

2.Произвести настройку *Параметров безопасности* на своем ПК.

3.Произвести настройку *Политики обновления* на своем ПК.

Лабораторная работа №17.

Тема: Администрирование системы через cmd в ос Windows XP

Введение

Консоль управления Microsoft Management Console (MMC) *группирует средства администрирования, которые используются для администрирования сетей, компьютеров, служб и других системных компонентов.*
Консоль MMC непосредственно не выполняет административные функции, однако предоставляет возможности интеграции в нее компонентов или системных приложений, выполняющие эти функции. Основной тип интегрируемых на консоль компонентов называется оснасткой, которые не могут выполняться отдельно без консоли. Среди других добавляемых элементов могут быть элементы управления ActiveX, ссылки на Web-страницы, папки, видов панели задач и собственно задачи для выполнения.
Дополнительные теоретические сведения об оснастках и других используемых для интеграции на консоль элементах будут добавлены в дальнейшем, в соответствующих разделах настоящей лабораторной работы.
Базовое окно консоли MMC представляет собой графическую форму с контекстными меню, реализующие дружественный пользовательский интерфейс. Имеется панель инструментов с командами создания, открытия и сохранения консолей и, кроме того, область описания и строка состояния в нижней части окна. Чтобы увидеть базовое окно, а также непосредственно саму консоль MMC, необходимо выполнить следующие действия:

- нажмите Пуск | Выполнить,
- наберите в появившемся окне MMC.exe (или просто mmc),
- нажмите Enter для ввода.

Новая консоль MMC представляет собой отдельное окно, разделенное на две вертикальные области, в левой из которых отображается дерево консоли с его корнем. Дерево консоли показывает доступные элементы и компоненты консоли. Правая область является областью сведений, которая содержит описания элементов и выполняемых ими функций. Содержание области сведений соответствует выбранному элементу в дереве консоли и может включать Web-страницы, графики, диаграммы, таблицы и столбцы.
Создавая надежные средства управления компьютерами сети, можно собрать и настроить собственную консоль MMC, выполняющую заданные функции администрирования.
После того как добавлены все необходимые элементы и компоненты консоли, панель главного меню, панель инструментов, а также область описания и строка состояния могут быть скрыты для предотвращения в дальнейшем нежелательных изменений. Созданные таким образом управляющие системы сохраняются в файлах с расширением .msc (Management Saved Console, сохраненная консоль управления) и могут быть, в частности, распространены в пределах всей системы посредством задания к ним доступа с помощью ярлыков или элементов меню Пуск.
Чтобы увидеть консоль управления локальным компьютером в качестве примера готовой и отлаженной консоли MMC, необходимо выполнить:

- нажмите Пуск | Выполнить,
- наберите в появившемся окне compmgmt.msc (или compmgmt),
- нажмите Enter для ввода.

Существует два основных режима доступа консоли администрирования, задающиеся непосредственно при ее создании:

- пользовательский, в котором можно администрировать систему, работая с уже существующими консолями,
- авторский, в котором можно создавать новые консоли или изменять существующие.

В свою очередь, имеется три уровня режима пользователя, что обусловливает всего четыре варианта предустановленного режима доступа:

- авторский режим;
- режим пользователя — полный доступ;
- режим пользователя — ограниченный доступ, многооконный;
- режим пользователя — ограниченный доступ, однооконный.

Консоль MMC, инициализированная в авторском режиме, предоставляет полный доступ ко всем ее возможностям, включая добавление и удаление оснасток, создание новых окон и панелей задач, а также просмотр любых частей дерева консоли и другие. Однако при выборе одного из трех режимов пользователя авторские возможности исключаются. В частности, если для консоли установлен параметр «пользовательский режим — полный доступ», то предоставляются все команды управления окном консоли и полный доступ к ее дереву, но запрещается добавление, удаление оснасток и изменение свойств консоли администрирования.
Изменения консоли MMC в авторском и пользовательском режимах сохраняются по-разному. При закрытии консоли в авторском режиме выводится диалоговое окно с предложением сохранить изменения. Однако в пользовательском режиме и снятом флажке «Не сохранять изменения для этой консоли» изменения будут сохранены автоматически при закрытии.
Если консоль открыта при соблюдении одного из следующих условий:

- в базовом окне при загрузке,
- с помощью команды контекстного меню Автор,
- в командной строке с параметром /a,

то предустановленный режим игнорируется, а открытие консоли осуществляется в авторском режиме.
Очевидно, что загрузка консоли MMC в авторском режиме не требуется рядовым пользователям. Системный администратор может настроить профили пользователей так, чтобы запретить им переход в авторский режим, как из командной строки, так и через контекстное меню. Кроме того, запрет перехода в авторский режим может быть организован при использовании возможностей групповой политики, при которой, в частности, осуществляется ограничение доступа к определенным оснасткам. Рассмотрению базовых возможностей оснастки групповой политики будет посвящена вторая часть настоящей лабораторной работы.
Прежде чем создавать новую консоль MMC, необходимо определить действия, для которых предназначена эта консоль, список администрируемых компонентов, оснасток и других элементов, которые потребуются для выполнения поставленных задач. Следует также рассмотреть необходимость создания видов панели задач. После принятия этих решений можно открыть новую консоль и начать добавлять элементы к дереву консоли. Полное руководство по созданию и настройке консолей MMC находится на Web-узле корпорации Майкрософт (http://www.microsoft.com).
В лабораторной работе предполагается ознакомление с основными принципами организации и построения консоли администрирования MMC, а также с базовыми возможностями основных инструментов системного администратора — оснасток

«Локальные пользователи и группы» и «Редактор объекта групповой политики» («Групповая политика»).
Перед началом выполнения заданий в среде ОС Windows XP необходимо выполнить следующее:

1. запустить виртуальную машину с ОС Windows XP и активировать справочное меню (Пуск | Справка и поддержка);
2. ознакомиться с описанием и возможностями запуска и применения консоли администрирования MMC;
3. ознакомиться возможностью получения сведений пункта 2 из альтернативного источника информации, доступного непосредственно в справке консоли администрирования MMC (Справка | Вызов справки);
4. ознакомиться с описанием и возможностями оснасток «Локальные пользователи и группы» и «Редактор объекта групповой политики» («Групповая политика»).

Порядок выполнения:
I. Создание консоли администрирования MMC в авторском режиме требует выполнения следующих действий:

- нажмите Пуск | Выполнить,
- наберите в появившемся окне MMC.exe (или просто mmc),
- нажмите Enter для ввода.

Содержание задания

Для придания уникального вида сохраненной (новой) консоли администрирования MMC в авторском режиме выполните следующие действия:

1. В меню Консоль выберите команду Параметры.
2. На вкладке Консоль в поле названия введите новый заголовок.

1. На вкладке Консоль из списка Режим консоли выберите пользовательский режим с полным доступом, в котором будет открываться консоль MMC при ее непосредственном запуске,
2. Для установленного в предыдущем пункте режима выполните указанные ниже действия:

- запретите изменение консоли MMC при ее непосредственном запуске, установив флажок «Не сохранять изменения для этой консоли»,
- сделайте активным диалоговое окно Вид | Настройка вида консоли MMC при запуске, установив флажок «Позволить пользователю настраивать вид консоли»,

1. Если необходимо удалить файлы, содержащие параметры отображения файлов консоли, на вкладке Очистка диска нажмите кнопку Удалить файлы.
2. Сохраните окончательно сконфигурированную консоль администрирования MMC, выбрав самостоятельно ее имя и путь к месту расположения в меню Консоль Закройте сконфигурированную и сохраненную консоль администрирования MMC, выполнив соответствующие необходимые действия.

В файловом менеджере Проводник ОС Windows XP выполните следующие инструкции:

- наведите манипулятором мышь на сохраненный файл консоли администрирования MMC и, дважды кликнув на нем, запустите консоль,
- откройте диалоговое окно Вид | Настройка вида и, изменяя положение флажков, обратите внимание на получаемый результат,
- изменив вид консоли MMC приемлемым образом, кликните OK для подтверждения полученного результата,
- в контекстном меню Консоль кликните Выход,
- снова запустите консоль администрирования MMC, кликнув манипулятором мышь на сохраненном файле консоли,
- изучите полученный результат,
- сделайте вывод о проделанной работе и апишите его в отчет.

- изолированные,
- расширения.

Изолированная оснастка (или просто оснастка) добавляется к дереву консоли MMC без предварительного добавления других элементов, то есть непосредственно в корень дерева консоли.
Оснастка расширения (или просто расширение) всегда добавляется к другой изолированной оснастке или расширению, которые уже имеются в дереве консоли MMC. Если для определенной оснастки разрешены расширения, то, как правило, они работают с объектами, управляемыми непосредственно этой оснасткой, например, с компьютером, принтером, модемом или другим внешним устройством.
В дереве консоли оснастки и расширения располагаются для удобства иерархически или по группам. При добавлении новой оснастки или расширения, они появляются в виде нового элемента в дереве консоли MMC или в виде нового пункта контекстного меню, дополнительной панели инструментов, страницы свойств, а также возможно мастера, организующего определенную последовательность действий, к уже установленной оснастке.
Другими элементами, по необходимости применимыми для интеграции на консоль администрирования MMC, являются виды панели задач и собственно задачи, которые могут включать в себя команды меню для элементов консоли и команды, запускаемые из командной строки. Кроме того, могут быть созданы команды, действующие как часть дерева консоли или открывающие другой компонент.
Прежде всего, перед добавлением указанных элементов к консоли MMC, необходимо определить их число. Если, в частности, требуется добавить несколько видов панели задач, то наряду с этим необходимо определить тип каждой панели (для отображения списка и задач или только задач), а также разделить задачи по интегрированным видам панели. Добавление видов панели задач и собственно задач осуществляется посредством работы мастера создания этих элементов. При этом важно помнить, что консоль MMC должна содержать, по крайней мере, одну оснастку, чтобы возможность интеграции появилась в принципе.
Отдельной возможностью, иногда необходимой при администрировании сетей, является добавление элементов и компонентов дерева консоли администрирования MMC в виде списка ярлыков в меню «Избранное».
Дополнительные сведения о добавлении различных элементов в дерево консоли администрирования MMC можно получить, воспользовавшись справкой ОС Windows XP (Пуск | Справка и поддержка) в разделе Общее представление о MMC \ Консоль MMC в авторском режим

Лабораторная работа №18.

1. **Тема: Мониторинг и оптимизация системы в ос Windows xp**

Введение

- **Мониторинг** — *система сбора или регистрации, хранения и анализа небольшого количества ключевых (явных или косвенных) признаков или параметров описания данного объекта для вынесения суждения о поведении или состоянии данного объекта в целом.*
- **Мониторинг** — *процесс наблюдения и регистрации данных о каком-либо объекте на неразрывно примыкающих друг к другу интервалах времени, в течение которых значения данных существенно не изменяются.*

Различают **Мониторинг параметров** и **Мониторинг состояния**объекта.
Мониторинг параметров — *наблюдение за какими либо параметрами. Результат мониторинга параметров представляет собой совокупность измеренных значений параметров, получаемых на неразрывно примыкающих друг к другу интервалах времени, в течение которых значения параметров существенно не изменяются.*
Мониторинг состояния — *наблюдение за состоянием объекта для определения и предсказания момента перехода в предельное состояние.*
Результат мониторинга состояния объекта представляет собой совокупность диагнозов составляющих его субъектов, получаемых на неразрывно примыкающих друг к другу интервалах времени, в течение которых состояние объекта существенно не изменяется.
Принципиальным отличием мониторинга состояния от мониторинга параметров является наличие интерпретатора измеренных параметров в терминах состояния — экспертной системы поддержки принятия решений о состоянии объекта и дальнейшем управлении.
Мониторинг в смысле I реализует программный модуль **Msinfo32**(в папке C:\Program Files\Common Files\Microsoft Shared\MSInfo).
После запуска программы открывается окно. В левой области окна «Сведения о системе» приведен список категорий, а в правой — подробные сведения о каждой из них. К этим категориям относятся:

- **Сведения о системе.** Содержит общие сведения о компьютере и операционной системе, такие как имя компьютера и его изготовитель, тип используемой BIOS, а также объем установленной памяти.
- **Ресурсы аппаратуры.** Содержит дополнительные сведения об оборудовании компьютера, предназначенные для ИТ-специалистов.
- **Компоненты**. Содержит перечень установленных дисководов, звуковых устройств, модемов и других компонентов.
- **Программная среда.** Содержит сведения о драйверах, сетевых подключениях и другую информацию, связанную с программами.

Пункты Ресурсы аппаратуры, Компоненты, Программная среда раскрываются и показывают конкретные значения параметров.
Чтобы найти определенные сведения о системе, введите ключевые слова в поле Что найти внизу этого окна. Например, для поиска IP-адреса компьютера в поле **найти** введите IP-адрес и нажмите кнопку Найти.
Примечание.
Чтобы сохранить сведения о системе, последовательно выберите пункты **Файл** и **Сохранить**, введите имя файла, после чего нажмите кнопку **Сохранить**. ОС Windows сохраняет эту информацию в файловом формате **NFO**. При наличии проблем с открытием

файла можно сохранить сведения о системе в формате **TXT**. Для этого последовательно выберите **Файл**, **Экспорт**, введите имя файла и нажмите кнопку **Сохранить**.
Мониторинг в смысле II реализует программный модуль **Taskmgr.exe**(в папке C:\Windows\System32) –"Диспетчер задач". Диспетчер задач отображает приложения, процессы и службы, которые в текущий момент запущены на компьютере. С его помощью можно контролировать производительность компьютера или завершать работу приложений, которые не отвечают.
После запуска программы открывается окно "Диспетчер задач Windows", содержащее вкладки: Приложения, Процессы, Быстродействие, Сеть, Пользователи.
Мониторинг состояния реализует программный модуль **Dxdiag.exe** (в папке C:\Windows\System32) –"Средство диагностики DirectX"
Отображает сведения о компонентах и драйверах интерфейса прикладного программирования приложений (API) Microsoft DirectX в системе. Позволяет проверить работу звуковой и графической подсистем.
Материал из Википедии — свободной энциклопедии.
DirectX (от англ. direct — прямой, непосредственный) — это набор **API** (интерфейс программирования приложений), разработанных для решения задач, связанных с программированием под Microsoft Windows. При выполнении **Dxdiag.exe**открывается окно со следующими вкладками: Система, Файлы DirectX, Дисплей, Звук, Музыка, Ввод, Сеть, Если ничего не помогло. Все вкладки, кроме Система и Файлы DirectX, содержат средства диагностики., оптимизировать работу ОС, а также автоматизировать устранение неполадок при настройке ОС.

Запустите на выполнение модули Msinfo32, Taskmgr.exe, Dxdiag.exe. Сверните появившиеся окна «Сведения о системе», «Средство диагностики DirectX» и «Диспетчер задач» на панель задач.

1. Разверните окно модуля «Сведения о системе» и последовательно просмотрите все категории сведений. При этом обратите внимание на то, что глобально все категории делятся на четыре класса «Ресурсы аппаратуры», «Компоненты», «Программная среда» и «Параметры обозревателя». Наиболее полезными с точки зрения сетевого администрирования являются категории «Конфликты/Совместное использование» и «Прерывания» в классе «Ресурсы аппаратуры», категория «Сеть» в классе «Компоненты», а также категории «Переменные среды», «Сетевые подключения» и «Службы» в классе «Программная среда». Необходимо отметить, что указанные классы ресурсов являются ценным источником системной информации, поскольку позволяют отслеживать аппаратные и программные изменения как локально, так и удаленно. Последнее может быть осуществлено посредством выбора «Удаленный компьютер…» в меню «Вид». Кроме того, отдельный интерес может представлять информация, собранная в классе «Параметры обозревателя».
2. Разверните окно следующего системного модуля «Диагностика DirectX», предназначенного для диагностирования аппаратных и программных компонентов компьютера, применяющихся для поддержки средств мультимедиа в играх и фильмах, и последовательно изучите все его вкладки. На вкладках «Дисплей», «Звук» и «Музыка» осуществите проверку соответствующих программных составляющих DirectX, а именно, интерфейсов DirectDraw, DirectSound и DirectMusic. Сохраните все сведения в текстовый файл для отчета. Обратите внимание на то, что системный модуль «Диагностика DirectX» также может быть вызван из меню «Сервис» программного модуля «Сведения о системе».
3. Универсальный системный модуль «Диспетчер задач» как правило является наиболее часто используемым компонентом ОС, предназначенным для

диагностики и мониторинга основных аппаратно-программных ресурсов системы, таких как центрального процессора, оперативной памяти, системных процессов. В частности, этот модуль позволяет управлять приложениями и процессами в оперативной памяти, снимать их с выполнения и назначать новое значение класса приоритета. Разверните окно системного модуля «Диспетчер задач» и последовательно ознакомьтесь со всеми его вкладками и меню. Выполните следующие действия:

- на вкладках «Приложения» и «Процессы» обратите внимание на количество работающих приложений и активных процессов,
- рядом с системным модулем «Диспетчер задач» разверните модуль «Сведения о системе» и откройте категорию «Выполняемые задачи» в классе «Программная среда»,
- в меню «Вид» в модуле «Диспетчер задач» добавьте следующие столбцы счетчиков: «память – максимум», «объем виртуальной памяти», «базовый приоритет», «счетчик потоков»,
- в модуле «Диспетчер задач» измените базовый приоритет процесса Dxdiag.exe на приоритет реального времени, перейдите в окно модуля «Сведения о системе», в меню «Вид» обновите системную информацию и обратите внимание на то, как изменилось значение в столбце «Приоритет» в категории «Выполняемые задачи»,
- на вкладке «Приложения» снимите с выполнения задачи «Сведения о системе» и «Средства диагностики DirectX», а на вкладке «Процессы» завершите процесс Taskmgr.exe.

- перенесите последовательность выполняемых действий по каждому из пунктов 1-3 в отчет (возможно приведение графических фрагментов, сделанных с экрана, в качестве демонстрационного материала),
- результаты ознакомления с возможностями системного модуля «Диспетчер задач» занесите в табл. 1 и 2.
- сделайте вывод о проделанной работе и запишите его в отчет.

Таблица 1. Результаты ознакомления с «Диспетчером задач»

№ п.п. Общая системная информация Количество

Работающих приложений:

Активных процессов:

Активных потоков:

Дескрипторов:

Таблица 12. Результаты ознакомления с «Диспетчером задач»

№ п.п.

Имя образа, .exe Пик. память, Кб Вирт. память, Кб Базовый приоритет Потоков

Taskmgr

HelpCtr

Dxdiag

Лабораторная работа №19.

Тема: Работа с Реестром ОС Windows XP

Введение

Реестр Windows или ***системный реестр*** *— иерархически построенная база данных параметров и настроек в большинстве операционных систем Microsoft Windows*.
(Материал из Википедии — свободной энциклопедии
http://ru.wikipedia.org/wiki/Реестр_Windows)

Основные свойства:

1. Реестр содержит информацию и настройки
   - для аппаратного обеспечения,
   - программного обеспечения,
   - профилей пользователей,
   - предустановки.
2. Большинство изменений в Панели управления, ассоциации файлов, системные политики, список установленного ПО фиксируются в реестре.
3. Все возможности ОС могут быть конфигурированы посредством Реестра.
4. Любое запускаемое в системе приложение может быть выполнено только через обращение к Реестру, поскольку именно там находятся все его параметры.
5. База данных Реестра хранится в системных файлах ОС, в частности, **system.dat** и **ntuser.dat**.

Историческая справка:

1. Реестр Windows был введён для упорядочения информации, хранившейся до этого во множестве INI-файлов.
2. Реестр, как древовидная иерархическая база данных впервые появился в Windows 3.1 (апрель 1992). Это был всего один двоичный файл, который назывался REG.DAT и хранился в каталоге C:\Windows\. Реестр Windows 3.1 имел только одну ветку HKEY_CLASSES_ROOT. Он служил для связи DDE объектов (Dynamic Data Exchange — механизм взаимодействия приложений в операционных системах Microsoft Windows и OS/2), а позднее и OLE объектов (Object Linking and Embedding — технология связывания и внедрения объектов в другие документы и объекты) .
3. Одновременно с появлением реестра в Windows 3.1 появилась программа REGEDIT.EXE для просмотра и редактирования реестра.
4. Первый реестр уже имел возможность импорта данных из *.REG файлов.
5. В базовой поставке шёл файл SETUP.REG, содержащий данные по основным расширениям и типам файлов.
6. Реестр Windows 3.1 имел ограничение на максимальный размер файла REG.DAT — 64 Кбайт. Если вдруг реестр превышал этот размер — то файл реестра (REG.DAT) приходилось удалять и собирать заново либо из *.REG файлов, либо вводить данные вручную.
7. Следующий шаг сделан в Windows NT 3.1 (июль 1993). Произошёл отказ от устаревших файлов MS-DOS: AUTOEXEC.BAT и CONFIG.SYS, а также от INI-файлов, как от основных файлов конфигурации. На «регистрационную базу» (реестр) была переведена вся конфигурация системы. Основой конфигурации системы стал реестр. Он имел 4 корневых раздела:

- HKEY_ LOCAL_MACHINE,
- HKEY_CURRENT_USER,
- HKEY_CLASSES_ROOT,
- HKEY_USERS.

8. Реестр стал «сборным»: на диске он хранился в файлах: DEFAULT, SOFTWARE, SYSTEM, а при запуске системы из этих файлов собиралась единая БД. В комплекте поставки оставался файл REGEDIT.EXE, который по-прежнему позволял просматривать и редактировать только ветку HKEY_CLASSES_ROOT, и появился файл REGEDT32.EXE, который позволял редактировать все ветки реестра.
9. Далее технология и идеология (назначение) реестра уже не менялись. Все последующие версии Windows (NT 3.5, 95, NT 4.0, 98, 2000, XP, Vista, 7) использовали реестр как основную БД, содержащую все основные данные по конфигурации как самой ОС, так и прикладных программ. Далее менялись названия файлов реестра и их расположение, а также название и назначение ключей.

Место хранения рестра Windows в XP и 7

1. После установки Windows XP на диске в каталоге C:\Windows\System32\Config\ хранятся файлы system, software, sam, security, default. Все имена файлов без расширений. Копия этих файлов хранится в каталоге C:\Windows\Repair\

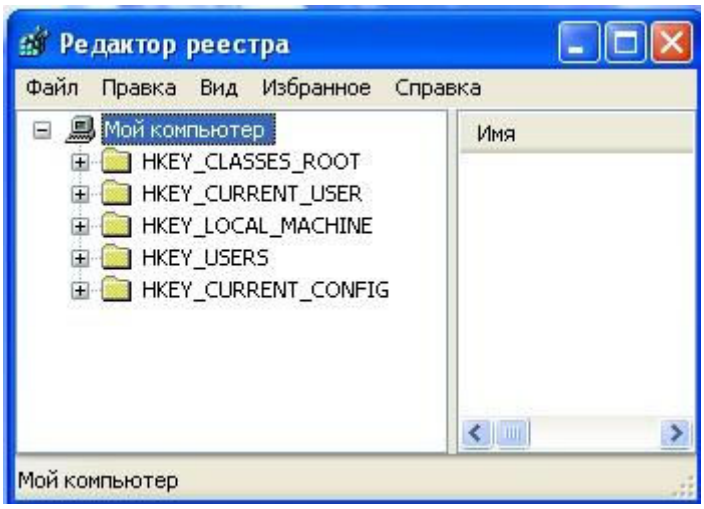


Рисунок 1 Реестр Windows XP

2. Файлы, используемые при построении «рабочей версии» реестра, могут храниться в каталогах:
   - %SystemDrive%\Documents and Settings\<Username>\ — файл «Ntuser.dat»
   - %SystemDrive%\Documents and Settings\<Username>\Local Settings\Application Data\Microsoft\Windows\ — файл «UsrClass.dat»
3. Кроме этого, могут появляться и другие файлы реестра, например, userdiff , userdiff.LOG, TempKey.LOG.
4. Можно провести некое примерное соответствие файлов и веток реестра, но оно не такое простое, полное и однозначное. Однако примерно можно сказать следующее:
   - Ветка реестра «HKEY_LOCAL_MACHINE\Software» формируется из файла «%SystemRoot%\system32\config\software».
   - Ветка реестра «HKEY_LOCAL_MACHINE\System\» формируется из файла «%SystemRoot%\system32\config\system».

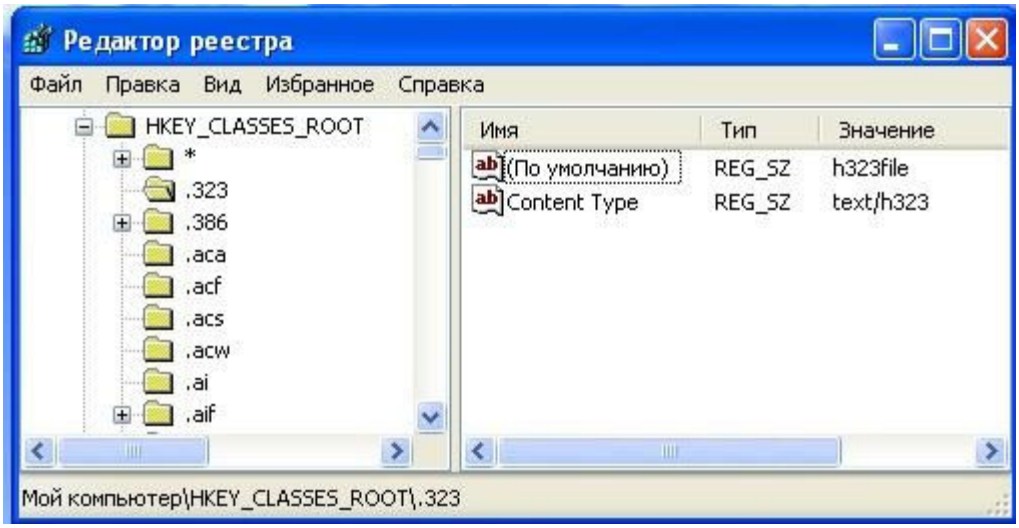
- Ветка реестра «HKEY_LOCAL_MACHINE\SAM\» формируется из файла «%SystemRoot%\system32\config\SAM».
- Ветка реестра «HKEY_LOCAL_MACHINE\SECURITY\» формируется из файла «%SystemRoot%\system32\config\SECURITY».
- Ветка реестра «HKEY_LOCAL_MACHINE\HARDWARE\» формируется в зависимости от оборудования(динамически).
- Ветка реестра «HKEY_USERS\<SID_пользователя>» формируется из файлов «%USERPROFILE%\ntuser.dat»
- Ветка реестра «HKEY_USERS\DEFAULT» формируется из файлов «%SystemRoot%\system32\config\default»

5. В Windows 7, согласно сведениям из HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\hivelist файлы реестра хранятся в следующих местах:
   - 01= Ветка реестра «HKEY_LOCAL_MACHINE\HARDWARE» формируется в зависимости от оборудования (динамически);
   - 02= Ветка реестра «HKEY_LOCAL_MACHINE\BCD00000000» формируется из файла «%SystemRoot%\Boot\BCD»
   - 03= Ветка реестра «HKEY_LOCAL_MACHINE\SYSTEM» формируется из файла «%SystemRoot%\System32\config\SYSTEM»
   - 04= Ветка реестра «HKEY_LOCAL_MACHINE\SOFTWARE» формируется из файла «%SystemRoot%\System32\config\SOFTWARE»
   - 05= Ветка реестра «HKEY_LOCAL_MACHINE\SECURITY» формируется из файла «%SystemRoot%\System32\config\SECURITY»
   - 06= Ветка реестра «HKEY_LOCAL_MACHINE\SAM» формируется из файла «%SystemRoot%\System32\config\SAM»
   - 07= Ветка реестра «HKEY_USERS\.DEFAULT» формируется из файла «%SystemRoot%\System32\config\DEFAULT»
   - 08= Ветка реестра «HKEY_USERS\S-1-5-18» формируется из файла «%SystemRoot%\System32\config\systemprofile\NTUSER.DAT» (относится к учетной записи system)
   - 09= Ветка реестра «HKEY_USERS\S-1-5-19» формируется из файла «%SystemRoot%\ServiceProfiles\LocalService\NTUSER.DAT» (относится к учетной записи LocalService)
   - 10= Ветка реестра «HKEY_USERS\S-1-5-20» формируется из файла «%SystemRoot%\ServiceProfiles\NetworkService\NTUSER.DAT» (относится к учетной записи NetworkService)
   - 11= Ветка реестра «HKEY_USERS\<SID_пользователя>» формируется из файла «%USERPROFILE%\NTUSER.DAT»
   - 12= Ветка реестра «HKEY_USERS\<SID_пользователя>_Classes» формируется из файла «%USERPROFILE%\AppData\Local\Microsoft\Windows\UsrClass.dat»
6. Резервные копии файлов реестра DEFAULT, SAM, SECURITY, SOFTWARE и SYSTEM находятся в папке «%SystemRoot%\System32\config\RegBack». Само резервное копирование производится силами Планировщика задач в 0 ч. 00 мин. каждые 10 дней по заданию «RegIdleBackup», расположенному в иерархии задач по пути «\Microsoft\Windows\Registry».

Ключи

1. Основными элементами структуры Реестра ОС являются ключи. Каждый ключ может иметь набор параметров, каждому из которых соответствует определенное значение, а также подключи – подчиненные ключи более низкого уровня. По

отношению к друг другу ключи и подключи организуются в системном Реестре в соответствии с отношением вида «предок-потомок».



2. Иерархическая структура Реестра ОС представляет собой дерево ключей, организованное в виде кустов или ульев (каждый из которых является двоичным файлом, называемым файлом куста), напоминающей структуру файлов и папок файловой системы (ФС). Корневой ключ (вершина дерева) и подключи по аналогии с ФС можно считать папками, а параметры Реестра – файлами, соответственно.
3. В качестве кустов корневого ключа HKEY_LOCAL_MACHINE (HKLM) и соответствующих им файлов кустов можно привести следующий пример (табл. 6.1). Каждый из файлов кустов HKLM имеет свой системный путь. В частности, файлы кустов HKLM\SOFTWARE и HKLM\SYSTEM находятся в системном каталоге %SYSTEMROOT%\System32\config.

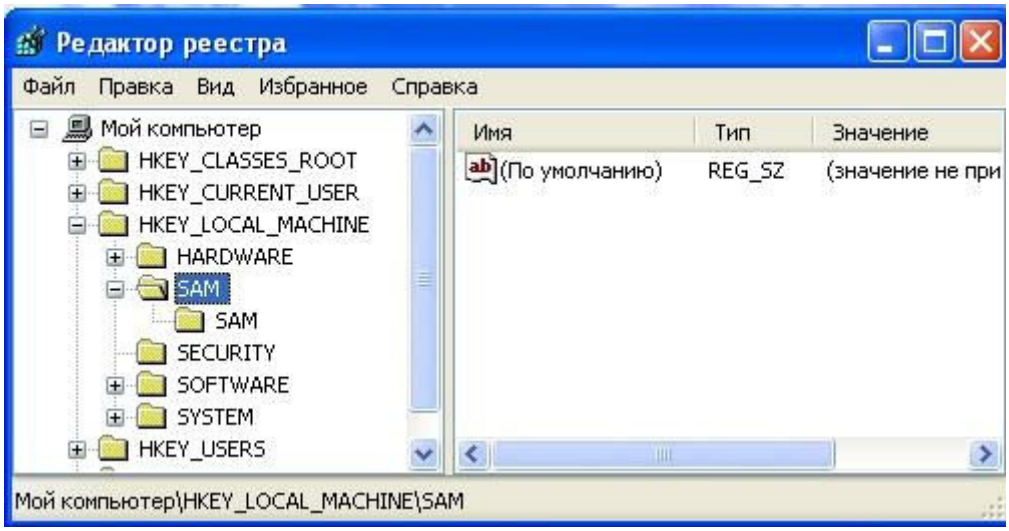


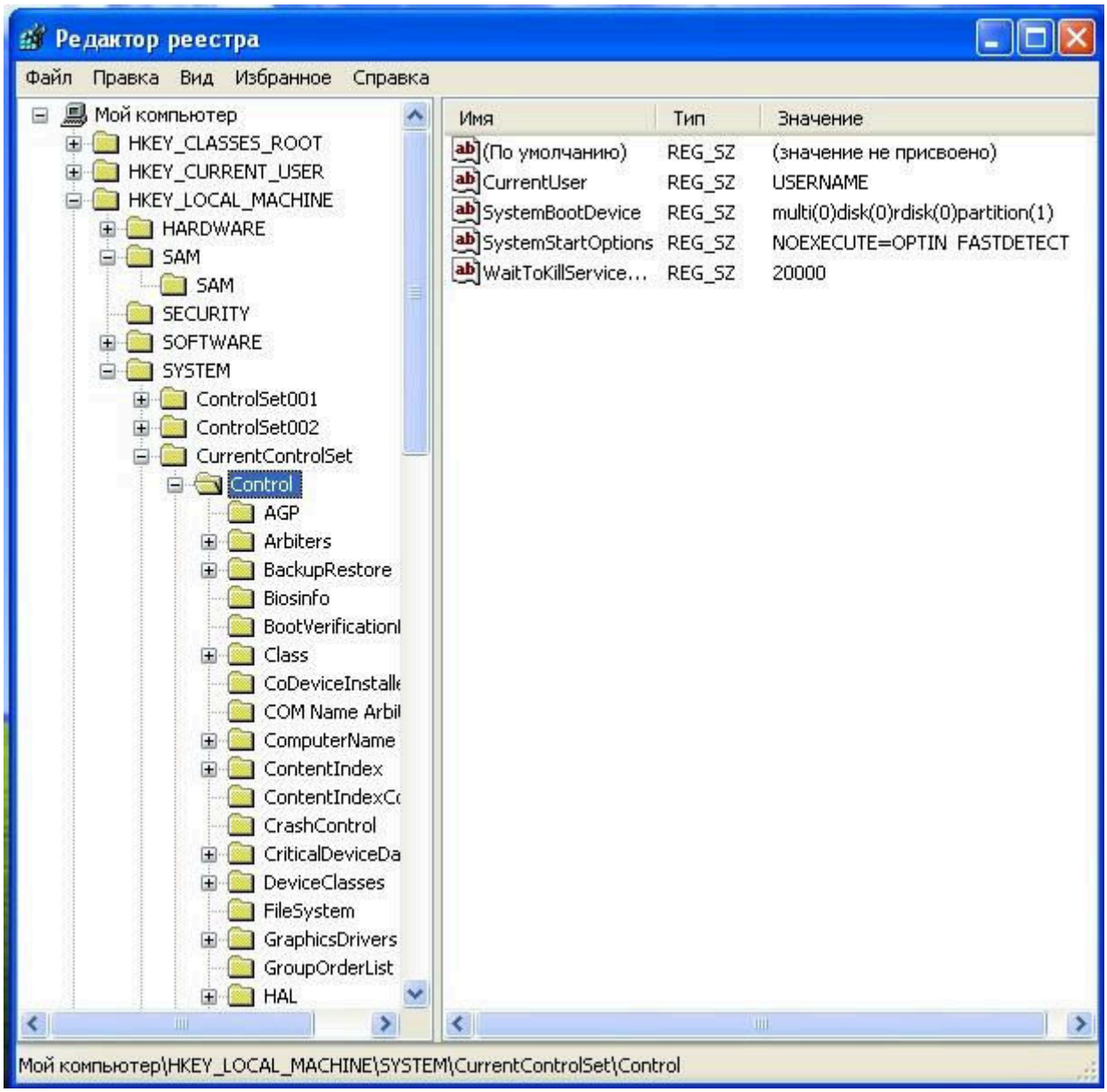
Таблица 6.1. Файлы кустов корневого ключа HKLM

№ п.п.	Куст	Файл куста
1.	HKLM\SAM	Sam.log

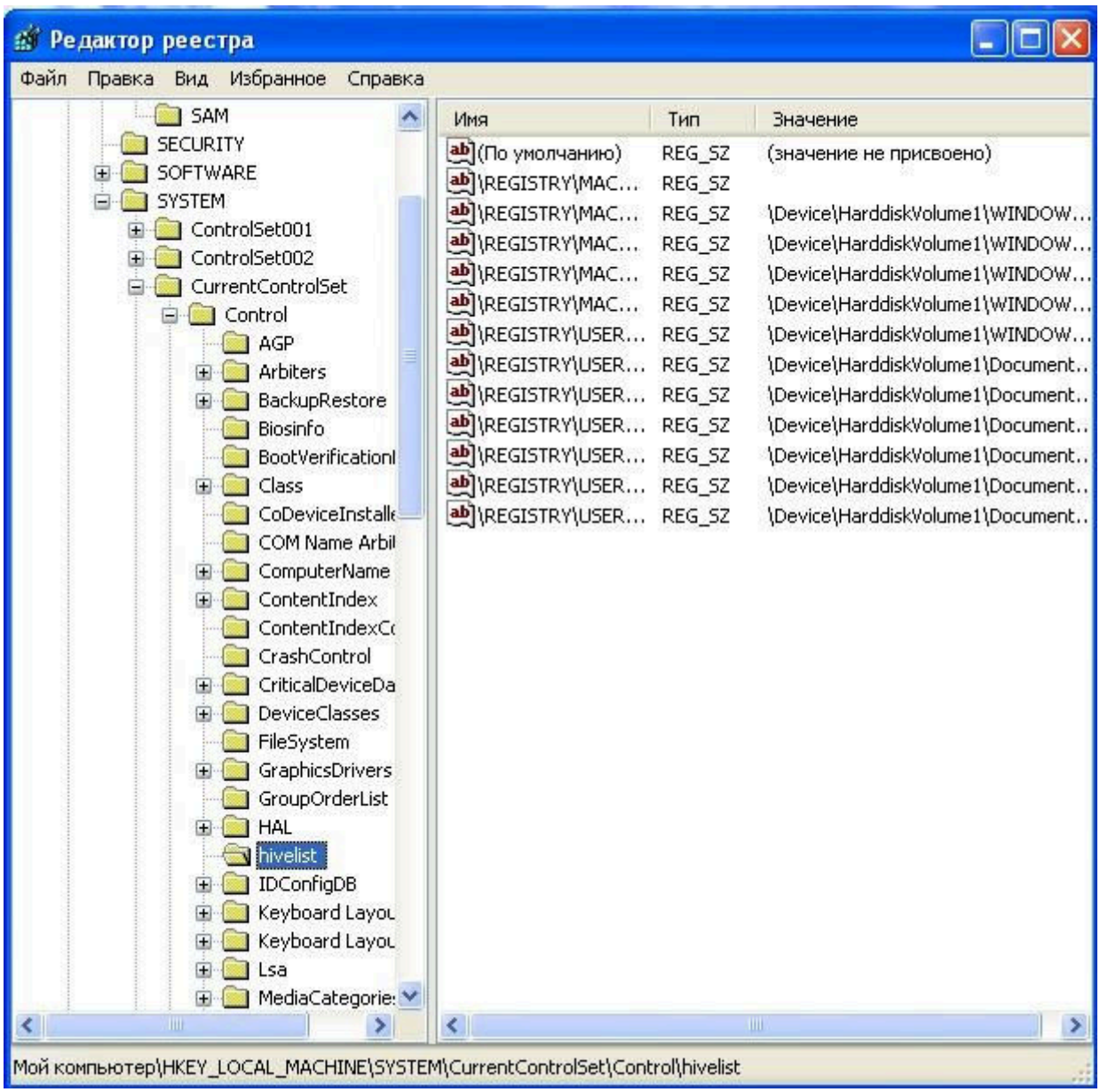
1.	HKLM\SECURITY	Security.log
1.	HKLM\SOFTWARE	Software.log, Software.sav
1.	HKLM\SYSTEM	System.log, System.sav

В таблице отображены не все кусты HKLM, а лишь те из них, которые являются постоянными Реестра ОС. В дополнение имеются два временных куста HKLM, образующиеся при старте системы.

4. Куст HKLM\SYSTEM корневого ключа HKLM является основным системным кустом, так как в него входит подключ \CurrentControlSet\Control, содержащий параметры, которые компонент ядра ОС, называемый «Менеджер конфигурации» (Configuration Manager), использует при инициализации Реестра.



Значение **hivelist** подключа \CurrentControlSet\Control используется системой при поиске остальных ее файлов куста.

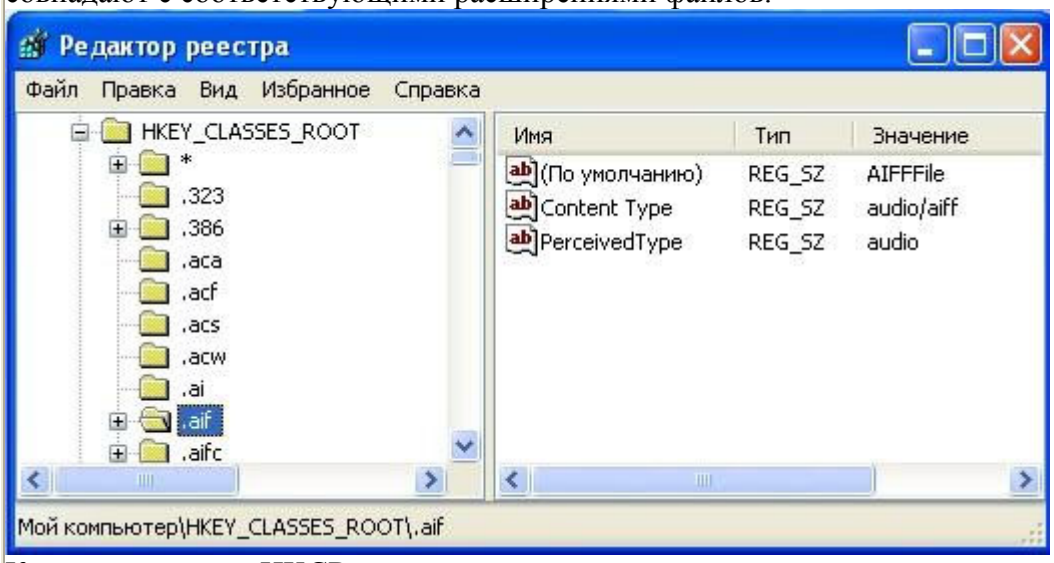
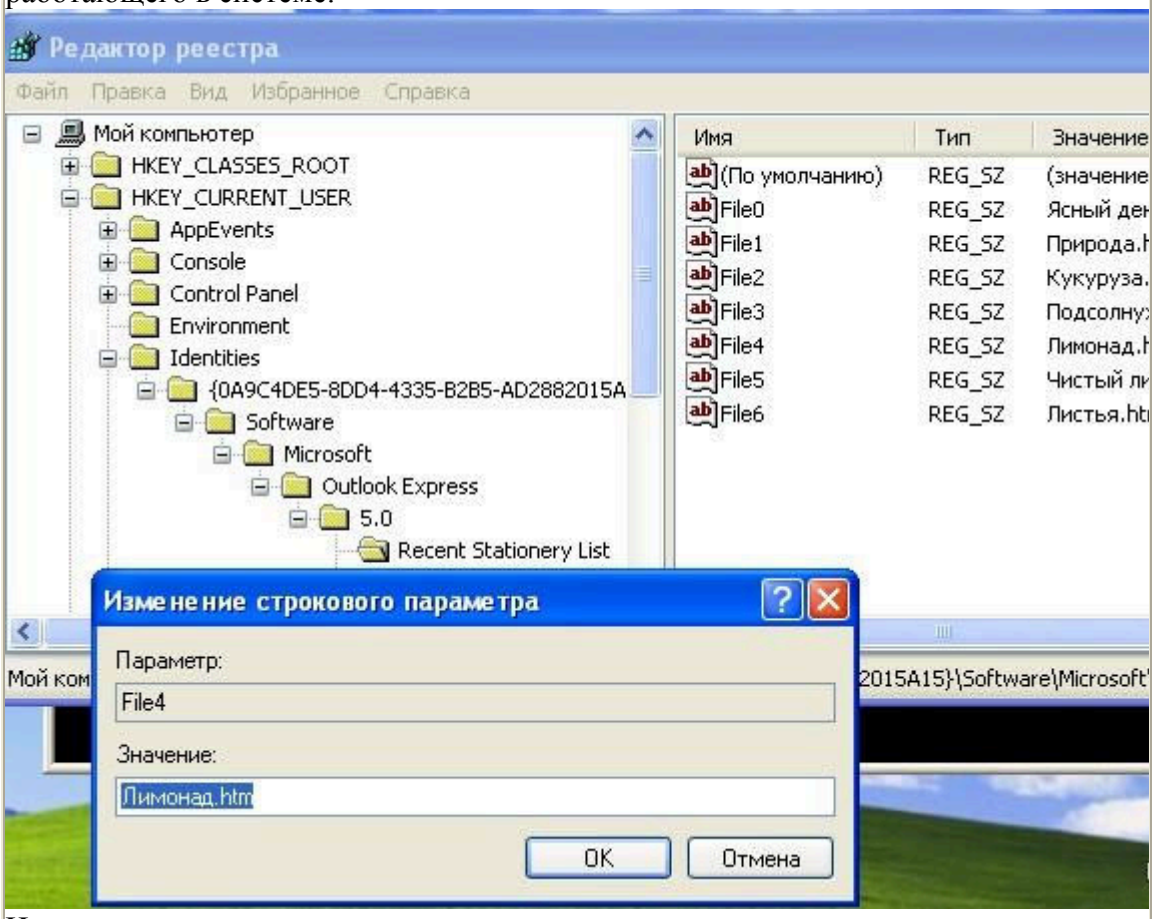


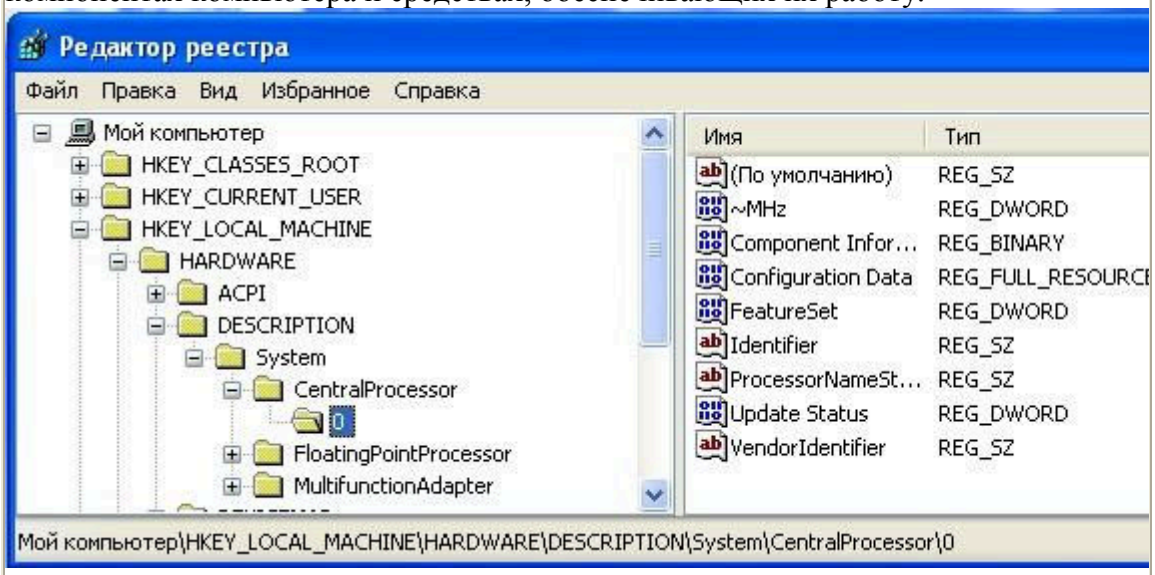
5. В рассмотренном примере одним из ключей системного Реестра был назван корневой ключ HKEY_LOCAL_MACHINE.

6. Реестр ОС имеет несколько корневых ключей высшего уровня, каждый из которых определяет некоторую категорию данных, хранимых в Реестре. Полный список корневых ключей, а также их краткое описание представлены ниже (табл. 6.2). Некоторые ключи и соответствующие им кусты являются временными. К их числу можно отнести корневые ключи HKU, HKDD и некоторые HKLM с соответствующими кустами HKLM\HARDWARE и HKLM\SYSTEM\Clone. ОС создает их каждый раз при загрузке и хранит в оперативной памяти до момента завершения сеанса работы.

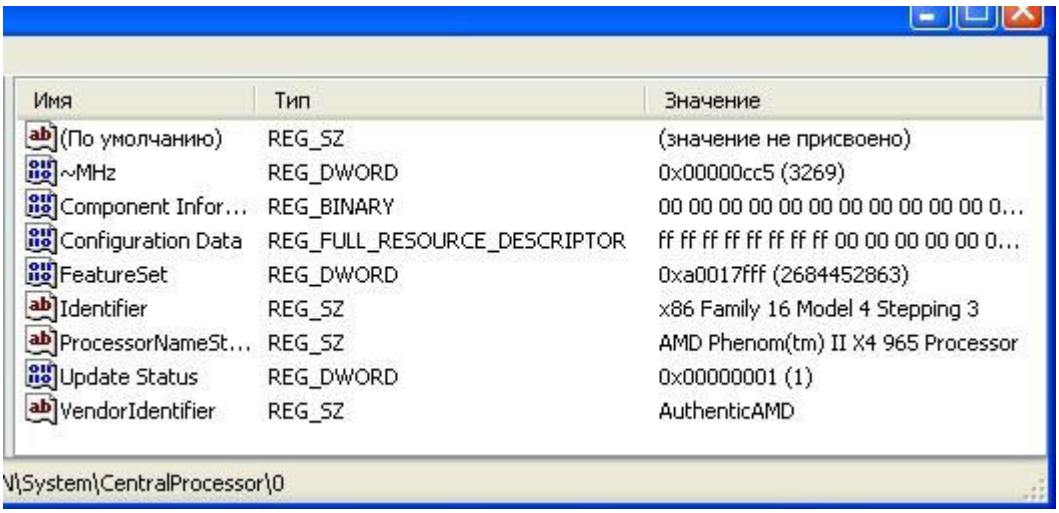
Таблица 6.2. Корневые ключи Реестра ОС Windows XP

№ п. п.	Ключ	Описание

1	HKCR	HKEY_CLASSES_ROOT. Подключи этого корневого ключа содержат основную информацию о типах файлов, зарегистрированных в системе. Названия подключей совпадают с соответствующими расширениями файлов.  Корневому ключу HKCR подчиняются описания различных программных средств обработки этих файлов, а также сведения обо всех категориях зарегистрированных объектов.
2	HKCU	HKEY_CURRENT_USER. Эта категория содержит описание параметров, меняющихся в зависимости от профиля пользователя, в данный момент работающего в системе.  Изменения, относящиеся к текущему пользователю, следует всегда вносить именно сюда, так как они автоматически копируются для длительного хранения при завершений работы компьютера и восстанавливаются в ходе начальной загрузки ОС.

3	HKLM	HKEY_LOCAL_MACHINE. Этот раздел отвечает за информацию об аппаратных компонентах компьютера и средствах, обеспечивающих их работу.  Здесь также хранится общая информация об установленном программном обеспечении.
4	HKU	HKEY_USERS. Этот раздел содержит подключи, соответствующие всем пользователям, зарегистрированным на данном компьютере. Когда один из пользователей начинает работу в системе, ОС автоматически копирует соответствующий ключ HKU в раздел HKCU. При завершении сеанса пользователя данные копируются обратно.
5	HKCC	HKEY_CURRENT_CONFIG. В этом разделе дублируется информация (текущий набор конфигурационных параметров аппаратуры) о некоторых устройствах компьютера, в первую очередь о видеоадаптере и принтере.
6	HKDD	HKEY_DYN_DATA. Этот раздел содержит текущую информацию о работе компьютера, обычно обновляемую в режиме реального времени. Основные подключи содержат данные об устройствах, работающих в настоящее время, а также сведения о текущем значении статистических параметров. Отобразить эти данные позволяет служебный модуль «Системный монитор».

6. Возвращаясь к вопросу о параметрах ключей и их значениях, следует сказать, что каждый ключ содержит как минимум одно значение какого-либо параметра. У параметра значения имеется имя, в то время как расширение файла похоже на его тип. Данные значения похожи на конкретное содержимое файла. Каждый ключ или подключ имеет одно или более значений, в свою очередь, каждое из которых характеризуется именем соответствующего параметра, типом и хранимыми в нем данными.
7. Имя параметра значения (или просто имя значения) представляет собой строку, содержащую до 512 символов в кодировке ANSI (или 256 символов в кодировке Unicode), за исключением символов, зарегистрированных для имен ОС Windows XP. Всего для значений предусмотрено пятнадцать различных типов, три из которых: REG_BINARY, REG_DWORD и REG_SZ являются основными и описывают



большинство всех значений в Реестре ОС.

8. Двоичные данные значений типа REG_BINARY, записанные в шестнадцатеричном виде, представляют собой строку байтов произвольной длины. Их обычно применяют в том случае, когда параметр должен хранить набор данных определенной структуры.

9. Значения типа REG_DWORD имеют длину данных в два машинных слова (четыре байта) и записываются в десятичной или шестнадцатеричной форме. Многие значения в Реестре принадлежат этому типу и используются в качестве логических флагов: 0 или 1, да или нет, истина или ложь; иногда значения этого типа встречаются в миллисекундах (1000 равно 1 секунде), описывающих время.
10. Ззначение типа REG_SZ представляет собой текст постоянной длины в виде строки символов, например, «Microsoft Windows XP». Каждая строка заканчивается символом null. Приложения не преобразуют значения этого типа, а транслируют и отображают их «как есть».

Подготовка к выполнению лабораторной работы

В предыдущих лабораторных работах были изучены различные способы настройки и оптимизации ОС Windows XP. Были рассмотрены основные служебные средства (системные программные модули и оснастки), посредством которых ОС может быть конфигурирована оптимальным образом. Однако, осуществленные действия являются лишь предварительными и обеспечивают достаточно поверхностную настройку ОС. Иногда подобных действий достаточно, чтобы получить приемлемый результат, но для более «тонкой» настройки ОС практически всегда требуется вмешательство в системный Реестр. В конечном счете такое вмешательство позволяет получить прирост в скорости реакции ОС на системные события и иногда значительно увеличить общую производительность системы в целом.

Оптимизация ОС посредством изменения значений системных параметров Реестра как правило подразумевает соответствующую квалификацию пользователя, чтобы гарантировать целостность Реестра и дальнейшую работоспособность системы. Некорректное или неумышленное изменение данных Реестра ОС может привести к непоправимым последствиям и обеспечить пользователя обременительной работой по повторной инсталляции системы. Чтобы этого избежать, необходимо перед проведением каких-либо манипуляций с Реестром ОС всегда осуществлять его предварительную архивацию и резервное сохранение в надежном месте на жестком диске. Выполнение

этого правила в последствии обеспечит «откат» системного Реестра и вернет систему в рабочее состояние до момента критических изменений.

На основе получаемых в настоящей лабораторной работе знаний и сведений по организации Реестра, его структуре, возможностям редактирования и изменения значений некоторых ключевых параметров ОС, предполагается выполнить ряд учебных заданий по конфигурированию ОС Windows XP и, тем самым, осуществить дополнительную «тонкую» настройку системы с целью ее оптимальной, быстрой и эффективной работы. Перед началом выполнения лабораторной работы в среде ОС Windows XP необходимо осуществить следующее:

1. запустить виртуальную машину с ОС Windows XP и активировать справочное меню (Пуск | Справка и поддержка);
2. ознакомиться с описанием Реестра и возможностями его применения в ОС Windows XP;
3. ознакомиться с описанием и возможностями служебного программного средства «Редактор Реестра» (Regedit), изучив справочный материал по данному приложению, находящийся в системном каталоге C:\Windows\Help\ в одноименном файле с расширением .chm;
4. воспользовавшись ключом /?, ознакомиться с описанием и возможностями консольной утилиты Reg.exe для работы с Реестром ОС, доступной из командной строки.

Задание 8.1. Изучение основных возможностей системного модуля «Редактор Реестра» ОС Windows XP на конкретных примерах

С одной стороны, настройка и конфигурирование ОС Windows XP возможна посредством штатных средств системы, например, изученных в предыдущих лабораторных работах, оснасток.

С другой стороны, настройка ОС возможна без помощи GUI, посредством служебного программного модуля «Редактор Реестра», причем таким образом, каким ее невозможно настроить с помощью GUI. Это делает «Редактор Реестра» наиболее мощным и, в то же время, опасным инструментом ОС, поскольку вносимые настройки применяются к системе без контроля с ее стороны. Контроль правильности вносимых настроек должен осуществляться тем, кто их вносит, то есть пользователем.

Поэтому тщательное изучение этого системного инструмента является необходимым для опытного пользователя и, тем более, IT-профессионала. Для изучения основных возможностей служебного приложения «Редактор Реестра» выполните следующее.

Содержание задания

1. Откройте «Редактор Реестра», одновременно нажав клавиши «WIN» (на ней изображен флаг-логотип MS Windows) + «R», введя в появившемся окне «Выполнить» строку Regedit и нажав Enter для подтверждения ввода.

Альтернативный способ запуска приложения состоит в следующем:

- нажмите Пуск | Выполнить,
- наберите в появившемся окне Regedit.exe (или просто Regedit),
- нажмите Enter для ввода.

2. В появившемся окне «Редактора Реестра» (рис. 9), обратите внимание на то, что с левой стороны окна расположена панель ключей, а с правой стороны – панель значений. Панель ключей отображает корневые ключи и подключи Реестра. Щелкая манипулятором мышь по корневым ключам Реестра, отобразите слева его иерархию. Панель значений справа демонстрирует настройки, содержащиеся в каждом из подключей. Щелкните по одному из них на панели ключей и найдите его значения на панели значений.

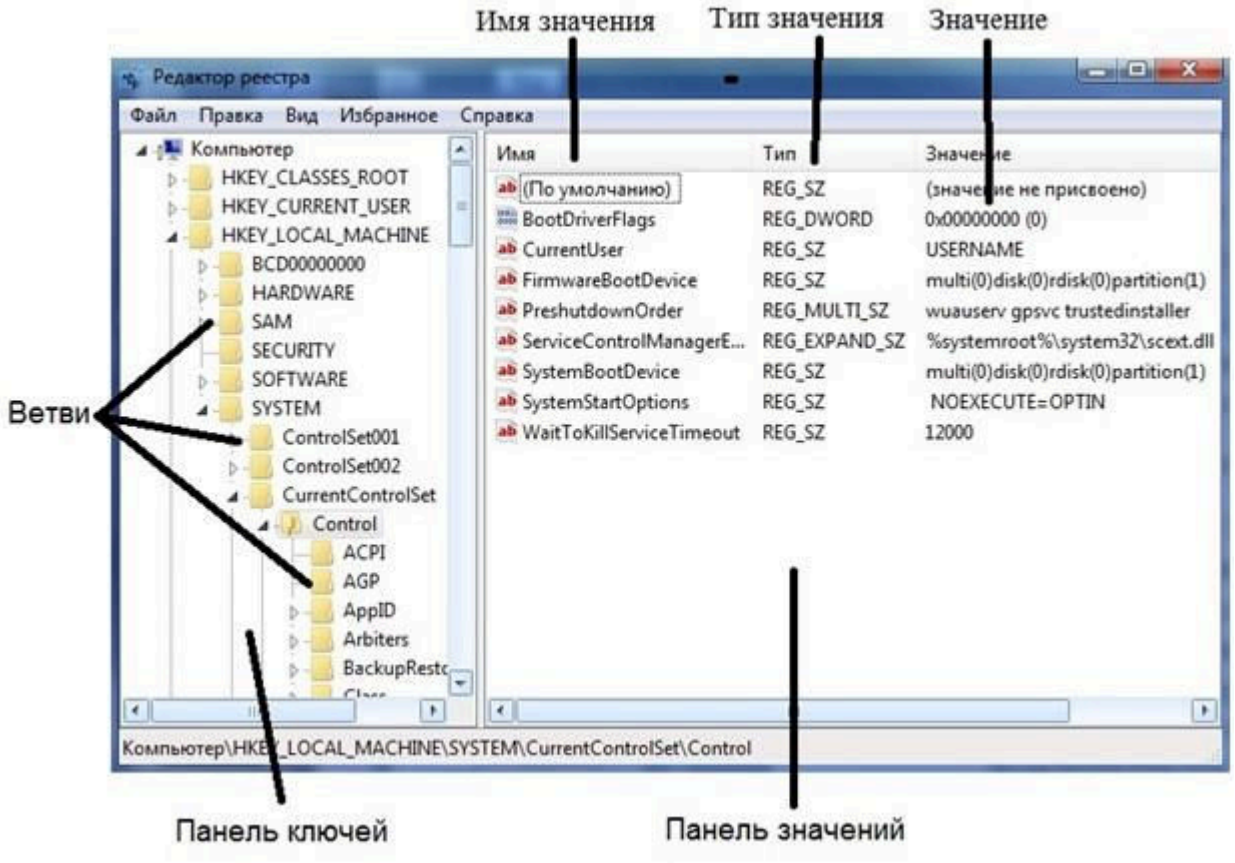


Рисунок 9 Окно редактора реестра

Термин «Ветвь» на рисунке 1 указывает на ключ и все его подключи. Разворачивать и сворачивать ключи и подключи также можно, щелкая манипулятором мышь на значках узла с соответствующим символом «+» или «-». На панели ключей выберите ветвь Реестра, изображенную на рис. 6.1 и разверните ее, щелкнув на узле с символом «+». На панели значений найдите параметр с именем CmdLine и определите строку данных этого значения.

3. Самостоятельно выберите в Реестре ОС какой-либо ключ (с соответствующими подключами), содержащий одновременно значения с основными системными типами REG_BINARY, REG_DWORD и REG_SZ. Обратите внимание на имеющуюся в редакторе возможность представления данных выбранного значения в двоичном виде (команда «Вывод двоичных данных» в меню «Вид»). Полученные данные перенесите в отчет.
4. На практике, иногда становится необходимым удаление из Реестра данных различных значений. Некорректно деинсталлированное приложение оставляет в Реестре «следы», которые иногда не позволяют повторно инсталлировать это приложение более поздней версии и, что не редкость, замедляют обращение к

Реестру и, как следствие, всю систему в целом. В частности, данное утверждение верно по отношению к широко известному пакету Nero, предназначенному для записи данных на оптические носители. Оставшиеся в Реестре ОС записи о данном программном продукте могут быть удалены из него вручную, но это кропотливая и сопряженная с опасностью процедура. Поэтому для безопасного удаления оставшихся в Реестре записей рекомендуется применять утилиту Nero General Clean Tool.

Однако, очень часто подобных утилит для специализированной чистки ненужных приложений не существует. Необходимо ручное вмешательство в Реестр ОС. Для этой цели в служебном модуле «Редактор Реестра» используются команды «Найти», «Найти далее» и «Удалить» меню «Правка».
Посредством имеющейся возможности поиска в Реестре можно осуществлять поиск

a. имен ключей,
b. имен значений,
c. строковых данных

с частичным или точным совпадением. При этом важно помнить о возможных последствиях, в случае ошибочного удаления нужного системного значения, и заранее создать резервную копию Реестра ОС.

Выберите пункт «Найти…» в меню «Правка». В поле «Найти:» введите строку hivelist («список файлов кустов») в качестве примера текста, который необходимо найти. При этом на поиск можно наложить определенные ограничения, установив соответствующие флажки:

- чтобы найти ключи, чьи имена содержат введенный текст, выберите пункт «имена разделов»,
- чтобы найти значения, чьи имена содержат введенный текст, выберите пункт «имена параметров»,
- чтобы найти значения REG_SZ, чьи данные содержат введенный текст, выберите пункт «значения параметров».

Осуществите поиск списка файлов кустов, нажав Enter или щелкнув манипулятором мышь по кнопке «Найти далее».

5. Чтобы каждый раз не осуществлять поиск одних и тех же необходимых данных, целесообразно воспользоваться быстрым доступом к ним, используя меню «Избранное». В этом меню может быть составлен список ключей, например, часто используемых для редактирования. Откройте меню «Избранное» и «добавьте в избранное» только что найденный список файлов кустов. Обратите внимание на то, что в меню стала активной команда «удалить из избранного». Она может быть использована для корректировки списка ключей в случае необходимости.
6. В «Редакторе Реестра», при условии, что права доступа ключей и значений позволяют это, имеется возможность их добавлять, удалять или переименовывать. Именно эти возможности служебного модуля делают Реестр мощным инструментом конфигурирования и оптимизации ОС.

В зависимости от типа значений «Редактор Реестра» отображает различные диалоговые окна для их редактирования. В частности, диалоговое окно «Изменение строкового параметра» открывается тогда, когда редактируется значение типа REG_SZ; диалоговое

окно «Изменение параметра DWORD» открывается в случае, если редактируется значение типа REG_DWORD, а для редактирования значений типа REG_BINARY используется диалоговое окно «Изменение двоичного параметра».
Чтобы изменить значение, щелкните манипулятором мышь на пункте «Изменить» в меню «Правка», а затем введите в поле «Значение» новые данные. Необходимо помнить, что все изменения остаются в системе незамеченными до тех пор, пока прикладная программа или сама ОС не решит перезагрузить измененное значение из Реестра. Иными словами, для того, чтобы изменения вступили в силу, необходимо закрыть работающее приложение и снова перезагрузить его.
Ниже приведены несколько примеров изменения Реестра ОС, которые являются базовыми и необходимыми для конфигурирования и оптимизации ОС Windows XP, но не достаточными для ее полноценной настройки. В диалоговом окне «Изменение строкового параметра» ключа HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AeDebug измените значение параметра Auto (тип значения REG_SZ) на 0, дважды кликнув по нему манипулятором мышь. Это изменение приведет к отключению служебного модуля «Доктор Ватсон», сомнительная польза от которого очевидна.
В диалоговом окне «Изменение параметра DWORD» ключа HKLM\SYSTEM\CurrentControlSet\Services\Cdrom при необходимости измените значение параметра AutoRun (тип значения REG_DWORD) на 0, тем самым, отключив автозапуск оптического привода.
В диалоговом окне «Изменение строкового параметра» ключа HKCU\Control Panel\Desktop измените значение параметра MenuShowDelay на любое число, менее 400, например, 50. Это число определяет задержку раскрытия вложенных меню Пуск. Чем это число меньше, тем меньше задержка.
Удалите из Реестра параметры Optional и связанный с ним Posix в ключе HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\SubSystems, щелкнув на них правой кнопкой манипулятора мышь и выбрав команду «Удалить» из появившегося списка команд. Эти параметры отвечают за запуск Unix-приложений в ОС Windows XP.
Воспользовавшись меню «Правка» изучаемого модуля, создайте новый строковый параметр EnableQuickReboot со значением равным 1 в ключе HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon для общего ускорения перезагрузки ОС Windows XP.
Откройте ветвь Реестра HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\ и создайте внутри новый раздел Explorer таким образом, чтобы получился ключ вида HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer. Обратите внимание на то, что создание раздела сопровождается созданием строкового параметра «по умолчанию» внутри него.
В конечном разделе Explorer создайте системный параметр с именем NoRemoteRecursiveEvents и значением типа REG_DWORD равным 1. Реализованная последовательность действий направлена на ускорение загрузки системного программного модуля «Мой компьютер».

7. Экспорт Реестра ОС или его части это одна из тех вещей, которые достаточно часто приходится делать системным администраторам и опытным пользователям. По сути, экспорт представляет собой копирование данных в другой файл. По отношению к Реестру, этот файл имеет расширение .reg.

Экспорт настроек в Reg-файл имеет практическую ценность. Прежде всего, это великолепный способ создать резервную копию системных настроек на случай их экстренного восстановления при необходимости. Это также хороший способ передавать настройки другим пользователям на другие компьютеры сети. Имея несколько Reg-файлов с различными настройками системы, возможно импортировать их одним двойным

щелчком мышью.
Для экспорта ветвей реестра выполните следующие инструкции:

- щелкните мышью на ключе, находящемся в вершине ветви, выбранной самостоятельно, которую необходимо экспортировать,
- в меню «Файл» выберите пункт «Экспорт», чтобы вывести на экран диалоговое окно «Экспорт файла Реестра»,
- в поле «Имя файла» введите имя файла для экспорта,
- выберите диапазон экспорта: чтобы создать копию всего реестра, щелкните на «Весь реестр», чтобы создать копию выделенной ветви, щелкните на «Выбранная ветвь»,
- в выпадающем списке «Тип файла» выберите тип файла для экспорта: «Файлы Реестра *.reg», «Файлы кустов Реестра *.*», «Текстовые файлы *.txt» или «Файлы Реестра Win9x/NT4 *.reg»,
- экспортируйте ветвь, мышью щелкнув на кнопке «Сохранить».

Последовательность вышеописанных действий фактически представляет собой один из способов создания резервной копии Реестра ОС. Сохранение Реестра перед его редактированием является принципиальным, поскольку обеспечивает дополнительный шанс на его восстановление в случае выхода системы из строя посредством непродуманных действий пользователя.
Обратная процедура импорта Реестра практически ни чем не отличается от простого открытия Reg-файла. Для этого необходимо щелкнуть мышью на пункте «Импорт» в меню «Файл», далее в выпадающем списке «Тип файла» выбрать тип файла, который предполагается импортировать, а затем в поле «Имя файла» ввести полный путь Reg-файла и подтвердить операцию, щелкнув по кнопке «Открыть».
Альтернативный способ импорта Реестра ОС заключается в следующем:

- в «Проводнике» дважды щелкните мышью на Reg-файле, чтобы внести его содержимое в Реестр,
- подтвердите внесение настроек в Реестр, щелкнув мышью по кнопке «Да», после чего должно последовать сообщение об успешном завершении операции.

Внимание! Файлы Реестра ОС Windows XP представляют собой пятую версию Reg-файлов. Другие ОС семейства Windows имеют другие версии Regфайлов. Поэтому не импортируйте Reg-файл, созданный в одной версии ОС Windows, в другую версию этой ОС. Это может привести к неработоспособности последней.
В ходе выполнения задания были рассмотрены основные функции и базовые понятия «Редактора Реестра»: изучены возможности редактирования Реестра, изменения значений параметров его ключей, а также создания новых разделов и подключей. Однако, существует ряд сложных задач, для выполнения которых полученных знаний не достаточно. Поэтому в дальнейшем предполагается осуществить изучение расширенных возможностей Реестра ОС, необходимых, в основном, для IT-профессионалов и опытных пользователей.
При выполнении задания используйте следующие инструкции:

- перенесите последовательность выполняемых действий по каждому из пунктов 1-7 в отчет (возможно приведение графических фрагментов, сделанных с экрана, в качестве демонстрационного материала),
- результаты изучения возможностей системного модуля «Редактор Реестра» занесите в табл. 6.3,
- сделайте вывод о проделанной работе и запишите его в отчет.

Таблица 3. Исследование ключа системного Реестра ОС Windows XP

Исследуемый ключ:				
(название ключа)				
№ п.п.	Значение ключа	Имя	Тип	Значение
1				
n				

Задание 8.2. Изучение некоторых специальных возможностей Реестра ОС Windows XP в системном модуле «Редактор Реестра» на конкретных примерах.

В данном задании предполагается изучить некоторые способы и инструменты устранения ошибок Реестра ОС, возникающих в процессе непродуманного или некорректного применения настроек в системе.
Большинство системных инструментов обладают возможностями, намного повышающими надежность ОС Windows XP по сравнению с предыдущими версиями. Однако, многие из них требуют выполнения различных подготовительных операций. В частности, в случае восстановления Реестра ОС из резервной копии сначала необходимо ее создать, воспользовавшись, например, потенциалом системного приложения «Редактор Реестра».
Для рассмотрения некоторых специальных возможностей Реестра ОС с применением служебного программного средства «Редактор Реестра» необходимо освоить следующее.

Содержание задания

1. Первое, что целесообразно изучить в контексте данного задания это создание резервных копий отдельных значений Реестра ОС, которые предполагается изменять чаще других. При этом возврат к первоначальному состоянию записи Реестра возможен при перезагрузке ОС с безопасном режиме и удалении некорректной копии измененного значения.

Чтобы осуществить создание резервной копии отдельной записи, необходимо выполнить следующее:

- выберите самостоятельно какую-либо запись Реестра ОС, значение которой предполагается изменять в дальнейшем; в качестве примера можно обратиться к значению, которое уже было изменено в предыдущем задании, а именно значение MenuShowDelay в ключе HKCU\Control Panel\Desktop,
- присвойте начальному значению какой-либо отличительный признак, переименовав его посредством команды «Переименовать» в меню «Правка», например, в Initial_MenuShowDelay; если предполагается часто изменять значение, то целесообразно маркировать его с использованием даты и времени,
- добавьте новое значение MenuShowDelay с оригинальными именем и типом, но с другими данными значения; таким образом, в системном Реестре образуется два значения (оригинал и резервная копия),

Новое значение системного параметра MenuShowDelay вступит в силу после перезагрузки ОС. Значение параметра Initial_MenuShowDelay при этом будет проигнорировано.
Для возврата Реестра ОС в исходное состояние достаточно удалить резервную копию MenuShowDelay и переименовать оригинальное значение.

2. Наряду с рассмотренным в предыдущем задании способом резервирования Реестра ОС или его части посредством экспорта данных в Reg-файл, в «Редакторе Реестра» имеется возможность резервирования отдельных файлов кустов, поскольку они лучшим образом подходят для создания резервных копий Реестра, чем Reg-файл. В частности, этот способ представляет собой гораздо более аккуратный путь резервирования ветвей. Преимущество данного способа заключается в том, что при импорте отдельного файла куста, содержащего необходимый ключ, «Редактор Реестра» полностью замещает текущий ключ содержимым файла куста. При этом удаляются все значения, внесенные в Реестр ОС после момента создания резервной копии в файле куста.

Экспорт ветвей в файлы кустов похож на экспорт их в Reg-файлы. Для этого просто необходимо выбрать другой тип файлов «Файлы кустов Реестра» в процессе их сохранения (пункт «Экспорт» в меню «Файл»). При этом необходимо задать расширение файла куста Реестра ОС, например, .dat или .hiv.
В качестве задания к данному пункту экспортируйте выбранную самостоятельно ветвь Реестра ОС, воспользовавшись процедурой, аналогично описанной при экспорте Реестра в Reg-файл и рассмотренной в предыдущем задании 8.1 лабораторной работы.
Для восстановления настроек Реестра ОС повторите процесс в обратном порядке. Из меню «Файл» выберите пункт «Импорт»; затем в выпадающем списке «Тип файлов» выберите «Файлы кустов Реестра», введите имя файла куста, в который настройки были сохранены, а затем щелкните мышью на кнопке «Открыть».
Отдельно необходимо отметить, что экспорт и импорт кустов отличается от их загрузки и выгрузки (команды «Загрузить куст» и «Выгрузить куст» в меню «Файл»). В случае импортирования файла куста вносятся изменения в рабочую часть Реестра ОС. В том случае, когда загружается файл куста, в Реестре создается полностью новая ветвь, не используемая ОС Windows XP. При этом ОС не читает и не изменяет настроек ветви. В частности, это обстоятельство может быть использовано с целью автономного изучения настроек ОС.
3. Дальнейшее изучение возможностей Реестра будет направлено на то из них, которое является прерогативой IT-профессионалов и системных программистов, а именно возможность настройки ассоциаций файлов, позволяющая управлять следующими аспектами их обработки в ОС Windows XP:

- какую пиктограмму ОС отображает рядом с именем файла;
- какое запускается приложение при двойном щелчке мышью;
- как «Проводник» отображает конкретные типы файлов в системе;
- какие команды появляются в контекстном меню файла;
- другие функции, например, такие как «всплывающие подсказки».

На рис. 6.2 изображены ключи Реестра, с которыми ОС сверяется, когда пользователь щелкает правой кнопкой мыши на текстовом файле и выбирает команду «Открыть» из контекстного меню. Вначале ОС ищет расширение файла в HKCR. Значение по умолчанию указывает на то, что класс программ, ассоциированный с расширением .txt, называется txtfile (рис. 6.2). Принимая во внимание эти данные, ОС далее ищет в HKCR\txtfile подключ shell, чтобы определить команды, которые следует добавить к контекстному меню. В изображенном на рисунке случае ОС Windows XP добавляет к

контекстному меню команду «Открыть» (Shell\Open). Когда пользователь выбирает эту команду «Открыть» на исполнение, система запускает команду, указанную в значении подключа command (Shell\Open\command).
Команда в подключе command обычно имеет вид «Исполняемое приложение», включающее полный путь и имя исполняемого файла, со следующими за ним опциями (например, %1, как показано на рис. 6.2).
Необходимо помнить, что при написании скриптов тип значения команды по умолчанию REG_SZ должен быть изменен на REG_EXPAND_SZ. Это позволит использовать переменные среды типа %SYSTEMROOT% (в противном случае используйте явное указание пути). %1 является указателем на целевой файл для открытия (заключите %1 в кавычки на случай, если путь и имя целевого файла содержат пробелы).

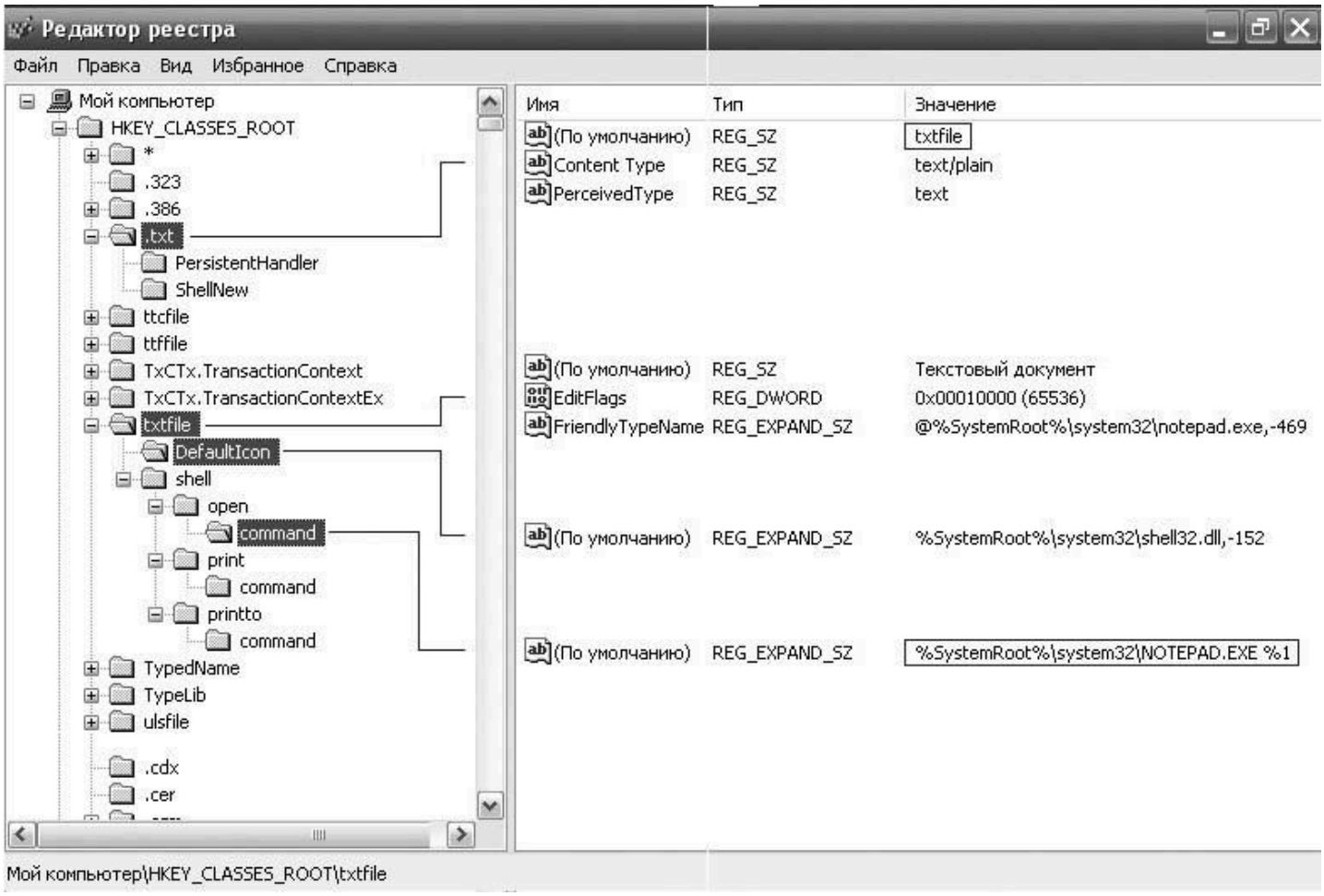


Рис. 10. Значение ключа расширения .txt, указывающее на соответствующий класс программ, с которыми это расширение ассоциировано
В качестве примера к изучаемому материалу, создайте в системном Реестре ОС Windows XP свой собственный обработчик произвольного расширения. Для этого выполните следующие действия:

- выберите самостоятельно произвольное расширение, состоящее из трех символов, обработчик которого предполагается создать,
- в разделе HKCR Реестра ОС создайте новый раздел с названием выбранного ранее расширения; при этом обратите внимание на то, как это уже сделано для других расширений в системе,

- значение строкового параметра (по умолчанию), соответствующего созданному разделу, должно содержать ссылку вида ***file, где *** – символы выбранного расширения, на раздел обработчика данного расширения,
- в разделе HKCR Реестра ОС создайте новый раздел обработчика расширения следующего вида ***file\shell\open\command – для команды открытия и ***file\shell\list\command – для команды просмотра файла; воспользуйтесь рис. 6.2. в случае необходимости,
- в разделах command, каждой из ветвей, создайте по одному расширяемому строковому параметру типа REG_EXPAND_SZ с наименованием (по умолчанию),
- удалите старые строковые параметры REG_SZ, создаваемые в разделе command по умолчанию (рис. 6.2.),
- в расширяемом строковом параметре раздела ***file\shell\list измените данные значения по умолчанию на «Мой просмотр»,
- в соответствующих разделах command измените значения расширяемых строковых параметров на команды для открытия файла и его просмотра. В частности, для открытия текстового файла можно воспользоваться приложением WordPad.exe, а для его просмотра выбрать NotePad.exe,
- проверьте работоспособность обработчика, выполнив следующее:
- выберите какой-либо файл с его стандартным расширением;
- поменяйте стандартное расширение на то, обработчик которого Вы только что создали;
- правой кнопкой манипулятора мышь выберите из контекстного меню команду с именем того файла (filename.***), который Вы собираетесь открыть или команду «Мой просмотр», чтобы просмотреть файл; при этом должно загрузиться соответствующее приложение обработчика.

4. Еще одной специальной возможностью Реестра, которая может существенно упростить восприятие ОС, является возможность настройки ее внешнего вида. В Реестре ОС существуют десятки, если не сотни, различных программных переключателей, позволяющих включить или отключить ту или иную визуальную опцию в системе. В частности, воспользовавшись некоторыми настройками Реестра ОС можно настроить главное меню «Пуск».

Настройка главного меню системы возможна стандартными средствами ОС, в частности, посредством GUI. Хотя в ОС имеется такая возможность, опытные пользователи и IT-профессионалы возможно захотят создать скрипт для автоматизации настроек этого меню. Системные администраторы вряд ли будут перенастраивать меню «Пуск» при каждой установке ОС Windows XP, особенно, когда парк обслуживаемых машин исчисляется сотнями. Скорее всего, написанный скрипт будет автоматически распространяться по сети. Все настройки главного меню «Пуск» находятся в системном Реестре в одном месте HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ Advanced.

Таблицы 4 и 5 описывают значения, которые можно добавлять в этот ключ. Причем первая таблица содержит значения для классического меню «Пуск», а вторая – для нового меню, соответственно. Большинство из этих значений принадлежит к типу REG_DWORD (данные имеют вид 0x01, 0x02 и т.д.), но некоторые из них имеют тип REG_SZ (символьные данные вида «NO» или «YES»).
Для настройки меню «Пуск» посредством Реестра ОС Windows XP, выполните следующие действия:

- самостоятельно выберите вид главного меню «Пуск» (классический или новый), соответствующие параметры которого будут применяться в Реестре ОС (табл. 4 или 5),
- самостоятельно определитесь какие именно параметры будут применены Вами для конфигурирования меню «Пуск» (в количестве не менее пяти штук),
- самостоятельно конфигурируйте меню «Пуск» с применением выбранных параметров,
- результаты конфигурирования меню «Пуск» зафиксируйте в виде графических фрагментов, сделанных с экрана командой Prt Screen.

При выполнении задания используйте следующие инструкции:

- перенесите последовательность выполняемых действий по каждому из пунктов 1-4 в отчет (возможно приведение графических фрагментов, сделанных с экрана, в качестве демонстрационного материала),
- результаты применения новых значений системных параметров Реестра ОС перенесите в отчет,
- сделайте вывод о проделанной работе и запишите его в отчет.

Таблица 4. Настройки классического меню «Пуск» ОС Windows XP

№ п.п.	Параметр	Описание
1	StartMenuAdminTools	«Администрирование» NO – Скрыть; YES – Отобразить;
2	CascadeControlPanel	«Панель управления» NO – Отобразить как ссылку; YES – Отобразить как меню;
3	CascadeMyDocuments	«Мои документы» NO – Отобразить как ссылку; YES – Отобразить как меню;
4	CascadeMyPictures	«Мои рисунки» NO – Отобразить как ссылку; YES – Отобразить как меню;
5	CascadePrinters	«Принтеры» NO – Отобразить как ссылку; YES – Отобразить как меню;
6	IntelliMenus	«Персонифицированное меню» 0x00 – не использовать; 0x01 – использовать;
7	CascadeNetwork-Connections	«Сетевые подключения» NO – Отобразить как ссылку; YES – Отобразить как меню;
8	Start_LargeMFUIcons	«Пиктограммы в меню «Пуск» 0x00 – Отобразить маленькими; 0x01 – Отобразить большими;
9	StartMenuChange	«drag'n'drop» 0x00 – Отключить; 0x01 – Включить;
10	StartMenuFavorites	«Избранное» 0x00 – Скрыть; 0x01 – Отобразить;
11	StartMenuLogoff	«Завершение сеанса» 0x00 – Скрыть; 0x01 – Отобразить;
12	StartMenuRun	Команда «Выполнить» 0x00 – Скрыть; 0x01 – Отобразить;
13	StartMenuScrollPrograms	Прокрутка меню «Программы» NO – Не использовать; YES – Использовать;

Таблица 5. Настройки нового меню «Пуск» ОС Windows XP

№ п.п.	Параметр	Описание
1	Start_ShowControlPanel	«Панель управления» 0x00 – Скрыть; 0x01 – Отобразить как ссылку; 0x02 – Отобразить как меню;
2	Start_EnableDragDrop	«drag'n'drop» 0x00 – Отключить: 0x01 – Включить;
3	StartMenuFavorites	«Избранное» 0x00 – Скрыть; 0x01 – Отобразить;
4	Start_ShowMyComputer	«Мой компьютер» 0x00 – Скрыть; 0x01 – Отобразить как ссылку; 0x02 – Отобразить как меню;
5	Start_ShowMyDocs	«Мои документы» 0x00 – Скрыть; 0x01 – Отобразить как ссылку; 0x02 – Отобразить как меню;
6	Start_ShowMyMusic	«Моя музыка» 0x00 – Скрыть; 0x01 – Отобразить как ссылку; 0x02 – Отобразить как меню;
7	Start_ShowMyPics	«Мои рисунки» 0x00 – Скрыть; 0x01 – Отобразить как ссылку; 0x02 – Отобразить как меню;
8	Start_ShowNetConn	«Сетевые подключения» 0x00 – Скрыть; 0x01 – Отобразить как ссылку; 0x02 – Отобразить как меню;
9	Start_AdminToolsTemp	«Администрирование» 0x00 – Скрыть; 0x01 – Отобразить в меню «Все программы» 0x02 – Отобразить в меню «Все программы» и меню «Пуск»;
10	Start_ShowHelp	«Справка и поддержка» 0x00 – Скрыть; 0x01 – Отобразить;
11	Start_ShowNetPlaces	«Сетевое окружение» 0x00 – Скрыть; 0x01 – Отобразить;
12	Start_ShowOEMLink	«Производитель» 0x00 – Скрыть; 0x01 – Отобразить;
13	Start_ShowPrinters	«Принтеры и факсы» 0x00 – Скрыть; 0x01 – Отобразить;
14	Start_ShowRun	Команда «Выполнить» 0x00 – Скрыть; 0x01 – Отобразить;
15	Start_ShowSearch	Команда «Найти» 0x00 – Скрыть; 0x01 – Отобразить;
16	Start_ScrollPrograms	Прокрутка меню «Программы» 0x00 – не использовать; 0x01 – использовать;

Рассмотренные в данном задании некоторые частные вопросы являются минимально необходимыми и достаточными для того, чтобы показать широкие возможности системного Реестра ОС Windows XP. Изучение Реестра ОС и инструментов для работы с ним является одной их важнейших задач при обучении IT-профессионалов и опытных пользователей, поскольку в дальнейшем позволит им, не прибегая к стандартным программным средствам ОС, оперировать системными настройками и фактически иметь низкоуровневый доступ к ОС. К сожалению, в рамках лабораторного практикума изучение всех возможностей Реестра не представляется возможным. Тем не менее, в глобальной сети Интернет на эту тему существует огромное количество литературы, конкретных системных настроек Реестра, **твиков** и **скриптов** на его основе, а также практических рекомендаций по конфигурированию ОС. Часть из этих потенциальных

возможностей, направленных на оптимизацию работы системы, будет рассмотрена в следующем учебном задании.

Изучение возможностей конфигурирования ОС Windows XP посредством специальных настроек Реестра, твиков и скриптов на его основе, направленных на оптимизацию работы системы.

Порядок выполнения:

I. Оптимизация ОС посредством Реестра является высшим пилотажем с точки зрения IT-профессионалов и опытных пользователей, поскольку требует тщательного изучения самого Реестра, инструментов для его редактирования, достаточно обширных знаний операционных систем, принципов их функционирования, а также достаточного опыта настройки ОС стандартными программными средствами и утилитами сторонних производителей.

Оптимизация как процесс предполагает настройку и конфигурирование ОС таким образом, чтобы получить стабильно работающую систему, с одной стороны, но при этом достаточно быструю и оптимально расходующую системные ресурсы, с другой стороны. Оптимизация ОС с использованием Реестра может быть осуществлена вручную, когда IT-профессионал самостоятельно вводит конкретные параметры и значения в Реестр для достижения заранее определенного результата, отчетливо зная при этом, что именно он изменяет и для чего он это делает. При этом он может облегчить ввод этих значений, написав сценарий или скрипт с целым рядом необходимых параметров и применив его к системе, одним из способов изученных ранее. Другой путь, назовем его автоматизированный, скорее подходит для тех лиц, кто не обладает значительными знаниями архитектуры ОС и принципов ее функционирования, а именно для рядовых пользователей и тех, кто только начинает приобретать знания в этой сфере. Этот путь заключается в том, чтобы использовать для настройки ОС, так называемые, **твики** Реестра ОС и написанные профессиональными программистами скрипты. Отчасти, это может гарантировать некоторую степень безопасного изменения настроек системы. Наконец, полностью автоматизированная настройка ОС возможна только в том случае, если твики Реестра или скрипты управляют автоматически процессом распространения настроек по сети и их применением в системе. Этот путь требует для своего осуществления не только достаточных знаний архитектуры и Реестра ОС, но дополнительно, как минимум, одного языка программирования, а также способов передачи данных по сетям. Как правило, багаж знаний и опыта, необходимый для его реализации, может быть достигнут через несколько лет практической работы в области компьютерных технологий и программирования в частности.

Другое, упоминаемое в рамках настоящей лабораторной работы, понятие «твик Реестра» (от англ. tweak - настройка) соответствует некоторой настройке программного обеспечения (ПО) или операционной системы, хранящейся в системном Реестре. Твики позволяют автоматизировать настройку ПО. К тому же, многие параметры легче настроить при помощи твика Реестра ОС, чем посредством GUI.

Твик Реестра ОС *представляет собой текстовый Reg-файл, который может быть создан в любом текстовом редакторе и сохранен в кодировке ANSI или Unicode.*

В качестве классического примера ниже приводится содержимое Reg-файла,

отключающего меню недавних документов. При этом обратите внимание на принцип построения и синтаксис данного файла. который использует следующие системные символы:

- ; – для справочной информации, которая является не исполняемой при внесении данных в Реестр ОС,
- [и] – для указания ключа Реестра, в который будет вносится его новое значение,
- « и » – для непосредственной идентификации параметра, подлежащего изменению.

Файл твика Реестра ОС Windows XP (Reg-файл) имеет стандартный вид:
Windows Registry Editor Version 5.00
;Отключить меню недавних документов
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer]
"NoRecentDocsMenu"=hex:01,00,00,00
Импорт приведенного твика Реестра и его интеграция в ОС осуществляется одним из способов, изученных ранее в предыдущем учебном задании.
Глобально все твики и настройки Реестра ОС можно разделить на несколько категорий, по их принадлежности к корневым ключам, в частности, HKLM и HKCU. По результатам, получаемым при применении настроек в системе, они подразделяются на: настройки пользователей и настройки интерфейса ОС. Отдельно можно определить настройки оптимизации ОС и конфигурирования системных служб. В рамках настоящей лабораторной работы последнюю категорию настроек предполагается рассмотреть более подробно.

Задание 8.3. Конфигурирование контекстного меню служебного программного средства «Мой компьютер» ОС Windows XP посредством применения твика Реестра.

Первая группа системных настроек имеет отношение в контекстному меню служебного приложения «Мой компьютер», возникающего при одиночном клике правой кнопкой манипулятора мышь на соответствующей иконке в меню «Пуск». Сущность данных настроек заключается в том, что при их применении контекстное меню приложения «Мой компьютер» приобретает несколько иной вид, а именно в нем становятся доступными некоторые системные программы (например, «Проводник») и функции. Польза такого новшества очевидна, поскольку фактически в один клик становятся доступными все необходимые системному администратору инструменты. Это существенно экономит время работы IT-профессионала или опытного пользователя.
Ниже приводится список настроек Реестра ОС, преобразующих контекстное меню штатного приложения «Мой компьютер» как показано на рис. 6.3. Данные системные настройки могут быть применены в ОС как по отдельности, так и в составе программного твика Реестра.

1. Настройка добавляет в контекстное меню приложения "Мой компьютер" команду "Администрирование":
   [HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\1]
   @="Администрирование"

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\1\command]
@="control admintools"

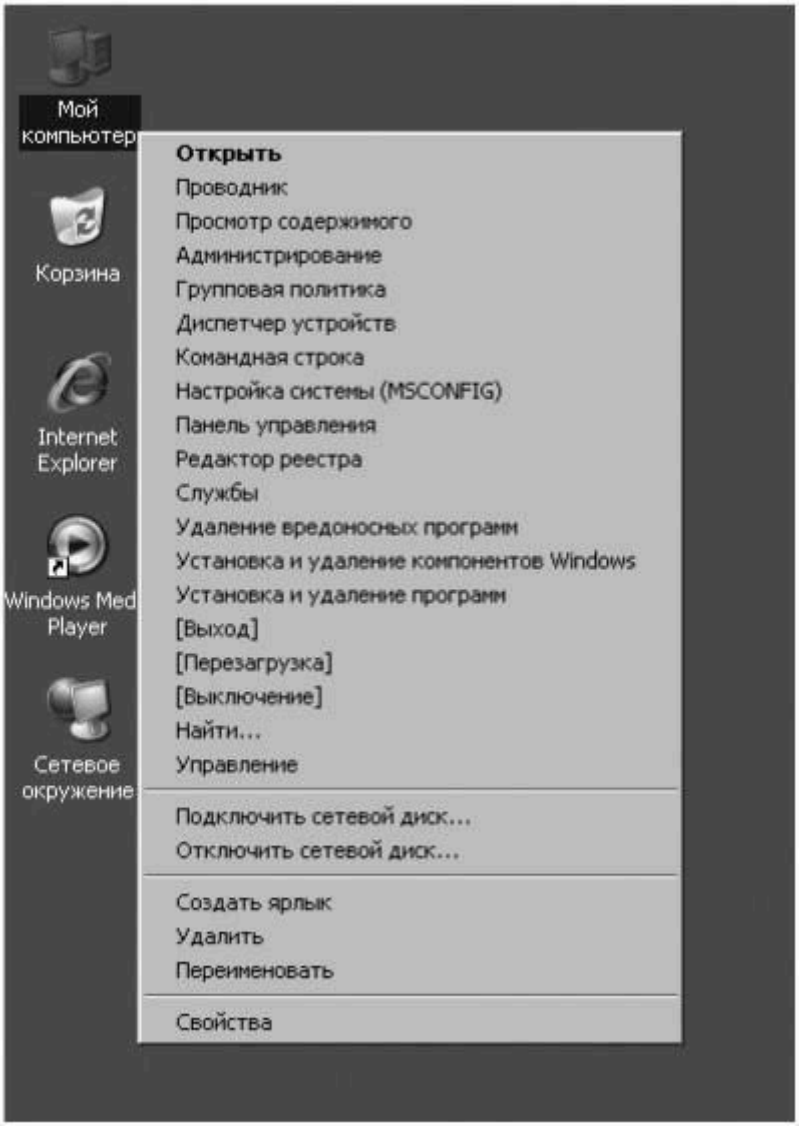


Рисунок 11 Внешний вид контекстного меню «Мой компьютер» после применения настроек в системном Реестре

2. Настройка добавляет в контекстное меню приложения "Мой компьютер" команду "Групповая политика":

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\11]
@="Групповая политика"
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\11\command]
@=hex(2):25,00,77,00,69,00,6e,00,64,00,69,00,72,00,25,00,5c,\
00,73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,6d,00,6d,00,63,00,\
2e,00,65,00,78,00,65,00,20,00,2f,00,73,00,20,00,25,00,53,00,79,00,73,00,74,\
00,65,00,6d,00,52,00,6f,00,6f,00,74,00,25,00,5c,00,73,00,79,00,73,00,74,00,\
65,00,6d,00,33,00,32,00,5c,00,67,00,70,00,65,00,64,00,69,00,74,00,2e,00,6d,\
00,73,00,63,00,20,00,2f,00,73,00,00,00

3. Настройка добавляет в контекстное меню приложения "Мой компьютер" команду "Диспетчер устройств":

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\2]
@="Диспетчер устройств"
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\2\command]
@=hex(2):25,00,77,00,69,00,6e,00,64,00,69,00,72,00,25,00,5c,\
00,73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,6d,00,6d,00,63,00,\
2e,00,65,00,78,00,65,00,20,00,2f,00,73,00,20,00,25,00,53,00,79,00,73,00,74,\
00,65,00,6d,00,52,00,6f,00,6f,00,74,00,25,00,5c,00,73,00,79,00,73,00,74,00,\
65,00,6d,00,33,00,32,00,5c,00,64,00,65,00,76,00,6d,00,67,00,6d,00,74,00,2e,\
00,6d,00,73,00,63,00,20,00,2f,00,73,00,00,00
```

4. Настройка добавляет в контекстное меню приложения "Мой компьютер"команду "Командная строка":

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\22]
@="Командная строка"
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\22\command]
@="cmd.exe"
```

5. Настройка добавляет в контекстное меню приложения "Мой компьютер" команду "Настройка системы":

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\3]
@="Настройка системы (MSCONFIG)"
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\3\command]
@="msconfig.exe /s"
```

6. Настройка добавляет в контекстное меню приложения "Мой компьютер" команду "Панель управления":

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\4]
@="Панель управления"
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\4\command]
@="rundll32.exe shell32.dll,Control_RunDLL"
```

7. Настройка добавляет в контекстное меню приложения "Мой компьютер" команду "Редактор реестра":

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\44]
@="Редактор реестра"
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-
```

08002B30309D}\shell\44\command]
@="Regedit.exe"

8. Настройка добавляет в контекстное меню приложения "Мой компьютер" команду "Службы":

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\5]
@="Службы"
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\5\command]
@=hex(2):25,00,77,00,69,00,6e,00,64,00,69,00,72,00,25,00,5c,00,73,00,79,00,\
73,\ 00,74,00,65,00,6d,00,33,00,32,00,5c,00,6d,00,6d,00,63,00,2e,00,65,00,78,\
00, 65,00,20,00,2f,00,73,00,20,00,25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,\
00,6f,00,6f,00,74, 00,25,00,5c,00,73,00,79,00,73,00,74,00,65,00,6d,00,33,00, 32,\
00,5c,00,73,00,65,00, 72,00,76,00,69,00,63,00,65,00,73,00,2e,00,6d,00,73, \
00,63,00,20,00,2f,00,73,00,00,00

9. Настройка добавляет в контекстное меню приложения "Мой компьютер" команду "Удаление вредоносных программ":

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\55]
@="Удаление вредоносных программ"
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\55\command]
@="mrt.exe"

10. Настройка добавляет в контекстное меню приложения "Мой компьютер" команду "Установка и удаление компонентов Windows":

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\6]
@="Установка и удаление компонентов Windows"
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\6\command]
@="rundll32 shell32,Control_RunDLL appwiz.cpl,,2"

11. Настройка добавляет в контекстное меню приложения "Мой компьютер" команду "Установка и удаление программ":

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\66]
@="Установка и удаление программ"
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\66\command]
@="control appwiz.cpl"

12. Настройка добавляет в контекстное меню приложения "Мой компьютер" команду "Выход":

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\7]
@="[Выход]"
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\7\command]
@="shutdown -l -f -t 0"

13. Настройка добавляет в контекстное меню приложения "Мой компьютер" команду "Перезагрузка":

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\77]
@="[Перезагрузка]"
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\77\command]
@="shutdown -r -f -t 0"

14. Настройка добавляет в контекстное меню приложения "Мой компьютер" команду "Выключение":

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\8]
@="[Выключение]"
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\shell\8\command]
@="shutdown -s -f -t 0"
Для создания твика Реестра ОС с целью конфигурирования контекстного меню приложения "Мой компьютер", выполните следующие действия.

Содержание задания

- сделайте резервную копию Реестра ОС одним из способов изученных в первой части лабораторной работы,
- из перечисленных выше настроек Реестра ОС выберите те (в количестве не менее пяти штук), которые наиболее подходят для Ваших персональных целей,
- самостоятельно создайте текстовый Reg-файл,
- откройте созданный твик-файл и напечатайте первой его строкой следующее Windows Registry Editor Version 5.00,
- отредактируйте твик-файл, скопировав в него выбранные ранее настройки Реестра ОС,
- сохраните созданный твик Реестра ОС.

Конфигурирование контекстного меню приложения "Мой компьютер" в ОС Windows XP осуществите следующим образом:

- примените созданный Вами твик Реестра в системе,
- перезагрузите ОС Windows XP, чтобы параметры вступили в силу,
- результаты применения твика Реестра ОС перенесите в отчет,
- сделайте вывод о проделанной работе и запишите его в отчет.

Задание 4. Конфигурирование ОС Windows XP с целью оптимизации ее работы и увеличения быстродействия.

Вторая группа системных настроек предназначена для оптимизации и увеличения быстродействия ОС Windows XP. Однако, оптимизация системы не ограничивается применением только данного набора настроек; существует ряд других системных твиков, позволяющих оптимизировать ее работу. Ниже представлен список части настроек Реестра ОС, наиболее интересных с точки зрения увеличения производительности некоторых подсистем. Как и в предыдущем случае каждая из них может быть использована в виде отдельного твика, а также при непосредственном редактировании Реестра ОС и внесении значений указанных параметров вручную.

1. Настройка позволяет оптимизировать расположение загрузочных файлов на жестком диске для ускорения их загрузки:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Dfrg\BootOptimizeFunction]
"Enable"="Y"

1. Настройка отключает сообщения об второстепенных ошибках в системе, но при этом уведомление о критических системных ошибках остается:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\PCHealth\ErrorReporting]
"AllOrNone"=dword:00000000
"IncludeMicrosoftApps"=dword:00000000 "IncludeWindowsApps"=dword:00000001
"IncludeKernelFaults"=dword:00000001
"DoReport"=dword:00000000
"ShowUI"=dword:00000000

1. Настройка отключает уведомления Центра обеспечения безопасности ОС Windows XP:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Security Center] ;Отключить предупреждения Брэндмауэра "FirewallDisableNotify"=dword:00000001
;Отключить предупреждения службы Автоматического обновления
"UpdatesDisableNotify"=dword:00000001
;Отключить предупреждения системы Антивирусной защиты
"AntiVirusDisableNotify"=dword:00000001

1. Настойка обеспечивает ускорение ОС при перезагрузке:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon]
"EnableQuickReboot"="1"

1. Настройка позволяет уменьшить использование доступного места на диске, отведенного для «Корзины», до 3% от общего пространства вместо 10%, отводимых системой по умолчанию:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\BitBucket]
"Percent"=dword:00000003

1. Настройка позволяет ускорить открытие служебного приложения "Мой компьютер":

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explo
rer]
"NoRemoteRecursiveEvents"=dword:00000001
```

1. Настройка ускоряет процесс выключения компьютера:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control]
"WaitToKillServiceTimeout"="2000"
```

1. Настройка позволяет зарезервировать оптимальный размер для главной таблицы размещения файлов MFT:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem]
"NtfsMftZoneReservation"=dword:00000003
```

1. Настройка осуществляет ускорение работы оптического привода:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem\CDFS]
"Prefetch"=dword:00004000
"PrefetchTail"=dword:00004000
"CacheSize"=hex:ff,ff,00,00
```

1. Настройка осуществляет исправление неполадок при отображении символов кириллицы в приложениях:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Nls\
CodePage]
"1250"="c_1251.nls"
"1251"="c_1251.nls"
"1252"="c_1251.nls"
"1253"="c_1251.nls"
```

1. Настройка позволяет осуществить отображение подробной системной информации в служебном модуле «Диспетчер устройств»:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manag-
er\Environment]
"DEVMGR_SHOW_DETAILS"=dword:00000001
```

1. Настройка позволяет осуществить очистку файла подкачки при выключении компьютера:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manag-er\Memory
Management]
"ClearPageFileAtShutdown"=dword:00000001
```

1. Настройка позволяет увеличить производительность ОС в целом:

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manag-er\Memory Management]
"DisablePagingExecutive"=dword:00000001

1. Настройка позволяет ускорить загрузку ОС:

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manag-er\MemoryManagement\PrefetchParameters]
"EnablePrefetcher"=dword:00000003

1. Настройка укоряет загрузку веб-страниц:

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\ServiceProvider]
"DnsPriority"=dword:00000001
"HostsPriority"=dword:00000001
"LocalPriority"=dword:00000001
"NetbtPriority"=dword:00000001

1. Настройка позволяет отключить проверку дисков при загрузке:

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager]
"AutoChkTimeOut"=dword:0000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager]
"BootExecute"=hex(7):61,00,75,00,74,00,6f,00,63,00,68,00,65,00,63,00,6b,00,20,\
00,61,00,75,00,74,00,6f,00,63,00,68,00,6b,00,20,00,2a,00,00,00,00,00
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon]
"SFCScan"=dword:00000000
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\MyComputer\cleanuppath]
@=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,00,74,00,25,\
00,5c,00,73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,63,00,6c,00,\
65,00,61,00,6e,00,6d,00,67,00,72,00,2e,00,65,00,78,00,65,00,20,00,2f,00,44,\
00,20,00,25,00,63,00,00,00

1. Настройка позволяет отключить автозапуск для всех типов оптических приводов и устройств в системе:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer]
"NoDriveTypeAutoRun"=dword:000000ff
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Cdrom]
"AutoRun"=dword:00000000
Для изучения результатов применения указанного комплекта системных настроек Реестра ОС выполните следующее.

Содержание задания

- сделайте резервную копию Реестра ОС одним из способов изученных в первой части лабораторной работы,
- последовательно примените к системе не менее пяти выбранных самостоятельно настроек Реестра ОС одним из способов изученных ранее в предыдущих заданиях лабораторной работы,

- перезагрузите ОС Windows XP, чтобы параметры вступили в силу, при этом каждый раз наблюдая за стабильностью работы ОС,
- результаты применения новых значений Реестра перенесите в отчет,
- сделайте вывод о проделанной работе и запишите его в отчет.

Примечание. Если после применения новых значений выбранных параметров ОС начала работать нестабильно, верните системные параметры в начальное состояние, восстановив Реестр ОС из сохраненной резервной копии.

Изученные в настоящей лабораторной работе возможности обеспечивают системного администратора базовым инструментарием тонкой настройки ОС Windows XP. Набор описанных настроек и твиков системного Реестра способствует оптимизации и ускорению работы ОС. Однако, конфигурирование системы посредством указанных настроек это лишь малая часть всех работ, направленных на обеспечение оптимальной и эффективной работы системы. Среди прочего, к числу подобного рода мероприятий следует отнести работы по настройке и оптимизации системных служб ОС Windows XP, в частности, за счет их включения или отключения посредством системного Реестра. В качестве примера ниже приведен текст твика Реестра для конфигурации служб ОС, оптимизированной с целью максимального быстродействия.
В заключение следует отметить, что для более подробного изучения рассмотренных в лабораторной работе вопросов целесообразно обратится к первоисточнику, книге «Реестр Microsoft Windows XP» автора Джерри Хонейкатт, который является признанным специалистом в этой области.
Кроме того, различные возможности настройки системы, твики Реестра ОС, скрипты для их системного применения в достаточном количестве можно найти в глобальной сети Интернет. Одним из сетевых ресурсов является специализированный сайт www.OSzone.net, в основном посвященный вопросам, связанным с операционными системами семейства Windows. На этом сайте опубликованы различные инструкции и рекомендации по конфигурированию ОС, в большом количестве приведены системные настройки и твики, а также специализированные обзоры и статьи по смежным тематикам.

Лабораторная работа № 20.

Тема: Организация пакетных файлов и сценариев в ОС Windows XP

Введение

Пакетный файл – *это неформатированный текстовый файл ASCII, содержащий одну или несколько команд ОС.*
Имена пакетных файлов имеют расширения .**cmd** или .**bat**. **ОС** при работе с пакетным файлом последовательно обрабатывает его команды после ввода его имени в строке командной оболочки или запуска из другой программы.
Сценарий – это программа, состоящая из набора инструкций для работы приложения или служебной утилиты. Сценарий – разновидность пакетного файла. Инструкции в сценариях обычно выражаются с использованием правил и синтаксиса соответствующего приложения или служебной утилиты в сочетании с простыми управляющими операторами, такими как операторы циклов и условные операторы.
Пакетные файлы и сценарии часто называют командными файлами, содержащими любые команды. Некоторые команды, такие как **For, Goto** и **If**, позволяют выполнять обработку условий в пакетных файлах. Другие команды позволяют управлять вводом и выводом, а также запускать другие пакетные файлы.
При организации пакетных файлов и сценариев применяют **переменные**, задающие поведение командной оболочки или ОС и **пакетные параметры** командного интерпретатора, которые используются в пакетном файле для получения информации о настройках среды.
Поведение среды командной оболочки или всей ОС задают с помощью двух типов переменных среды: **системных** и **локальных**.

- **Системные переменные** *определяют поведение глобальной среды ОС*.
- **Локальные переменные** *определяют поведение среды в конкретном экземпляре командного интерпретатора Cmd.exe*.

Системные переменные среды задаются заранее в ОС Windows XP и доступны для всех ее процессов. Только пользователи с привилегиями администратора могут изменять эти переменные.
Локальные переменные среды доступны в случае, когда пользователь, для которого они были созданы, входит в систему. В частности, локальные переменные реестра **HKEY_CURRENT_USER** подходят только для текущего пользователя, но определяют поведение глобальной среды ОС.
В следующем списке представлены различные типы переменных в порядке убывания их приоритета:

1. встроенные системные переменные,
2. системные переменные реестра **HKEY_LOCAL_MACHINE**,
3. локальные переменные реестра **HKEY_CURRENT_USER**,
4. все переменные среды и пути, указанные файле **Autoexec.bat**,
5. все переменные среды и пути, указанные в сценарии входа в систему, если он имеется,
6. переменные, используемые интерактивно в пакетном файле или сценарии.

Чтобы иметь возможность подставить значение в переменную среды из командной строки или в пакетном файле (сценарии), следует заключить имя соответствующей переменной (Приложени. 1) в символы процентов (%), например **Set MyPath=%CD%.** Символы

процентов указывают на то, что командный интерпретатор должен обратиться к значению переменной без посимвольного ее разложения и сравнения.
Командный интерпретатор Cmd.exe может оперировать переменными с %0 по %9. При использовании пакетных параметров переменная %0 заменяется именем пакетного файла, а переменные с %1 по %9 — на соответствующие аргументы командной строки. Для доступа к переменным больше %9 еобходимо воспользоваться командой Shift. Параметр %* ссылается на все аргументы, которые передаются пакетному файлу, за исключением %0.
В качестве примера, рассмотрим копирование содержимого из каталога 1 (*Folder1*) в каталог 2 (*Folder2*), где параметр %1 заменяется значением *Folder1*, а параметр %2 соответственно значением *Folder2*. В пакетном файле **Mybatch.bat** следует ввести следующую строку:
Xcopy %1*.* **%2**
Используйте пакетный файл **Mybatch.bat** следующим образом:
Mybatch.bat *C:\folder1 D:\folder2*
Результат будет таким же, как и при записи в пакетный файл строки:
Xcopy *C:\folder1**.* *D:\folder2*\
С пакетными параметрами можно также использовать модификаторы. **Модификаторы** *используют информацию о текущем диске и каталоге как часть или полное имя файла (каталога)*.
Синтаксис модификатора: %~*xy*, где *x* — символьное сокращение действия, определяемое модификатором, *y* — идентификатор переменной (в диапазоне от 1до 9).
В табл. 1и 2 описаны модификаторы, выполняемые ими действия, и даны возможные комбинации модификаторов и квалификаторов для получения более сложных результатов. В этих таблицах %1 и переменную среды PATH можно заменить другими значениями пакетных параметров.
Таблица 1. Модификаторы и выполняемые ими действия

№ п.п.	Модификатор	Описание
1	%~1	расширение %1 и удаление любых кавычек (" ")
2	%~f1	замена %1 полным путем
3	%~d1	замена %1 именем диска
4	%~p1	замена %1 путем
5	%~n1	замена %1 именем файла
6	%~x1	замена %1 расширением имени файла
7	%~s1	замена путем, содержащим только короткие имена
8	%~a1	Замена %1 атрибутами файла
9	%~t1	замена %1 датой и временем модификации файла
10	%~z1	замена %1 размером файла
11	%~$PATH:1	поиск в каталогах, перечисленных в переменной среды PATH, замена %1 полным именем первого найденного файла. Если переменная среды не определена или поиск не обнаружил файлов, модификатор выдает пустую строку.

Таблица 2. Комбинации модификаторов и квалификаторов

№ п.п.	Модификатор	Описание
1	%~dp1	замена %1 именем диска и путем

2		%~nx1 замена %1 именем файла и расширением
3	%~dp$PATH:1	поиск в каталогах, перечисленных в переменной среды PATH, и замена %1 именем диска и путем к первому найденному файлу.
4	%~ftza1	замена %1 строкой, аналогичной результату работы команды Dir

Еще один модификатор, являющийся уникальным, имеет вид %*. Он представляет все аргументы, переданные пакетному файлу. Этот модификатор не используется в комбинации с модификатором %~.
Конвейеры команд и «каналы», рассмотренные в предыдущих лабораторных работах (Часть I, Приложения 1 и 2) являются инструментами для расширения функционала пакетных файлов и сценариев при их построении и организации.
Сервер сценариев ОС Windows XP *позволяет быстро запустить пакетный файл или сценарий, имя которого введено в командной строке оболочки.*
Сервер сценариев

- служит контроллером средств обработки сценариев в ОС Windows XP;
- не требует много памяти;
- является идеальным средством, как для интерактивных, так и для пакетных сценариев.

Существуют две версии сервера сценариев, доступных в окне командной оболочки:

- **Wscript.exe** — позволяет задавать параметры выполнения сценариев в окне свойств;
- **Cscript.exe** — позволяет задавать параметры выполнения сценариев с помощью ключей командной строки.

Для разработки сценариев **ОС Windows XP** следует использовать редакторы сценариев **JScript** или **VBScript** (в составе **Visual Basic Scripting Edition**). При запуске сценария из командной строки, сервер сценария читает и передает содержимое указанного файла зарегистрированному обработчику сценариев. Для определения языка сценария используется расширение имени файла (.**vbs** для **VBScript**, .**js** для **JScript**). Благодаря этому, разработчик сценария не обязан знать точные программные идентификаторы (**ProgID**) различных обработчиков сценариев. Сопоставление расширения имени файла сценария с программным идентификатором и запуск конкретного обработчика сценариев осуществляется непосредственно сервером сценариев **ОС Windows XP**.
Простейшим сценарием, не требующим применения среды Visual Basic, является сценарий входа в систему, представляющий собой файл, связываемый с одной или несколькими учетными записями пользователей. Обычно сценарий входа является пакетным файлом, который автоматически выполняется при каждом входе пользователя в систему. Сценарии входа используются для настройки рабочей среды пользователя при входе и позволяют администратору задавать основные параметры рабочей среды пользователя без непосредственного его участия.
Поскольку пакетные файлы могут включать в себя любые команды, их конвейеры и «каналы», при большом количестве условий и циклов последствия некорректной работы пакетного файла могут быть непредсказуемыми для ОС, и возможно как следствие разрушительными. Поэтому для организации пакетного файла разработчику необходимо четко представлять себе, что именно и каким образом должно происходить в системе при

работе этого файла, какая последовательность действий реализуется в результате выполнения задуманного сценария и как на эти действия реагирует ОС. Помимо рассмотренных в предыдущих лабораторных работах команд, которые могут быть использованы при организации пакетного файла, существует ряд дополнительных, функционал которых напоминает операторы языков программирования высокого уровня. К их числу относятся: **At, Call, Doskey, Echo, Endlocal, For, Goto, If, Pause, Rem, Set, Setlocal и Shift**.
В настоящей лабораторной работе предполагается ознакомление с основными командами, используемыми в качестве инструментов организации пакетных файлов, создание командного файла в формате ASCII, реализующего определенный сценарий работы системы, а также оценка возможности использования его в качестве сценария входа в систему.
Лабораторная работа выполняется на виртуальной машине в среде ОС Windows XP.

Задание 26. Команда Set

Команда **Set** предназначена для отображения и установки переменных среды в ОС Windows XP.
Синтаксис команды **Set**:
Set [*переменная*=[*строка*]],
где параметр:
переменная — задает имя переменной, значение которой требуется присвоить или изменить.
строка — задает строковое значение для указанной переменной.
Команда SET поддерживает два дополнительных ключа:
SET /A выражение
SET /P variable=[promptString]
Ключ /A указывает, что строка справа от знака равенства является числовым выражением, значение которого вычисляется. Обработчик выражений очень прост и поддерживает следующие операции, перечисленные в порядке убывания приоритета (Табл. 3).
Таблица 3. Операции обработчика выражений команды Set

№ п.п.	Операция	Описание
1	( )	группировка
2	! ~ -	унарные операторы
3	* / % +	арифметические операторы
4	<< >>	двоичный сдвиг
5	&	двоичное И
6	^	двоичное исключающее ИЛИ
7	|	двоичное ИЛИ
8	= *= /= %= += -= &= ^= |= <<= >>=	присвоение
9	,	разделитель операторов

При использовании любых логических или двоичных операторов необходимо заключить строку выражения в кавычки. Любые нечисловые строки в выражении рассматриваются как имена переменных среды, значения которых преобразуются в числовой вид перед использованием. Если переменная с указанным именем не определена в системе, вместо нее подставляется нулевое значение. Это позволяет выполнять арифметические операции

со значениями переменных среды, причем не нужно вводить знаки % для получения значений. Если команда SET /A вызывается из командной строки, а не из пакетного файла, она выводит окончательное значение выражения. Слева от любого оператора присваивания должно стоять имя переменной среды. Числовые значения рассматриваются как десятичные, если перед ними не стоит префикс 0x для шестнадцатеричных чисел, и 0 для восьмеричных чисел. Например, числа 0x12, и 022 обозначают десятичное число 18. Обратите внимание на запись восьмеричных числе: 08 и 09 не являются допустимыми числами, так как в восьмеричной системе исчисления цифры 8 и 9 не используются. Если команда с ключом /a вызывается из командной строки, а не из пакетного файла, она выводит окончательное значение выражения.
Ключ /P позволяет установить значение переменной для входной строки, введенной пользователем. Показывает указанное приглашение promptString перед чтением введенной строки. Приглашение promptString может быть пустым.
Подстановка переменной среды может быть расширена следующим образом:
%PATH:str1=str2%
расширит действие переменной среды PATH, заменяя каждое вхождение "str1" в расширенном результате на "str2". "str2" может быть пустой строкой для эффективного удаления вхождений "str1" из расширенного вывода. "str1" может начинаться со звездочки, и в этом случае это будет соответствовать любому началу расширенного вывода до первого вхождения оставшейся части "str1". Можно также использовать строки расширения.
%PATH:~10,5%
расширит действие переменной среды PATH, затем использует только 5 символов, которые начинаются с 11-го символа (пропустив 10) расширенного результата. Если длина не указана, по умолчанию используется оставшееся значение переменной. Если оба значения (длина и число пропускаемых символов) отрицательны, следующим используемым значением будет длина значения переменной среды, добавленной к указанному значению пропуска или указанной длины.
%PATH:~-10%
извлечет последние 10 символов переменной PATH.
%PATH:~0,-2%
извлечет все символы переменной PATH, за исключением 2-х последних.
Наконец, добавлена поддержка связывания времени выполнения для переменных среды окружения. По умолчанию эта поддержка отключена. Ключ /V командной строки CMD.EXE позволяет включать и выключать ее. Связывание времени выполнения для переменных среды окружения полезно при обходе ограничений раннего связывания, которое происходит при первом чтении текстовой строки, а не при ее выполнении.
Следующий пример демонстрирует возникающую проблему при использовании раннего связывания переменных:
```
set VAR=before
if "%VAR%" == "before" (set VAR=after
if "%VAR%" == "after" @echo Тело внутреннего оператора сравнения)
```
Данное сообщение не будет выводиться, т.к. %VAR% в ОБОИХ выражениях if подставляется в момент первого использования в первом if, в том числе и в тело первого ветвления if, которое является составным выражением. В if внутри составного выражения в действительности сравниваются значения "before" и "after", что заведомо ложно.
Следующий пример демонстрирует подобную ошибку:
```
set LIST=
for %i in (*) do set LIST=%LIST% %i
echo %LIST%
```
в данном случае список файлов текущей папки никогда не будет построен. Вместо этого, значением переменной LIST будет имя последнего найденного файла. И вновь, это

случилось потому, что %LIST% подставляется всего один раз - в момент обработки выражения FOR, когда список еще пуст. Фактически, приведенный фрагмент эквивалентен следующему примеру:
for %i in (*) do set LIST= %i
в котором имя последнего найденного файла сохраняется в переменной LIST.
Связывание времени выполнения для переменных среды окружения происходит при использовании специального символа (восклицательного знака), обозначающего проведение сопоставления во время выполнения. Если включена поддержка связывания времени выполнения, то для достижения ожидаемых результатов приведенные выше фрагменты должны быть изменены следующим образом:
set VAR=before
if "%VAR%" == "before" (set VAR=after
if "!VAR!" == "after" @echo Тело внутреннего оператора сравнения)
set LIST=
for %i in (*) do set LIST=!LIST! %i
echo %LIST%
При включенной расширенной обработке команд доступны несколько переменных среды, которые не отображаются в списке, отображаемом при вызове команды SET. Значения этих переменных вычисляются динамически каждый раз при их извлечении. Если пользователь явно задает переменные с одним из этих имен, то это определение перекрывает соответствующее динамическое определение, описанное ниже:
%CD% - раскрывается в строку текущей директории.
%DATE% - раскрывается в текущую дату, используя тот же формат команды DATE.
%TIME% - раскрывается в текущую дату, используя формат команды TIME.
%RANDOM% - раскрывается в случайное десятичное число между 0 и 32767.
%ERRORLEVEL% - раскрывается в текущее значение ERRORLEVEL
%CMDEXTVERSION% - раскрывается в текущее значение версии расширенной обработки команд.
%CMDCMDLINE% - раскрывается в исходную командную строку, которая вызвала командный процессор.
Таблица 4. Динамические значения команды Set

Значение	Описание действия
	%Cd% раскрывается в строку текущей директории
	%Date% раскрывается в текущую дату
	%Time% раскрывается в текущее время
%Random%	раскрывается в случайное десятичное число в диапазоне от 0 до 32767
%Errorlevel%	раскрывается в текущее значение ErrorLevel
%Cmdextversion%	раскрывается в текущее значение версии расширенной обработки команд
%Cmdcmdline%	раскрывается в исходную командную строку, которая вызвала текущее окно командной оболочки

Содержание задания

1. Отобразите переменные среды двумя способами: из командной оболочки и окна свойств системы (Пуск | Панель управления | Система).
2. Задайте переменную среды, содержащую определенный путь к месту назначения, выбранный самостоятельно.

3. Проверьте наличие в системе переменной среды, заданной в предыдущем пункте задания.
4. Выведите значение выражения, определенного в соответствии с вариантом задания (табл. 8), в качестве переменной среды Result.
5. Задайте переменную среды с различными вариантами динамически формируемых значений (табл. 4). Варианты динамических значений выберете самостоятельно.

При выполнении задания используйте следующие инструкции:

- по каждому из пунктов задания в окне командной оболочки наберите соответствующую команду с необходимыми ключами,
- нажмите **Enter** для ввода,
- изучите полученный результат и сделайте вывод о проделанной работе,
- запишите полученную информацию в отчет, заполнив табл. 5.

Таблица 5. Результаты выполнения команды Set

№ п.п.	Команда с ключами	Результат и вывод по способу применения команды
1		
2		
3		
4		
5		

Задание. Команды Rem, Echo

Команда **Rem** добавляет комментарии в пакетных файлах и не выводит комментарии на экран. При обработке пакетной программы строки, начинающиеся с Rem, пропускаются.
Синтаксис команды **Rem**:
Rem [*текст*],
где параметр:
текст — задает строку символов, используемую в качестве комментария.
Команда **Echo** выводит на экран сообщения или задание режима вывода на экран.
Синтаксис команды **Echo**:
Echo [{**on** | **off**}] [*сообщение*],
где параметр:
{**on** | **off**} — включает или отключает режим отображения на экране информации о работе команд.
сообщение — задает текст для вывода на экран.
Для вывода сообщений из нескольких строк без вывода дополнительных команд между ними следует использовать несколько последовательных команд **Echo** после команды **Echo off**:

```
Cls
@Echo off
Echo.
Rem ****** Эта пакетная программа ******
Rem ***** иллюстрирует возможности *****
Rem ********** команды Echo **********
Echo.
```

Echo ******** This batch program ********
Echo *** illustrates possibilities of ***
Echo ********* the Echo command *********
Echo.
Pause

Содержание задания

1. Создайте пакетный файл, воспользовавшись любым текстовым редактором. Имя пакетного файла выберете самостоятельно.
2. Введите в созданный пакетный файл текст, приведенного выше примера.
3. Сохраните текст пакетного файла.

При выполнении задания используйте следующие инструкции:

- воспользовавшись командой Start и указав путь к пакетному файлу,
- запустите его на выполнение, нажав Enter для ввода,
- изучите пример и полученный с его помощью результат, обратив внимание
- на то, что команда Echo с точкой (.) в конце выводит на экран пустую строку, а символ «коммерческое И» (@) перед командой Echo отключает режим отображения команд.
- сделайте соответствующий вывод и запишите его в отчет.

Задание. Утилита For

Утилита **For** командной среды предназначена для выполнения циклических операций для заданного множества операндов.
Синтаксис команды **For**:
For {*% переменная*| *%%переменная*} in (*множество*) do *команда* [*ПараметрыКоманднойСтроки*]
где параметр:
%переменная - любые символы, кроме цифр 0–9 используется для выполнения команды из строки в окне командной оболочки;
%%переменная - любые символы, используется для выполнения команды в пакетном файле. Переменные должны учитывать регистр.
(*множество*) — обязательный параметр, задающий один или группу файлов, каталогов, диапазон значений или текстовых строк, указывающих на файлы, подлежащие обработке заданной командой. Например, допустимыми являются следующие варианты: (*.doc), (*.doc *.txt *.me), (jan*.doc jan*.rpt feb*.doc feb*.rpt), (ar??1991.* ap??1991.*). Скобки являются обязательными для обозначения множества.
команда — обязательный параметр, задающий команду, которая будет выполнена для каждого операнда множества.
ПараметрыКоманднойСтроки — задает параметры командной строки, которые используются с указанной командой.
Если расширения командного интерпретатора разрешены, что делается по умолчанию, то существуют дополнительные формы команды **For**.
For /D {*% переменная*| *%%переменная*} in (*множество*) do *команда* [*ПараметрыКоманднойСтроки*]
Если множество содержит подстановочные знаки, команда выполняется для всех подходящих имен каталогов, а не имен файлов.
For /R [[*диск* :]*путь*] {*%* | *%%*} *переменная* in (*множество*) do *команда* [*ПараметрыКоманднойСтроки*]

Выполнение команды для каталога [диск:]путь, а также для всех подкаталогов этого пути. Если после ключа /R не указано имя каталога, выполнение команды начинается с текущего каталога. Если вместо набора указана только точка (.), команда выводит список всех подкаталогов.

For /L {% | %%} *переменная* in (*НачальноеЗначение*#,*шаг*#, *КонечноеЗначение*#) do *команда* [*ПараметрыКоманднойСтроки*]

Множество раскрывается в последовательность чисел с заданными началом, концом и шагом приращения. Так, набор (1,1,5) раскрывается в (1 2 3 4 5), а набор (5,-1,1) заменяется на (5 4 3 2 1)

FOR /F ["ключи"] %переменная **IN** (*множество*) **DO** команда [параметры]

FOR /F ["options"] %variable **IN** ("literal string") **DO** command [command-parameters]

FOR /F ["options"] %variable **IN** ('command') **DO** command [command-parameters]

или, если использован параметр **usebackq**:

FOR /F ["options"] %variable **IN** (filenameset) **DO** command [command-parameters]

FOR /F ["options"] %variable **IN** ('literal string') **DO** command [command-parameters]

FOR /F ["options"] %variable **IN** (`command`) **DO** command [command-parameters]

Набор содержит имена одного или нескольких файлов, которые по очереди открываются, читаются и обрабатываются. Обработка состоит в чтении файла, разбивки его на отдельные строки текста и выделения из каждой строки заданного числа подстрок (в том числе нуля). Затем найденная подстрока используется в качестве значения переменной при выполнении основного тела цикла. По умолчанию ключ /F выделяет из каждой строки файла первое слово, очищенное от окружающих его пробелов. Пустые строки в файле пропускаются. Необязательный параметры "ключи" служит для переопределения заданных по умолчанию правил обработки строк. Ключи представляют собой заключенную в кавычки строку, содержащую указанные параметры. Ключевые слова:

eol=c - определение символа комментариев в конце строки (допускается задание только одного символа).

skip=n - число пропускаемых при обработке строк в начале файла.

delims=xxx - определение набора разделителей для замены заданных по умолчанию пробела и знака табуляции.

tokens=x,y,m-n - определение номеров подстрок, выделяемых из каждой строки файла и передаваемых для выполнения в тело цикла. При использовании этого ключа создаются дополнительные переменные. Формат m-n представляет собой диапазон подстрок с номерами от m по n. Если последний символ в строке tokens= является звездочкой, создается дополнительная переменная, значением которой будет весь оставшийся текст в строке после обработки последней подстроки.

usebackq - применение новой семантики, при которой строки, заключенные в обратные кавычки, выполняются как команды, строки, заключенные в прямые одиночные кавычки, являются строкой литералов команды, а строки, заключенные в двойные кавычки, используются для выделения имен файлов в списках имен файлов.

Поясняющий пример:

FOR /F "eol=; tokens=2,3* delims=, " %i in (myfile.txt) do @echo %i %j %k

Эта команда обрабатывает файл myfile.txt, пропускает все строки, которые начинаются с символа точки с запятой, и передает вторую и третью подстроки из каждой строки в тело цикла, причем подстроки разделяются запятыми и/или пробелами. В теле цикла переменная %i используется для второй подстроки, %j - для третьей, а %k получает все оставшиеся подстроки после третьей.

Имена файлов, содержащие пробелы, необходимо заключать в двойные кавычки. Для

того чтобы использовать двойные кавычки, необходимо использовать параметр usebackq, иначе двойные кавычки будут восприняты как границы строки для обработки.

Переменная %i явно описана в инструкции for, а переменные %j и %k описываются неявно с помощью ключа tokens=. Ключ tokens= позволяет извлечь из одной строки файла до 26 подстрок, при этом, не допускается использование переменных больших чем буквы 'z' или 'Z'. Следует помнить, что имена переменных **FOR** являются глобальными, поэтому одновременно не может быть активно более 52 переменных.

Синтаксис команды **FOR /F** также позволяет обработать отдельную строку, с указанием параметра filenameset, заключенным в одиночные кавычки. Строка будет обработана как единая строка из входного файла.

Наконец, команда **FOR /F** позволяет обработать строку вывода другой команды. Для этого следует ввести строку вызова команды в апострофах вместо набора имен файлов в скобках. Строка передается для выполнения обработчику команд CMD.EXE, а вывод этой команды записывается в память и обрабатывается так, как будто строка вывода взята из файла. Например, следующая команда:

FOR /F "usebackq delims==" %i IN (`set`) DO @echo %i

выведет перечень имен всех переменных среды, определенных в настоящее время в системе.

Операции подстановки ссылок на переменные команды FOR также расширены.
Допускается применение следующих синтаксических конструкций:

%~I - из переменной %I удаляются обрамляющие кавычки (")
%~fI - переменная %I расширяется до полного имени файла
%~dI - из переменной %I выделяется только имя диска
%~pI - из переменной %I выделяется только путь к файлу
%~nI - из переменной %I выделяется только имя файла
%~xI - из переменной %I выделяется расширение имени файла
%~sI - полученный путь содержит только короткие имена
%~aI - переменная %I расширяется до атрибутов файла
%~tI - переменная %I расширяется до даты /времени файла
%~zI - переменная %I расширяется до размера файла
%~$PATH:I - проводится поиск по каталогам, заданным в переменной среды PATH, и переменная %I заменяется на полное имя первого найденного файла. Если переменная PATH не определена или в результате поиска не найден ни один файл, то этот модификатор заменяется на пустую строку

При объединении нескольких операторов можно получить следующие результаты:

%~dpI - переменная %I заменяется только на имя диска и путь
%~nxI - переменная %I заменяется только на имя файла и его расширение
%~fsI - переменная %I заменяется только на полный путь с краткими именами
%~dp$PATH:I - проводится поиск по каталогам, заданным в переменной среды PATH, и переменная %I заменяется на имя диска и путь к первому найденному файлу.
%~ftzaI - переменная %I заменяется на строку, выдаваемую командой DIR

В приведенных выше примерах переменные %I и PATH можно заменить на любые другие допустимые значения. В таких синтаксических конструкциях за символами %~ может следовать любое имя переменной цикла FOR.

Рекомендуется использовать заглавные буквы для имен таких переменных, например, %I, что делает их более заметными, а также позволяет отличать их от обозначений модификаторов, которые не различают используемый регистр.

Содержание задания

6. Скопируйте файлы каталога, путь к которому задайте самостоятельно, в точку назначения, заданную путем d:\Temp\. При копировании воспользуйтесь любым методом, изученным ранее.
7. К каждому из файлов, местоположение которых определено путем d:\Temp\, добавьте символ «!» в начале имени, воспользовавшись командой циклической обработки данных.
8. Подсчитать количество каталогов на локальном диске, воспользовавшись командой циклической обработки данных, в процессе выполнения выводя результат в переменную среды, выбранную самостоятельно. Проверьте полученный результат в файловом диспетчере Total Commander (Файл | Подсчитать занимаемое место), предварительно выделив содержимое локального диска.
9. Модифицируйте пакетный файл, полученный в предыдущем задании, воспользовавшись командой циклической обработки данных таким образом, чтобы в процессе его выполнения отображалось определенное количество раз выражение «********* the For command **********».

При выполнении задания используйте следующие инструкции: по каждому из пунктов задания в окне командной оболочки наберите соответствующую команду с необходимыми ключами, нажмите Enter для ввода, изучите полученный результат и сделайте вывод о проделанной работе, запишите полученную информацию в отчет, заполнив табл. 6.
Таблица 6. Результаты выполнения команды For

№ п.п.	Команда с ключами	Результат и вывод по способу применения команды
1		
2		
3		
4		
5		

Задание. Команда If

Обработка условий в пакетных файлах осуществляется командой **If**.
IF [NOT] ERRORLEVEL *число команда*
IF [NOT] *строка1==строка2 команда*
IF [NOT] EXIST *имя_файлакоманда*
NOT Обращает истинность условия: истинное условие становится ложным, а ложное - истинным.
ERRORLEVEL *число* Условие является истинным, если код возврата последней выполненной программы не меньше указанного числа.
строка1==строка2 Это условие является истинным, если указанные строки совпадают.
EXIST *имя_файла* Это условие является истинным, если файл с указанным именем существует.
команда Задает команду, выполняемую при истинности условия. За этой командой может следовать ключевое слово ELSE, служащее для указания команды, которая должна выполняться в том случае, если условие ложно.
Предложение **ELSE** должно располагаться в той же строке, что и команда, следующая за ключевым словом **IF**. Например:
I**F EXIST** *имя_файла* (**del** *имя_файла*) **ELSE** (**echo** *имя_файла* missing.)
Следующий пример содержит ОШИБКУ, поскольку команда **del** должна заканчиваться переходом на новую строку:

IF EXIST *имя_файла* **del** *имя_файла* **ELSE echo** *имя_файла* **missing**
Следующий пример также содержит ОШИБКУ, поскольку команда **ELSE** должна располагаться в той же строке, что и команда, следующая за **IF**:
IF EXIST *имя_файла* **del** *имя_файла*
ELSE echo *имя_файла* **missing**
Вот правильный пример, где все команды расположены в одной строке:
IF EXIST *имя_файла* (**del** *имя_файла*) **ELSE echo** *имя_файла* missing
Изменение команды **IF** при включении расширенной обработки команд:
IF [**/I**] *строка1 оператор_сравнения строка2 команда*
IF CMDEXTVERSION *число команда*
IF DEFINED *переменная команда*
где *оператор_сравнения* принимает следующие значения:
Таблица 7. Допустимые значения *оп_сравнения*

Оператор	Описание	Оператор	Описание
EQU	Равно	LEQ	меньше или равно
NEQ	не равно	GTR	больше
LSS	Меньше	GEQ	больше или равно

а ключ **/I**, если он указан, задает сравнение текстовых строк без учета регистра. Ключ **/I** можно также использовать и в форме *строка1==строка2* команды **IF**. Сравнения проводятся по общему типу данных, так что если строки 1 и 2 содержат только цифры, то обе строки преобразуются в числа, после чего выполняется сравнение чисел.
Условие **CMDEXTVERSION** применяется подобно условию **ERRORLEVEL**, но значение сравнивается с внутренним номером версии текущей реализации расширенной обработки команд. Первая версия имеет номер 1. Номер версии будет увеличиваться на единицу при каждом добавлении существенных возможностей расширенной обработки команд. Если расширенная обработка команд отключена, условие **CMDEXTVERSION** никогда не бывает истинно.
Условие **DEFINED** применяется подобно условию **EXISTS**, но принимает в качестве аргумента имя переменной среды и возвращает истинное значение, если эта переменная определена.
Строка %**ERRORLEVEL**% будет развернута в строковое представление текущего значения кода ошибки ERRORLEVEL, за исключением ситуации, когда уже имеется переменная среды с именем ERRORLEVEL; в подобном случае подставляется значение этой переменной. Например, с помощью данной строки можно выполнить следующее:
goto answer%**ERRORLEVEL**%
:**answer0**
echo Получен код возврата 0
:**answer1**
echo Получен код возврата 1
Допускается и применение описанных выше операторов числового сравнения:
IF %ERRORLEVEL% LEQ 1 **goto okay**
Строка %**CMDCMDLINE**% будет развернута в исходную командную строку, переданную **CMD.EXE** до любой обработки, за исключением ситуации, когда уже определена переменная среды с именем **CMDCMDLINE,** в подобном случае подставляется значение этой переменной.
Строка %**CMDEXTVERSION**% будет развернута в строку, представляющую собой текущее значение **CMDEXTVERSION**, за исключением ситуации, когда уже имеется переменная среды с именем **CMDEXTVERSION**; в подобном случае подставляется значение этой переменной.

Содержание задания

Модифицируйте пакетный файл, полученный в предыдущем задании таким образом, чтобы выполнялись следующие условия:

10. Если не существует каталог d:\Temp\MyFont\, создайте его любым способом, изученным ранее. В противном случае выведите сообщение «Folder exists» (Каталог существует).
11. Если в каталоге d:\Temp\MyFont\ не существует файлов-шрифтов, скопируйте любые три одним из методов, изученных ранее, из системного каталога c:\Windows\Fonts\. В противном случае выведите сообщение «Fonts exist» (Шрифты присутствуют).
12. Если в каталоге d:\Temp\MyFont\ существует файлы, удалите каталог вместе с его содержимым, изученным ранее способом и выведите сообщение «Folder deleted». В противном случае выведите сообщение «Folder is empty. Deleting is senseless» (Каталог пуст. Удаление бессмысленно).

При выполнении задания используйте следующие инструкции:

- по каждому из пунктов задания в командном файле наберите соответствующий код из команд с необходимыми ключами,
- сохраните модифицированный пакетный файл, воспользовавшись командой Start и указав путь к пакетному файлу,
- запустите его на выполнение, нажав Enter для ввода,
- изучите полученный результат и сделайте вывод о проделанной работе,
- запишите полученную информацию в отчет, заполнив табл. 8.

Таблица 8. Результаты выполнения команды If

№ п.п.	Команда с ключами	Результат и вывод по способу применения команды
1		
2		
3		

Задание. Команда Goto

Команда Goto, эквивалентная безусловному переходу в языке высокого уровня, в пакетной программе реализует передачу правления ОС Windows XP в строку, определенную символьной меткой. Когда метка найдена, выполнение продолжается со следующей за ней строки. Команда Goto неразрывно связана с описанной выше командой If, в совокупности обеспечивающей возможность сложных программных конструкций с переходами.
Синтаксис команды Goto:
Goto *метка*,
где параметр:
метка это строка в пакетной программе, на которую выполняется переход. В пакетном файле она должна начинаться с двоеточия (:). Если строка начинается с двоеточия, все присутствующие в ней команды обработаны не будут. Синтаксически она может включать пробелы, но не может включать другие разделители, такие как точка с запятой

или знак равенства. При этом используются только первые восемь знаков метки (метки: hithere0, hithere01 и hithere02 воспринимаются интерпретатором команд как эквивалентные).
Если расширения командного интерпретатора разрешены и в команде Goto используется метка :Eof, управление будет передано в конец текущего пакетного файла для выхода из него без назначения метки. В синтаксис команды обязательно должно быть включено двоеточие (:), то есть Goto :Eof.
Дополнительная информация по данной команде, а также примеры ее использования доступны в справке ОС Windows XP (Пуск | Справка и поддержка) в соответствующем разделе. Справку также можно получить, набрав в окне командной оболочки строку Goto /? и нажав Enter для ввода.

Содержание задания

18. Модифицируйте существующий пакетный файл, введя в него следующий текст:

```
Pause
Echo.
Format A:
If not Errorlevel 1 Goto End
Echo.
Echo *** Error of formatting ***
Rem *** Ошибка форматирования ***
:End
Echo.
Echo *** The end of batch program ***
Rem *** Конец пакетной программы ***
Echo.
Pause
```

19. Сохраните текст пакетного файла.

При выполнении задания используйте следующие инструкции: воспользовавшись командой **Start** и указав путь к пакетному файлу,

- запустите его на выполнение, нажав **Enter** для ввода,
- изучите пример и полученный с его помощью результат,
- сделайте вывод о проделанной работе и запишите его в отчет.

Задание 31. Команда Call

Вызов одного пакетного файла из другого без завершения его выполнения осуществляется командой Call. Эта команда эквивалентна вызову процедуры из основного тела программы. Она принимает метки в качестве объекта вызова и используется только в сценариях или пакетных файлах; при вызове из командной строки команда Call игнорируется.
Синтаксис команды Call:
Call [[*диск*:][*путь*] *имя_файла* [*пакетные_параметры*]] [:*метка* [*аргументы*]],
где параметр:
[*диск*:][*путь*] *имя_файла* — задает имя и расположение пакетного файла.
пакетные_параметры — задает данные командной строки, используемые программой пакетной обработки, включая параметры командной строки, имена файлов, пакетные параметры (%0-%9) или переменные (например, %baud%).

:*метка* — указывает метку, на которую должно быть передано управление программы пакетной обработки. При использовании с этим параметром создается новый контекст пакетного файла, а управление передается инструкции, следующей за указанной меткой.
аргументы — задает данные командной строки, которые передаются в новую программу пакетной обработки, начинающуюся с :метки, включая параметры командной строки, имена файлов, пакетные параметры или переменные.
Необходимо отметить, что при использовании команды Call символы перенаправления ввода-вывода и «каналы» не допускаются. Кроме того, может быть реализована подстановка переменных на основе модификаторов. При этом разрешаются все варианты синтаксических конструкций, приведенных в табл. 1и 2.
Дополнительная информация по данной команде, а также примеры ее использования доступны в справке ОС Windows XP (Пуск | Справка и поддержка) в соответствующем разделе. Справку также можно получить, набрав в окне командной оболочки строку Call /? и нажав Enter для ввода.

Содержание задания

23. Создайте новый (дочерний) пакетный файл, воспользовавшись любым текстовым редактором. Имя пакетного файла выберите самостоятельно.
24. Введите в дочерний пакетный файл процедуру форматирования гибкого диска, учитывающую переход в начало процедуры в случае ошибки, из приведенного выше примера.
25. Модифицируйте родительский пакетный файл, удалив из него лишние команды и добавив ссылку на дочерний пакетный файл для его вызова.
26. Сохраните тексты обоих пакетных файлов.

При выполнении пунктов 1-4 задания используйте следующие инструкции:

- воспользовавшись командой Start и указав путь к родительскому файлу, запустите его на выполнение, нажав Enter для ввода,
- изучите полученный результат и сделайте вывод о проделанной работе,
- запишите полученную информацию в отчет.

5. Вспомните команду форматирования **Format** и ее параметры.
6. Модифицируйте родительский и дочерний файлы таким образом, чтобы осуществилась передача из родительского файла двух значений параметров (%переменная) команды Format (табл. 9), находящейся внутри дочернего файла. Обратите внимание на то, что в таблице 9 передаваемые параметры команды Format имеют числовое (%0-%9), а не символьное представление.
7. Сохраните тексты обоих пакетных файлов.

При выполнении пунктов 5-7 задания используйте следующие инструкции:

- воспользовавшись командой Start и указав путь к родительскому файлу с параметрами для команды Format, запустите его на выполнение, нажав Enter для ввода,
- изучите полученный результат и сделайте вывод о проделанной работе,
- перенесите тексты модифицированных пакетных файлов, а также значения используемых пакетных параметров в отчет.

Варианты для заданий 26, 31.

Вар. №	Задание 26.				Задание 31.	
	Переменная среды				Параметры команды Format	
	a	b	c	Result	%1	%2
1	24	11	35	(a + b – c) * 10	/v:System	/a:512
2	AA	01	C1	a * 5 – b / 5 + c	/a:512	/q
3	12	33	10	a * 4 / c – b * 2	/v:IICT	/a:1024
4	25	A3	B4	a – b * 3 – c / 2	/a:1024	/q
5	49	02	65	a – b * b + c / 5	/v:VMgroup	/a:2048
6	21	99	12	(b *(a + c)) / 3	./a:2048	/q
7	BC	BC	CB	10 + (a – b) * c	/v:MyCore	/a:4096
8	01	94	04	(b – c) / (a * 9)	/a:4096	/q
9	84	D2	2A	a / 10 – (b * c)	/v:Useless	/a:8192
10	10	39	92	a * a + b – c / 2	/v:MyDisk	/q
11	D1	CC	1C	a * b * c – b / c	/v:Temp	/q
12	FF	00	F1	(a – c) * b + 25	/q	/v:Apps
13	CA	DA	FA	a * b * (FA – c)	/v:Double	/q
14	45	78	87	((c – b) *a) / 8	/q	/v:HomeUse
15	88	88	00	(a – b) / (1 – c)	/v:MyDocs	/q
16	75	93	02	(b + c) * 2 –a	/q	/v:VMComp
17	A1	CD	0E	a*a*b*b*c*c	/v:MyList	/a:512
18	C4	EA	E3	(a + b + c) * a	/a:512	/q
19	44	55	33	c *(b – a) / 10	/v:Admin	/a:1024
20	B3	E5	DE	(a * b * c) / 25	/a:1024	/q
21	BB	ED	AE	(a – b) * 2 – c	/v:SysCore	/a:2048
22	55	56	31	(b – a) * 31 – c	/a:2048	/q
23	D1	EC	EE	(c – b) * 5 / a	/v:Kernel	/a:4096
24	D6	E6	FE	(a + b) * c – 11	/a:4096	/q
25	71	65	32	(a – b) *c / 32	/v:User	/a:8192
26	84	32	10	(a * (b + c)) / 5	/a:8192	/q

Таблица 9. Варианты для заданий 1 и 6.

Задание 32. Команды Setlocal, Endlocal

Следующие две команды и последние из основного набора, предназначены для задания начала и конца области определения локальных переменных среды в пакетном файле. Изменения среды являются локальными для пакетного файла и задаются командой **Setlocal**. Локальное окружение используется до тех пор, пока не встретится команда **Endlocal** или не будет достигнут конец пакетного файла (Eof), при этом командный интерпретатор восстанавливает первоначальные параметры.
Синтаксис команд **Setlocal** и **Endlocal**:
Setlocal {enableextension | disableextensions} {enabledelayedexpansion | disabledelayedexpansion},
:
Endlocal,
где параметр:

enableextension — включает расширения командного интерпретатора до появления соответствующей команды **Endlocal**, вне зависимости от состояния расширений командного интерпретатора перед командой **Setlocal**.
disableextensions — выключает расширения командного интерпретатора до появления соответствующей команды **Endlocal**, вне зависимости от состояния расширений командного интерпретатора перед командой **Setlocal**.
enabledelayedexpansion — включает расширения переменной среды с задержкой до появления соответствующей команды **Endlocal**, вне зависимости от состояния расширений командного интерпретатора перед командой **Setlocal**.
disabledelayedexpansion — выключает расширения переменных среды с задержкой до появления соответствующей команды **Endlocal**, вне зависимости от состояния расширений командного интерпретатора перед командой **Setlocal**.
Вертикальное двоеточие (:) иллюстрирует последовательность определенных команд, расположенных между **Setlocal** и **Endlocal**, ограничивающие область локальных переменных среды. Кроме того, что допускается использование нескольких вложенных пар команд **Setlocal** и **Endlocal**.
Дополнительная информация по данным командам, а также примеры их использования доступны в справке ОС Windows XP (Пуск | Справка и поддержка) в соответствующих разделах. Справку также можно получить, набрав в окне командной оболочки строку Setlocal /? или Endlocal /? и нажав Enter для ввода.

Содержание задания

5. Модифицируйте существующий пакетный файл, введя в него следующий текст, иллюстрирующий локальное изменение переменных среды:
   ```
   @Echo off
   Echo.
   Echo *** Local changing the environment variables ***
   Rem *** Локальное изменение переменных среды ***
   Setlocal
   Path=c:\Windows\system32\help;%path%
   Call help>c:\help.out
   Endlocal
   Start notepad c:\help.out
   Pause
   ```
6. Сохраните текст пакетного файла.

При выполнении задания используйте следующие инструкции:

- воспользовавшись командой Start и указав путь к пакетному файлу, запустите его на выполнение, нажав Enter для ввода,
- изучите пример и полученный с его помощью результат,
- сделайте вывод о проделанной работе и запишите его в отчет.

Задание. Создание пакетного файла, реализующего определенную последовательность действий в ОС Windows XP

Порядок выполнения:
I. Создайте новый пакетный файл, воспользовавшись любым текстовым редактором. Имя пакетного файла выберете самостоятельно.
II. Изучите выбранный вариант задания табл. 10.
III. Синтезируйте алгоритм работы пакетного файла.

IV. Выберите необходимый набор команд для реализации алгоритма.
V. С помощью выбранного набора команд запрограммируйте сценарий в виде пакетного файла, реализующего определенную последовательность действий в среде ОС Windows XP.
VI. Сохраните текст пакетного файла.
При выполнении задания используйте следующие инструкции:

- воспользовавшись командой Start и указав путь к пакетному файлу, запустите его на выполнение, нажав Enter для ввода,
- изучите полученный результат,
- перенесите алгоритм, блок-схему и текст разработанного пакетного файла в отчет.
- сделайте вывод о проделанной работе и запишите его в отчет.

Дополнительную информацию по возможностям командной оболочки, а также все множество команд доступных при работе с ней наряду с параметрами и примерами применения можно получить в справке ОС Windows XP (Пуск | Справка и поддержка) в разделах «Общие сведения о командной оболочке», «Справочник по параметрам командной строки» и «Новые средства командной строки».

Варианты для задания.

Таблица 10. Варианты для задания 33.

Задание 33	
Вар. №	**Описание пакетного файла**
1	Пакетный файл, предназначенный для резервного копирования файлов с определенным расширением из разных каталогов с возможностью создания резервного каталога, в случае его отсутствия в системе. Расширение файлов для копирования задается в качестве пакетного параметра. Резервное копирование осуществляется каждый четверг в 22:00. В течение 3 минут после копирования выводится сообщение «Резервное копирование в каталог <путь> завершено» и далее происходит автоматическое выключение системы с принудительным закрытием всех работающих приложений.
2	Пакетный файл, предназначенный для организации процесса поиска и отображения текстового файла. Поиск осуществляется по всем локальным дискам. Имя текстового файла задается пакетным параметром. После того как необходимый файл найден, в автоматическом режиме осуществляется его отображение в текстовом процессоре «Блокнот».
3	Пакетный файл, предназначенный для копирования каталога с его содержимым в заданное место назначения. Копируемый каталог и место назначения задаются в качестве пакетных параметров. После копирования каталога файл-отчет, содержащий информацию о количестве скопированных файлов и их месте расположения, в автоматическом режиме загружается в текстовый процессор «Блокнот».
4	Пакетный файл, предназначенный для перемещения каталога с его содержимым в заданное место назначения с запросом на удаление, перемещаемого каталога. Перемещаемый каталог и место назначения задаются в качестве пакетных параметров. После перемещения каталога в отдельный файл выводится отчет, содержащий два дерева каталогов тех

	мест, откуда и куда было осуществлено перемещение. В конце выводится сообщение вида «Отчет о перемещении находится в каталоге <путь>».
5	Пакетный файл, предназначенный для копирования каталога и включенных в него файлов, расположенных в месте, заданном определенным путем. Полный путь расположения и маска копируемых файлов задаются в качестве пакетных параметров. Если в результирующем каталоге уже находятся копирующиеся файлы, то повторное копирование должно сопровождаться выдачей предупреждающего сообщения о существовании файлов. В конце выводится сообщение вида «Копирование файлов из каталога <путь> в каталог <путь> завершено».
6	Пакетный файл, предназначенный для создания отчета, содержащего «Software part» (программная часть), включающую информацию о содержимом корневых каталогов всех логических дисков в системе и «Hardware part» (аппаратная часть), включающую сведения о конфигурации компьютера и ОС, сведения о безопасности, параметры оборудования, такие как ОЗУ, дисковое пространство, сетевые карты и другие. Файл-отчет копируется в некоторый сетевой каталог, задаваемый пакетным параметром, под именем, отражающим имя компьютера, с которого получен этот отчет. В конце выводится сообщение вида «Отчет находится в сетевом каталоге <путь>».
7	Пакетный файл, предназначенный для углубленной проверки жесткого диска с созданием файла отчета, путь к которому задается в качестве пакетного параметра. Проверка жесткого диска осуществляется ежедневно в 21:00. В течение 20 секунд по окончании проверки диска выводится сообщение «Проверка диска завершена. Файл-отчет находится в каталоге <путь >» и далее осуществляться автоматическая перезагрузка системы.
8	Пакетный файл, предназначенный для резервного копирования файлов системной папки Windows с возможностью создания резервного каталога, в случае его отсутствия в системе. Путь к резервному каталогу задается в качестве пакетного параметра. Резервное копирование осуществляется ежедневно в 23:00. В течение 2 минут после копирования выводится «Резервное копирование в каталог <путь> завершено» и далее происходит автоматическое выключение системы с принудительным закрытием всех работающих приложений.
9	Пакетный файл, предназначенный для архивирования и шифрования указанного каталога с его содержимым. Архивируется каталог-источник с помощью существующего в системе архиватора (например, WinRar), вызов которого осуществляется непосредственно из пакетного файла. Архив в дальнейшем шифруется и сохраняется в определенном месте на жестком диске. Пути к каталогу-источнику и месту назначения задаются в качестве пакетных параметров. В конце выводится сообщение вида «Шифрованный архив сохранен в каталог <путь>».
10	Пакетный файл, предназначенный для удаления файлов по маске, расположенных в месте, заданном определенным путем. Полный путь расположения и маска удаляемых файлов задаются в качестве пакетных параметров. В процессе необходимо осуществлять запрос на подтверждение удаления. В конце выводится сообщение вида «Стерто файлов: <количество> из каталога <путь>».
11	Пакетный файл, предназначенный для организации процесса поиска и сравнения оригинального и резервной копии (.bak) одного и того же файла. Если оригинальный файл найден, то осуществляется его сравнение с

	резервной копией. Отличия, найденные при сравнении, передаются в отчет, который сохраняется в определенном месте. Имя резервной копии файла и путь к месту назначения, где сохраняется отчет о сравнении, задаются в качестве пакетных параметров. В конце выводится сообщение вида «Отчет сохранен в каталог <путь>».
12	Пакетный файл, предназначенный для создания отчета, содержащего «Software part» (программная часть), включающую информацию об присутствующих в системе загруженных драйверах и «Hardware part» (аппаратная часть), включающую сведения о конфигурации компьютера и ОС, сведения о безопасности, параметры оборудования, такие как ОЗУ, дисковое пространство и другие. Файл-отчет копируется в некоторый сетевой каталог, задаваемый пакетным параметром, под именем, отражающим IP-адрес компьютера, с которого получен этот отчет. В конце выводится сообщение вида «Отчет находится в каталоге <путь>».
13	Пакетный файл, предназначенный для подсчета файлов в каталоге, заданном определенным путем. Полный путь расположения и расширение подсчитываемых файлов задаются в качестве пакетных параметров. Организовать отчет с возможностью дописывания в него информации вида «Каталог <путь> содержит <количество> файлов с <расширение> расширением»
14	Пакетный файл, предназначенный для резервного копирования каталога, заданного определенным путем, и содержащихся в нем файлов с возможностью создания резервного каталога, в случае его отсутствия в системе. Пути к каталогу-источнику и месту назначения задаются в качестве пакетных параметров. Резервное копирование осуществляется ежедневно в 23:59. В течение 5 секунд после копирования выводится сообщение «Резервное копирование в каталог <путь> завершено» и далее происходит автоматическая перезагрузка системы.
15	Пакетный файл, предназначенный для отражения статистики по атрибутам файлов в каталоге, заданном определенным путем. В каталоге файлы с определенным атрибутом подсчитываются, а их количество передается в текстовый файл статистики с дописыванием в него информации вида «Файлов с атрибутом <атрибут>: <количество>». Пути к каталогу и месту назначения, где сохраняется файл статистики, задаются в качестве пакетных параметров. В конце выводится сообщение вида «Отчет сохранен в каталог <путь>».
16	Пакетный файл, предназначенный для копирования файлов с определенным расширением и путем в заданное место назначения с их последующим шифрованием. Расширение файлов для копирования, полный путь расположения и путь к месту назначения задаются в качестве пакетных параметров. В конце выводится сообщение вида «Копирование файлов из каталога <путь> в каталог <путь> завершено. Шифрование скопированных данных завершено».
17	Пакетный файл, предназначенный для организации процесса поиска и сравнения оригинального и резервной копии (.bak) одного и того же файла. Имя резервной копии файла передается в качестве пакетного параметра. Если оригинальный файл найден, то осуществляется его сравнение с резервной копией. Отличия, найденные при сравнении, передаются в отчет, который, в автоматическом режиме загружается в текстовый процессор «Блокнот».
18	Пакетный файл, предназначенный для создания отчета, включающего

	информацию о присутствующих в системе загруженных драйверах и отображающего список приложений и служб, выполняющихся на компьютере. Файл-отчет копируется в некоторый сетевой каталог, задаваемый пакетным параметром, под именем, отражающим MAC-адрес компьютера, с которого получен этот отчет. В конце выводится сообщение вида «Отчет находится в сетевом каталоге <путь>».
19	Пакетный файл, предназначенный для копирования файлов, определяемых маской и путем, в заданное место назначения с их последующим архивированием. Архивирование осуществляется с помощью доступного в системе архиватора (например, WinRar), вызов которого осуществляется непосредственно из пакетного файла. Маска файлов для копирования, полный путь расположения и путь к месту назначения задаются в качестве пакетных параметров. В конце выводится сообщение вида «Копирование файлов из каталога <путь> в каталог <путь> завершено. Архивирование скопированных данных завершено».
20	Пакетный файл, предназначенный для удаления файлов с определенным расширением, расположенных в месте, заданном определенным путем. Полный путь расположения и расширение удаляемых файлов задаются в качестве пакетных параметров. В процессе необходимо осуществлять запрос на подтверждение удаления. После удаления в отдельный файл выводится отчет, содержащий список удаленных файлов с их полным путем, который, в свою очередь, в автоматическом режиме загружается в текстовый процессор «Блокнот».
21	Пакетный файл, предназначенный для создания отчета событий и их свойств из журнала событий на локальном компьютере. Журналы событий: Application, System и Security. Файл-отчет копируется в некоторый сетевой каталог, задаваемый пакетным параметром, под именем, отражающим IP-адрес компьютера, с которого получен этот отчет. В конце выводится сообщение вида «Отчет находится в сетевом каталоге <путь>». Примечание: если необходимо, смените сервер сценариев ОС Windows XP.
22	Пакетный файл, предназначенный для перемещения файлов с определенным расширением из каталога-источника в заданное место назначения. Расширение файлов, каталог-источник и место назначения задаются в качестве пакетных параметров. После перемещения отчет, содержащий список перемещенных файлов с путем, загружается в текстовый процессор «Блокнот».
23	Пакетный файл, предназначенный для резервного копирования каталога, заданного определенным путем, и содержащихся в нем файлов с возможностью создания резервного каталога, в случае его отсутствия в системе. Пути к каталогу-источнику и месту назначения задаются в качестве пакетных параметров. Резервное копирование осуществляется ежедневно в 23:30. В течение 10 секунд после копирования выводится сообщение «Резервное копирование в каталог <путь> завершено» и далее происходит автоматическая перезагрузка системы.
24	Пакетный файл, предназначенный для копирования дерева каталогов из каталога-источника в заданное место назначения. Копируемый каталог-источник и место назначения задаются в качестве пакетных параметров. После копирования отчет, содержащий дерево каталогов с местом его расположения, в автоматическом режиме загружается в текстовый процессор «Блокнот».
25	Пакетный файл, предназначенный для организации процесса поиска и

	копирования файлов с определенным расширением. Поиск осуществляется по всем локальным дискам. Расширение файлов и место назначения, куда необходимо копировать файлы, задаются в качестве пакетных параметров. В конце выводится сообщение вида «Файлы найдены и скопированы в каталог <путь>».
26	Пакетный файл, предназначенный для копирования системных и скрытых файлов из каталога-источника в заданное место назначения. Каталог-источник и место назначения задаются в качестве пакетных параметров. После копирования отчет, содержащий список скопированных файлов с путем, в автоматическом режиме загружается в текстовый процессор «Блокнот».

Лабораторная работа №21.

Тема: Организация консоли администрирования в ОС Windows XP

Введение

Консоль управления Microsoft Management Console (MMC) *группирует средства администрирования, которые используются для администрирования сетей, компьютеров, служб и других системных компонентов.*
Консоль MMC непосредственно не выполняет административные функции, однако предоставляет возможности интеграции в нее компонентов или системных приложений, выполняющие эти функции. Основной тип интегрируемых на консоль компонентов называется оснасткой, которые не могут выполняться отдельно без консоли. Среди других добавляемых элементов могут быть элементы управления ActiveX, ссылки на Web-страницы, папки, видов панели задач и собственно задачи для выполнения.
Дополнительные теоретические сведения об оснастках и других используемых для интеграции на консоль элементах будут добавлены в дальнейшем, в соответствующих разделах настоящей лабораторной работы.
Базовое окно консоли MMC представляет собой графическую форму с контекстными меню, реализующие дружественный пользовательский интерфейс. Имеется панель инструментов с командами создания, открытия и сохранения консолей и, кроме того, область описания и строка состояния в нижней части окна. Чтобы увидеть базовое окно, а также непосредственно саму консоль MMC, необходимо выполнить следующие действия:

- нажмите Пуск | Выполнить,
- наберите в появившемся окне MMC.exe (или просто mmc),
- нажмите Enter для ввода.

Новая консоль MMC представляет собой отдельное окно, разделенное на две вертикальные области, в левой из которых отображается дерево консоли с его корнем. Дерево консоли показывает доступные элементы и компоненты консоли. Правая область является областью сведений, которая содержит описания элементов и выполняемых ими функций. Содержание области сведений соответствует выбранному элементу в дереве консоли и может включать Web-страницы, графики, диаграммы, таблицы и столбцы.
Создавая надежные средства управления компьютерами сети, можно собрать и настроить

собственную консоль MMC, выполняющую заданные функции администрирования. После того как добавлены все необходимые элементы и компоненты консоли, панель главного меню, панель инструментов, а также область описания и строка состояния могут быть скрыты для предотвращения в дальнейшем нежелательных изменений. Созданные таким образом управляющие системы сохраняются в файлах с расширением .msc (Management Saved Console, сохраненная консоль управления) и могут быть, в частности, распространены в пределах всей системы посредством задания к ним доступа с помощью ярлыков или элементов меню Пуск.
Чтобы увидеть консоль управления локальным компьютером в качестве примера готовой и отлаженной консоли MMC, необходимо выполнить:

- нажмите Пуск | Выполнить,
- наберите в появившемся окне compmgmt.msc (или compmgmt),
- нажмите Enter для ввода.

Существует два основных режима доступа консоли администрирования, задающиеся непосредственно при ее создании:

- пользовательский, в котором можно администрировать систему, работая с уже существующими консолями,
- авторский, в котором можно создавать новые консоли или изменять существующие.

В свою очередь, имеется три уровня режима пользователя, что обусловливает всего четыре варианта предустановленного режима доступа:

- авторский режим;
- режим пользователя — полный доступ;
- режим пользователя — ограниченный доступ, многооконный;
- режим пользователя — ограниченный доступ, однооконный.

Консоль MMC, инициализированная в авторском режиме, предоставляет полный доступ ко всем ее возможностям, включая добавление и удаление оснасток, создание новых окон и панелей задач, а также просмотр любых частей дерева консоли и другие. Однако при выборе одного из трех режимов пользователя авторские возможности исключаются. В частности, если для консоли установлен параметр «пользовательский режим — полный доступ», то предоставляются все команды управления окном консоли и полный доступ к ее дереву, но запрещается добавление, удаление оснасток и изменение свойств консоли администрирования.
Изменения консоли MMC в авторском и пользовательском режимах сохраняются по-разному. При закрытии консоли в авторском режиме выводится диалоговое окно с предложением сохранить изменения. Однако в пользовательском режиме и снятом флажке «Не сохранять изменения для этой консоли» изменения будут сохранены автоматически при закрытии.
Если консоль открыта при соблюдении одного из следующих условий:

- в базовом окне при загрузке,
- с помощью команды контекстного меню Автор,
- в командной строке с параметром /a,

то предустановленный режим игнорируется, а открытие консоли осуществляется в авторском режиме.

Очевидно, что загрузка консоли ММС в авторском режиме не требуется рядовым пользователям. Системный администратор может настроить профили пользователей так, чтобы запретить им переход в авторский режим, как из командной строки, так и через контекстное меню. Кроме того, запрет перехода в авторский режим может быть организован при использовании возможностей групповой политики, при которой, в частности, осуществляется ограничение доступа к определенным оснасткам. Рассмотрению базовых возможностей оснастки групповой политики будет посвящена вторая часть настоящей лабораторной работы.
Прежде чем создавать новую консоль ММС, необходимо определить действия, для которых предназначена эта консоль, список администрируемых компонентов, оснасток и других элементов, которые потребуются для выполнения поставленных задач. Следует также рассмотреть необходимость создания видов панели задач. После принятия этих решений можно открыть новую консоль и начать добавлять элементы к дереву консоли. Полное руководство по созданию и настройке консолей ММС находится на Web-узле корпорации Майкрософт (http://www.microsoft.com).
В лабораторной работе предполагается ознакомление с основными принципами организации и построения консоли администрирования ММС, а также с базовыми возможностями основных инструментов системного администратора — оснасток «Локальные пользователи и группы» и «Редактор объекта групповой политики» («Групповая политика»).
Перед началом выполнения заданий в среде ОС Windows XP необходимо выполнить следующее:

1. запустить виртуальную машину с ОС Windows XP и активировать справочное меню (Пуск | Справка и поддержка);
2. ознакомиться с описанием и возможностями запуска и применения консоли администрирования ММС;
3. ознакомиться возможностью получения сведений пункта 2 из альтернативного источника информации, доступного непосредственно в справке консоли администрирования ММС (Справка | Вызов справки);
4. ознакомиться с описанием и возможностями оснасток «Локальные пользователи и группы» и «Редактор объекта групповой политики» («Групповая политика»).

Задание 35. Изменение параметров и способов настройки консоли администрирования ММС

Порядок выполнения:
I. Создание консоли администрирования ММС в авторском режиме требует выполнения следующих действий:

- нажмите Пуск | Выполнить,
- наберите в появившемся окне MMC.exe (или просто mmc),
- нажмите Enter для ввода.

Возможны следующие альтернативные варианты авторского запуска консоли администрирования:
А. запуск из командной строки, используя синтаксис:
Mmc *путь\имя_файла*.**msc /a**,
где параметр:
путь\имя_файла.**msc** — запускает консоль ММС с одновременным открытием файла сохраненной консоли с именем *имя_файла*.**msc** (Приложение 2). Если файл консоли не указан, будет открыта новая консоль ММС.

/a — открывает консоль MMC в авторском режиме.
Дополнительными параметрами команды могут быть:
/64 — открывает 64-разрядную версию консоли MMC (MMC64). Этот параметр используется только при работе в ОС Windows XP 64-Bit Edition.
/32 — открывает 32-разрядную версию консоли MMC (MMC32). При работе в ОС Windows XP 64-Bit Edition в окне консоли MMC, запущенной с этим параметром, открываются 32-разрядные оснастки.
В. запуск из файлового менеджера Проводник ОС Windows XP:

- наведите манипулятор мышь на файл с расширением .msc, находящийся в системной папке ОС (%systemroot%\system32\),
- кликните правой кнопкой мыши на файле и из контекстного меню выберите Автор.

II. Настройка параметров консоли администрирования MMC предназначена для ее конфигурирования с целью придания ей уникального вида.

Содержание задания

Для придания уникального вида сохраненной (новой) консоли администрирования MMC в авторском режиме выполните следующие действия:

1. В меню Консоль выберите команду Параметры.
2. На вкладке Консоль в поле названия введите новый заголовок.
3. На вкладке Консоль выполните следующие действия:

- нажмите кнопку Сменить значок,
- в поле Имя файла введите путь к файлу, содержащему значки (например, %systemroot%\system32\shell32.dll),
- в поле Текущий значок выберите необходимый значок,
- кликните OK для ввода и Применить для подтверждения.

4. На вкладке Консоль из списка Режим консоли выберите пользовательский режим с полным доступом, в котором будет открываться консоль MMC при ее непосредственном запуске,
5. Для установленного в предыдущем пункте режима выполните указанные ниже действия:

- запретите изменение консоли MMC при ее непосредственном запуске, установив флажок «Не сохранять изменения для этой консоли»,
- сделайте активным диалоговое окно Вид | Настройка вида консоли MMC при запуске, установив флажок «Позволить пользователю настраивать вид консоли»,

6. Если необходимо удалить файлы, содержащие параметры отображения файлов консоли, на вкладке Очистка диска нажмите кнопку Удалить файлы.
7. Сохраните окончательно сконфигурированную консоль администрирования MMC, выбрав самостоятельно ее имя и путь к месту расположения в меню Консоль | Сохранить как... При сохранении обратите внимание на то, что файлы консоли по умолчанию размещаются в папке «Администрирование», имеющей полный путь %Pathname%\Главное меню\Программы\Админи-стрирование\.
8. Закройте сконфигурированную и сохраненную консоль администрирования MMC, выполнив соответствующие необходимые действия.

В файловом менеджере Проводник ОС Windows XP выполните следующие инструкции:

- наведите манипулятором мышь на сохраненный файл консоли администрирования MMC и, дважды кликнув на нем, запустите консоль,
- откройте диалоговое окно Вид | Настройка вида и, изменяя положение флажков, обратите внимание на получаемый результат,
- изменив вид консоли MMC приемлемым образом, кликните OK для подтверждения полученного результата,
- в контекстном меню Консоль кликните Выход,
- снова запустите консоль администрирования MMC, кликнув манипулятором мышь на сохраненном файле консоли,
- изучите полученный результат,
- сделайте вывод о проделанной работе и запишите его в отчет.

Задание. Добавление различных элементов и компонентов к дереву консоли администрирования MMC

Основным, интегрируемым на консоль компонентом является оснастка. Оснастки существуют двух видов:

- изолированные,
- расширения.

Изолированная оснастка (или просто оснастка) добавляется к дереву консоли MMC без предварительного добавления других элементов, то есть непосредственно в корень дерева консоли.
Оснастка расширения (или просто расширение) всегда добавляется к другой изолированной оснастке или расширению, которые уже имеются в дереве консоли MMC. Если для определенной оснастки разрешены расширения, то, как правило, они работают с объектами, управляемыми непосредственно этой оснасткой, например, с компьютером, принтером, модемом или другим внешним устройством.
В дереве консоли оснастки и расширения располагаются для удобства иерархически или по группам. При добавлении новой оснастки или расширения, они появляются в виде нового элемента в дереве консоли MMC или в виде нового пункта контекстного меню, дополнительной панели инструментов, страницы свойств, а также возможно мастера, организующего определенную последовательность действий, к уже установленной оснастке.
Другими элементами, по необходимости применимыми для интеграции на консоль администрирования MMC, являются виды панели задач и собственно задачи, которые могут включать в себя команды меню для элементов консоли и команды, запускаемые из командной строки. Кроме того, могут быть созданы команды, действующие как часть дерева консоли или открывающие другой компонент.
Прежде всего, перед добавлением указанных элементов к консоли MMC, необходимо определить их число. Если, в частности, требуется добавить несколько видов панели задач, то наряду с этим необходимо определить тип каждой панели (для отображения списка и задач или только задач), а также разделить задачи по интегрированным видам панели. Добавление видов панели задач и собственно задач осуществляется посредством работы мастера создания этих элементов. При этом важно помнить, что консоль MMC должна содержать, по крайней мере, одну оснастку, чтобы возможность интеграции появилась в принципе.
Отдельной возможностью, иногда необходимой при администрировании сетей, является добавление элементов и компонентов дерева консоли администрирования MMC в виде

списка ярлыков в меню «Избранное».
Дополнительные сведения о добавлении различных элементов в дерево консоли администрирования MMC можно получить, воспользовавшись справкой ОС Windows XP (Пуск | Справка и поддержка) в разделе Общее представление о MMC \ Консоль MMC в авторском режиме \ Оснастки \ Создание консолей.

Содержание задания

Первым необходимым компонентом, добавляемым к дереву консоли администрирования MMC при ее организации и построении, является оснастка. Для добавления оснастки в авторском режиме выполните следующие действия:

1. Создайте новую Консоль управления MMC одним их описанных в пункте I текущего учебного задания способов.
2. В меню Консоль выберите команду Добавить или удалить оснастку.
3. В диалоговом окне Добавить/удалить оснастку нажмите кнопку Добавить вкладки Изолированная оснастка. Список Оснастки в диалоговом окне Добавить/удалить оснастку определяет элемент дерева консоли, к которому выполняется добавление элементов. В этом списке можно найти любой элемент дерева консоли. Обратите внимание на то, что по умолчанию это Корень консоли.
4. В диалоговом окне Добавить изолированную оснастку, выберите оснастки Службы из списка доступных в системе, кликнув на ней манипулятором мышь и нажав кнопку Добавить. Для добавления другой оснастки из списка, повторите указанные действия настоящего пункта повторно.
5. Для некоторых оснасток в процессе их инсталляции выводится диалоговое окно Выбор целевого компьютера, определяющее чем, устанавливаемая оснастка, будет управлять в дальнейшем — локальным или сетевым компьютером. Выберите Локальный компьютер, установив переключатель в соответствующее положение.
6. Нажмите Готово, Закрыть и затем кликните OK для подтверждения ввода.
7. Скройте меню и панель инструментов оснастки Службы, выполнив действия указанные ниже:

- В меню Вид выберите команду Настроить,
- В группе Оснастка снимите флажок Меню,
- В группе Оснастка снимите флажок Панели инструментов.

При устанавливании или снятии флажков, соответствующие им меню и панели инструментов отображаются или скрываются, причем, для всех оснасток консоли, включая текущую. Если переключение флажков не приводит к изменению вида консоли, тогда текущая оснастка не имеет специальных меню или панелей инструментов.

8. Не закрывая консоль администрирования MMC, сохраните ее, выбрав команду Сохранить в меню Консоль.

Для добавления расширений к уже установленной в предыдущем задании оснастке Службы выполните следующее:

9. В меню Консоль выберите команду Добавить или удалить оснастку.
10. В диалоговом окне Добавить/удалить оснастку выберите вкладку Расширение. На этой вкладке можно выбрать любой элемент дерева консоли из списка Оснастки, которые могут быть расширены, и просмотреть Доступные расширения, которые могут быть включены или отключены. После подключения расширение

автоматически размещается в дереве консоли под оснасткой, к которой оно относится. Если дерево консоли содержит больше одного экземпляра оснастки, к которой подключено расширение, все остальные экземпляры автоматически получают это расширение.

11. Среди Доступных расширений оснастки Службы удалите флажок с расширения Расширенный вид и отметьте к чему привело это действие. Повторите аналогичные действия с другими расширениями данной оснастки и изучите получаемый результат.
12. Не закрывая консоль администрирования MMC, сохраните ее. В окне консоли администрирования выполните следующие инструкции:

- последовательно перебирая доступные в системе оснастки, найдите те из них, которые обладают дополнительным меню, панелью инструментов или расширениями,
- изучите полученный результат и сделайте вывод о проделанной работе,
- запишите полученную информацию в отчет, заполнив табл. 13.

Таблица 13. Результат поиска оснастки

№ п.п.	Оснастка	Результат поиска и вывод по способу применения дополнительных меню, панелей инструментов и/или расширений оснасток
1		
n		

Задание. Создание нового вида панели задач консоли администрирования MMC

Содержание задания

Для добавления видов панелей задач и собственно задач в авторском режиме выполните следующее:

1. Создайте новую Консоль управления MMC одним их описанных в пункте I текущего учебного задания способов.
2. Добавьте оснастку Службы в корень консоли MMC.
3. В дереве консоли кликните манипулятором мышь на этой оснастке.
4. В меню Действие или кликнув правой кнопкой манипулятора на оснастке, выберите команду Новый вид панели задач.
5. Следуйте инструкциям «Мастера создания вида панели задач», чтобы добавить на консоль новую панель вида.
6. Если сразу после создания вида панели задач необходимо создать задачи, установите флажок «Запустить мастер создания новой задачи» на последнем экране «Мастера создания вида панели задач».
7. Следуйте инструкциям «Мастера создания новой задачи», чтобы добавить
8. на консоль новую задачу к существующей панели вида.
9. В дереве консоли кликните элемент или компонент (в нашем случае это оснастка), связанный с видом панели задач, затем в меню Действие выберите команду Правка вида панели задач.

10. На вкладке Задачи нажмите кнопку Создать.
11. Повторите инструкции пункта 7 настоящего задания.
12. Не закрывая консоль администрирования ММС, сохраните ее.

Измените вид панели задач сохраненной консоли администрирования ММС, выполнив следующие действия:

- введите новое имя,
- введите новое описание,
- установите переключатель Стиль для области сведений в положение, соответствующее новому формату списка,
- удалите соответствующий флажок, чтобы скрыть стандартную вкладку,
- установите переключатель Стиль для описания задачи в положение, соответствующее новому стилю задачи,
- выберите новое значение ширины для вертикального списка или высоты для горизонтального списка,
- нажмите кнопку Параметры и установите переключатель в одно из необходимых положений,
- нажмите ОК для подтверждения ввода,
- изучите полученный результат,
- сделайте вывод о проделанной работе и запишите его в отчет.

Задание. Добавление элементов и компонентов дерева консоли в виде списка ярлыков в меню «Избранное»

Содержание задания

Для добавления элемента или компонента в авторском режиме выполните следующее:

1. Создайте новую Консоль управления ММС одним их описанных в пункте I текущего учебного задания способов.
2. В дереве консоли кликните элемент или компонент (в нашем случае это оснастка), который нужно добавить в список «Избранное».
3. В области сведений выберите вкладку вида панели задач, которую нужно добавить, в случае, если для элемента или компонента, указанного в дереве консоли, настроен вид панели задач. В противном случае в области сведений вкладки не видны.
4. Выберите в меню Избранное команду Добавить в избранное.
5. В поле Создать в диалогового окна Добавление в папку «Избранное» выполните указанные ниже действия:

- создайте новую папку с названием, выбранным самостоятельно, кликнув папку, которая будет выступать в качестве родительской для создаваемой папки и нажав кнопку Создать папку,
- нажмите кнопку ОК для ввода,
- в поле Имя папки введите имя, под которым будет добавлен элемент,
- кликните ОК для подтверждения ввода.

6. Не закрывая консоль администрирования ММС, сохраните ее. Упорядочите «Избранное» сохраненной консоли администрирования ММС, выполнив следующие действия:

- добавьте новую папку, введя ее имя в соответствующее поле и кликнув ОК для подтверждения ввода,
- переместите элемент, созданный в пункте 5 настоящего задания, в новую, только что созданную, папку и кликните ОК для ввода,
- переименуйте выбранный элемент и нажмите клавишу Enter для подтверждения ввода,
- удалите все элементы, расположенные ниже папки «Избранное»,
- нажмите Закрыть для завершения задания,
- изучите полученный результат,
- сделайте вывод о проделанной работе и запишите его в отчет.

Задание. Ознакомление с оснасткой «Локальные пользователи и группы»

Порядок выполнения:
Первым шагом при создании пользовательской консоли администрирования является придание ей функций контроля и управления учетными записями пользователей и групп. В среде ОС Windows XP учетные записи локальных пользователей и групп управляются одноименной оснасткой «Локальные пользователи и группы».
Учетная запись пользователя содержит все сведения, определяющие пользователя. К этим сведениям относятся имя пользователя и пароль, требуемые для входа в систему, имена групп, членом которых он является, а также права и разрешения, которые он имеет при работе и доступе к системным ресурсам системы.
Существует два основных типа учетных записей пользователей, доступных на компьютере с ОС Windows XP: учетная запись администратора и учетная запись с ограниченными правами. Кроме того, штатная учетная запись гостя доступна для пользователей, не имеющих собственных учетных записей в системе.
Учетная запись администратора предназначена для тех уполномоченных пользователей, кто может вносить изменения на уровне системы, устанавливать приложения и иметь полный доступ ко всем файлам и другим учетным записям. Она позволяет:

- создавать, изменять и удалять любые учетные записи,
- создавать и изменять имена и пароли пользователей,
- обеспечить наличие, по крайней мере, одного пользователя с учетной записью администратора, когда уже нет других учетных записей с подобными правами.

Учетная запись с ограниченными правами предназначена для пользователей, которым запрещается изменять большинство настроек системы, а также удалять важные файлы. Пользователь с такой учетной записью:

- не может устанавливать приложения и оборудование, но имеет доступ к уже существующим программам в системе,
- не может изменять имя и тип собственной учетной записи,
- может создавать, изменять или удалять собственный пароль,
- может изменять рисунок, назначенный учетной записи.

Необходимо помнить, что некоторые приложения могут работать некорректно для пользователей с ограниченными правами. В этом случае следует сменить тип учетной записи на администратора.
Учетная запись гостя предназначена для пользователей, не имеющих собственных учетных записей в системе. У записи гостя нет пароля, что позволяет быстро входить в систему, в частности, для проверки электронной почты или просмотра Интернета.
Пользователь с учетной записью гостя:

- не может устанавливать приложения и оборудование, но имеет доступ к уже существующим программам в системе,
- не может изменить собственной тип учетной записи,
- может изменять рисунок, назначенный учетной записи.

Совокупность пользователей, компьютеров и контактов называется группой. Существуют группы безопасности, которые используются для управления доступом или в качестве списков рассылки и группы распространения, применяемые только в электронной почте. Набор учетных записей пользователей представляет собой учетную запись группы. При включении учетной записи пользователя в определенную группу соответствующий пользователь получает все права и разрешения, предоставленные этой группе.

Содержание задания

1. Создайте пользовательскую консоль администрирования одним из изученных в предыдущем учебном задании способов и сконфигурируйте ее должным образом, приняв во внимание ее целевую принадлежность.
2. Добавьте оснастку «Локальные пользователи и группы» в корень консоли администрирования.
3. Создайте две учетные записи для двух разных пользователей. Имена, описание и пароли выберите самостоятельно.
4. Установите флажки «Потребовать смену пароля при следующем входе в систему» для первого пользователя и «Запретить смену пароля пользователем» для второго.
5. Создайте локальную группу, имя и описание которой выберите самостоятельно.
6. Поместите в новую локальную группу созданных ранее пользователей, воспользовавшись диалоговым окном Свойства каждого из них.
7. Поместите в новую локальную группу пользователя Администратор, воспользовавшись диалоговым окном Свойства этой группы (см. пункт 5 секции). Обратите внимание на то, что в этой локальной группе уже присутствуют учетные записи двух созданных ранее пользователей.
8. Выбрав самостоятельно имя и путь к месту расположения, сохраните и закройте консоль администрирования MMC.

При выполнении заданий секции используйте следующие инструкции:

- перенесите последовательность выполняемых действий по каждому из пунктов 1-8 в отчет (возможно приведение графических фрагментов, сделанных с экрана, в качестве демонстрационного материала),
- дважды войдите в систему под именами пользователей, соответствующих созданным учетным записям,
- обратите внимание на то, каким образом осуществляется влияние установленных в пункте 4 флажков на работу компьютера при смене пользователя,
- сделайте вывод о проделанной работе и запишите его в отчет.

Задание. Взаимосвязь утилиты «Учетные записи пользователей» с оснасткой «Локальные пользователи и группы» при смене типа учетной записи

Содержание задания

1. В Панели управления загрузите утилиту «Учетные записи пользователей», позволяющую изменять параметры и пароли учетных записей пользователей непосредственно в ОС Windows XP.
2. Загрузите пользовательскую консоль администрирования, созданную в Секции А текущего задания, с возможностью ее редактирования.
3. Создайте новую ограниченную учетную запись, воспользовавшись утилитой «Учетные записи пользователей».
4. Перейдите в оснастку «Локальные пользователи и группы» загруженной консоли администрирования и отметьте, в какую из локальных групп была по умолчанию определена вновь созданная учетная запись.
5. В окне утилиты «Учетные записи пользователей» измените тип данной учетной записи с ограниченного на административный.
6. Снова перейдите в оснастку «Локальные пользователи и группы» загруженной консоли администрирования и убедитесь в том, что смена типа учетной записи привела к перемещению пользователя из группы Пользователи в Администраторы, тем самым, расширяя его права.
7. Удалите этого пользователя из групп Пользователи и Администраторы.
8. Сохраните и закройте консоль администрирования MMC.

При выполнении заданий используйте следующие инструкции:

- перенесите последовательность выполняемых действий по каждому из пунктов 1-8 в отчет (возможно приведение графических фрагментов, сделанных с экрана, в качестве демонстрационного материала),
- обратите внимание на то, каким образом осуществляется взаимосвязь утилиты «Учетные записи пользователей» с оснасткой «Локальные пользователи и группы» при смене типа учетной записи,
- сделайте вывод о проделанной работе и запишите его в отчет.

Задание. Возможности оснастки «Локальные пользователи и группы» при работе с профилями пользователей

Профиль пользователя определяет параметры конфигурации рабочей среды, например, индивидуальную настройку экрана, сетевых подключений и доступа к принтеру. Эти параметры могут быть заданы самим пользователем или системным администратором. Существуют следующие виды профилей пользователя:

- локальный профиль, представляющий собой машинную запись об авторизованном пользователе, автоматически создаваемую на компьютере при первом входе на рабочую станцию или сервер, и хранящийся на локальном жестком диске. Любые изменения локального профиля действительны только в рамках компьютера, на котором они произведены.
- перемещаемый профиль это находящийся на сервере профиль пользователя, который загружается на локальный компьютер при входе в систему и обновляется на локальном компьютере и сервере при выходе из нее. Если локальная копия профиля является более новой, чем копия на сервере, то пользователь имеет возможность использовать ее при очередном входе в систему.
- обязательный профиль это перемещаемый профиль, в котором можно задать конкретные параметры для отдельных пользователей или их группы. Этот профиль

загружается всякий раз, когда осуществляется вход в систему и не обновляется при выходе из нее. Он может быть изменен только членом группы Администратор.

Содержание задания

1. Загрузите пользовательскую консоль администрирования, созданную в Секциях А и В текущего задания, с возможностью ее редактирования.
2. На локальном диске D: создайте папку, выбрав ее имя самостоятельно, для хранения в ней профилей пользователей. Эта папка верхнего уровня, которая будет содержать индивидуальные папки профилей пользователей.
3. Создайте вложенную внутрь папку, предназначенную для непосредственного хранения профиля пользователя.
4. Скопируйте профиль Администратора в созданную в пункте 3 секции папку, воспользовавшись системными средствами ОС Windows XP в окне свойств системы (Панель управления | Система). При этом в диалоговом окне Копирование профиля разрешите использование для Всех пользователей.
5. В существующую консоль добавьте локальную оснастку «Общие папки». В доступных командах вложенной директории «Общие ресурсы» оснастки кликните Новый общий ресурс…и, следуя инструкциям Мастера создания общей папки, откройте Всем пользователям полный доступ для директорий, созданных в пунктах 2 и 3 секции. При этом новый общий ресурс должен появиться в столбце «Общая папка» в области сведений консоли администрирования. На вкладке Безопасность диалогового окна Свойства новых общих ресурсов нажмите кнопку Дополнительно и во вновь появившемся диалоговом окне установите флажок Наследовать от родительского объекта…
6. В виде профиля пользователя подключите только что скопированный в нестандартную папку профиль Администратора к одному из ранее созданных в Секции А пользователей, воспользовавшись его диалоговым окном Свойства.
7. В виде домашней папки подключите сетевой диск «H» на нестандартную папку с профилем Администратора к другому из ранее созданных в Секции А пользователей, воспользовавшись его диалоговым окном Свойства.
8. Создайте Новый вид панели задач таким образом, чтобы имелась возможность добавлять нового пользователя или группу, а изменять, удалять или переименовывать их возможности не было.
9. Сохраните и закройте консоль администрирования MMC.

При выполнении заданий секции используйте следующие инструкции:

- перенесите последовательность выполняемых действий по каждому из пунктов 1-9 в отчет (возможно приведение графических фрагментов, сделанных с экрана, в качестве демонстрационного материала),
- дважды войдите в систему под именами пользователей, соответствующих созданным учетным записям,
- проверьте результат подключения профилей в пунктах 6 и 7 секции: нажмите Пуск | Выполнить, наберите в появившемся окне SendTo и нажмите Enter для ввода, в строке Адрес: появившегося окна отметьте полный путь папки SendTo, удостоверяющий правильную загрузку подключенного профиля.
- обратите внимание на то, каким образом осуществляется влияние подключенных в профилей на работу компьютера при смене пользователя,
- сделайте вывод о проделанной работе и запишите его в отчет.

Задание. Основные возможности оснастки «Редактор объекта групповой политики»

Административное средство Active Directory предназначено для решения повседневных задач сетевого управления, в число которых входят: создание, удаление, изменение, перемещение и предоставление разрешений на объекты каталога. Объектами управления могут являться подразделения, пользователи, контакты, группы, компьютеры, принтеры, а также объекты общих файлов.
Active Directory обеспечивает системного администратора иерархическим представлением сети и едиными возможностями управления всеми сетевыми объектами ОС Windows 2000 Server, где она является основой групповой политики. Для реализации указанных возможностей в ОС Windows 2000 Server имеются три оснастки «Пользователи и компьютеры Active Directory», «Домены и доверие Active Directory» и «Сайты и службы Active Directory», первые две из которых предназначены для создания объектов групповой политики домена или подразделения, третья — соответственно, для создания объекта групповой политики сайта. Для запуска этих оснасток из ОС Windows XP необходимо осуществить подключение к удаленному столу сервера с установленными средствами администрирования ОС Windows 2000 Server, поскольку служба каталогов Active Directory в ОС Windows XP является по умолчанию не доступной.
Понятие групповая политика в области администрирования применяется не только к пользователям и компьютерам, но и к серверам, контроллерам доменов и другим компьютерам под управлением ОС Windows XP. По умолчанию групповая политика применяется к домену и влияет на все его компьютеры и пользователи. Обеспечение групповой политики подразумевает конфигурирование системы и задания набора условий и параметров групповой политики, приводящих к определенным ограничениям или разрешениям в ОС.
Параметры групповой политики определяют различные компоненты конфигурации пользователя, в частности, окружения пользовательского рабочего стола (программы, доступные пользователям, программы, отображающиеся на рабочем столе, и параметры меню Пуск) и конфигурации компьютера, включая параметры, применяемые вне зависимости от того, кто работает на этих компьютерах. Чтобы создать частную конфигурацию компьютера для определенной группы пользователей используется оснастка «Редактор объекта групповой политики».
Набор указанных параметров групповой политики содержится в объекте групповой политики, который, в свою очередь, связан с выбранными объектами Active Directory — сайтами, доменами или подразделениями. Объектами групповой политики являются документы, создаваемые оснасткой. Они хранятся на уровне домена и оказывают влияние на пользователей и компьютеры доменов и подразделений. Кроме того, каждый компьютер с ОС Windows XP имеет единственную, хранящуюся локально, группу параметров, которая называется локальным объектом групповой политики.
Все многообразие параметров групповой политики содержится в соответствующих расширениях оснастки «Групповая политика», посредством которых системному администратору предоставляются широкие возможности по управлению процессами и ресурсами ОС Windows XP:

- на основе реестра, используя расширение Административные шаблоны (А). При этом создается файл, содержащий параметры реестра, записанные в область базы данных реестра пользователя, в разделе HKEY_CURRENT_USER и в разделе HKEY_LOCAL_MACHINE для локального компьютера.
- посредством расширения Назначение сценариев (В). Групповая политика указывает сценарии входа/выхода пользователей из системы и загрузки/завершения работы.

- используя Редактор перенаправления папок (C). Групповая политика имеет возможность перенаправить системные папки «Мои документы» и «Мои рисунки», из папки «Documents and Settings» локального компьютера в новое место расположения в сети.
- на основе расширения Установка программ (D), которое позволяет назначать, публиковать и восстанавливать приложения.
- посредством расширения Параметры безопасности (E), позволяющей устанавливать ограничения на использование программ, политику отрытого ключа, а также осуществлять управление политикой безопасности IP.

Административные шаблоны это текстовые файлы с расширением .adm, содержащие сведения о политике для элементов, расположенных в папке «Административные шаблоны» оснастки. В ОС Windows XP доступно четыре файла административных шаблонов, приведенных в табл. 14.
Файлы административные шаблонов состоят из иерархии категорий и подкатегорий, которые вместе определяют отображение параметров групповой политики. В них содержатся следующие сведения:

- размещение параметров реестра, соответствующих каждому параметру административного шаблона групповой политики,
- величина параметров или ограничений, связанных с каждым параметром административного шаблона,
- значение по умолчанию для большинства параметров,
- объяснение функции каждого параметра,
- версии ОС Windows, поддерживающие каждый параметр.

Таблица 14. Административные шаблоны ОС Windows XP

Шаблон (.adm)	**Справка по параметрам**	**Описание**
System	%systemroot%\help\system.chm	В групповой политике шаблон установлен по умолчанию для клиентов ОС Windows 2000 и XP
Inetres	%systemroot%\ help\inetres.chm	В групповой политике шаблон Internet Explorer установлен по умолчанию для клиентов ОС Windows 2000 и XP
Wmplayer	%systemroot%\help\wmplay.chm	Параметры WMP для клиентов ОС Windows 2000 и XP
Conf	%systemroot%\help\conf1.chm	Параметры программы NetMeeting для клиентов ОС Windows 2000 и XP

В среде ОС Windows XP имеется возможность использования сценариев посредством двух серверов Wscript.exe или Cscript.exe, поддерживающих как Visual Basic Scripting Edition (расширение .vbs), так и JScript (расширение .js) файлы. В частности, в средствах оснастки «Групповая политика» имеется два расширения, расположенные в узлах консоли «Конфигурация компьютера | Конфигурация Windows» или «Конфигурация пользователя | Конфигурация Windows», позволяющие развертывать сценарии с использованием указанных серверов ОС Windows XP. Эти расширения следующие:

- сценарии (запуск/завершение) — расширение, посредством которого можно указать локально выполняемый сценарий при запуске и завершении работы компьютера.
- сценарии (вход/выход из системы) — расширение, посредством которого можно указать выполняемый сценарий при входе и выходе пользователя из системы. Эти сценарии запускаются с правами пользователя, а не администратора.

Перенаправление папки используется для перемещения некоторых специальных папок, например «Мои документы» и «Мои рисунки», в заданное место в сети, для их последующего доступа с разных узлов. В ОС Windows XP возможны следующие специальные папки для перенаправления (табл. 15).

Таблица 15. Специальные папки ОС Windows XP

Специальная папка	Примечания
Application Data	Параметры групповой политики управляют поведением папки «Application Data» при включении кэширования на стороне клиента. Параметры расположены в дереве консоли «Групповая политика» в Административных шаблонах\Сеть\Автономные файлы.
Рабочий стол	Папка может быть перенаправлена независимо от всех остальных специальных папок.
Мои документы	Особенности и преимущества перенаправления этой папки описаны ниже.
Мои документы\ Мои рисунки	Эта папка может быть перенаправлена независимо от предыдущей папки «Мои документы» или совместно с ней, как это происходит по умолчанию. Именно эта комбинация является рекомендуемой.
Главное меню	При перенаправлении папки «Главное меню» ее подпапки всегда перенаправляются вместе с ней.

Некоторые из преимуществ, описанных ниже, относятся к перенаправлению любой специальной папки, однако перенаправление папки «Мои документы» может быть особенно удобным, поскольку со временем размер этой папки может увеличиваться.

- При использовании перемещаемого профиля пользователя его частью является только сетевой путь к папке «Мои документы», но не сама папка. Поэтому ее содержимое не нужно копировать и перемещать между клиентом и сервером каждый раз при входе пользователя в систему или его выходе, что делает процессы входа и выхода сравнительно быстрее.
- Даже если пользователь входит в сеть с различных компьютеров, все его документы всегда доступны.
- Технология автономных файлов обеспечивает пользователям доступ к папке «Мои документы» даже при отсутствии подключения к сети. Это особенно полезно для пользователей, использующих мобильные компьютеры.
- Имеется возможность архивировать данные, хранящиеся на сервере, при управлении перемещаемыми профилями. Это является более безопасным, поскольку не требуется вмешательство пользователя.
- Системный администратор может устанавливать дисковые квоты с помощью групповой политики, ограничивая дисковое пространство, выделенное пользователю для специальных папок.

- Данные пользователя могут быть перенаправлены на жесткий диск локального компьютера с другого жесткого диска, на котором хранятся системные файлы ОС. Это может обезопасить пользовательские файлы, если необходимо будет ее переустанавливать.

Кроме всего прочего, в ОС Windows XP имеется возможность предоставления исключительных прав на специальные папки. Если на вкладке Параметры диалогового окна свойств каждой папки установить флажок Предоставить права монопольного доступа к папке «Мои документы», пользователь и локальная система получают полный контроль над папкой, и никто другой, включая администратора, не будет иметь на нее никаких прав. В противном случае, если этот параметр отключен, то разрешения для папки не изменяются, а используются, применяемые по умолчанию разрешения. Еще одна возможность заключается в том, что на специальные папки могут быть расширены дополнительные разрешения, полный список которых доступен в справке ОС Windows XP.

Установка программного обеспечения является неотъемлемой процедурой при работе с любой ОС. Для этого используется одноименная оснастка «Установка программного обеспечения», которая помогает определить способ установки и сопровождения приложений. Также с ее помощью можно управлять приложением внутри объекта групповой политики посредством службы каталогов Active Directory.

Приложения управляются в одном из двух режимов: назначения или публикации. Приложение назначается, когда необходимо, чтобы оно было установлено на всех узлах сети. Например, требуется, чтобы на всех компьютерах аудитории было установлено одно и тоже приложение. Поскольку объект групповой политики управляет всеми пользователями аудитории, при назначении приложения в объекте групповой политики, оно одновременно объявляется на всех компьютерах, но при этом фактически не устанавливается. Устанавливаются лишь только необходимые данные для создания ярлыка этого приложения в меню Пуск, а в реестре осуществляется связывание расширения его документа с ним самим. При первом выборе приложения на загрузку, а также, если пользователь, не запускавший приложение ранее, выбирает его документ для работы, приложение устанавливается автоматически с одновременным открытием этого документа.

Назначенное в системе приложение можно удалить, но оно будет объявлено снова при следующем входе. Если выбрать его в меню Пуск, оно будет повторно автоматически установлено.

Приложение публикуется, если необходимо сделать его доступным для тех пользователей, управляемых объектом групповой политики, кто хочет установить это приложение. При этом у пользователей имеется выбор самостоятельно решать, устанавливать приложение или нет. Например, если приложение публикуется для пользователей, желающих его установить, им следует для этого открыть компонент «Установка и удаление программ» на панели управления и произвести установку. В случае если пользователям не удалось установить приложение с помощью этого компонента, но файлы с соответствующим расширением связаны с приложением, оно будет установлено при первой попытке открыть файл с этим расширением.

Безопасность компьютера, уязвимость системы безопасности, а также различного вида угрозы заботят не только профессионалов в области информационных технологий, но и рядовых пользователей компьютеров. Многие организации и отдельные пользователи имеют постоянные подключения к Интернету, что подвергает их компьютеры рискам заражения вирусами, несанкционированного проникновения, атак на службы и другим угрозам.

Существуют некоторые правила, называемые политиками или параметрами безопасности,

которые предназначены для обеспечения защиты ресурсов одного или нескольких компьютеров в сети. Параметры безопасности позволяют контролировать:

- проверку подлинности пользователей при входе в сеть или отдельный узел,
- ресурсы, которые пользователи могут использовать,
- включение и отключение записи действий пользователя или группы в журнале событий,
- принадлежность к группам.

Настраиваются параметры безопасности, используя средства диспетчера настройки безопасности. К этим средствам относятся:

- шаблоны безопасности,
- анализ и настройка безопасности,
- программа командной строки Secedit.exe,
- локальная политика безопасности,
- расширение «Параметры безопасности» для групповой политики.

Расширение «Параметры безопасности» позволяет пользователям изменять настройку безопасности в оснастке «Групповая политика», влияющей, в свою очередь, одновременно на все узлы сети посредством объекта групповой политики. Однако чтобы установить или изменить отдельные параметры безопасности на отдельных компьютерах, используется средство «Локальная политика безопасности», включающее политику аудита, назначение прав пользователя и локальные параметры безопасности.
Чтобы применить несколько параметров безопасности единовременно, имеется возможность определить их с помощью шаблонов безопасности и затем применить к системе с помощью средства «Анализ и настройка безопасности» или программы Secedit.exe (Пуск | Выполнить | Secedit.exe), а также импортировать готовый шаблон в соответствующую локальную или групповую политику.
В заключение следует отметить, что в пакете обновления SP2 для ОС Windows XP с целью повышения безопасности вводятся некоторые изменения параметров безопасности. Обобщая нововведения, параметры безопасности сгруппированы по соответствующим областям (табл. 16).

Таблица 16. Области безопасности ОС Windows XP

Область безопасности	Описание
Политики учетных записей	Политика паролей, политика блокировки учетной записи и политика Kerberos
Локальные политики	Политика аудита, назначение прав пользователя и параметры безопасности
Журнал событий	Параметры журналов событий приложений, системных событий и событий безопасности
Группы с ограниченным доступом	Состав групп с особыми требованиями к безопасности
Системные службы	Параметры запуска и разрешения для системных служб
Реестр	Разрешения для разделов реестра
Файловая система	Разрешения для файлов и папок

Содержание задания

Прежде чем непосредственно перейти к ознакомлению с возможностями изучаемой оснастки, следует обратить внимание на имеющиеся в системе способы ее открытия в изолированном виде. Имеется возможность открыть оснастку через меню Пуск | Выполнить. Для этого необходимо ввести gpedit.msc и нажать Enter для ввода. Кроме того, загрузка оснастки возможна в среде командной оболочки, различные варианты которой отображены ниже (табл. 17). Поэкспериментируйте с имеющимися возможностями загрузки изолированной оснастки «Групповая политика».

Таблица 17. Варианты загрузки групповой политики посредством командной оболочки

№ п.п.	Пример команды	Описание
1	gpedit.msc /gpcomputer:"Имя_компьютера"	редактирование групповой политики локального компьютера.
2	gpedit.msc /gpcomputer:"Имя_компьютера.WingTipToys.com"	редактирование локального объекта групповой политики на локальном компьютере, имя которого задается в формате DNS.
3	gpedit.msc /gpobject:"LDAP://CN={31B2F340016D-11D2-945F00C04FB984F9},CN=Policies,CN=System,DC=WingTipToys,DC=com" в фигурных скобках {16-байтное число} — глобальный уникальный идентификатор GUID	редактирование объекта групповой политики с применением Active Directory.

1. Загрузите пользовательскую консоль администрирования, созданную в задании 39, с возможностью ее редактирования.
2. Добавьте оснастку «Редактор объекта групповой политики» в корень консоли администрирования, выбрав в поле Объекта групповой политики значение Локальный компьютер.
3. Откройте поочередно все ветви дерева консоли в узле Политика «Локальный компьютер» и изучите где и какие параметры групповой политики располагаются. Обратите внимание на то, что некоторые параметры находятся в состоянии Включено, другие, напротив — Отключено или Не определено (Не задано).
4. В левой части окна консоли выберите «Политика Локальный компьютер | Конфигурация компьютера | Конфигурация Windows | Параметры безопасности | Локальные политики | Назначение прав пользователей / Параметры безопасности».
5. Изучите локальные политики Назначение прав пользователя и Параметры безопасности.
6. Сохраните и закройте консоль администрирования MMC.

При выполнении пункта 6 используйте следующие инструкции:

- последовательно перебирая каждую из локальных политик, изучите ее содержимое, принадлежность к пользователю, а также состояние, в котором она находится,
- запишите полученную информацию в отчет, заполнив табл. 18.
- сделайте вывод о проделанной работе и запишите его в отчет.

Таблица 18. Локальная политика безопасности ОС Windows XP

№ п.п.	Политика		Привилегией обладает	Состояние
	название	описание		
1	**Изменение системного времени**	**Определяет, какие пользователи и группы могут изменять время и дату компьютера.**	**Администратор Опытный пользователь**	**Включено**
n				

Задание. Возможности оснастки «Групповая политика» при настройке локального узла

Как утверждалось ранее, с помощью параметров узла Конфигурация компьютера устанавливаются политики, которые применяются к компьютеру независимо от того, кто его использует для входа в сеть, в то время как, с помощью параметра Конфигурация пользователя устанавливаются политики, которые применяются к каждому пользователю, работающему на компьютере.
Таким образом, существует ряд параметров групповой политики предназначенных исключительно для изменений, направленных на создание специальных настроек среды пользователя ОС Windows XP, в частности, для придания системе уникального вида. В качестве примера, с помощью групповой политики можно удалить значки с рабочего стола, изменить содержимое меню Пуск и упростить структуру панели управления.

Содержание задания

1. Загрузите пользовательскую консоль администрирования, созданную ранее, с возможностью ее редактирования.
2. В левой части окна консоли выберите «Политика Локальный компьютер | Конфигурация пользователя | Административные шаблоны | Панель задач и меню «Пуск» / Панель управления / Рабочий стол».
3. Изучите параметры политик | Административные шаблоны | Панель задач и меню «Пуск» / Панель управления / Рабочий стол.
4. Выберите любые пять параметров в каждой из политик Панель задач и меню «Пуск» / Панель управления / Рабочий стол. Измените выбранные параметры на противоположные, перезагрузите компьютер и отметьте полученные визуальные изменения графического интерфейса.
5. Сохраните и закройте консоль администрирования ММС.

При выполнении пункта 3 используйте следующие инструкции:

- самостоятельно последовательно выбирая политики указанных административных шаблонов, в общем количестве не менее десяти, измените их состояние, делая его активным,
- исследуйте влияние смены состояния на внешний вид соответствующего элемента графического интерфейса ОС Windows XP,
- запишите полученную информацию в отчет, заполнив табл. 19.
- сделайте вывод о проделанной работе и запишите его в отчет.

- продемонстрируйте преподавателю конечный результат изменения параметров административных шаблонов, применяемых для настройки уникального вида элементов графического интерфейса.

Таблица 19. Настройка некоторых административных шаблонов

№ п.п.	Административный шаблон		Результат при активации политики
	Название шаблона	Описание политики	
1	Рабочий стол	Не показывать значок Internet Explorer (IE)	При изменении состояния политики в положение Включено значок IE убирается с рабочего стола и панели быстрого запуска
n			

Задание 44. Возможности оснасток, предназначенных для диагностики, мониторинга, настройки и оптимизации

В широком смысле аудитом называется регистрация каких-либо действий, процессов или событий, предназначенная для обеспечения комплексной безопасности чего-либо. В частности, средства аудита в среде ОС Windows XP предназначены для отслеживания действий пользователей путем регистрации системных событий определенных типов в журнале безопасности сервера или рабочей станции. Кроме того, отображение и фиксация системных событий необходимы для определения злоумышленников или попыток поставить под угрозу данные операционной системы. Примером события, подлежащего аудиту, является неудачная попытка доступа к системе.
Наиболее общими типами событий, подлежащих аудиту в ОС, являются:

- доступ к таким объектам, как файлы и папки;
- управление учетными записями пользователей и групп;
- вход пользователей в систему и выход из нее.

Чтобы обеспечить возможность аудита в среде ОС Windows XP, сперва необходимо выбрать политику аудита, указывающую категории событий аудита, связанных с безопасностью. При инсталлировании ОС все категории аудита по умолчанию выключены; включая их последовательно администратор может создать политику аудита, удовлетворяющую всем требованиям организации.
К числу категорий событий, предназначенных для контроля, относятся:

- аудит событий входа в систему;
- аудит управления учетными записями;
- аудит доступа к службе каталогов;
- аудит входа в систему;
- аудит доступа к объектам;
- аудит изменения политики;
- аудит использования привилегий;
- аудит отслеживания процессов и системных событий.

В частности, если выбран аудит доступа к объектам как часть политики аудита, необходимо включить либо категорию аудита доступа к службе каталогов (для аудита объектов на контроллере домена), либо категорию аудита доступа к объектам (для аудита объектов на рядовой сервер или рабочую станцию).
Кроме того, с целью уменьшения риска угроз системной безопасности в целом администратор должен предпринять следующие базовые шаги, направленные на обеспечение аудита в системе. Основные события аудита и угрозы безопасности, отображаемые при помощи этого события, сведены в табл. 20.
Таблица 20. Основные события аудита в ОС Windows XP

№ п.п.	Событие аудита	Потенциальная угроза
1	Аудит отказов входа/выхода.	Случайный взлом пароля
2	Аудит успехов входа/выхода.	Вход с украденным паролем
3	Аудит успехов использования привилегий, управления пользователями и группами, изменений политик безопасности, перезагрузки, выключения и системных событий.	Неправильное использование привилегий
4	Аудит успехов и отказов событий доступа к файлам и объектам. Аудит успехов и отказов диспетчера файлов в доступе подозрительным пользователям или группам к важным файлам для чтения и записи.	Неправильный доступ к важным файлам
5	Аудит успехов и отказов событий доступа к принтерам и объектам. Аудит успехов и отказов диспетчера печати в доступе подозрительным пользователям или группам к принтерам.	Неправильный доступ к принтерам
6	Аудит успехов и отказов доступа для записи к программным файлам с расширениями .exe и .dll. Аудит успехов и отказов для отслеживания процессов в системе при запуске подозрительных программ.	Эпидемия вирусов

Формирование политики аудита объектов в системе осуществляется посредством оснастки «Групповая политика»; в частности, с ее помощью устанавливается и настраиваются параметры политики аудита.

Содержание задания

1. Откройте консоль администрирования и создайте в оснастке новые задачи «Сведения о системе», «Диагностика DirectX» и «Диспетчер задач», вызовите соответствующие системные модули (указанные в задании 34) для выполнения задач диагностирования и мониторинга ОС Windows XP. Убедитесь в работоспособности этих задач.
2. Локально добавьте на открытую консоль администрирования системную оснастку «Групповая политика».
3. В дереве консоли «Политика «Локальный компьютер» щелкните манипулятором мышь по папке «Конфигурация компьютера | Конфигурация Windows | Параметры безопасности | Локальные политики | Политика аудита» для настройки локальных политик аудита.

4. Настройте политики аудита. Для этого в области сведений дважды щелкните на политике аудита, для которой необходимо изменить параметры аудита и установите один или оба флажка («успех» или «отказ») для успешных или неуспешных системных событий, которые необходимо регистрировать. Повторите действия указанные в текущем пункте секции для других политик аудита в случае необходимости.
5. Настройте аудит файлов и папок. Для этого измените параметры «успех» или «отказ» категории событий «Аудит доступа к объектам». Укажите файлы или папки для аудита, выполнив следующие действия:

- на вкладке «Безопасность» команды «Свойства» файла или папки нажмите кнопку «Дополнительно»,
- на вкладке «Аудит» нажмите кнопку «Добавить»,
- в диалоговом окне «Выбор: пользователь, компьютер или группа» выберите имя пользователя или группы, для действий которых требуется производить аудит файлов и папок, и нажмите кнопку OK для подтверждения выбора;
- в появившемся диалоговом окне «Элемент аудита» в группе «Доступ» установите флажки «успех», «отказ» или оба эти флажка одновременно напротив действий, для которых требуется провести аудит,
- выберите из выпадающего меню «Применить:» опцию «Для этой папки и ее подпапок» (или любую другую опцию на Ваш выбор), а затем нажмите кнопку OK и Примерить для подтверждения ввода.

Если указанные выше действия выполнить не удалось по причине отсутствия вкладки «Безопасность» в «Свойствах» объекта, выполните следующее:

- в дереве консоли «Политика «Локальный компьютер» щелкните манипулятором мышь по папке «Конфигурация пользователя | Административные шаблоны | Компоненты Windows | Проводник»;
- в области сведений дважды щелкните на «Удалить вкладку «Безопасность», измените системный параметр на «Отключено» на одноименной вкладке и подтвердите выбор, кликнув OK;
- выберите команду Панель управления в меню Пуск, откройте компонент «Свойства папки» на панели управления, дважды щелкнув по нему мышью, и на вкладке «Вид» в группе «Дополнительные параметры | Файлы и папки» снимите флажок «Использовать простой общий доступ к файлам (рекомендуется)».

6. Не закрывая консоль администрирования MMC, сохраните ее.

При выполнении заданий секции используйте следующие инструкции:

- перенесите последовательность выполняемых действий по каждому из пунктов 1-5 в отчет (возможно приведение графических фрагментов, сделанных с экрана, в качестве демонстрационного материала),
- сделайте вывод о проделанной работе и запишите его в отчет.

Задание. Возможности оснастки «Просмотр событий»

Поскольку, аудит предполагает регистрацию различного рода системных событий (табл. 21), имеющих место в операционной среде, их регистрация в ОС Windows XP осуществляется в журналах трех основных типов, описание которых представлено после таблицы.

В журнале приложений содержатся данные, относящиеся к работе приложений. Записи этого журнала создаются самими приложениями. События, вносимые в журнал, определяются разработчиками соответствующих приложений.
Журнал безопасности содержит записи о таких событиях, как успешные и безуспешные попытки доступа в ОС, а также о событиях, относящихся к использованию системных ресурсов. В частности, после разрешения аудита входа в систему сведения обо всех попытках входа заносятся в журнал безопасности. Именно в этом журнале аккумулируются данные по системным событиям, аудит которых был настроен в предыдущей секции учебного задания.
В журнале системы содержатся события системных компонентов ОС. Так, например, в журнале системы регистрируются сбои при загрузке драйвера или других программных компонентов в момент запуска системы.
В дополнение к существующим ОС Windows XP имеет в своем распоряжении еще два журнала: службы каталогов и службы репликации файлов, запись событий в которые выполняется в случае, если компьютер настроен в качестве контроллера домена.

Таблица 21. Типы системных событий в ОС Windows XP

№ п.п.	Тип события	Описание
1	Ошибка	Возникает при серьезных трудностях, связанных с потерей данных или функциональности ОС (например, при сбое загрузки службы в момент ее запуска).
2	Предупреждение	Возникает при событии, которое в момент записи в журнал не было существенным, но может привести к ошибкам в будущем (например, если на диске осталось мало свободного места).
3	Уведомление	Возникает при событии, описывающее удачное завершение действия приложением, драйвером или службой (например, после успешной загрузки драйвера).
4	Аудит успехов	Возникает при событии, которое соответствует успешно завершенному действию, связанному с поддержкой безопасности ОС (например, в случае успешного входа пользователя в систему).
5	Аудит отказов	Возникает при событии, которое соответствует неудачно завершенному действию, связанному с поддержкой безопасности ОС (например, в случае неудачной попытки доступа пользователя к сетевому диску).

В журнале службы каталогов содержатся события, заносимые службой каталогов ОС Windows XP. Например, проблемы соединения между сервером и общим каталогом записываются в этот журнал.
Журнал службы репликации файлов содержит записи о системных событиях, внесенных службой репликации файлов ОС Windows XP. В этот журнал записываются неудачи при репликации файлов, а также события, которые происходят пока контроллеры домена обновляются данными об изменениях из общей папки Sysvol, где хранится серверная копия общих файлов, реплицируемых между всеми контроллерами домена.
Кроме того, существует журнал DNS-сервера, в который записываются сообщения об системных событиях, зарегистрированных службой DNS. В этот журнал записываются события, связанные с разрешением DNS-имен IP-адресам.

В ОС Windows XP за регистрацию системных событий в описанных выше журналах отвечает специальная служба, называемая службой журнала событий, которая загружается автоматически при старте системы. Эта служба контролирует ведение журналов и осуществляет внесение в них соответствующих записей системных событий в реальном масштабе времени. При этом любой пользователь может просматривать журналы приложений и системы, однако журналы безопасности доступны только системному администратору, который предварительно должен настроить параметры системных событий аудита (табл. 20), воспользовавшись компонентом «Групповая политика».

Содержание задания

1. Локально добавьте на открытую консоль администрирования новую системную оснастку «Просмотр событий».
2. В дереве консоли щелкните манипулятором мышь по оснастке «Просмотр событий» и обратите внимание на появившиеся три журнала и их текущие размеры в области сведений справа. Последовательно перебирая журналы приложений, безопасности и системы, отметьте в них наличие всех указанных выше типов системных событий (табл. 20). При этом обратите внимание на то, что такие типы событий как аудиты отказов и успехов присущи только журналу безопасности, который был Вами настроен в предыдущей секции. Остальные типы событий встречаются как в журнале приложений, так и в журнале системы.
3. Воспользовавшись меню «Вид» изучаемой оснастки, отфильтруйте:

- в журнале приложений событие «Уведомление» за прошедшее время,
- в журнале безопасность событие «Аудит отказов» за IV квартал,
- в журнале система событие «Ошибка» за последнюю неделю, с сортировкой по дате «от старых к новым».

4. В окне журнала событий системы удалите столбцы «Пользователь», «Компьютер» и «Категория», оставив остальные.
5. Воспользовавшись системой поиска, найдите событие типа «Предупреждение» с кодом 1003 от источника DHCP в журнале событий системы.
6. Создайте собственный журнал событий, содержащий только сведения об ошибках приложений и программ. Установите максимальный размер этого журнала в 128 Кб и возможность затирания старых событий по необходимости. Сохраните созданный журнал в двоичном виде с расширением .evt.
7. Создайте инструмент для регистрации событий аудита любого компьютера рабочей группы или домена и осуществите просмотр системных событий другого узла локальной сети с его помощью.
8. Не закрывая консоль администрирования MMC, сохраните ее.

При выполнении заданий секции используйте следующие инструкции:

- перенесите последовательность выполняемых действий по каждому из пунктов 1-6 в отчет (возможно приведение графических фрагментов, сделанных с экрана, в качестве демонстрационного материала),
- сделайте вывод о проделанной работе и запишите его в отчет.

Лабораторная работа №22

Тема: Программа «Просмотр событий» в ос Windows XP

Целью лабораторной работы является получение навыков по созданию политики аудита и управлению аудитом ресурсов и событий защищенной операционной системы Windows-XP.
1.2. В результате выполнения лабораторной работы должны быть приобретены знания:
- назначения и возможностей использования аудита защищенной операционной системы Windows-XP;
- принципа разработки политики аудита.
1.3. В процессе выполнения лабораторной работы необходимо овладеть навыками:
- разработки политики аудита для таких объектов операционной системы Windows-XP как файлы, принтеры и системные события;
- настройки и управления аудитом для таких объектов операционной системы Windows-XP как файлы, принтеры и системные события;
- просмотра и управления журналами аудита.
1.4. Используемое оборудование и программное обеспечение - ПЭВМ IBM PC, операционная система Windows-XP.

2. Основные теоретические сведения

2.1. Изучите понятие и свойства аудита ресурсов и событий

Аудит — одна из функций групповой политики Windows XP Professional. Подсистема аудита (auditing) — это инструмент, предназначенный для поддержания безопасности в сети и позволяющий отслеживать действия пользователей, а также действия операционной системы Windows XP Professional, называемые событиями (events). Средствами аудита можно задать режим, при котором Windows XP Professional регистрирует события в журнале безопасности (security.log). В нем хранятся записи об успешных и неудачных попытках входа в систему и о таких событиях, как создание, открытие и закрытие файлов или других объектов. Запись аудита в журнале безопасности содержит данные о:
• выполненных операциях;
• пользователях, выполнивших операцию;
• успешном или неуспешном выполнении операции, кроме того, указывается время, когда произошло данное событие.

Планирование политики аудита

Понятие о политике аудита

Политика аудита (audit policy) задает типы событий системы безопасности, которые Windows XP Professional регистрирует в журнале безопасности каждого компьютера. Журнал безопасности позволяет отслеживать заданные события.
Система Windows XP Professional регистрирует событие в журнале безопасности того компьютера, на котором событие происходит. Так, каждый раз, когда кто-нибудь пытается войти в систему и попытка входа оказывается неудачной, Windows XP Professional регистрирует событие в журнале безопасности данного компьютера.

Можно настроить политику аудита для данного компьютера на выполнение следующих операций:
• выявление успешных или неудачных действий, например попыток входа пользователей в систему или отдельного пользователя прочитать указанный файл, изменения учетной записи пользователя или принадлежности к группе, изменение параметров безопасности;
• исключение или минимизация риска неавторизованного использования ресурсов.
Для просмотра событий, зарегистрированных системой Window-XP Professional в журнале безопасности, используется утилита. Просмотр событий (Event Viewer). Можно также сохранять файлы журнала в архиве для выяснения закономерностей за указанный период времени — например, чтобы определить частоту использования принтеров или файлов или выявление попыток неавторизованного использования ресурсов.

Определение событий, подлежащих регистрации

При планировании политики аудита нужно определить, какие события следует регистрировать и на каких компьютерах надо установить аудит. По умолчанию аудит отключен.
Можно регистрировать следующие типы событий:
• попытки доступа к файлам и папкам;
• вход в систему и выход из нее;
• выключение компьютера с Windows XP Professional;
• запуск компьютера с Windows XP Professional;
• изменения учетных записей пользователей и групп;
• попытки изменения объектов Active Directory (только если компьютер с Windows XP Professional является частью домена).
После того как типы регистрируемых событий заданы, нужно определить, регистрировать ли успешные действия, неудавшиеся попытки или оба вида событий.
Отслеживание успешных действий дает информацию о том, как часто система Windows XP Professional или пользователи обращаются к конкретным файлам, принтерам или другим объектам; эта информация пригодится для планирования ресурсов.
Регистрация неудачных попыток позволяет выявить слабые места в защите системы. Так, если зафиксировано несколько неудачных попыток входа в систему под определенным именем пользователя, особенно в нерабочее время, можно предположить, что неавторизованный пользователь пытается «взломать» систему. Кроме того, при выборе политики аудита руководствуйтесь правилами:
• Определите, нужно ли отслеживать закономерности загрузки системы. Если да, то предусмотрите сохранение журналов событий в архиве. Это позволит контролировать распределение загрузки по времени и заранее планировать увеличение ресурсов системы.
• Часто просматривайте журналы безопасности. Обязательно установите расписание и регулярно просматривайте журналы безопасности, так как выполнение аудита само по себе не предупреждает о слабых местах в защите системы.
• Задайте информативную и работоспособную политику аудита. Всегда регистрируйте попытки доступа к жизненно важным и конфиденциальным данным. Регистрируйте только те события, которые дают существенную информацию. Это снижает потребление ресурсов компьютера до минимума и упрощает поиск нужной информации. Регистрация большого числа событий может привести к чрезмерной трате системных ресурсов Windows XP Professional.

Реализация политики аудита

Аудит — мощный инструмент для отслеживания событий, происходящих на компьютерах в вашем офисе. Прежде чем применять аудит, следует продумать требования к нему и установить политику аудита. Далее можно выполнять аудит файлов, папок и принтеров.

Конфигурация аудита

На компьютерах с Windows XP Professional политики аудита устанавливаются для каждого компьютера в отдельности. Для установки и администрирования аудита необходимо:
• иметь право пользователя Управление аудитом и журналом безопасности (Manage Auditing And Security Log) на том компьютере, на котором планируется установить политику аудита или просмотреть журнал безопасности. По умолчанию в Windows XP Professional такие права имеет группа Администраторы (Administrators);
• разместить файлы и папки, аудит которых планируется, на томах с файловой системой NTFS.

Настройка аудита

Настройка аудита выполняется в два этапа.
1. Задание политики аудита. Политика аудита разрешает аудит объектов, но не инициирует аудит заданных объектов.
2. Разрешения аудита заданных ресурсов. Для файлов, папок, принтеров и объектов Active Directory назначаются конкретные события, подлежащие регистрации. После этого Windows XP Professional начинает отслеживать заданные события и регистрировать их в журнале.

Установка политики аудита

На первом этапе установки политики аудита в Windows XP Professional необходимо выбрать типы событий, подлежащих регистрации. Для каждого регистрируемого события в параметрах указывается, какие попытки следует отслеживать — успешные или неудачные. Политика аудита на локальном компьютере устанавливается средствами оснастки ***Групповая политика***, которую можно запустить, используя ***Консоль управления MMC*** (Microsoft Management Console) и добавив в консоль оснастку ***Групповая политика*** (Group Policy). В таблице 4.1 перечислены типы событий, регистрируемые в Windows XP Professional.

Типы событий, регистрируемых Windows XP Professional

Таблица 4.1

Событие	Описание
Вход в систему под заданной учетной записью	Контроллер домена получил запрос на проверку учетной записи пользователя (применяется только в тех случаях, когда компьютер с Windows XP Professional входит в домен Microsoft Windows 2000)
Управление учетными записями	Администратор создал, изменил или удалил учетную запись пользователя или группу. Некоторая учетная запись была переименована, отключена или включена, или для нее был

	назначен или изменен пароль
Доступ к службе каталогов	Пользователь получил доступ к объекту Active Directory. Для регистрации событий этого типа нужно сконфигурировать аудит для определенных объектов Active Directory (Active Directory можно применять, только если компьютер с Windows XP Professional входит в домен Microsoft Windows 2000)
Вход в систему	Пользователь локально вошел в систему или вышел из нее, или подключился к компьютеру через сеть (или отключился от него)
Доступ к объектам	Пользователь получил доступ к файлу, папке или принтеру. Определенные файлы, папки или принтеры должны быть настроены для аудита. В этом случае регистрируется доступ пользователей к файлам, папкам и принтерам
Изменение системной политики	Изменены пользовательские параметры безопасности, права пользователя или политика аудита
Использование привилегий	Пользователь применил права, например изменил системное время (в этом случае не подразумеваются права, связанные с регистрацией в системе или с выходом из нее)
Отслеживание процесса	Программа выполнила действие. Эта информация важна главным образом для программистов, которым нужно детально проследить выполнение программы
Системные события	Пользователь перезапустил или выключил компьютер, или произошло событие, повлиявшее на безопасность Windows XP Professional или на журнал безопасности. Например, журнал аудита переполнился и Windows XP Professional начинает игнорировать поступающие сообщения о событиях

После настройки параметров политики аудита имейте в виду, что изменения в политике аудита компьютера вступают в силу только после перезагрузки.

Аудит доступа к файлам и папкам

Если требуется выявить слабые места в защите корпоративной сети, можно установить аудит файлов и папок, расположенных на разделах NTFS. Для аудита доступа пользователей к файлам и папкам прежде всего следует настроить политику аудита на регистрацию доступа к объектам, в том числе файлов и папок. При настройке политики аудита на регистрацию доступа к объектам включается аудит конкретных файлов и папок. При этом также указывается, какие виды доступа, пользователи и группы подлежат аудиту.
В таблице 4.2 перечислены действия пользователей, вызывающие соответствующие события, что поможет определить, в каких случаях следует включать аудит этих событий.

События для файлов и папок, вызываемые действиями пользователей.

Таблица 4.2

Событие	Действие пользователя, вызвавшее событие
Переход в папку/Выполнение файла (Traverse Folder/Execute File)	Запуск программы или получение доступа к папке при смене текущей папки
Получение списка файлов/Чтение данных (List Folder/Read Data)	Просмотр содержимого файла или папки

Чтение атрибутов (Read Attributes)	Просмотр атрибутов файла или папки
Чтение расширенных атрибутов (Read Extended Attributes)	Просмотр атрибутов файла или папки
Создание файлов/Запись данных (Create Files/Write Data)	Изменение содержимого файла или создание новых файлов в папке
Создание папок/Добавление данных (Create Folders/Append Data)	Создание папок внутри папок
Запись атрибутов (Write Attributes)	Изменение атрибутов файла или папки
Запись расширенных атрибутов (Write Extended Attributes)	Изменение атрибутов файла или папки
Удаление вложенных папок и файлов (Delete Subfolders And Files)	Удаление файла или папки внутри папки
Удаление (Delete)	Удаление файла или папки
Чтение разрешений (Read Permissions)	Просмотр прав владельца файла на файл или папку
Смена разрешений (Change Permissions)	Изменение прав доступа к файлу или папке
Смена владельца (Take Ownership)	Изменить право владельца на файл или папку

Аудит доступа к принтерам

При необходимости отслеживать использование конкретных принтеров включите аудит доступа к принтерам. Чтобы включить аудит доступа к принтерам, в политике аудита настройте аудит доступа к объектам, что включает принтеры.
Включите аудит конкретных принтеров и укажите, какие виды .доступа регистрировать и какие пользователи будут иметь доступ. Для выбранного принтера аудит включается в той же последовательности, что и при установке параметров аудита файлов и папок.
В таблице 4.3 перечислены события, аудит которых возможен для принтеров, и соответствующие этим событиям действия пользователей.

События для принтеров, вызываемые действиями пользователей. Таблица 4.3

Событие	Действие пользователя, вызвавшее событие
Печать (Print)	Печать файла
Управление принтерами (Manage Printers)	Изменение параметров принтера, приостановка печати, разрешение совместного использования принтера, удаление принтера из системы
Управление документами (Manage Documents)	Изменение параметров заданий; приостановка или продолжение печати, изменение порядкового номера в очереди или удаление документов; разрешение совместного использования принтера; изменение параметров принтера
Чтение разрешений (Read Permissions)	Просмотр прав доступа к принтеру
Смена разрешений (Change Permissions)	Изменение прав доступа к принтеру
Смена владельца	Смена владельца принтера

(Take Ownership)	

2.2. Изучите возможности управления журналом безопасности

Использование утилиты ***Просмотр событий*** (Event Viewer)
Утилита ***Просмотр событий*** (Event Viewer) применяется для решения различных задач администрирования, в том числе для просмотра журналов аудита, созданных в результате установки политики аудита и обновляемых при возникновении заданных событий.
Утилиту Просмотр событий (Event Viewer) также используют для просмотра содержимого файлов журнала безопасности и поиска конкретных событий в файлах журнала.
Утилита ***Просмотр событий*** (Event Viewer) необходима для просмотра информации, содержащейся в журналах (logs) Windows XP Professional. По умолчанию эта утилита позволяет просматривать три журнала (таблица 4.4).

Журналы Windows XP Professional.

Таблица 4.4

Журнал	Описание
Журнал приложений	Хранит сообщения об ошибках, предупреждения или информацию, генерируемые приложениями, например СУБД или программой электронной почты. События, регистрируемые в журнале, задаются разработчиками соответствующих программ
Журнал безопасности	Содержит информацию об успешных или неудачных попытках выполнения операций, аудит которых включен. Эти события регистрируются Windows XP Professional в соответствии с политикой аудита
Системный журнал	Хранит сообщения об ошибках, предупреждения и данные, генерируемые Windows XP Professional. События, регистрируемые в журнале, задаются в Windows XP Professional

Примечание Если установлены дополнительные службы, могут быть добавлены и соответствующие журналы событий.

Просмотр журнала безопасности

В журнале безопасности (security log) хранится информация о событиях, которые отслеживаются политикой аудита, например неудачные и успешные попытки регистрации в системе.
Windows XP Professional регистрирует события в журнале безопасности того компьютера, на котором событие произошло. Эти события можно просматривать с любого компьютера при наличии прав администратора на компьютере, на котором произошло событие. Чтобы просмотреть журнал безопасности удаленного компьютера, откройте консоль MMC и выберите просмотр событий удаленного компьютера.
При первом запуске утилиты ***Просмотр событий*** (Event Viewer) автоматически отображаются все события, зарегистрированные в выбранном журнале. Чтобы показать нужные события, можно использовать команду ***Фильтр*** (Filter). Кроме того, для поиска

конкретных событий применяют команду ***Найти*** (Find).
Чтобы выполнить отбор или поиск событий, запустите утилиту ***Просмотр событий*** (Event Viewer), затем в меню ***Вид*** (View) щелкните пункт ***Фильтр*** (Filter) или ***Найти*** (Find). Параметры в окнах фильтра и поиска практически не отличаются.
В таблице 4.5 перечислены параметры вкладки Фильтр (Filter), используемые для отбора нужных событий, и команды Найти (Find), применяемые для поиска нужных событий.

Управление журналами аудита

Сравнивая данные журналов, записанные в разное время, можно выявлять закономерности в работе Windows XP Professional. Их анализ позволяет определять загруженность ресурсов и планировать их расширение. Кроме того, в журналах фиксируются попытки неавторизованного использования ресурсов. Windows XP Professional позволяет изменять размер файлов журнала и задавать действия системы при переполнении журнала.

Параметры для фильтрации и поиска событий.

Таблица 4.5

Параметр	Описание
Типы событий (Event Types)	Типы событий для просмотра
Источник события (Event Source)	Программа или драйвер компонента, вызвавший событие
Категория (Category)	Тип события, например попытка входа, выхода или системное событие
Код события (Event ID)	Идентификационный номер события. Он упрощает сотрудникам службы технической поддержки контроль событий
Пользователь (User)	Имя учетной записи пользователя
Компьютер (Computer)	Имя компьютера
«С» и «До» (From and To)	Интервал времени, за который вы хотите просмотреть события [только на вкладке Фильтр (Filter)]
Восстановить значения по умолчанию (Restore Defaults)	Отменяет все изменения на этой вкладке и восстанавливает значения по умолчанию
Описание (Description)	Текстовый фрагмент в описании события (только в диалоговом окне Найти (Find)
Направление поиска (Search Direction)	Направление, в котором программа поиска будет просматривать журнал (вверх или вниз; только в диалоговом окне Найти (Find)
Найти далее (Find Next)	Программа поиска находит и отображает следующую запись, удовлетворяющую условиям поиска

Параметры каждого журнала аудита можно настраивать в отдельности. Чтобы изменить параметры журнала, выберите его название в окне утилиты ***Просмотр событий*** (Event Viewer), а затем в меню ***Действие*** (Action) щелкните пункт ***Свойства*** (Properties). В диалоговом окне ***Свойства*** (Properties) для каждого типа журнала аудита настраиваются следующие параметры:
- предельный размер каждого журнала, который может изменяться от 64 кбайт до 4194240 кбайт (4 Гбайт). По умолчанию размер журнала составляет 512 кбайт;
- действия Windows XP Professional при достижении файлом журнала предельного

размера. Чтобы настроить эти действия, выберите один из вариантов, перечисленных в таблице 4.6.

Архивация журналов

Архивация журналов безопасности позволяет вести учет событий, связанных с безопасностью. На многих предприятиях принято сохранять архивированные журналы в течение некоторого времени, чтобы иметь возможность просмотреть информацию по безопасности за требуемый период.
Чтобы сохранить, очистить или просмотреть архивированный журнал, выберите нужный журнал в окне утилиты ***Просмотр событий*** (Event Viewer) и выполните одно из действий, перечисленных в таблице 4.7.

Варианты обработки заполненных файлов журнала аудита.

Таблица 4.6

Параметр	Описание
Удалять старые события по необходимости (Overwrite Events As Needed)	Если установлен этот параметр, можно потерять информацию при переполнении журнала до того, как его сохранят. Однако при этом не требуется обслуживания
Удалять события, произошедшие более, чем V дней назад (Overwrite Events Older Than X Days)	Если установлен этот параметр, можно потерять информацию при переполнении журнала до того, как его сохранят. Будет утрачена только та информация, которая поступила более V дней назад. При применении этого параметра необходимо указать число дней (по умолчанию 7)
Не удалять события (Do Not Overwrite Events)	Если установлен этот параметр, нужно очищать журнал вручную. При заполнении журнала Windows XP Professional прекращает регистрацию событий, сохраняя уже имеющиеся записи журнала безопасности

Действия для архивации, очистки или просмотра файла журнала.

Таблица 4.7

Действие	Выполните
Сохранить журнал в архиве	Щелкните Сохранить файл журнала как (Save Log File As), затем введите имя файла
Очистить журнал	Для очистки журнала щелкните Стереть все события (Clear All Events). При этом Windows XP Professional генерирует запись в журнале безопасности о том, что журнал очищен
Просмотреть архиви-рованный журнал	Щелкните Новый вид журнала (New Log View); укажите вид выбранного журнала

Лабораторная работа № 23.

Тема: Службы Windows XP

Цель работы: научиться настраивать службы, оптимизировать производительность компьютера с помощью служб.

Теоретические сведения. Служба (Service) — это программа, работающая в фоновом режиме и обеспечивающая функционирование системы. Например, служба *Удаленный реестр* позволяет получить доступ к реестру данного компьютера с помощью сети.

Для управления службами служит оснастка *Службы*. Чтобы открыть ее, щелкните правой кнопкой на значке *Мой компьютер* и выберите команду *Управление*. В появившемся окне откройте элемент *Службы и приложения* и щелкните на значке *Службы.*

После запуска оснастки *Службы* можно просматривать список установленных служб, выполнять их запуск и остановку, изменять параметры работы выбранной службы. Все описанные действия можно выполнить в окне свойств конкретной службы, которое вызывается двойным щелчком на ее имени.

Окно свойств службы содержит несколько вкладок. На вкладке *Общие* можно узнать основные сведения о выбранной службе: имя, описание, файл службы и ее текущее состояние. Можно изменять состояние службы с помощью одной из кнопок: *Пуск, Стоп, Пауза, Продолжить.*

Вкладка *Вход в систему* позволяет задать имя учетной записи, с помощью которой запускается служба.

На вкладке *Восстановление* можно задавать действия при возникновении сбоев в работе службы. Об ошибках работы служб можно узнать, просматривая системный журнал.

Вкладка *Зависимости* позволяет определить для данной службы список служб, необходимых для ее работы, а также список зависимых служб.

Задание:

1. Просмотрите список установленных служб.

2. Просмотрите основные сведения о выбранной службе.

3. Просмотрите список служб, необходимых для работы выбранной службы.

Работающие службы используют память, процессорное время и другие ресурсы, что неизбежно сказывается на производительности компьютера. Отключив ненужные службы, можно несколько увеличить скорость работы системы.

Несколько аргументов против отключения служб:

· После отключения некоторых важных служб система может работать нестабильно, вплоть до полной остановки или перезагрузки.

· Многие службы используют минимум ресурсов, и их отключение не приведет к увеличению производительности.

Для повышения производительности можно отключить ненужные службы. Перед отключением выполните следующие действия.

· Создайте контрольную точку восстановления или выполните архивацию состояния системы.

· Посмотрите вкладку *Зависимости* в свойствах отключаемой службы. Если есть включенные службы, которые зависят от данной, то не следует отключать такую службу.

· Запомните текущее состояние службы, чтобы вернуть его при возникновении ошибок, связанных с ее отключением.

Рассмотрим службы, которые можно отключить, если система не нуждается в их функциях:

· Вторичный вход в систему — позволяет запускать программы от имени другого пользователя (команда *Запуск* от имени контекстного меню). При отключении службы эта функция будет недоступна.

· Диспетчер сеанса справки для удаленного рабочего стола — обеспечивает работу *Удаленного помощника*. При отключении данной службы *Удаленный помощник* будет недоступен.

· Серийный номер переносного медиа-устройства — можно отключить, если не используются подобные устройства.

· Служба индексирования — индексирует содержимое файлов для последующего быстрого поиска. Это одна из немногих служб, которая может существенно влиять на производительность недостаточно быстрых компьютеров. Рекомендуется отключать индексирование не остановкой данной службы, а с помощью соответствующих настроек панели поиска.

Не стоит без особой необходимости полностью отключать службы, имеющие тип запуска *Вручную*. Подобные службы запускаются по мере надобности, а в остановленном состоянии они не используют ресурсы.

Отключая сетевые службы, обязательно запишите их прежнее состояние, потому что придется вернуть его при необходимости подключить компьютер к сети или Интернету.

Задание:

1. Отключите службу *беспроводной настройки*.

2. Отключите службу *вторичного входа в систему*.

Вопросы к защите лабораторной работы:

1. Что такое *служба*?

2. Как открыть оснастку *Службы*?

3. Как открыть окно свойств конкретной службы?

4. Перечислите вкладки окна свойств конкретной службы и их назначение.

Лабораторная работа №24

Тема: Диагностика операционной системы

Задание:

1. Настройка системы с помощью команды msconfig

2. Средства диагностики системы, команда dxdiag

1. Настройка системы с помощью команды msconfig

- *Общие*

- Обычный запуск – загрузка всех драйверов устройств и запуск всех служб

- Диагностический запуск – загрузка только основных драйверов и запуск основных служб

- Выборочный запуск – какие файлы обрабатывать при запуске системы:

- Обрабатывать файл SYSTEM.INI

- Обрабатывать файл WIN.INI

- Загружать системные службы

- Загружать элементы автозагрузки

- *SYSTEM . INI* и *WIN . INI* - -Эти вкладки позволяют редактировать файлы конфигурирования добавляя или удаляя командные строки с помощью кнопок «Вверх», «Вниз», «Добавить» и «Удалить» Если галочка напротив строки стоит, то данная строка выполняется. Если галочка не стоит, то строка выводится как комментарий

- *Boot . ini* - В этой вкладке можно изменить содержание файла Autoexec.bat, поставив, галочки над той строкой, которую нужно загрузить. Если галочка напротив строки не стоит, то указанная в данной строке программа не загружается;

- *Службы* – позволяет включать и отключать службы;

- Автозагрузка – используется для загрузки системных элементов и программ при включении компьютера, а также для отмены их запуска при включении компьютера;

также можно отследить присутствие некоторых вирусов; поставив галочку над приложением, оно будет загружаться при включении компьютера, если галочка не стоит, то приложение грузиться не будет

- *Автозагрузка* – включает в себя множество программ, запускающихся при входе в систему, для включения элемента в процессе запуска нужно поставить флажок. Включает в себя 2 кнопки:

- Включить все – включает все элементы в списке

- Отключить все – отключает все элементы из списка

- Галочка напротив программы отвечает за запуск программы при загрузке системы

- Поставить какое-нибудь приложение в автозагрузку можно с помощью пункта “Автозагрузка” из меню “Пуск”. Для этого надо просто перетащить туда ярлык к программе

2. Средства диагностики системы, команда dxdiag

- *Система*

- Подробные сведения о системе: Информация о версии Windows, имя компьютера, выбранный язык, процессор, память, версия DirectX

- *Файлы DirectX*

- Сведения о файлах DirectX, об их версиях, атрибутах, языке, размере

- *Дисплей*

- Сведения о видеокарте и дисплее, а также выполнение проверки DirectDraw и Direct 3D

- *Музыка*

- Сведения о звуковой карте, ее драйвере и о музыкальных портах, а также выполнение проверки Directmusic

- *Ввод*

- Информация об устройствах ввода и о их драйверах

- *Сеть*

- Информация о сети, а также выполнение проверки Directplay

- *Если ничего не помогло*

- На этой странице находится список средств, которые могут помочь при устранении неполадок

Файл подкачки – это скрытый файл на жестком диске, для хранения частей программ, файлов и данных, не помещающихся в оперативной памяти. Оперативная память и файл подкачки составляют виртуальную память

Цифровая подпись – это способ обозначения авторства в сообщениях, файлах и других цифровых объектах. Процесс подписания цифровых данных вызывает их преобразования, а так же добавляет в тег подписи некоторых скрытых сведений, контролируемых создателем

Цифровая подпись драйверов – это подтверждения качества драйверов, эта подпись удостоверяет, то что драйвер прошел соответствующую проверку и не был изменен или заменен в процессе установки других программ

Если при проверки окажется что драйверы имеют цифровую подпись (MicrosoftWindowsHardwareQualityLabs), средство диагностики DirectXможет попытаться установить соединение через Интернет для загрузки новых сертификатов WHQL. При этом никакие данные с вашего компьютера не извлекается.

Лабораторная работа 25.

Тема: Восстановление операционной системы Windows XP

Цель: Научиться производить резервное архивирование и восстановление операционной системы.

Ход выполнения работы:

Теоретические сведения.

1. **Резервное копирование**

Многие программы-настройщики (иначе Твикеры) предлагают создать резервный диск восстановления Windows. То же предлагает сделать Антивирус Касперского, дабы восстановить работу Windows после серьезной вирусной атаки.

Можно заархивировать содержимое папки \Windows\System32\config через другой компьютер либо же с помощью загрузочной версии Windows, чтобы в случае появления сообщения "\Windows\System32\config файл поврежден" можно было его распаковать обратно и тем самым восстановить работу Windows.

Подобная ошибка появляется из-за повреждения кластеров, но может произойти из-за экстренного завершения работы.

При повреждении кластеров может помочь проверка на ошибки системного диска с исправлением ошибок, ее можно произвести с помощью другого компьютера, либо же Загрузочной версии Windows, но такой метод является экстренным и не желаемым,

поскольку Windows скрывает поврежденные кластера, вместо того чтобы восстанавливать их. В этом случае оптимальным вариантом будет использование HDD Regenerator'a, поскольку он именно восстанавливает поврежденные кластера.

В некоторых случаях на системном диске повреждается файл NTLDR (NT Loader). В следствие чего появляется сообщение: "NTLDR is Missing". Чтобы исправить данную ошибку в некоторые сборки Windows XP включается загрузочная программа "Исправить "NTLDR is Missing"". В Windows Vista / 7 данной ошибки не наблюдается в связи с отсутствием файла NTLDR, его заменяет BootMGR (Boot Manager).

Основные средства восстановления работоспособности:  HDD Regenerator Acronis Rescue Media Hiren's Boot CD Windows XP Portable Edition

Задание 1

2. **Резервное копирование реестра в Windows XP**

Способ 1.

Не используйте этот способ для экспорта всего реестра или его основных разделов, ☞таких как HKEY_CURRENT_USER и т.п.

Прежде, чем начать редактирование реестра вручную с помощью REGEDIT, или REG-файла не помешает **сохранить ту часть реестра**: раздел или подраздел, которую вы будете изменять. Для этого:

- Запустите REGEDIT. "**Пуск- Выполнить- REGEDIT**".
- Найдите ветвь реестра содержащую ключ значение которого вы будете редактировать и кликните на ней, в левой части окна **REGEDIT**.
- В главном меню выберите "**Файл-Экспорт**" и укажите имя файла. Либо кликните правой кнопкой и укажите "**Экспортировать**".

Альтернативный вышеприведенному способ состоит в том, что можно выполнить команду или командный файл определённого содержания. Например, сохраним настройки популярной программы Mozilla или Google:

Выполните

Для Mozilla:

Пуск – Выполнить – и введите команду:

Regedit /e mozilla1.reg HKEY_CURRENT_USER\Software\Mozilla\FireFox\ u

Regedit /e mozilla2.reg HKEY_LOCAL_MACHINE\Software\ Mozilla\FireFox

Вся необходимая информация будет помещена в файлы mozilla1.reg и mozilla2.reg.

Для Google

Regedit /e Chrome1.reg HKEY_CURRENT_USER\Software\ Google \Chrome\ u

Regedit /e Chrome2.reg HKEY_LOCAL_MACHINE\Software\ Google \Chrome

Вся необходимая информация будет помещена в файлы mozilla1.reg и mozilla2.reg.

Способ 2.

Для резервного копирования всего реестра используйте программу архивации данных "**Программы-Стандартные-Служебные-Архивация данных**" или просто введите команду *%SystemRoot%\system32\ntbackup.exe* в «**Пуск-Выполнить»**

В открывшемся окне нажмите кнопку **Далее**

В открывшемся окне поставьте галочку в пункте **Архивация файлов и параметров и** нажмите **Далее**

В открывшемся окне выберите пункт **Предоставить возможность выбора объектов для архивации и** нажмите **Далее.**

В открывшемся окне выберите папки или документы, которые должны быть заархивированы и нажмите **Далее**.

В открывшемся окне выберите место сохранения архива и нажмите **Далее и в новом окне нажмите Готово.** После нажатия кнопки Готово начнется процесс архивации.

Программа архивации позволяет архивировать и восстанавливать так называемые данные состояния системы, что включает в себя следующие системные компоненты:

- реестр;
- базу данных регистрации классов COM+
- загрузочные файлы : Ntldr и Ntdetect.com
- системные файлы;

Задание 2.

Пошаговые инструкции для архивации реестра Windows XP:

1. Войдите в систему с необходимыми правами, например, как администратор.
2. Запустите NTbackup ("Пуск – Стандартные – Служебные - Архивация данных").
3. Если NTbackup запустилась в режиме мастера, перейдите в "Расширенный режим".
4. Выберите закладку "Архивация".
5. В левом окне найдите и пометьте "птичкой" строку "Диск C:\Windows\System32".
6. Нажмите кнопку "Архивировать" и выберите "Дополнительно".
7. Снимите "галочку" с пункта "Автоматически архивировать защищенные системные файлы вместе с состоянием системы". Таким образом мы заархивируем только файлы реестра, что произойдёт быстро и займёт немного места на диске, примерно 17-20Мб.
8. На этой же вкладке "Тип архива" установите "Обычный".
9. "ОК" и нажмите "Архивировать". После архивации вы сможете просмотреть отчет.
10. Отчёты об архивации накапливаются в папке

x:\Documents and Settings\%User%\Local Settings\Application Data\Microsoft\Windows NT\NTBackup\data\

в пронумерованных файлах backup01.log, backup02.log и т.д.

NTbackup можно использовать и из командной строки, но мы не будем рассматривать этот способ, так как восстановить данные с командной строки нам не удастся и , кроме того, при архивации вместе с реестром будут заархивированы и все системные файлы, необходимые для загрузки Windows XP. А это потребует более долгого времени и займёт заметно больше места на жестком диске.

Восстановление реестра в Windows XP

В данном разделе мы практически повторим предыдущий, но с точки зрения восстановления реестра, а не архивации.

Задание 3

Способ 1.

При архивации части реестра, мы с помощью REGEDIT экспортировали данные в REG-файл. Теперь, чтобы извлечь их и восстановить исходный вид части реестра выполним следующие шаги:

1. Запустите REGEDIT. "Пуск-Выполнить-REGEDIT". 2. В главном меню выберите "Файл-Импорт" и укажите имя файла из задания 1 .

Или можно выполнить команду или командный файл определённого содержания. Например, восстановим настройки программы Mozilla:

Выбираем Пуск – Выполнить и вводим команду:

regedit -s mozilla1.reg regedit -s mozilla2.reg

Вся необходимая информация будет взята из файлов MOZILLA1.REG и MOZILLA2.REG.

Способ 2.

Пошаговые инструкции для полного восстановления реестра Windows XP:

1. Войдите в систему с необходимыми правами, например, как администратор.
2. Запустите NTbackup.
3. Если NTbackup запустилась в режиме мастера, нажмите кнопку "Расширенный" в окне мастера архивации.
4. Перейдите на вкладку "Восстановление и управление носителем"
5. Установите в списке "Установите флажки для всех объектов, которые вы хотите восстановить" флажок для объекта "Состояние системы". Это позволит восстановить данные состояния системы вместе с остальными данными, отмеченными в текущем задании восстановления.
6. Отчёты о проделанной работе находятся в папке x:

\Documents and Settings\%User%\Local Settings\Application Data\Microsoft\Windows NT\NTBackup\data\ в пронумерованных файлах типа backup01.log, backup02.log и т.д.

Восстановление повреждённого реестра когда Windows XP не загружается

А теперь мы посмотрим, что нужно делать, когда из-за ошибок в реестре Windows XP не загружается.

Описываемая процедура не гарантирует полное восстановление системы к предыдущему состоянию; однако, мы сможем восстановить наши данные.

Разрушенные файлы системного реестра могут вызывать ряд различных сообщений об ошибках.

Попробуйте при загрузке Windows XP нажать F8 и выбрать вариант "Загрузка последней удачной конфигурации" (Boot Using Last Known Good Configuration). При этом восстанавливаются только данные в разделе реестра HKLM\System\CurrentControlSet. Любые изменения в других разделах реестра сохраняются. Загрузка последней удачной конфигурации позволяет восстановить реестр в случае неполадок, вызванных, например, новым, несовместимым с имеющимся оборудованием, драйвером. Неполадки, возникшие вследствие повреждения или ошибочного удаления драйверов или файлов, не могут быть устранены таким образом.

Итак, при попытке запуска Windows XP вы получаете сообщение об ошибке, например, одно из указанных ниже:

Windows XP could not start because the following file is missing or corrupt:
\WINDOWS\SYSTEM32\CONFIG\SYSTEM

Windows XP could not start because the following file is missing or corrupt:
\WINDOWS\SYSTEM32\CONFIG\SOFTWARE

Stop: c0000218 {Registry File Failure} The registry cannot load the hive (file):
\SystemRoot\System32\Config\SOFTWARE or its log or alternate

Очень хорошо, теперь настала пора применить ваши знания на практике. Если вы когда-либо выполняли NTBACKUP и завершили системное копирование успешно, то вы можете сразу приступить к **4-ому шагу**.

Шаг 2.

Чтобы выполнить процедуру, описанную в этом разделе, вы должны войти как администратор, или как пользователь приравненный к администратору. Т.е. пользователь имеющий учетную запись в группе Администраторы.

Выполняем следующие действия:

1. Перегрузите компьютер.
2. При загрузке Windows XP нажмите F8.
3. Выберите безопасный режим.

Если вы используете проводник в качестве файл-менеджера, то придётся выполнить несколько действий, чтобы сделать папку System Restore видимой:

1. Запускаем "Проводник".
2. В меню "Сервис" выбираем "Свойства папки" и далее закладку "Вид".

3. Раскрываем опцию "Скрытые файлы и папки" и щёлкаем на "Показывать скрытые файлы и папки".
4. Далее щёлкаем на "Применить" и "Ок".

Теперь:

1. Открываем раздел жёсткого диска где установлена Windows XP и находим папку System Volume Information. Примечание: Это скрытая системная папка. Она содержит одну или более папок с именами вида _restore {GUID} , например, _restore{87BD3667-3246-476B-923F-F86E30B3E7F8}
2. Откройте папку, которая была создана НЕ в текущее время. Это может быть одна или больше папок, имена которых начинаются с "RP". Это - точки восстановления.
3. Откройте выбранную папку и затем папку с именем Snapshot. Например, c:\System Volume Information_restore{DBB3294C-F5C9-43A9-9010-A75010CD2631}\RP2\snapshot
4. Из папки Snapshot в папку C:\Windows\Tmp, уже созданную на первом этапе, скопируйте следующие файлы:

- REGISTRY_USER_.DEFAULT
- REGISTRY_MACHINE_SECURITY
- REGISTRY_MACHINE_SOFTWARE
- REGISTRY_MACHINE_SYSTEM
- REGISTRY_MACHINE_SAM

Эти файлы созданы службой восстановления системы - System Restore. Так как на предыдущем шаге мы использовали файлы системного реестра, созданные при начальной установке Windows XP, то этот "новый" системный реестр не знает, что "старые" точки восстановления существуют и доступны. При загрузке Windows XP создана новая папка с новым GUID и с новым System Volume Information, и создана новая точка восстановления, которая включает копию файлов нового системного реестра. Вот почему важно не использовать самую новую папку, особенно, если время ёе создания - текущее время.

Таким образом конфигурация существующей системы не знает о предыдущих точках восстановления. Нам нужна предыдущая, "старая" копия системного реестра от предыдущей, "старой" точки восстановления, чтобы сделать все предыдущие, "старые" точки восстановления доступными. Я надеюсь, что вы меня поняли.

Файлы системного реестра были скопированы из папки Snapshot в папку C:\Windows\Tmp чтобы сделать их доступными, когда мы будем находиться в Recovery Console. Мы будем использовать эти файлы, чтобы заменить ими файлы текущего системного реестра в папке C:\Windows\System32\Config. Дело в том, что в Recovery Console папка с System Volume Information в общем случае недоступна.

Контрольные вопросы:

1. Перечислите программы для создания резервных копий и восстановления данных. Опишите основные возможности данных программ.

Практические работы по дисциплине

ОП.04 Операционные системы

Практическое занятие № 1

Тема: Создание образа операционной системы

Цель занятия: изучить структуру операционной системы, назначение ее отдельных компонентов, функции операционных систем, их классификацию и критерии оценки.

Задание: составить конспект основных теоретических положений.

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Операционная система (operating system) – комплекс программ, предоставляющий пользователю удобную среду для работы с компьютерным оборудованием.

Операционная система позволяет запускать пользовательские программы; управляет всеми ресурсами компьютерной системы – процессором (процессорами), оперативной памятью, устройствами ввода вывода; обеспечивает долговременное хранение данных в виде файлов на устройствах внешней памяти; предоставляет доступ к компьютерным сетям.

Для более полного понимания роли операционной системы рассмотрим составные компоненты любой вычислительной системы (рис.1.1).

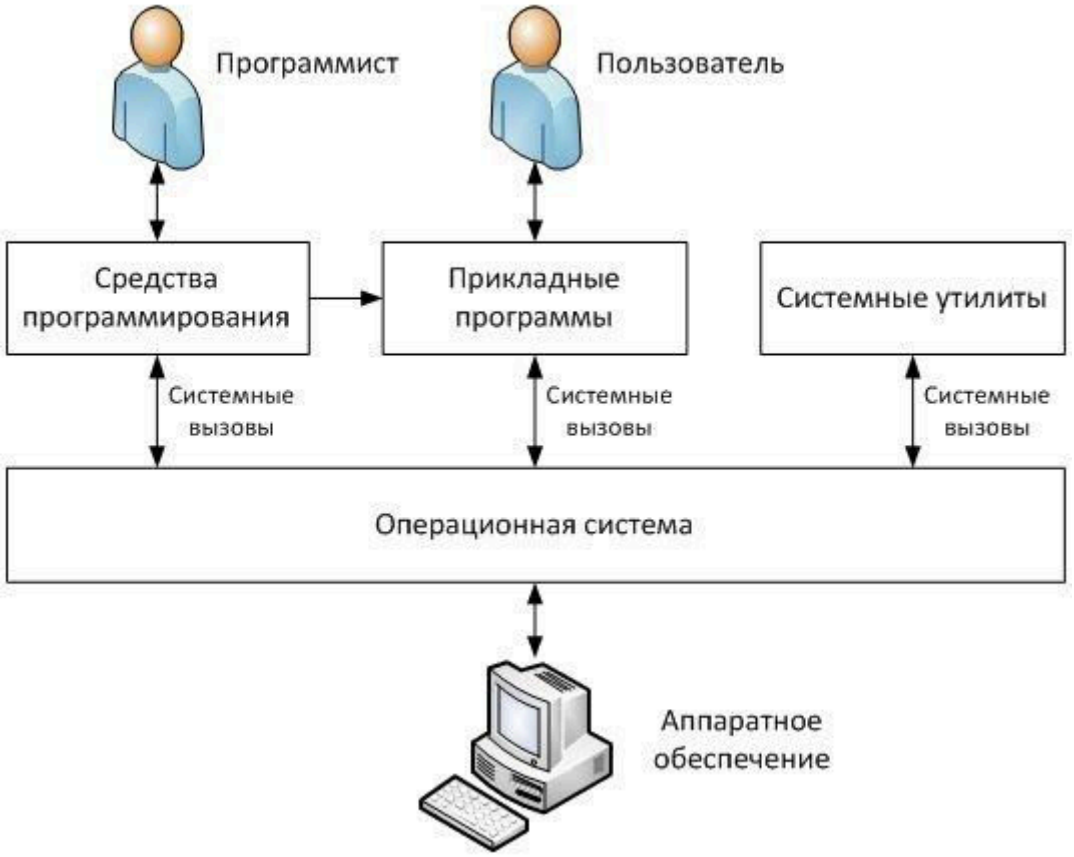


Рис. 1.1. Компоненты вычислительной системы

Все компоненты можно разделить на два больших класса – программы или программное обеспечение (ПО, software) и оборудование или аппаратное обеспечение (hardware). Программное обеспечение делится на прикладное, инструментальное и системное.

Цель создания вычислительной системы – решение задач пользователя. Для решения определенного круга задач создается прикладная программа (приложение, application). Примерами прикладных программ являются текстовые редакторы и процессоры (Блокнот, Microsoft Word), графические редакторы (Paint, Microsoft Visio), электронные таблицы (Microsoft Excel), системы управления базами данных (Microsoft Access, Microsoft SQL Server), браузеры (Internet Explorer) и т. п. Все множество прикладных программ называется прикладным программным обеспечением (application software).

Создается программное обеспечение при помощи разнообразных средств программирования (среды разработки, компиляторы, отладчики и т. д.), совокупность которых называется инструментальным программным обеспечением. Представителем инструментального ПО является среда разработки Microsoft Visual Studio.

Основным видом системного программного обеспечения являются операционные системы. Их основная задача – обеспечить интерфейс (способ взаимодействия) между пользователем и приложениями с одной стороны, и аппаратным обеспечением с другой. К системному ПО относятся также системные утилиты – программы, которые выполняют строго определенную функцию по обслуживанию вычислительной системы, например, диагностируют состояние системы, выполняют дефрагментацию файлов на диске, осуществляют сжатие (архивирование) данных. Утилиты могут входить в состав операционной системы.

Взаимодействие всех программ с операционной системой осуществляется при помощи системных вызовов (system calls) – запросов программ на выполнение операционной системой необходимых действий. Набор системных вызовов образует API – Application Programming Interface (интерфейс прикладного программирования).

Функции операционной системы

К основным функциям, выполняемым операционными системами, можно отнести:

- обеспечение выполнения программ – загрузка программ в память, предоставление программам процессорного времени, обработка системных вызовов;
- управление оперативной памятью – эффективное выделение памяти программам, учет свободной и занятой памяти;
- управление внешней памятью – поддержка различных файловых систем;
- управление вводом-выводом – обеспечение работы с различными периферийными устройствами;
- предоставление пользовательского интерфейса;
- обеспечение безопасности – защита информации и других ресурсов системы от несанкционированного использования;
- организация сетевого взаимодействия.

Структура операционной системы

Современные процессоры имеют минимум два режима работы – привилегированный (supervisor mode) и пользовательский (user mode).

Отличие между ними заключается в том, что в пользовательском режиме недоступны команды процессора, связанные с управлением аппаратным обеспечением, защитой оперативной памяти, переключением режимов работы процессора. В привилегированном режиме процессор может выполнять все возможные команды.

Приложения, выполняемые в пользовательском режиме, не могут напрямую обращаться к адресным пространствам друг друга – только посредством системных вызовов.

Все компоненты операционной системы можно разделить на две группы – работающие в привилегированном режиме и работающие в пользовательском режиме, причем состав этих групп меняется от системы к системе.

Основным компонентом операционной системы является ядро (kernel). Функции ядра могут существенно отличаться в разных системах; но во всех системах ядро работает в привилегированном режиме (который часто называется режим ядра, kernel mode).

Ядро - это основная, «самая системная» часть операционной системы. Имеются разные определения ядра. Согласно одному из них, ядро — это ***резидентная*** часть системы, т.е. к ядру относится тот программный код, который постоянно находится в памяти в течение всей работы системы. Остальные модули ОС являются ***транзитными***, т.е. подгружаются в память с диска по мере необходимости на время своей работы. К транзитным частям системы относятся:

- ***утилиты*** (utilities) — отдельные системные программы, решающие частные задачи, такие как форматирование и проверку диска, поиск данных в файлах, мониторинг (отслеживание) работы системы и многое другое;
- ***системные библиотеки подпрограмм***, позволяющие прикладным программам использовать различные специальные возможности, поддерживаемые системой (например, библиотеки для графического вывода, для работы с мультимедиа и т.п.);
- ***интерпретатор команд*** — программа, выполняющая ввод команд пользователя, их анализ и вызов других модулей для выполнения команд;
- ***системный загрузчик*** — программа, которая при запуске ОС (например, при включении питания) обеспечивает загрузку системы с диска, ее инициализацию и старт;
- другие виды программ, в зависимости от конкретной системы.

Особую роль в структуре системы играют ***драйверы устройств***. Эти программы, предназначенные для обслуживания конкретных периферийных устройств, несомненно, можно отнести к ядру системы: они почти всегда являются резидентными и работают в режиме ядра. Но в отличие от самого ядра, которое изменяется только при появлении новой версии ОС, набор используемых драйверов весьма мобилен и зависит от набора устройств, подключенных к данному компьютеру. В некоторых системах (например, в ранних версиях UNIX) для подключения нового драйвера требовалось перекомпилировать все ядро. В большинстве современных ОС драйверы подключаются к ядру в процессе загрузки системы, а иногда разрешается даже загрузка и выгрузка драйверов в ходе работы системы.

В качестве программного интерфейса системы, т.е. средств для обращения прикладных программ к услугам ОС, используется документированный набор ***системных вызовов*** или ***функций API*** (Applied Programming Interface). Между этими двумя терминами есть некоторая разница. Под системными вызовами понимаются функции, реализуемые непосредственно программами ядра системы. При их выполнении происходит переход из режима пользователя в режим ядра, а затем обратно. В отличие от этого, API-функции определяются как функции, описанные в документации ОС, независимо от того, выполняются ли они ядром или же системными библиотеками, работающими в режиме пользователя. В Windows часто несколько разных API-функций обращаются к одному и тому же недокументированному системному вызову, но имеют различные обрамляющие части, работающие в режиме пользователя.

Там, где различие между двумя этими понятиями несущественно, можно использовать нейтральный термин «***системные функции***».

Существует два основных вида ядер – монолитные ядра (monolithic kernel) и микроядра (microkernel). В монолитном ядре реализуются все основные функции операционной системы, и оно является, по сути, единой программой, представляющей собой совокупность процедур. В микроядре остается лишь минимум функций, который должен быть реализован в привилегированном режиме: планирование потоков, обработка прерываний, межпроцессное взаимодействие. Остальные функции операционной системы по управлению приложениями,

памятью, безопасностью и пр. реализуются в виде отдельных модулей в пользовательском режиме.

Ядра, которые занимают промежуточные положение между монолитными и микроядрами, называют гибридными (hybrid kernel).

Примеры различных типов ядер:

- монолитное ядро – MS-DOS, Linux, FreeBSD;
- микроядро – Mach, Symbian, MINIX 3;
- гибридное ядро – NetWare, BeOS, Syllable.

Кроме ядра в привилегированном режиме (в большинстве операционных систем) работают драйверы (driver) – программные модули, управляющие устройствами.

В состав операционной системы также входят:

- системные библиотеки (system DLL – Dynamic Link Library, динамически подключаемая библиотека), преобразующие системные вызовы приложений в системные вызовы ядра;
- пользовательские оболочки (shell), предоставляющие пользователю интерфейс – удобный способ работы с операционной системой.

Пользовательские оболочки реализуют один из двух основных видов пользовательского интерфейса:

- текстовый интерфейс (Text User Interface, TUI), другие названия – консольный интерфейс (Console User Interface, CUI), интерфейс командной строки (Command Line Interface, CLI);
- графический интерфейс (Graphic User Interface, GUI).

Пример реализации текстового интерфейса в Windows – интерпретатор командной строки cmd.exe; пример графического интерфейса – Проводник Windows (explorer.exe).

Классификация операционных систем

Классификацию операционных систем можно осуществлять несколькими способами.

1. По способу организации вычислений:

- системы пакетной обработки (batch processing operating systems) – целью является выполнение максимального количества вычислительных задач за единицу времени; при этом из нескольких задач формируется пакет, который обрабатывается системой;
- системы разделения времени (time-sharing operating systems) – целью является возможность одновременного использования одного компьютера несколькими пользователями; реализуется посредством поочередного предоставления каждому пользователю интервала процессорного времени;
- системы реального времени (real-time operating systems) – целью является выполнение каждой задачи за строго определённый для данной задачи интервал времени.

2. По типу ядра:

- системы с монолитным ядром (monolithic operating systems);
- системы с микроядром (microkernel operating systems);
- системы с гибридным ядром (hybrid operating systems).

3. По количеству одновременно решаемых задач:

- однозадачные (single-tasking operating systems);
- многозадачные (multitasking operating systems).

4. По количеству одновременно работающих пользователей:
- однопользовательские (single-user operating systems);
- многопользовательские (multi-user operating systems).

5. По количеству поддерживаемых процессоров:
- однопроцессорные (uniprocessor operating systems);
- многопроцессорные (multiprocessor operating systems).

6. По поддержке сети:
- локальные (local operating systems) – автономные системы, не предназначенные для работы в компьютерной сети;
- сетевые (network operating systems) – системы, имеющие компоненты, позволяющие работать с компьютерными сетями.

7. По роли в сетевом взаимодействии:
- серверные (server operating systems) – операционные системы, предоставляющие доступ к ресурсам сети и управляющие сетевой инфраструктурой;
- клиентские (client operating systems) – операционные системы, которые могут получать доступ к ресурсам сети.

8. По типу лицензии:
- открытые (open-source operating systems) – операционные системы с открытым исходным кодом, доступным для изучения и изменения;
- проприетарные (proprietary operating systems) – операционные системы, которые имеют конкретного правообладателя; обычно поставляются с закрытым исходным кодом.

9. По области применения:
- операционные системы мэйнфреймов – больших компьютеров (mainframe operating systems);
- операционные системы серверов (server operating systems);
- операционные системы персональных компьютеров (personal computer operating systems);
- операционные системы мобильных устройств (mobile operating systems);
- встроенные операционные системы (embedded operating systems);
- операционные системы маршрутизаторов (router operating systems).

Требования к операционным системам

Основное требование, предъявляемое к современным операционным системам – выполнение функций, перечисленных выше в параграфе "Функции операционных систем". Кроме этого очевидного требования существуют другие, часто не менее важные:

- расширяемость – возможность приобретения системой новых функций в процессе эволюции; часто реализуется за счет добавления новых модулей;
- переносимость – возможность переноса операционной системы на другую аппаратную платформу с минимальными изменениями;
- совместимость – способность совместной работы; может иметь место совместимость новой версии операционной системы с приложениями, написанными для старой версии, или совместимость разных операционных систем в том смысле, что приложения для одной из этих систем можно запускать на другой и наоборот;
- надежность – вероятность безотказной работы системы;
- производительность – способность обеспечивать приемлемые время решения задач и время реакции системы.

Предыстория ОС

Вскоре после того, как в конце 40-х годов XX века были созданы первые электронные компьютеры, очень остро встала проблема повышения эффективности использования оборудования, и прежде всего центрального процессора.

Типичный компьютер первого — второго поколений представлял собой большую комнату, уставленную шкафами и увитую кабелями. Каждое из основных устройств — центральный процессор, оперативная память, накопители на магнитных лентах, устройства ввода с перфокарт, принтер — занимало один или несколько «шкафов» или «тумб», наполненных радиолампами и механическими частями.

Все это стоило больших денег, потребляло бешеное количество электроэнергии и регулярно ломалось.

В таких условиях машинное время стоило очень дорого. Тем не менее, обычная практика использования ЭВМ не способствовала экономии. Как правило, программист, разрабатывающий программу, заказывал ежедневно несколько часов машинного времени и в течение этого времени монопольно использовал машину. Выполнив очередной запуск отлаживаемой программы (которую надо было каждый раз вводить либо с клавиатуры, либо, в лучшем случае, с перфокарт), пользователь получал распечатку (чаще всего в виде массива цифр), анализировал результаты, вносил изменения в программу и снова запускал ее. Таким образом, в ходе сеанса отладки дорогостоящее оборудование простаивало 99% времени, пока программист осмысливал результаты и работал с устройствами ввода/вывода. Кроме того, сбой при вводе одной перфокарты мог потребовать начать сначала всю работу программы.

Возникла великая идея — использовать сам компьютер для повышения эффективности работы с ним же.

Одно из ответвлений этой идеи — создание языков и систем программирования — рассматривается в отдельных курсах. Другим важным шагом стало возложение на специальную компьютерную программу части тех функций, которые до этого выполнял оператор или сам программист.

Программы такого рода назывались обычно ***мониторами*** (не путать с монитором как устройством вывода, который в то время был редчайшей экзотикой!). Монитор принимал команды, состоящие, как правило, из 1-2 букв названия и 1-3 аргументов, заданных 8-ричными или 16-ричными числами. Типичными командами были, например:

- загрузка данных с перфокарт по указанному адресу памяти;
- просмотр и корректировка (с пишущей машинки) значений в указанном диапазоне адресов;
- пошаговое выполнение программы с выдачей результатов каждой команды на пишущую машинку;
- запуск программы с указанного адреса с заданием адресов контрольных точек остановки.

Несмотря на убогость, по нынешним меркам, подобных средств, они в свое время значительно повысили производительность работы программистов. Однако кардинального повышения загрузки процессора не произошло.

Временем широкого распространения мониторов в мире были 50-е годы прошлого века (в СССР — 60-е годы). В настоящее время нечто подобное можно встретить на самых примитивных микропроцессорных контроллерах.

Пакетные ОС

Историю собственно ОС можно начать с появления в конце 50-х годов первых систем, организующих работу по пакетному принципу.

Важнейшим организационным изменением, происшедшим на этом этапе развития, стало массовое изгнание программистов из машинных залов, как фактора, лишь вносящего сумятицу в работу.

Теперь от программиста требовалось собрать пакет перфокарт, содержащий его программу, данные к ней, а также управляющие перфокарты. Эти карты на специально разработанном ***языке управления заданиями*** (JCL, Job Control Language) объясняли операционной системе, чье это задание, что нужно сделать с программой (например, передать ее транслятору с Фортрана), что предпринять в случае успешной трансляции (вероятно, пустить на решение), что — при наличии ошибок (например, перейти к другой программе), откуда взять исходные данные (например, с такого-то цилиндра магнитного диска). Кроме того, там могли быть даже указания на то, сколько метров бумаги можно выделить на распечатку и какое максимальное время может занять работа программы.

Обойтись без столь подробных инструкций было нельзя, потому что программист не присутствовал при запуске задания и не мог вмешаться лично.

Подготовленный пакет передавался, вместе с другими подобными пакетами, оператору ЭВМ, перед которым стояли две основные задачи: чтобы в устройстве ввода не переводились пакеты заданий и чтобы в принтере не кончилась бумага. Когда процессор заканчивал обработку задания и печать его результатов, он вводил следующий пакет и приступал к его обработке. Так достигалась основная цель пакетного режима — исключить простои процессора из-за нерасторопности людей.

В скором времени разработчики ОС осознали, что вычерпаны далеко не все резервы повышения загрузки процессора. Операции ввода и печати требовали лишь очень небольшой доли от полной производительности процессора. Кроме того, в ходе работы программы случались обращения к периферийным устройствам (например, к магнитным лентам и, позднее, дискам), при выполнении которых процессор опять простаивал. Целесообразно было найти способ, чтобы в эти периоды ожидания загрузить процессор другой работой. Но для этого необходимо, чтобы в памяти процессора находились сразу несколько программ, тогда ОС смогла бы переключать процессор на выполнение той программы, которая в данный момент может работать.

Такая организация работы, когда в памяти находятся несколько программ и система в определенные моменты переключает выполнение с одной программы на другую, была названа ***мультипрограммированием***. Эта важная идея в разных воплощениях пережила те пакетные системы, в которых она впервые была реализована, и является основой для функционирования практически всех современных ОС.

Среди наиболее развитых пакетных ОС с мультипрограммированием нельзя не назвать OS/360, основную ОС знаменитого в 60-70 гг. семейства ЭВМ IBM 360/370.

ОС с разделением времени

На рубеже 60-70 гг. распространенным и не слишком дорогим периферийным устройством становятся мониторы (сначала монохромные и работающие только в текстовом режиме). При этом процессор и ОЗУ остаются самыми дорогими и громоздкими устройствами вычислительной системы. В этих условиях возникает и быстро приобретает популярность принципиально новый тип ОС — ***системы с разделением времени***.

К одной ЭВМ подключается несколько десятков рабочих мест, оборудованных дисплеем (монитор + клавиатура) и совместно использующих вычислительные ресурсы ЭВМ. Процессорное время делится на кванты длительностью в несколько десятков миллисекунд и по истечении каждого кванта процессор может быть переключен на обслуживание другого процесса, другого дисплея. Поскольку теперь подготовку текстов программ выполняют сами программисты за дисплеями, а работа по редактированию текста требует очень малых затрат процессорного времени, процессор успевает обслужить все рабочие места практически без ощутимой задержки. Большая часть времени процессора уделяется небольшому числу рабочих

мест, где в данный момент запущены на выполнение программы. При этом, разумеется, средняя скорость работы каждой программы уменьшается, по крайней мере во столько раз, сколько программ выполняется одновременно.

Режим разделения времени стал огромным облегчением для программистов, которые вновь смогли в некоторой степени почувствовать себя «хозяевами» ЭВМ и получили возможность запускать программы на трансляцию и отладку хоть каждые 5 минут. Это позволило сократить сроки разработки и отладки программ.

Для трудоемких вычислительных заданий, предусматривающих счет по ранее отлаженным программам, режим разделения времени менее эффективен, чем пакетный, поскольку частое переключение процессора между выполняемыми программами требует дополнительных затрат времени.

Системы разделения времени используются в режиме диалога с пользователем, поэтому вместо громоздких, детализированных операторов JCL в них используются более простые команды, выполняющие элементарные действия — запуск программы, выдача на экран файла или каталога, копирование или удаление файла и т.п. Пользователю не нужно предвидеть заранее все возможные исходы выполнения команды, гораздо проще увидеть результат выполнения на экране и после этого принять решение, какую команду выполнять следующей. В то же время, некоторые часто повторяющиеся последовательности команд удобно описать один раз в виде «пакетного задания» и затем использовать при необходимости. В этом плане системы разделения времени сохраняют те удобные возможности, которые предоставляли пакетные системы.

Первоначально в качестве аппаратной основы систем разделения времени должны были использоваться «большие» ЭВМ, которые позднее стало принято называть «мейнфреймами» (mainframes). Позднее, по мере прогресса вычислительной техники, это стало по плечу даже миниЭВМ (так назывался в те годы класс компьютеров, занимавших всего лишь один-два небольших шкафчика). Следует особо упомянуть серию миниЭВМ PDP-11, имевшую широчайшее распространение во всем мире в течение полутора десятков лет.

Этот период (70-е годы в мире, 80-е в СССР) характерен глубоким развитием теории и практики создания мощных ОС, содержащих развитые средства управления процессами и памятью, реализующих многопользовательский режим работы. Из большого числа подобных систем особого упоминания заслуживает UNIX — единственная система, благополучно дожившая до нашего времени.

Однозадачные ОС для ПЭВМ

В середине 70-х годов был изобретен микропроцессор, а к началу 80-х микропроцессоры стали догонять по функциональным характеристикам ранее использовавшиеся «большие» процессоры. Эта ситуация сделала почти бесполезным режим разделения времени: зачем делить один процессор между многими задачами и многими пользователями, если проще и дешевле дать отдельный микропроцессор каждому пользователю? Разделение времени осталось целесообразным разве что в отношении суперкомпьютеров.

Появление и бурное распространение персональных компьютеров (ПК) вызвало к жизни новое поколение ОС, которые оказались во много раз проще своих предшественниц. Ненужной оказалась многопользовательская защита. На первых порах показалась ненужной и многозадачность. Все это можно было расценить как явный регресс в развитии ОС.

Наиболее популярной ОС для ранних восьмиразрядных ПК была система CP/M известной тогда фирмы Digital Research, однако с появлением в начале 80-х знаменитой машины IBM PC лидерство было прочно перехвачено системой MS-DOS фирмы Microsoft.

Многозадачные ОС для ПК с графическим интерфейсом

Быстрое развитие технологии привело к тому, что к концу 80-х годов ПК оказались в состоянии решать значительно более сложные и трудоемкие задачи, чем раньше. При этом

многие из достижений прежних этапов развития ОС оказались вновь востребованными, но теперь уже в новых условиях, среди которых надо назвать резкое повышение мощности процессоров и объема памяти, появление высококачественных графических мониторов и развитие сетевых технологий.

Стала реальной такая вещь, как многозадачная ОС для ПК. На смену ОС, которые выполняли текстовые команды, вводимые пользователем с клавиатуры, пришли системы, в которых взаимодействие с пользователем основано на использовании GUI (Graphical User Interface, графический интерфейс пользователя).

Значительная часть ПК работает в составе локальных вычислительных сетей. Это привело к тому, что вопросы защиты данных пользователя вновь приобрели первостепенное значение.

Критерии оценки ОС

При сравнительном рассмотрении различных ОС в целом или их отдельных подсистем возникает вечный вопрос — какая из них лучше и почему, какая архитектура системы предпочтительнее, какой из алгоритмов эффективнее, какая структура данных удобнее и т.п.

Очень редко можно дать однозначный ответ на подобные вопросы, если речь идет о практически используемых системах. Система или ее часть, которая хуже других систем во всех отношениях, просто не имела бы права на существование. На самом деле имеет место типичная многокритериальная задача: имеется несколько важных критериев качества, и система, опережающая прочие по одному критерию, обычно уступает по другому критерию. Сравнительная важность критериев зависит от назначения системы и условий ее работы.

1. Надежность

Этот критерий вообще принято считать самым важным при оценке программного обеспечения, и в отношении ОС его действительно принимают во внимание в первую очередь.

Под надежностью понимается, прежде всего, ее ***живучесть*** операционной системы, т.е. способность сохранять хотя бы минимальную работоспособность в условиях аппаратных сбоев и программных ошибок. Высокая живучесть особенно важна для ОС компьютеров, встроенных в аппаратуру, когда вмешательство человека затруднено, а отказ компьютерной системы может иметь тяжелые последствия.

Во-вторых, способность, как минимум, диагностировать, а как максимум, компенсировать хотя бы некоторые типы аппаратных сбоев. Для этого обычно вводится избыточность хранения наиболее важных данных системы.

В-третьих, ОС не должна содержать собственных (внутренних) ошибок. Это требование редко бывает выполнимо в полном объеме (программисты давно сумели доказать своим заказчикам, что в любой большой программе всегда есть ошибки, и это в порядке вещей), однако следует хотя бы добиться, чтобы основные, часто используемые или наиболее ответственные части ОС были свободны от ошибок.

Наконец, к надежности системы следует отнести ее способность противодействовать явно неразумным действиям пользователя. Обычный пользователь должен иметь доступ только к тем возможностям системы, которые необходимы для его работы. Если же пользователь, даже действуя в рамках своих полномочий, пытается сделать что-то очень странное (например, отформатировать системный диск), то самое малое, что должна сделать ОС, это переспросить пользователя, уверен ли он в правильности своих действий.

2. Эффективность

Эффективность любой программы определяется двумя группами показателей, которые можно обобщенно назвать «время» и «память». При разработке системы приходится принимать много непростых решений, связанных с оптимальным балансом этих показателей.

Важнейшим показателем временно́й эффективности является ***производительность*** системы, т.е. усредненное количество полезной вычислительной работы, выполняемой в единицу времени. С другой стороны, для диалоговых ОС не менее важно ***время реакции*** системы на действия пользователя. Эти показатели могут в некоторой степени противоречить друг другу. Например, в системах разделения времени увеличение кванта времени повышает производительность (за счет сокращения числа переключений процессов), но ухудшает время реакции.

В программировании известна аксиома: выигрыш во времени достигается за счет проигрыша в памяти, и наоборот. Это в полной мере относится к ОС, разработчикам которых постоянно приходится искать баланс между затратами времени и памяти.

Забота об эффективности долгое время стояла на первом месте при разработке программного обеспечения, и особенно ОС. К сожалению, оборотной стороной стремительного увеличения мощности компьютеров стало ослабление интереса к эффективности программ. В настоящее время эффективность является первостепенным требованием разве что в отношении систем реального времени.

3. Удобство

Этот критерий наиболее субъективен. Можно предложить, например, такой подход: система или ее часть удобна, если она позволяет легко и просто решать те задачи, которые встречаются наиболее часто, но в то же время содержит средства для решения широкого круга менее стандартных задач (пусть даже эти средства не столь просты). Пример: такое частое действие, как копирование файла, должно выполняться при помощи одной простой команды или легкого движения мыши; в то же время для изменения разделов диска не грех почитать руководство, поскольку это может понадобиться даже не каждый год.

Разработчики каждой ОС имеют собственные представления об удобстве, и каждая ОС имеет своих приверженцев, считающих именно ее идеалом удобства.

4. Масштабируемость

Термин «масштабируемость» (scalability) означает возможность настройки системы для использования в разных вариантах, в зависимости от мощности вычислительной системы, от набора конкретных периферийных устройств, от роли, которую играет конкретный компьютер (сервер, рабочая станция или изолированный компьютер) от назначения компьютера (домашний, офисный, исследовательский и т.п.).

Гарантией масштабируемости служит продуманная модульная структура системы, позволяющая в ходе установки системы собирать и настраивать нужную конфигурацию. Возможен и другой подход, когда под общим названием объединяются, по сути, разные системы, обеспечивающие в разумных пределах программную совместимость. Примером могут служить версии Windows NT/2000/XP, Windows 95/98 и Windows CE.

В некоторых случаях фирмы, производящие программное обеспечение, искусственно отключают в более дешевых версиях системы те возможности, которые на самом деле реализованы, но становятся доступны, только если пользователь покупает лицензию на более дорогую версию. Но это уже вопрос, связанный не с технической стороной дела, а с маркетинговой политикой.

5. Способность к развитию

Чтобы сложная программа имела шансы просуществовать долго, в нее изначально должны быть заложены возможности для будущего развития.

Одним из главных условий способности системы к развитию является хорошо продуманная модульная структура, в которой четко определены функции каждого модуля и его взаимосвязи с другими модулями. При этом создается возможность совершенствования отдельных модулей с минимальным риском вызвать нежелательные последствия для других частей системы.

Важным требованием к развитию ОС является ***совместимость версий снизу вверх***, означающая возможность безболезненного перехода от старой версии к новой, без потери ранее наработанных прикладных программ и без необходимости резкой смены всех навыков пользователя. Обратная совместимость — сверху вниз — как правило, не гарантируется, поскольку в ходе развития система приобретает новые возможности, не реализованные в старых версиях. Программа из Windows 3.1 будет нормально работать и в Windows XP; наоборот — вряд ли.

Совместимость версий — благо для пользователя, однако на практике она часто приводит к консервации давно отживших свой век особенностей или же просто неудачных решений, принятых в ранней версии системы. В документации подобные архаизмы помечаются как «устаревшие» (obsolete), но полного отказа от них, как правило, не происходит (а вдруг где-то еще работает прикладная программа, написанная двадцать лет назад с использованием именно этих средств?).

Как правило, наиболее консервативной стороной любой ОС являются не алгоритмы, а структуры системных данных, поэтому дальновидные разработчики заранее строят структуры «на вырост»: закладывают в них резервные поля, используют переменные вместо некоторых констант, устанавливают количественные ограничения с большим запасом и т.п.

6. Мобильность

Под ***мобильностью*** (portability) понимается возможность переноса программы (в данном случае ОС) на другую аппаратную платформу, т.е. на другой тип процессора и другую архитектуру компьютера. Здесь имеется в виду перенос с умеренными трудозатратами, не требующий полной переработки системы.

Свойство мобильности не столь однозначно положительно, как может показаться. Чтобы программа была мобильна, при ее разработке следует отказаться от глубокого использования особенностей конкретной архитектуры (таких, как количество и функциональные возможности регистров процессора, нестандартные команды и т.п.). Мобильная программа должна быть написана на языке достаточно высокого уровня (часто используется язык С), который можно реализовать на компьютерах любой архитектуры. Платой за мобильность всегда является некоторая потеря эффективности, поэтому немобильные системы распространены достаточно широко.

С другой стороны, история системного программирования усеяна останками замечательных, эффективных и удобных, но немобильных ОС, которые вымерли вместе с процессорами, для которых они предназначались. В то же время мобильная система UNIX продолжает процветать четвертый десяток лет, намного пережив те компьютеры, для которых она первоначально создавалась. Примерно 5-10% исходных текстов UNIX написаны на языка ассемблера и должны переписываться заново при переносе на новую архитектуру. Остальная часть системы написана на С и практически не требует изменений при переносе.

Некоторым компромиссом являются ***многоплатформенные*** ОС (например, Windows NT), изначально спроектированные для использования на нескольких аппаратных платформах, но не гарантирующие возможность переноса на новые, не предусмотренные заранее архитектуры.

2. КОНТРОЛЬНЫЕ ВОПРОСЫ

2.1. Дайте определение понятию "операционная система".
2.2. Назовите примеры прикладного, инструментального и системного программного обеспечения.
2.3. Дайте определение понятий "системный вызов", "API", "драйвер", "ядро".
2.4. Какие виды ядер вы знаете? К каким видам относятся ядра известных вам операционных систем?
2.5. Чем ядро отличается от операционной системы?
2.6. Приведите несколько способов классификации операционных систем.
2.7. Назовите требования к современным операционным системам и объясните, что они означают.
2.8. Назовите основные функции операционных систем.

Практическое занятие № 2

Тема: Программный интерфейс и файловая система ОС Windows

Цель работы: ознакомиться с назначением и функциями программного интерфейса Windows API, ознакомиться с файловыми системами операционных систем Windows.

Задание: составить конспект «Программный интерфейс», составить сравнительную таблицу файловых систем для операционных систем Windows.

1. Теоретические сведения

Программный интерфейс операционных систем Windows

Программный интерфейс - система унифицированных связей, предназначенных для обмена информацией между компонентами одной или нескольких вычислительных систем. Программный интерфейс задает набор необходимых процедур, их параметров и способов обращения.

Для программиста – разработчика программных средств программный интерфейс понимается, прежде всего, как API (англ. Application Programming Interface - интерфейс программирования приложений; по-русски чаще произносят [апи́]) — набор методов (функций), который программист может использовать для доступа к функциональности программного компонента (программы, модуля, библиотеки). API является важной абстракцией, описывающей функциональность «в чистом виде», безотносительно того, как реализована эта функциональность.

Прикладные программы – антивирусы, офисные приложения или игры – не могут напрямую обращаться к процессору или осуществлять вывод результатов на экран монитора. Если бы в код программы нужно было добавлять модули для выполнения всех таких стандартных операций, разработка программного обеспечения превратилась бы в довольно трудную задачу, а размер каждой программы был невообразимо большим. Поэтому на практике каждое приложение использует готовые функции операционной системы, которые и позволяют отображать картинку на экране или, например, выполнять распечатку текста.

Вызов этих функций происходит через специальные программные интерфейсы (API), которые связывают прикладные программы с операционной системой и устройствами компьютера. Если программе требуется записать файл на диск, она обращается к функции

сохранения данных Windows. Приложение дает указание операционной системе: «Сохрани файл АБВГД на жесткий диск!», а Windows принимает эту команду и проделывает всю необходимую для записи файла работу: находит на жестком диске свободную область, резервирует это место под будущий файл и затем записывает туда информацию. Кроме того, система информирует программу о результате операции: прошло ли создание файла успешно или же возникли какие-либо проблемы. Таким образом, операционная система выступает «посредником» между программой и устройствами компьютера. В Windows функции посредника выполняет Win32 или 32-разрядный API. Для современных версий Windows самым известным графическим программным интерфейсом является DirectX – он служит для поддержки различных графических режимов.

Программы, разработанные для Windows, в большинстве случаев не будут работать под управлением других операционных систем, например, Linux или Mac OS. При этом верно и обратное, то есть программа для Mac OS не будет работать под управлением Windows. Основная причина этого кроется в том, что в системах используются уникальные API, на работу с которыми рассчитаны прикладные программы.

Для программиста – системного интегратора программный интерфейс это, прежде всего способ (правила, протокол, стандарт, язык) обмена данными между различными программами и/или информационными системами. Различные программные средства разрабатываются с помощью различных языков программирования и используют различные системы управления базами данных. В этой ситуации единственным способом организации обмена данными является их внешнее представление (экспорт) в каком-либо общепринятом формате и, соответственно, наличие средств понимания (импорта) внешних данных.

API – интерфейс прикладного программирования (иногда интерфейс программирования приложений). Другими словами, это те возможности (функции, переменные, константы, классы), которые предоставляет приложение для использования прикладными программами.

API определяет функциональность, которую предоставляет программа (модуль, библиотека), при этом API позволяет абстрагироваться от того, как именно эта функциональность реализована.

Если программу (модуль, библиотеку) рассматривать как чёрный ящик, то API — это множество «ручек», которые доступны пользователю данного ящика, которые он может вертеть и дёргать, при этом ящик будет производить какие-то определенные действия понятные и необходимые пользователю, но пользователь, при этом, не имеет даже представления о их реализации.

Программные компоненты взаимодействуют друг с другом посредством API. При этом обычно компоненты образуют иерархию – высокоуровневые компоненты используют API низкоуровневых компонентов, а те, в свою очередь, используют API ещё более низкоуровневых компонентов.

API операционных систем

Практически все операционные системы (Unix, Windows, Mac OS, и т. д.) имеют API, с помощью которого программисты могут создавать приложения для этой операционной системы. Главный API операционных систем – это множество системных вызовов.

В индустрии программного обеспечения общие стандартные API для стандартной функциональности имеют важную роль, так как они гарантируют, что все программы, использующие общий API, будут работать одинаково хорошо или, по крайней мере,

типичным привычным образом. В случае API графических интерфейсов это означает, что программы будут иметь похожий пользовательский интерфейс, что облегчает процесс освоения новых программных продуктов.

Widows API

Windows API – общее наименование целого набора базовых функций интерфейсов программирования приложений операционных систем семейств Windows (от Windows 3.11 до Windows 98) и Windows NT корпорации «Microsoft». Является самым прямым способом взаимодействия приложений с Windows. Для создания программ, использующих Windows API, «Microsoft» выпускает SDK, который называется Platform SDK и содержит документацию, набор библиотек, утилит и других инструментальных средств.

Windows API был изначально спроектирован для использования в программах, написанных на языке C (или C++). Работа через Windows API — это наиболее близкий к системе способ взаимодействия с ней из прикладных программ. Более низкий уровень доступа, необходимый только для драйверов устройств, в текущих версиях Windows предоставляется через Windows Driver Model.

Win32 – 32x разрядный API для современных версий Windows. Самая популярная ныне версия. Базовые функции этого API реализованы в DLL kernel32.dll и advapi32.dll; базовые модули GUI — в user32.dll и gdi32.dll. Win32 появился вместе с Windows NT и затем был перенесён (в несколько ограниченном виде) в системы серии Windows 9x. В современных версиях Windows, происходящих от Windows NT, работу Win32 GUI обеспечивают два модуля: csrss.exe (Client/Server Runtime Subsystem), работающий в пользовательском режиме, и win32k.sys в режиме ядра. Работу же системных Win32 API обеспечивает ядро — ntoskrnl.exe

Win64 — 64-разрядная версия Win32, содержащая дополнительные функции для использования на 64-разрядных компьютерах. Win64 API можно найти только в 64-разрядных версиях Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows Server 2008 R2 и Windows 7.

Структура API-программ

Центральным понятием программирования в среде Windows является сообщение. Система посылает сообщение приложению, а то, в свою очередь, должно правильно отреагировать на него. Получателями сообщений являются функции окон приложения, на программирование которых и уходит большая часть времени при разработке API-приложений.

Классическая структура API-программы определяется четырьмя компонентами: инициализация; цикл ожидания, или цикл обработки сообщений; функция главного окна; другие функции. В простейшем случае последний компонент может отсутствовать. Два первых компонента располагаются в функции WinMain, остальные реализуются отдельными функциями.

Windows API спроектирован для использования в языке Си для написания прикладных программ, предназначенных для работы под управлением операционной системы MS Windows. Работа через Windows API — это наиболее близкий к операционной системе

способ взаимодействия с ней из прикладных программ. Более низкий уровень доступа, необходимый только для драйверов устройств, в текущих версиях Windows предоставляется через Windows Driver Model.

Windows API представляет собой множество функций, структур данных и числовых констант, следующих соглашениям языка Си. Все языки программирования, способные вызывать такие функции и оперировать такими типами данных в программах, исполняемых в среде Windows, могут пользоваться этим API. В частности, это языки C++, Pascal, Visual Basic и многие другие.

Для облегчения программирования под Windows, в компании Microsoft и сторонними разработчиками было предпринято множество попыток создать библиотеки и среды программирования, частично или полностью скрывающие от программиста особенности Windows API, и предоставляющие ту или иную часть его возможностей в более удобном виде. В частности, сама Microsoft в разное время предлагала библиотеки Active Template Library (ATL)/Windows Template Library (WTL), Microsoft Foundation Classes (MFC), .Net/WinForms/WPF. Компания Borland (ныне Embarcadero, её преемник в части средств разработки) предлагала OWL и VCL. Есть кросс-платформенные библиотеки, такие как Qt, Tk и многие другие. Весомая часть этих библиотек сконцентрирована на облегчении программирования графического интерфейса пользователя.

Для облегчения переноса на другие платформы программ, написанных с опорой на Windows API, сделана библиотека Wine.

Файловые системы операционных систем Windows

Одной из важнейших характеристик операционной системы помимо управления памятью, ресурсами компьютера и задачами является поддержка файловой системы — основного хранилища системной и пользовательской информации. Каждая файловая система имеет свои достоинства и недостатки, которые мы обсудим ниже.

В различных операционных системах фирмы Microsoft поддерживаются различные файловые системы (табл. 1).

Таблица 1

Операционная система	Файловая система
Microsoft Windows 2000	NTFS, FAT16, FAT32
Microsoft Windows NT	NTFS, FAT16
Microsoft Windows 98	FAT16, FAT32
Microsoft Windows 95 OSR2	FAT16, FAT32
Microsoft Windows 95	FAT16
Microsoft MS-DOS	FAT16

Как видно из данной таблицы, наиболее популярной и широко используемой файловой системой является FAT.

Файловые системы FAT

FAT16

Файловая система FAT16 начала свое существование еще во времена, предшествовавшие MS-DOS, и поддерживается всеми операционными системами Microsoft для обеспечения совместимости. Ее название File Allocation Table (таблица расположения файлов) отлично отражает физическую организацию файловой системы, к основным характеристикам которой можно отнести то, что максимальный размер поддерживаемого тома (жесткого диска или раздела на жестком диске) не превышает 4095 Мбайт. Во времена MS-DOS 4-гигабайтные жесткие диски казались несбыточной мечтой (роскошью были диски объемом 20-40 Мбайт), поэтому такой запас был вполне оправданным.

Том, отформатированный для использования FAT16, разделяется на кластеры. Размер кластера по умолчанию зависит от размера тома и может колебаться от 512 байт до 64 Кбайт. В табл. 2 показано, как размер кластера зависит от размера тома. Отметим, что размер кластера может отличаться от значения по умолчанию, но должен иметь одно из значений, указанных в табл. 2.

Таблица 2

Размер тома (Мбайт)	Число секторов в кластере	Размер кластера (Кбайт)
0-32	1	0,5 (512 байт)
33-64	2	1
65-128	4	2
129-255	8	4
256-511	16	8
512-1023	32	16
1024-2047	64	32
2048-4095	128	64

Не рекомендуется задействовать файловую систему FAT16 на томах больше 511 Мбайт, так как для относительно небольших по объему файлов дисковое пространство будет использоваться крайне неэффективно (файл размером в 1 байт будет занимать 64 Кбайт). Независимо от размера кластера файловая система FAT16 не поддерживается для томов больше 4 Гбайт.

На рис. 1 показано, как организован том при использовании файловой системы FAT16.

Рис. 1. Организация томов FAT 16

В файловой системе FAT16 кластеры могут иметь различное значение. Например, это может быть свободный (неиспользуемый) кластер, кластер, занятый файлом, дефектный кластер или последний кластер файла.

Единственным различием между корневым и другими каталогами является то, что первый располагается в определенном месте и имеет фиксированное число вхождений. Каждый каталог и файл используют одно или более вхождений. Например, если число фиксированных вхождений для корневого каталога равно 512 и создано 100 подкаталогов, в корневом каталоге можно создать не более 412 файлов (512 – 100).

Для каждого файла и каталога в файловой системе хранится информация (в табл. 3 приведены данные для коротких имен файлов).

Таблица 3

Информация	Размер
Имя файла или каталога (xxxxxxxx.yyy)	88 бит
Байт атрибутов	8 бит
Резервный байт	8 бит
Время создания	24 бит
Дата создания	16 бит
Дата последнего доступа	16 бит
Два резервных байта	16 бит
Время последней модификации	16 бит
Дата последней модификации	16 бит
Номер начального кластера в FAT	16 бит
Размер файла	32 бит
Итого	32 байт

В структуре каталогов файлу отводится первый незанятый кластер, доступный на томе. Номер начального кластера позволяет определить местонахождение файла: каждый кластер содержит указатель на следующий кластер или значение FFFF, указывающее на то, что это последний кластер в цепочке кластеров, занимаемых файлом. Расположение файлов по кластерам показано на рис. 2.

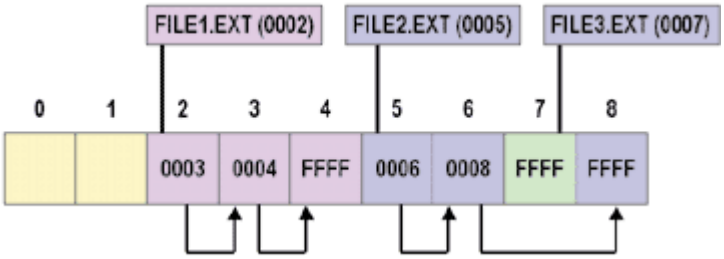


Рис. 2. Расположение файлов по кластерам

Как видно из рис. 2, в папке расположены три файла. Первый из них — FILE1.EXT занимает три кластера (файл не фрагментирован, кластеры 2, 3 и 4 расположены последовательно), второй файл — FILE2.EXT фрагментирован и располагается в кластерах 5, 6 и 8, а третий — FILE2.EXT занимает всего один кластер. Вхождение для каждого файла содержит адрес его начального кластера (2, 5 и 7 соответственно). Последний кластер каждого файла (4, 8 и 7) в качестве адреса следующего кластера содержит значение FFFF, указывающее на то, что это последний кластер для данного файла.

Так как все вхождения имеют одинаковый размер информационного блока, они различаются по байту атрибутов. Один из битов в данном байте может указывать, что это каталог, другой — что это метка тома. Для пользователей доступны четыре бита, позволяющих управлять атрибутами файла — архивный (archive), системный (system), скрытый (hidden) и доступный только для чтения (read-only) .

FAT32

Начиная с Microsoft Windows 95 OEM Service Release 2 (OSR2) в Windows появилась поддержка 32-битной FAT. Для систем на базе Windows NT эта файловая система впервые

стала поддерживаться в Microsoft Windows 2000. Если FAT16 может поддерживать тома объемом до 4 Гбайт, то FAT32 способна обслуживать тома объемом до 2 Тбайт. Размер кластера в FAT32 может изменяться от 1 (512 байт) до 64 секторов (32 Кбайт). Для хранения значений кластеров FAT32 требуется 4 байт (32 бит, а не 16, как в FAT16). Это означает, в частности, что некоторые файловые утилиты, рассчитанные на FAT16, не могут работать с FAT32.

Основным отличием FAT32 от FAT16 является то, что изменился размер логического раздела диска. FAT32 поддерживает тома объемом до 127 Гбайт. При этом, если при использовании FAT16 с 2-гигабайтными дисками требовался кластер размером в 32 Кбайт, то в FAT32 кластер размером в 4 Кбайт подходит для дисков объемом от 512 Мбайт до 8 Гбайт (табл. 4).

Таблица 4

Размер раздела (Гбайт)	Размер кластера по умолчанию (Кбайт)
Менее 8	4
8 и более, но менее 16	8
16 и более, но менее 32	16
32 и более	32

Это соответственно означает более эффективное использование дискового пространства — чем меньше кластер, тем меньше места требуется для хранения файла и, как следствие, диск реже становится фрагментированным.

При применении FAT32 максимальный размер файла может достигать 4 Гбайт минус 2 байта. Если при использовании FAT16 максимальное число вхождений в корневой каталог ограничивалось 512, то FAT32 позволяет увеличить это число до 65 535.

FAT32 накладывает ограничения на минимальный размер тома — он должен быть не менее 65 527 кластеров. При этом размер кластера не может быть таким, чтобы FAT занимала более 16 Мбайт–64 Кбайт / 4 или 4 млн. кластеров.

При использовании длинных имен файлов данные, необходимые для доступа из FAT16 и FAT32, не перекрываются. При создании файла с длинным именем Windows создает соответствующее имя в формате 8.3 и одно или более вхождений в каталог для хранения длинного имени (по 13 символов из длинного имени файла на каждое вхождение). Каждое последующее вхождение хранит соответствующую часть имени файла в формате Unicode. Такие вхождения имеют атрибуты «идентификатор тома», «только чтение», «системный» и «скрытый» — набор, который игнорируется MS-DOS; в этой операционной системе доступ к файлу осуществляется по его «псевдониму» в формате 8.3.

Файловая система NTFS

В состав Microsoft Windows 2000 входит поддержка новой версии файловой системы NTFS, которая, в частности, обеспечивает работу с сервисами каталогов Active Directory, точки пересчета (reparse points), средства защиты информации, контроль за доступом и ряд других возможностей.

Как и при использовании FAT, основной информационной единицей в NTFS является кластер. В табл. 5 показаны размеры кластеров по умолчанию для томов различной емкости.

Таблица 5

Размер тома (Мбайт)	Число секторов в кластере	Размер кластера (Кбайт)
512 и менее	1	0,5 (512 байт)
513-1024 (1 Гбайт)	2	1024
1025-2048 (2 Гбайт)	4	2048
Более 2049	8	4096

При формировании файловой системы NTFS программа форматирования создает файл Master File Table (MTF) и другие области для хранения метаданных. Метаданные используются NTFS для реализации файловой структуры. Первые 16 записей в MFT зарезервированы самой NTFS. Местоположение файлов метаданных $Mft и $MftMirr записано в загрузочном секторе диска. Если первая запись в MFT повреждена, NTFS считывает вторую запись для нахождения копии первой. Полная копия загрузочного сектора располагается в конце тома. В табл. 6 перечислены основные метаданные, хранимые в MFT.

Таблица 6

Системный файл	Имя файла	Запись MFT	Назначение
Master file table	$Mft	0	Содержит одну базовую файловую запись для каждого файла или каталога на томе NTFS. Если эта информация занимает более одной записи, создаются дополнительные записи
Master file table 2	$MftMirr	1	Копия первых четырех записей MFT. Гарантирует доступ к MFT в случае, если первый сектор поврежден
Log file	$LogFile	2	Содержит список действий, необходимых для восстановления NTFS. Размер зависит от размера тома. Используется Windows 2000 для восстановления файловой системы в случае системных сбоев
Volume	$Volume	3	Содержит информацию о томе — метку и номер версии
Attribute definitions	$AttrDef	4	Таблица имен атрибутов и описания
Root file name index	$	5	Корневой каталог
Cluster bitmap	$Bitmap	6	Информация о том, какие кластеры заняты
Boot sector	$Boot	7	Содержит код загрузки для загрузочных томов
Bad cluster file	$BadClus	8	Информация о дефектных секторах
Security file	$Secure	9	Содержит уникальные дескрипторы для всех файлов
Upcase table	$Upcase	10	Используется для преобразования символов нижнего регистра в соответствующие Unicode-символы верхнего регистра
NTFS extension file	$Extend	11	Используется различными службами операционной системы — службой квот, службой пересчета и идентификаторами объектов
		12-15	Зарезервировано для будущих версий

Остальные записи MFT содержат записи для каждого файла и каталога, расположенных на данном томе.

Обычно один файл использует одну запись в MFT, но если у файла большой набор атрибутов или он становится слишком фрагментированным, то для хранения информации о нем могут потребоваться дополнительные записи. В этом случае первая запись о файле, называемая базовой записью, хранит местоположение других записей. Данные о файлах и каталогах небольшого размера (до 1500 байт) полностью содержатся в первой записи.

Атрибуты файлов в NTFS

Каждый занятый сектор на NTFS-томе принадлежит тому или иному файлу. Даже метаданные файловой системы являются частью файла. NTFS рассматривает каждый файл (или каталог) как набор файловых атрибутов. Такие элементы, как имя файла, информация о его защите и даже данные в нем, являются атрибутами файла. Каждый атрибут идентифицируется кодом определенного типа и, опционально, именем атрибута.

Если атрибуты файла вмещаются в файловую запись, они называются резидентными атрибутами. Такими атрибутами всегда являются имя файла и дата его создания. В тех случаях, когда информация о файле слишком велика, чтобы вместиться в одну MFT-запись, некоторые атрибуты файла становятся нерезидентными. Резидентные атрибуты хранятся в одном или более кластерах и представляют собой поток альтернативных данных для текущего тома (об этом — чуть ниже). Для описания местонахождения резидентных и нерезидентных атрибутов NTFS создает атрибут Attribute List.

В табл. 7 показаны основные атрибуты файлов, определенные в NTFS. В будущем этот список может быть расширен.

Таблица 7

Атрибут	Описание
Standard Information	Содержит такую стандартную информацию, как дата создания и число ссылок
Attribute List	Список местонахождения всех атрибутов, размещенных вне записи MFT
File Name	Повторяющийся атрибут для длинных и коротких имен файлов. Длинное имя файла состоит из 13-255 символов Unicode, короткое представлено в формате 8.3. Возможно использование дополнительных имен, например, в формате POSIX
Security Descriptor	Информация о том, кто является владельцем файла и кто имеет доступ к файлу
Data	Данные, содержащиеся в файле. NTFS позволяет использовать более одного атрибута этого типа для каждого файла. Каждый файл обычно имеет один неименованный атрибут типа Data. Помимо этого файл может иметь дополнительные именованные атрибуты данного типа, каждый из которых использует собственный синтаксис
Object ID	Уникальный в рамках тома идентификатор файла. Используется сервисом слежения за ссылками
Logged Tool Stream	Схож с потоковыми операциями, но в данном случае изменения заносятся в протокол. Используется Encrypting File System (EFS) — файловой системой с шифрованием, поддерживаемой в Windows 2000
Reparse Point	Используется при монтировании томов, драйверами-фильтрами и в ряде других случаев
Index Root	Используется для реализации каталогов и других индексов

Index Allocation	Используется для реализации каталогов и других индексов
Bitmap	Используется для реализации каталогов и других индексов
Volume Information	Используется только в системном файле $Volume. Содержит версию тома
Volume Name	Используется только в системном файле $Volume. Содержит метку тома

Файловая система CDFS

В ОС Windows 95 и старше обеспечивается поддержка файловой системы CDFS, отвечающей стандарту ISO'9660, описывающему расположение информации на CD-ROM. Поддерживаются длинные имена файлов в соответствии с ISO'9660 Level 2.

При создании CD-ROM для использования под управлением Windows следует иметь в виду следующее:

все имена каталогов и файлов должны содержать менее 32 символов;

все имена каталогов и файлов должны состоять только из символов верхнего регистра;

глубина каталогов не должна превышать 8 уровней от корня;

использование расширений имен файлов не обязательно.

Universal Disk Format

Поддержка файловой системы UDF является одним из новшеств в Windows 2000. Universal Disk Format — это файловая система, отвечающая стандарту ISO'13346 и используемая для обмена данными с накопителями CD-ROM и DVD. В настоящее время поддерживаются только диски версий UDF 1.02 и 1.50.

Сравнение файловых систем

Под управлением Microsoft Windows 2000 возможно использование файловых систем FAT16, FAT32, NTFS или их комбинаций. Выбор операционной системы зависит от следующих критериев:

того, как используется компьютер;

аппаратной платформы;

размера и числа жестких дисков;

безопасности информации.

Файловые системы FAT

Цифры в названии файловых систем — FAT16 и FAT32 — указывают на число бит, необходимых для хранения информации о номерах кластеров, используемых файлом. Так, в FAT16 применяется 16-битная адресация и, соответственно, возможно использование до 216 адресов. В Windows 2000 первые четыре бита таблицы расположения файлов FAT32 необходимы для собственных нужд, поэтому в FAT32 число адресов достигает 228.

В табл. 8 показаны размеры кластеров для файловых систем FAT16 и FAT32.

Таблица 8

Размер диска	Размер кластера FAT16 (Кбайт)	Размер кластера FAT32 (Кбайт)
260-511 Мбайт	8	Не поддерживается
512-1023 Мбайт	16	4
1024-2 Гбайт	32	4
2-8 Гбайт	Не поддерживается	4
8-16 Гбайт	Не поддерживается	8
16-32 Гбайт	Не поддерживается	16
Более 32 Гбайт	Не поддерживается	32

Помимо существенных отличий в размере кластера FAT32 также позволяет корневому каталогу расширяться (в FAT16 число вхождений ограничено 512 и может быть даже ниже при использовании длинных имен файлов).

Преимущества FAT16

Среди преимуществ FAT16 можно отметить следующие:

файловая система поддерживается операционными системами MS-DOS, Windows 95, Windows 98, Windows NT, Windows 2000, а также некоторыми операционными системами UNIX;

существует большое число программ, позволяющих исправлять ошибки в этой файловой системе и восстанавливать данные;

при возникновении проблем с загрузкой с жесткого диска система может быть загружена с флоппи-диска;

данная файловая система достаточно эффективна для томов объемом менее 256 Мбайт.

Недостатки FAT16

К основным недостаткам FAT16 относятся:

корневой каталог не может содержать более 512 элементов. Использование длинных имен файлов существенно сокращает число этих элементов;

FAT16 поддерживает не более 65 536 кластеров, а так как некоторые кластеры зарезервированы операционной системой, число доступных кластеров — 65 524. Каждый кластер имеет фиксированный размер для данного логического устройства. При достижении максимального числа кластеров при их максимальном размере (32 Кбайт) максимальный объем поддерживаемого тома ограничивается 4 Гбайт (под управлением Windows 2000). Для поддержания совместимости с MS-DOS, Windows 95 и Windows 98 объем тома под FAT16 не должен превышать 2 Гбайт;

не поддерживается резервная копия загрузочного сектора;

в FAT16 не поддерживается встроенная защита файлов и их сжатие;

на дисках большого объема теряется много места за счет того, что используется максимальный размер кластера. Место под файл выделяется исходя из размера не файла, а кластера.

Преимущества FAT32

Среди преимуществ FAT32 можно отметить следующие:

выделение дискового пространства выполняется более эффективно, особенно для дисков большого объема;

корневой каталог в FAT32 представляет собой обычную цепочку кластеров и может находиться в любом месте диска. Благодаря этому FAT32 не накладывает никаких ограничений на число элементов в корневом каталоге;

за счет использования кластеров меньшего размера (4 Кбайт на дисках объемом до 8 Гбайт) занятое дисковое пространство обычно на 10-15% меньше, чем под FAT16;

FAT32 является более надежной файловой системой. В частности, она поддерживает возможность перемещения корневого каталога и использования резервной копии FAT. Помимо этого загрузочная запись содержит ряд критичных для файловой системы данных.

Недостатки FAT32

Основные недостатки FAT32:

размер тома при использовании FAT32 под Windows 2000 ограничен 32 Гбайт;

тома FAT32 недоступны из других операционных систем — только из Windows 95 OSR2 и Windows 98;

не поддерживается резервная копия загрузочного сектора;

в FAT32 не поддерживается встроенная защита файлов и их сжатие.

Файловая система NTFS

При работе в Windows 2000 и старше Microsoft рекомендуется отформатировать все разделы жесткого диска под NTFS, за исключением тех конфигураций, когда используется несколько операционных систем (кроме Windows 2000 и Windows NT). Применение NTFS вместо FAT позволяет использовать функции, доступные в NTFS. К ним, в частности, относятся:

возможность восстановления. Эта возможность «встроена» в файловую систему. NTFS гарантирует сохранность данных за счет того, что использует протокол и некоторые алгоритмы восстановления информации. В случае системного сбоя NTFS использует протокол и дополнительную информацию для автоматического восстановления целостности файловой системы;

сжатие информации. Для томов NTFS Windows 2000 поддерживает сжатие отдельных файлов. Такие сжатые файлы могут использоваться Windows-приложениями без предварительной распаковки, которая происходит автоматически при чтении из файла. При закрытии и сохранении файл снова упаковывается;

помимо этого можно выделить следующие преимущества NTFS:

- некоторые функции операционной системы требуют наличия NTFS;

- скорость доступа намного выше — NTFS минимизирует число обращений к диску, требуемых для нахождения файла;

- защита файлов и каталогов. Только на томах NTFS возможно задание атрибутов доступа к файлам и папкам;

- при использовании NTFS Windows 2000 поддерживает тома объемом до 2 Тбайт;

- файловая система поддерживает резервную копию загрузочного сектора — она располагается в конце тома;

- NTFS поддерживает систему шифрования Encrypted File System (EFS), обеспечивающую защиту от неавторизованного доступа к содержимому файлов;

- при использовании квот можно ограничить объем дискового пространства, занимаемого пользователями.

Недостатки NTFS

Говоря о недостатках файловой системы NTFS, следует отметить, что:

NTFS-тома недоступны в MS-DOS, Windows 95 и Windows 98. Помимо этого ряд функций, реализованных в NTFS под Windows 2000, недоступен в Windows 4.0 и более ранних версиях;

для томов небольшого объема, содержащих много файлов небольшого размера, возможно снижение производительности по сравнению с FAT.

Файловая система и скорость

Для томов небольшого объема FAT16 или FAT32 обеспечивает более быстрый доступ к файлам по сравнению с NTFS, так как:

FAT обладает более простой структурой;

размер каталогов меньше;

FAT не поддерживает защиту файлов от несанкционированного доступа — системе не нужно проверять права доступа к файлам.

NTFS минимизирует число обращений к диску и время, необходимое для нахождения файла. Кроме того, если размер каталога достаточно мал, чтобы поместиться в одной записи MFT, вся запись считывается за один раз.

Одно вхождение в FAT содержит номер кластера для первого кластера каталога. Для просмотра файла FAT требуется поиск по всей файловой структуре.

Сравнивая скорость операций, выполняемых для каталогов, содержащих короткие и длинные имена файлов, следует учитывать, что скорость операций для FAT зависит от самой операции и размера каталога. Если FAT ищет несуществующий файл, поиск выполняется по всему каталогу — эта операция занимает больше времени, чем поиск по структуре, основанной на B-деревьях, используемой в NTFS. Среднее время, необходимое для поиска файла, в FAT выражается как функция от N/2, в NTFS — как log N, где N — это число файлов.

Ряд следующих факторов влияет на скорость чтения и записи файлов под управлением Windows 2000:

фрагментация файла. Если файл сильно фрагментирован, NTFS обычно требуется меньше обращений к диску, чем FAT для нахождения всех фрагментов;

размер кластера. Для обеих файловых систем размер кластера по умолчанию зависит от объема тома и всегда выражается степенью числа 2. Адреса в FAT16 — 16-битные, в FAT32 — 32-битные, в NTFS — 64-битные;

размер кластера по умолчанию в FAT базируется на том факте, что таблица расположения файлов может иметь не более 65 535 вхождений — размер кластера представляет собой функцию от объема тома, деленного на 65 535. Таким образом, размер кластера по умолчанию для тома FAT всегда больше, чем размер кластера для тома NTFS того же объема. Отметим, что больший размер кластера для томов FAT означает, что тома FAT могут быть менее фрагментированными;

расположение файлов небольшого размера. При использовании NTFS файлы небольшого размера содержатся в MFT-записи. Размер файла, помещающегося в одну запись MFT, зависит от числа атрибутов этого файла.

Максимальный размер тома

Максимальный размер тома зависит от используемой файловой системы. Windows 2000 позволяет форматировать тома для трех файловых систем: FAT16, FAT32 и NTFS.

Максимальный размер томов FAT

FAT16 поддерживает до 65 535 кластеров на одном томе. Из этого факта вытекают ограничения, показанные в табл. 9.

Таблица 9

Описание	Ограничение
Максимальный размер файла	2^{32} — 1 байт
Максимальный размер тома	4 Гбайт
Число файлов в томе	216

Под управлением Windows NT и Windows 2000 размер кластера FAT16 для томов размером от 2 до 4 Гбайт равен 64 Кбайт. Этот размер кластера используется для обеспечения совместимости с некоторыми приложениями (например, с программами установки, которые неверно высчитывают объем свободного пространства). Поэтому рекомендуется использовать FAT32 для томов размером от 2 до 4 Гбайт.

FAT32 работает с томами, на которых есть как минимум 65 527 кластеров, а максимальное число кластеров, поддерживаемых этой файловой системой для одного тома, — 4 177 918. Windows 2000 позволяет создавать тома размером до 32 Гбайт.

В табл. 10 показаны основные ограничения FAT32.

Таблица 10

Описание	Ограничение
Максимальный размер файла	2^{32} — 1 байт
Максимальный размер тома	32 Гбайт
Число файлов в томе	Около 4 млн.

Максимальный размер томов NTFS

Теоретически NTFS поддерживает тома с числом кластеров до 232. Но, тем не менее, помимо отсутствия жестких дисков такого объема существуют и другие ограничения на максимальный размер тома.

Одним из таких ограничений является таблица разделов. Индустриальные стандарты ограничивают размер таблицы разделов 232 секторами. Другим ограничением является размер сектора, который обычно равен 512 байт. Поскольку размер сектора может измениться в будущем, текущий размер дает ограничение на размер одного тома — 2 Тбайт (232 x 512 байт = 241). Таким образом, 2 Тбайт является практическим пределом для физических и логических томов NTFS.

В табл. 11 показаны основные ограничения NTFS.

Таблица 11

Описание	Ограничение
Максимальный размер файла	2^{64} — 1 Кбайт
Максимальный размер тома	2 Тбайт
Число файлов в томе	2^{32} — 1 Кбайт

Управление доступом к файлам и каталогам

При использовании томов NTFS можно устанавливать права доступа к файлам и каталогам. Эти права доступа указывают, какие пользователи и группы имеют доступ к ним и какой уровень доступа допустим. Такие права доступа распространяются как на

пользователей, работающих за компьютером, на котором располагаются файлы, так и на пользователей, обращающихся к файлам через сеть, когда файл располагается в каталоге, открытом для удаленного доступа.

Под NTFS можно также устанавливать разрешения на удаленный доступ, объединяемые с разрешениями на доступ к файлам и каталогам. Помимо этого файловые атрибуты (только чтение, скрытый, системный) также ограничивают доступ к файлу.

Под управлением FAT16 и FAT32 тоже возможно устанавливать атрибуты файлов, но они не обеспечивают права доступа к файлам.

В версии NTFS, используемой в Windows 2000, появился новый тип разрешения на доступ — наследуемые разрешения. Вкладка Security содержит опцию Allow inheritable permissions from parent to propagate to this file object, которая по умолчанию находится в активном состоянии. Данная опция существенно сокращает время, требуемое на изменение прав доступа к файлам и подкаталогам. Например, для изменения прав доступа к дереву, содержащему сотни подкаталогов и файлов, достаточно включить эту опцию — в Windows NT 4 необходимо изменить атрибуты каждого отдельного файла и подкаталога.

Напомним, что для томов FAT можно управлять доступом только на уровне томов и такой контроль возможен только при удаленном доступе.

Сжатие файлов и каталогов

В Windows 2000 поддерживается сжатие файлов и каталогов, расположенных на NTFS-томах. Сжатые файлы доступны для чтения и записи любыми Windows-приложениями. Для этого нет необходимости в их предварительной распаковке. Используемый алгоритм сжатия схож с тем, который используется в DoubleSpace (MS-DOS 6.0) и DriveSpace (MS-DOS 6.22), но имеет одно существенное отличие — под управлением MS-DOS выполняется сжатие целого первичного раздела или логического устройства, тогда как под NTFS можно упаковывать отдельные файлы и каталоги.

Алгоритм сжатия в NTFS разработан с учетом поддержки кластеров размером до 4 Кбайт. Если величина кластера больше 4 Кбайт, функции сжатия NTFS становятся недоступными.

Самовосстановление NTFS

Файловая система NTFS обладает способностью самовосстановления и может поддерживать свою целостность за счет использования протокола выполняемых действий и ряда других механизмов.

NTFS рассматривает каждую операцию, модифицирующую системные файлы на NTFS-томах, как транзакцию и сохраняет информацию о такой транзакции в протоколе. Начатая транзакция может быть либо полностью завершена (commit), либо откатывается (rollback). В последнем случае NTFS-том возвращается в состояние, предшествующее началу транзакции. Для того чтобы управлять транзакциями, NTFS записывает все операции, входящие в транзакцию, в файл протокола, перед тем как осуществить запись на диск. После того как транзакция завершена, все операции выполняются. Таким образом, под управлением NTFS не может быть незавершенных операций. В случае дисковых сбоев незавершенные операции просто отменяются.

Под управлением NTFS также выполняются операции, позволяющие «на лету» определять дефектные кластеры и отводить новые кластеры для файловых операций. Этот механизм называется cluster remapping.

Файловая система ReFS

ReFS (Resilient file system, предварительное название Protogon) — локальная файловая система, используемая в Windows Server 2012, Windows Server 2012 R2, Windows 8, Windows 8.1. Является дальнейшим развитием NTFS. Protogon поддерживает точки повторной обработки (reparse points) — технологию, которая ранее содержалась только в файловой системе NTFS. Через точки повторной обработки реализована поддержка символьных ссылок и точек монтирования в Windows, так что Protogon также поддерживает их. Protogon не поддерживается Windows 7 и более ранними системами.

Основные особенности

Улучшенная надёжность хранения информации на диске структур. ReFS использует B+ деревья (принцип, сходный с хранением данных в реляционных СУБД) как для метаданных, так и для содержимого файлов. Размеры файлов, томов, количество файлов в каталоге ограничены лишь 64-битным числом, что соответствует размеру файла в 16 эксбибайт, размеру тома в 1 йобибайт (при использовании кластеров данных размером 64 КиБ). Свободное место на диске описывается 3 отдельными иерархическими таблицами для малых, средних и больших фрагментов свободного пространства. Имена файлов и длина пути ограничена 32 кибибайтами, для их хранения используется Unicode.

Согласно предложению МЭК, назвать 2^{60} байт "эксабайт" общепринято, но формально неверно, как приставка *экса-*, означает умножение на 10^{18}, а не 2^{60}. Правильной для 2^{60} является двоичная приставка *эксби-*. Таким образом, 1 ЭБ = 10^{18} Б, а 1 ЭиБ = 2^{60} Б.

Таблица 12

Измерения в байтах							
ГОСТ 8.417-2002			**Приставки СИ**		**приставки МЭК**		
Название	**Символ**	**Степень**	**Название**	**Степень**	**Название**	**Символ**	**Степень**
байт	Б	10^0	-	10^0	байт	В Б	2^0
килобайт	кБ	10^3	кило-	10^3	кибибайт	KiB КиБ	2^{10}
мегабайт	МБ	10^6	мега-	10^6	мебибайт	MiB МиБ	2^{20}
гигабайт	ГБ	10^9	гига-	10^9	гибибайт	GiB ГиБ	2^{30}
терабайт	ТБ	10^{12}	тера-	10^{12}	тебибайт	TiB ТиБ	2^{40}
петабайт	ПБ	10^{15}	пета-	10^{15}	пебибайт	PiB ПиБ	2^{50}

эксабайт	ЭБ	10^{18}	экса-	10^{18}	эксбибайт	EiB	ЭиБ	2^{60}
зеттабайт	ЗБ	10^{21}	зетта-	10^{21}	зебибайт	ZiB	ЗиБ	2^{70}
йоттабайт	ЙБ	10^{24}	йотта-	10^{24}	йобибайт	YiB	ЙиБ	2^{80}

Поддержка стратегии Copy-on-write (копирование при записи, выделение при записи) для метаданных, при которой любые транзакции файловой системы не перезаписывают старые метаданные, а записываются в новый блок и организуются в пачки. Для всех метаданных в ReFS используются 64-битные контрольные суммы, хранящиеся независимо. Данные файлов могут иметь контрольную сумму в отдельном потоке (атрибут «integrity»). В случае, если содержимое файлов или метаданных не соответствует контрольным суммам, не требуется отключение файловой системы для удаления или восстановления таких данных. За счет встроенных проверок ReFS не требует регулярного использования утилит проверки диска типа CHKDSK.

Совместимость со старыми API, поддержка многих особенностей NTFS, например, шифрование BitLocker, Access Control Lists, USN Journal, уведомления об изменениях, символьные ссылки, junction point, точки монтирования, reparse point, снапшоты томов, идентификаторы файлов, NTFS oplock. ReFS интегрируется с технологией виртуализации носителей данных Storage Spaces, которая позволяет применять зеркалирование и объединять несколько физических носителей, как в рамках одного ПК, так и через сеть. При использовании зеркалирования ReFS может обнаруживать и исправлять сбойные копии файлов в процессе data scrubbing, при котором проводится фоновая сверка контрольных сумм.

Многие возможности NTFS не поддерживаются в ReFS, включая именованные потоки файлов (после проверки на Windows 10 TP ADS работает с ReFS. Для поддержки ReFS нужно вносить изменения в реестр), NTFS Distributed Link Tracking (DLT), короткие имена файлов (формат 8.3), сжатие файлов, шифрование на уровне файлов Encrypting File System, Transactional NTFS, жёсткие ссылки, extended attributes, и дисковые квоты. Разреженные файлы (Sparse files) поддерживаются в RTM.

2. Порядок выполнения работы

2.1. Изучить главные особенности программного интерфейса ОС Windows.

2.2. Изучить структуру API-программ.

2.3. Ознакомиться с назначением и основными особенностями файловых систем ОС Windows.

2.4. Составить сравнительную таблицу файловых систем ОС Windows в виде:

Файловая система	Операционная система	Размер тома	Размер кластера (Кб)	Максимальный размер файла	Максимальный размер тома

3. Контрольные вопросы

3.1. Дайте понятие программного интерфейса.
3.2.В чем отличительная особенность программного интерфейса Windows API.
3.3. Какова структура API-программ?
3.4.Что такое файловая система, какую функцию она выполняет в среде операционной системы?
3.5. Расскажите об основных особенностях файловой системы FAT16.
3.6.Расскажите об основных особенностях файловой системы FAT32.
3.7.Расскажите об основных особенностях файловой системы NTFS.

Практическая работа № 3

Тема: Процесс загрузки операционной системы. Настройка компьютерной системы средствами программы SETUP

Цель работы: Ознакомиться с основными опциями программы SETUP

Задание: научиться осуществлять предварительную настройку компьютерной системы средствами программы SETUP

1. Теоретические сведения

Слово BIOS является аббревиатурой, которая в расшифровке выглядит как Basic Input/Output System, что в переводе с английского означает "Базовая система ввода/вывода".

Впервые эта система была разработана в 1981 году. Она отвечает за все операции ввода/вывода для Windows. Самый первый вариант BIOS на компьютерах IBM, и его функции несколько отличались от того, что делает сегодняшний BIOS. Тот BIOS в определенной мере исполнял роль драйверов, то есть связывал операционную систему с периферийными устройствами. Но со временем периферия становилось все совершеннее, впрочем, как и сам компьютер. BIOS уже не мог выполнять всех тех задач, которые были на него первоначально возложены, поэтому появились драйверы, программы, взаимодействующие с операционной системой.

BIOS в общем случае представляет собой набор правил, определяющих, как происходит конфигурирование компонент компьютера при его включении, как его устройства взаимодействуют друг с другом, как осуществляется простейший ввод/вывод данных.

Именно BIOS определяет ход процесса загрузки компьютера, тестирование и первичную настройку присоединенных устройств. Только после этого управление компьютером передается сначала загрузчику операционной системы, а затем самой операционной системе (если таковая найдется). Далее, уже в процессе работы компьютера, именно BIOS обеспечивает базовые функции ввода/вывода и функции взаимодействия устройств между собой. Кстати, именно BIOS производит распределение ресурсов компьютера между устройствами.

Таким образом, можно выделить следующие задачи BIOS, которые им решаются:

- инициализация и начальное тестирование аппаратных средств компьютера (POST-тестирование);
- настройка и конфигурирование аппаратных средств и системных ресурсов;
- распределение системных ресурсов;
- индентификация и конфигурирование устройств PCI;
- начальная загрузка (первый этап) операционной системы;

- отработка базовых функций программных обращений;

- обработка программных прерываний от системных устройств;

- базовые функции ввода/вывода и функции взаимодействия устройств между собой;

- поддержка управления энергопотреблением компьютера, автоматическое включение, выключение, перевод в "спящий режим" и т.п.

Из всего этого можно сделать вывод, что от настроек BIOS в значительной степени зависит то, как будет работать компьютер.

Справедливости ради стоит отметить, что многие операционные системы берут на себя некоторые "мелкие" функции BIOS (простейший пример - настройка системного времени), однако ни одна из них не "суется" в такие настройки, как соотношение частот системной шины и частоты процессора, время доступа к оперативной памяти, подаваемое напряжение на процессор и т.д.

BIOS как набор правил оформлен в виде нескольких программ. Программой, с помощью которой производится настройка BIOS, является BIOS Setup. Иногда некоторые одни и те же опции имеют отличающиеся названия на разных системных платах при одинаковой базовой версии BIOS. Просто производители системных плат к стандартному BIOS (например, AMI BIOS 8.0) добавляют свою собственную модификацию BIOS Setup, в которой могут использоваться свои названия опций, смысл которых остается тем же.

Очевидно, что набор программ BIOS должен оставаться в памяти компьютера и после его выключения (чтобы быть доступным сразу после включения компьютера). Кроме того, очевидно, что этот набор не может содержаться на каком-то носителе, так как именно BIOS производит настройку взаимодействия с этим самыми носителями. И, наконец, BIOS должен быть достаточно надежно защищен от перезаписи, так как повреждение (умышленное или неумышленное) BIOS может привести к полному выходу из строя и повреждению компьютера. Компьютер может просто "сгореть".

Исходя из необходимости выполнения этих условий BIOS аппаратно записан в специальной микросхеме постоянной памяти на системной плате. Что касается возможности перезаписи BIOS (перепрошивки), то изначально таковая вообще была исключена. На старых системных платах перепрошить BIOS нельзя. Сейчас же на современных системных платах используются специальные микросхемы BIOS с возможностью перезаписи - так называемые flash-микросхемы. При этом выделяют две группы таких перезаписываемых микросхем:

- EPROM (от англ. Erasable Programmable Read Only Memory) - перезаписываемые микросхемы постоянной памяти, содержимое которых может быть стерто с помощью ультрафиолетового излучения. Соответственно для этого необходимо использование специального оборудования. Такие микросхемы применяли для BIOS до появления более совершенных микросхем, в которых стирание может быть произведено электрическим зарядом.

- EEPROM (от англ. Electrically Erasable Programmable Read Only Memory) - перезаписываемые микросхемы постоянной памяти, содержимое которых может быть стерто с помощью электрического сигнала. При этом никакого специального оборудования не требуется. Микросхему можно перезаписать, не вынимая ее из компьютера.

Необходимо также отметить, что, помимо основной микросхемы BIOS, расположенной на системной плате, все чаще свои микросхемы имеют платы расширения, подключаемые к ней. Особенно часто свои BIOS имеют современные видеокарты.

Наличие или отсутствие собственных BIOS обусловливается обычно сложностью устройств, для которых они предназначены. При настройке основного BIOS (на системной плате) можно разрешить или запретить использование собственных BIOS'ов плат расширения.

Для того чтобы обозначить эту систему, есть специальный термин Firmware, что означает: аппаратно-реализованное программное обеспечение. Само, название говорит о том, что программа уже вшита в одну из микросхем, на системной плате, ее установкой не надо заниматься, да и нежелательно, она уже от производителя настроена для работы в оптимальном режиме. После включения компьютера, она запускается автоматически, выполняя множество разнообразных задач. Что конкретно делает BIOS вовремя загрузки, первым делом он проверяет узлы компьютера, которые отвечают за запуск, это оперативная память и центральный процессор. Проверяется их состояние и работоспособность, например, если отключить кулер обдувающий процессор, BIOS сразу об этом оповестит. После проверки основных узлов, продолжается это, кстати, недолго, и если все нормально, то практически ничего на экране монитора не отражается. Потом проходит проверка настроек, определяется частота оперативки, соответствует ли она заданной, а также время включения всех вентиляторов, расположенных внутри системного блока и подключенных к материнской плате. Идет выбор, в каком режиме энергосбережения компьютер будет работать дальше, если долгое время простаивал. Хотя, кое-что из этих настроек можно поменять и в Windows, но основные работоспособные параметры компьютера BIOS выставит сам, перед запуском операционной системы. Если вы уже ознакомлены с БИОСом, то, не боясь, можете менять некоторые его настройки. Можно отключить, или наоборот подключить интерфейсы на материнской плате, это или жесткий диск, или привод DVD, или USB-накопители, например внешний жесткий диск, или флешка. Также через BIOS, можно выбрать к чему в первую очередь будет обращаться компьютер. По умолчанию стоит такая последовательность, жесткий диск, привод DVD, USB-накопители. Так вот эту последовательность можно поменять, и поставить первым привод DVD, это позволит без лишней возни, после перезагрузки запустить диск с операционной системой для ее переустановки. После проверок всех узлов, BIOS загружает Windows, система запускается вместе со всеми необходимыми для работы драйверами, и потом BIOS становится ненужным, до следующего включения или перезагрузки.

Годы шли, и BIOS менялся, чтобы соответствовать развивающейся технике. Но менялся не только внешний вид, но и внутреннее содержимое. В начале 90-х годов к его основным функциям, такие как загрузка с привода DVD, а также автоматическая настройка плат расширения. Появился UEFI. Это Unified Extensible Firmware Interface. На русском – Расширяемый Интерфейс Встроенного ПО. Разрабатываться он начал уже относительно давно, с 2001 года, занималась этим компания Intel, и предназначался изначально для серверного процессора Itanium. Что подтолкнуло разработчиков заняться uefi boot? А то, что Itanium был принципиально новым оборудованием и никакая версия BIOS, не работала с ним, и никакие доработки тут помочь не могли. Первоначально появилась EFI, и первый кто ее начал использовать оказалась компания Apple. Она начала ставить EFI на все выпускаемые ПК и ноутбуки. С 2006 года эта компания при сборке компьютеров и ноутбуков использует интеловские процессоры. За год до этого к аббревиатуре EFI, была добавлена еще одна буква U, за этой буквой скрывается слово Unified, слово говорит о том, что разработкой интерфейса uefi bios занимается сразу несколько компаний. К ним относятся Dell, HP, IBM, и Phoenix Insyde, ну и, конечно же, вездесущая компания Microsoft, потому что именно она является основным разработчиком операционных систем.

В далеком прошлом, когда программисты разрабатывали BIOS, никто из них не задумывался над тем, сколько еще ей придется работать. Но время шло, техника менялась, и многие новинки приходилось вписывать в старую систему, и все может быть и дальше шло также, если бы не начали вылезать проблемы. Одна из них заключается в том, что современные

жесткие диски уже могут достигать размеров 6 Тб и более, а вот старая BIOS видит только 2 Тб. Получается, что компьютеры, оснащенные BIOS, ограничены в объемах памяти, кроме этого каждый производитель системных плат делал свои интерфейсы, что путало пользователей. Все совсем иначе выглядит с применением UEFI, объем жесткого диска, тут можно сказать неограничен, кроме того в новой системе единый для всех интерфейс. Это облегчает жизнь не только пользователей, но и разработчиков программ которые запускаются до загрузки Windows. В UEFI есть множество новых дополнительных функций, которые были недоступны в старых версиях, это например резервное копирование данных. И убраны некоторые лишние функции, уже неиспользуемые в наше время.

Особенности интерфейса UEFI

Главное меню графического интерфейса UEFI-Bios

Сразу хотелось бы отметить графический UEFI, новый разноцветный интерфейс, радующий глаз. Все меню состоит из квадратных кнопок с рисунками, и даже если вы не знаете английского, то по рисункам определите, что скрыто за кнопками. Основное меню, называемое Click Bios, состоит из пяти пунктов, или пяти кнопок. Щелкать по ним вы можете мышкой, чего не было доступно в старом BIOS, там все управлялось только с клавиатуры. — Первая кнопка GreenPower, она зеленого цвета, в центре нее изображена лампа, здесь при желании можно поменять настройки энергосбережения процессора. — Вторая - «Служебная программа», после нажатия на нее откроется дополнительное меню со вспомогательными программами uefi bios. Всего их шесть:

Практическая работа № 4

Тема: Linux, работа с файлами и каталогами

1. Работа с файловой системой

1.1. Команда ls

Синтаксис:

`ls [-acltuFR] [`*`файлы_или_каталоги`*`]`

Описание:

При указании в аргументах *`файла`* выводится информация о нем согласно указанным опциям, в случае *`каталога`* — то же для файлов каталога. Если аргументов нет, — выдается листинг текущего каталога.

Опции:

`-a`	Выводить информацию обо всех файлах в каталоге (по системному соглашению файлы, имя которых начинается с точки, являются скрытыми, и командой `ls` не показываются
`-c`	Использовать время последней модификации i-node, связанного с файлом (создание файла, изменение прав, и т.д.), а не время модификации
`-i`	Выдать номер i-node, связанного с файлом
`-l`	Выдавать информацию о файлах в длинном формате: тип файла (обычный файл -, каталог *`d`*, сокет *`s`*, символьная ссылка *`l`*, символьное устройство *`c`*, блочное устройство *`b`*, FIFO *`p`*), права доступа (чтение — *`r`*, запись — *`w`*, исполнение — *`x`*; первые 3 символа относятся к владельцу, следующие 3 — к членам группы, владеющей файлом, и последние 3 — ко всем остальным), владелец, группа, размер в байтах, дата модификации, имя файла
`-t`	Отсортировать листинг по времени
`-u`	Использовать время последнего доступа к файлу, а не время модификации
`-F`	После каждого имени выдавать значок, характеризующий тип последнего: каталог (/), исполняемый файл (*), FIFO (|), символьная ссылка (@), сокеты семейства AF_UNIX (=)
`-R`	Выдавать листинг рекурсивно по подкаталогам

Примеры:

Получение полного листинга текущего каталога, включая скрытые файлы.

```
$ ls -la
итого 224
drwx------ 16 guest guest 4096 Map 21 18:16 .
drwxr-xr-x  3 root root 4096 Map 21 17:53 ..
-rw-------  1 guest guest   21 Map 21 17:54 .bash_history
-rw-r--r--  1 guest guest   33 Дек 16 22:42 .bash_logout
```

```
-rw-r--r--  1 guest guest  176 Дек 16 22:42 .bash_profile
-rw-r--r--  1 guest guest  124 Мар 16 22:42 .bashrc
drwxr-xr-x  2 guest test  4096 Мар 21 17:53 Desktop
```

7.1.2. Команда pwd

Синтаксис:

```
pwd
```

Описание:

Выводит имя текущего каталога.

Примеры:

Определяем текущий каталог.

```
$ pwd
/home/guest
```

7.1.3. Команда cd

Синтаксис:

```
cd [каталог]
```

Описание:

Сменить текущий *каталог* на указанный в аргументе. При отсутствии аргументов происходит переход в домашний каталог пользователя `$HOME`.

Примеры:

Переход из текущего каталога в каталог `/usr/share/doc/`.

```
$ pwd
/home/guest
$cd /usr/share/doc/
$pwd
/usr/share/doc/
```

7.1.4. Команда cp

Синтаксис:

```
cp [-i] файл ... файл_или_каталог

cp -r [-i] каталог ... каталог
```

Описание:

Копирует *файлы* или *каталог*, указанный в первых параметрах, в файл или каталог, указанный в последнем.

Опции:

`-i`	Интерактивно — запрашивается подтверждение на перезапись существующих файлов
`-r`	Рекурсивное копирование каталога

Примеры:

Рекурсивное копирование каталога `a` в `b`:

```
$ cp -r a b
```

Интерактивное копирование файлов из каталога `a` в `b`:

```
$ cp -i a/* b
cp: overwrite b/1 (yes/no)? y
cp: overwrite b/2 (yes/no)? n
cp: overwrite b/2 (yes/no)? y
```

7.1.5. Команда ln

Синтаксис:

`ln [-fs]` *`файл_или_каталог ...`* `[`*`ссылка_или_каталог`*`]`

Описание:

Устанавливает ссылку в файловой системе (символьную или жесткую). Если второй операнд является уже существующим каталогом, то ссылки создаются внутри него. В случае, если второй аргумент отсутствует, ссылка создается в текущем каталоге с именем источника.

Опции:

`-f`	Устанавливать ссылку вместо существующего файла
`-s`	Символьная ссылка

Примеры:

Сделать символьную ссылку `b` на `a`:

```
$ln -s a b
$ls -l
итого 8
-rw-rw-r-- 1 guest guest 0 Map 21 18:57 a
lrwxrwxrwx 1 guest guest 1 Map 21 18:57 b -> a
```

Сделать жесткую ссылку между `a` и `c`:

```
$ln a c
$ls -l
итого 12
-rw-rw-r-- 2 guest guest 0 Map 21 18:57 a
lrwxrwxrwx 1 guest guest 1 Map 21 18:57 b -> a
```

```
-rw-rw-r-- 2 guest guest 0 Map 21 18:57 c
$ ls -il
итого 12
4247456 -rw-rw-r-- 2 guest guest 0 Map 21 18:57 a
4247457 lrwxrwxrwx 1 guest guest 1 Map 21 18:57 b -> a
4247456 -rw-rw-r-- 2 guest guest 0 Map 21 18:57 c
```

7.1.6. Команда mv

Синтаксис:

`mv [-i] файл_или_каталог файл_или_каталог`

Описание:

Переименовать *файл* или *каталог*, указанный в первом аргументе, в *файл* или *каталог*, указанный во втором.

Опции:

`-i`	Интерактивно — запрашивать разрешение на перезапись уже существующих объектов.

Примеры:

Переименовать `b` в `d`:

```
$mv b d
$ ls -l
итого 12
-rw-rw-r-- 2 guest guest 0 Map 21 18:57 a
-rw-rw-r-- 2 guest guest 0 Map 21 18:57 c
lrwxrwxrwx 1 guest guest 1 Map 21 18:57 d -> a
```

Попытка переименования файла самого в себя:

```
$ mv d c
mv: `d' и `c' - один и тот же файл
```

7.1.7. Команда rm

Синтаксис:

`rm [-f] [-i] файл...`

`rm -r [-f] [-i] каталог...`

Описание:

Удалить файлы или каталоги. Если права доступа не позволяют этого сделать немедленно, то последние выводятся в восьмеричной форме и требуется подтверждение операции.

Опции:

`-f`	Не спрашивать подтверждений, удалить все, что возможно

`-i`	Запрашивать подтверждение на каждый удаляемый файл
`-r`	Выполнить удаление рекурсивно, включая каталоги

Примеры:

Рекурсивное удаление каталога `.mozilla/firefox/gb16cbr1.default/Cache` в принудительном режиме:

```
$ rm -rf .mozilla/firefox/gb16cbr1.default/Cache
```

7.1.8. Команда mkdir

Синтаксис:

```
mkdir [-p] каталог
```

Описание:

Создать *каталог*[7].

Опции:

`-p`	Создаются также и все несуществующие к этому моменту родительские каталоги указанного места

Примеры:

Создать каталог `/tmp/a/b/c`:

```
$ mkdir -p /tmp/a/b/c
$ ls -R /tmp
/tmp:
a             gconfd-guest

/tmp/a:
b

/tmp/a/b:
c

/tmp/a/b/c:
```

7.1.9. Команда rmdir

Синтаксис:

```
rmdir [-p] каталог
```

Описание:

Удалить пустой *каталог*.

Опции:

`-p`	Удалить также и все пустые родительские каталоги

Примеры:

Удалить каталог `/tmp/a/b/c`:

```
$ ls -R /tmp
/tmp:
a             gconfd-guest

/tmp/a:
b

/tmp/a/b:
c

/tmp/a/b/c:

$ rmdir -p /tmp/a/b/c
 $ ls -R /tmp
/tmp:
 gconfd-guest
```

Практическая работа № 5

Тема: Linux управление пользователями. Работа с учетными записями пользователей.

Введение

Учётная запись пользователя - это запись, которая содержит сведения, необходимые для идентификации пользователя при подключении к системе, а также информацию для авторизации и учёта. Это имя пользователя и пароль (или другое аналогичное средство аутентификации - например, биометрические характеристики). Пароль или его аналог, как правило, хранится в зашифрованном или хэшированном виде (в целях его безопасности).

Для повышения надёжности могут быть, наряду с паролем, предусмотрены альтернативные средства аутентификации - например, специальный секретный вопрос (или несколько вопросов) такого содержания, что ответ может быть известен только пользователю. Такие вопросы и ответы также хранятся в учётной записи.

Учётная запись может содержать следующие дополнительные анкетные данные о пользователе:

- имя;
- фамилию;
- отчество;
- псевдоним (ник);
- пол;
- национальность;
- расовую принадлежность;
- вероисповедание
- группу крови;
- резус-фактор;
- возраст;
- дату рождения;
- адрес электронной почты;
- домашний адрес;
- рабочий адрес;
- нетмейловый адрес;
- номер домашнего телефона;
- номер рабочего телефона;
- номер мобильного телефона;
- номер ICQ;
- идентификатор Skype, ник в IRC;
- другие контактные данные систем обмена мгновенными сообщениями;
- адрес домашней страницы и/или блога в Интернете или интранете;
- сведения о хобби;
- сведения о круге интересов;
- сведения о семье;
- сведения о перенесённых болезнях;
- сведения о политических предпочтениях;
- и многое другое

Конкретные категории данных, которые могут быть внесены в такую анкету, определяются администраторами системы.

Учётная запись может также содержать одну или несколько фотографий или аватар пользователя. Учётная запись пользователя также может учитывать различные статистические характеристики поведения пользователя в системе: давность последнего входа в систему, продолжительность последнего пребывания в системе, адрес использованного при подключении компьютера, интенсивность использования системы, суммарное и (или) удельное количество определённых операций, произведённых в системе, и так далее.

Создание учетных записей пользователей

В операционной системе Windows 7 можно создавать несколькими способами как учетные записи пользователей для компьютеров, состоящих в рабочих группах, так и учетные записи пользователей для компьютеров, которые входят в состав домена. Домены, рабочие группы и домашние группы представляют разные методы организации компьютеров в сети. Основное их различие состоит в том, как осуществляется управление компьютерами и другими ресурсами.

Рабочая группа - это группа компьютеров, подключенных к сети, которые совместно используют ресурсы. При настройке сети операционная система Windows автоматически создает рабочую группу и присваивает ей имя по умолчанию.

Домен - это группа компьютеров одной сети, имеющих единый центр, использующий единую базу пользователей, единую групповую и локальную политики, единые параметры безопасности, ограничение времени работы учётной записи и прочие параметры, значительно упрощающие работу системного администратора организации, если в ней эксплуатируется большое число компьютеров.

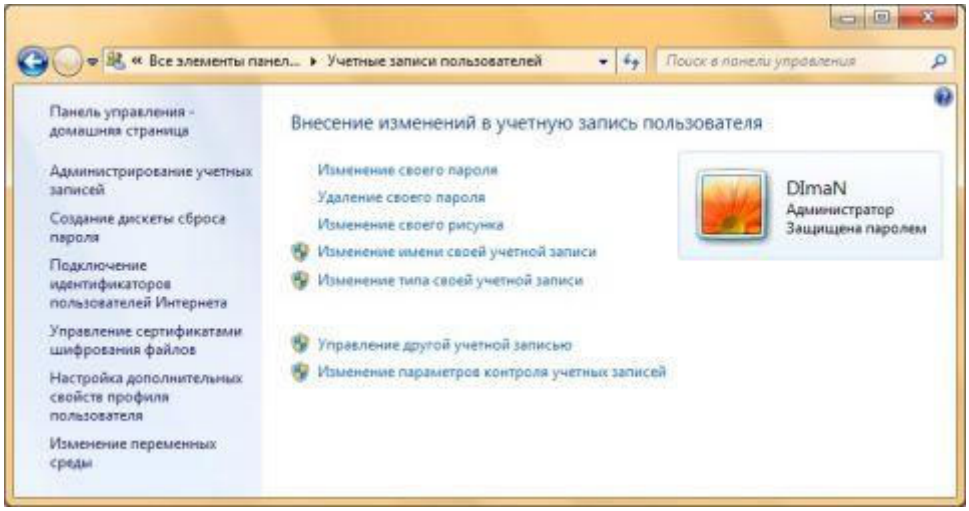
Создание учетных записей пользователей для компьютеров, состоящих в рабочей группе

В операционной системе Windows 7 для компьютеров, которые состоят в рабочей или домашней группе, учетные записи можно создавать следующими способами:

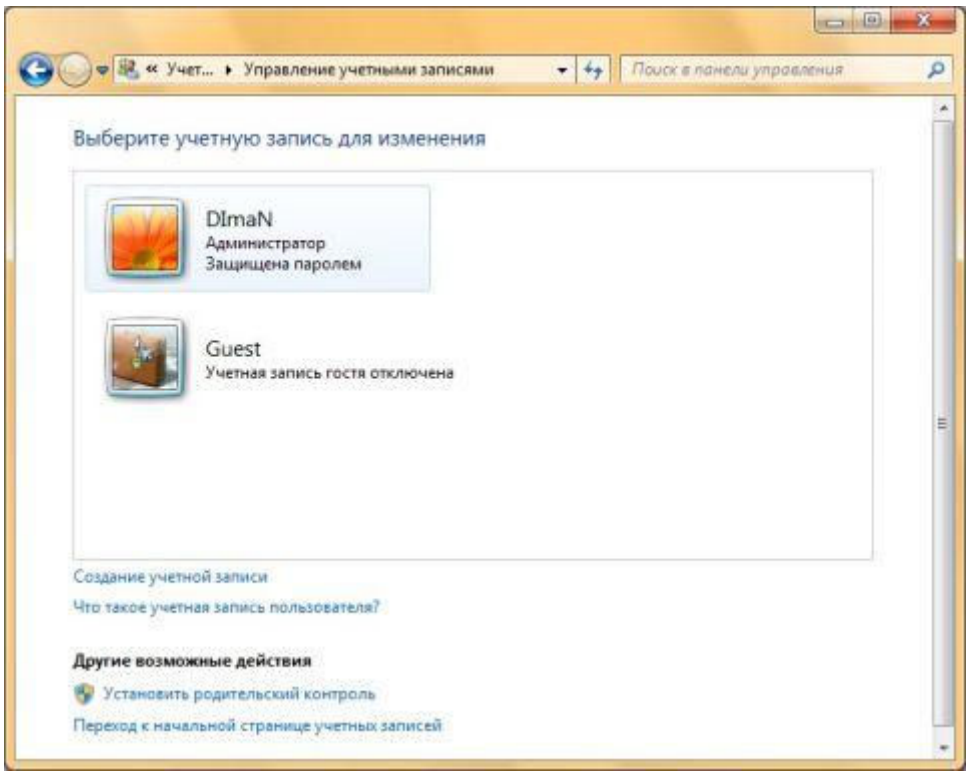
Создание учетной записи при помощи диалога "Управление учетными записями пользователей"

Для того чтобы создать учетную запись при помощи диалога **"Учетные записи пользователей"**, нужно сделать следующее:

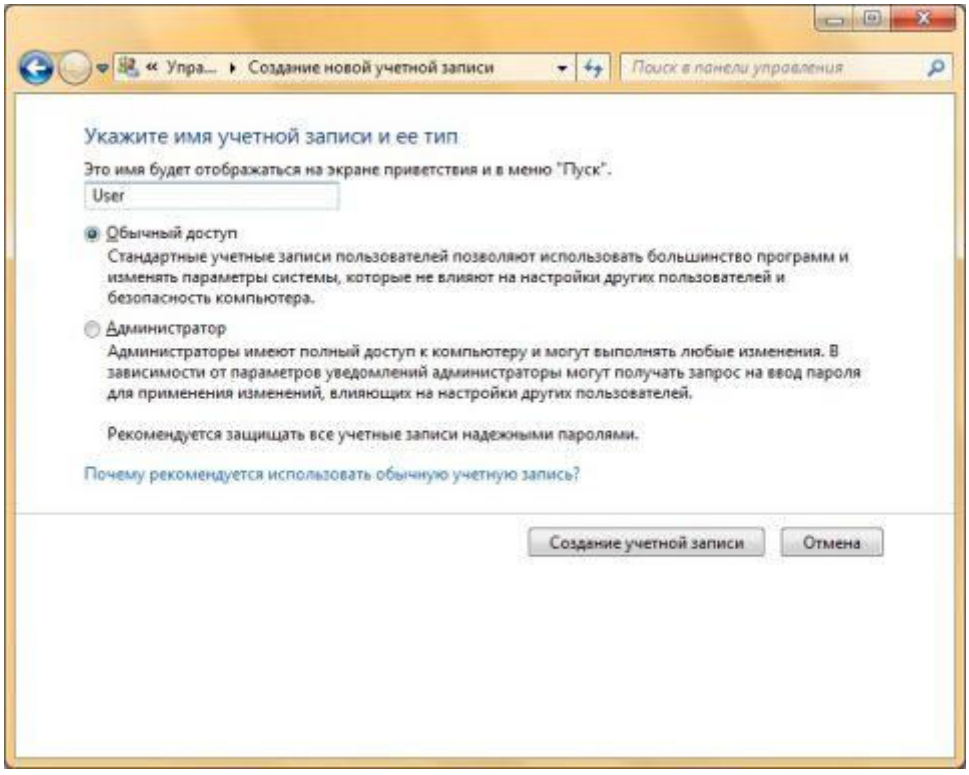
1. Нажмите на кнопку **"Пуск"** для открытия меню, откройте **"Панель управления"** и из списка компонентов панели управления выберите **"Учетные записи пользователей"**;



2. В диалоге **"Учетные записи пользователей"** перейдите по ссылке **"Управление другой учетной записью"**, а затем нажмите на **"Создание учетной записью"**;



3. Здесь нужно будет ввести имя для учетной записи, выбрать тип учетной записи и нажать на кнопку **"Создание учетной записи"**;



Имя пользователя не должно совпадать с любым другим именем пользователя или группы на данном компьютере. Оно может содержать до 20 символов верхнего или нижнего регистров, за исключением следующих: " / \ [ ] : ; / =, + * ? <> @, а также имя пользователя не может состоять только из точек и пробелов.

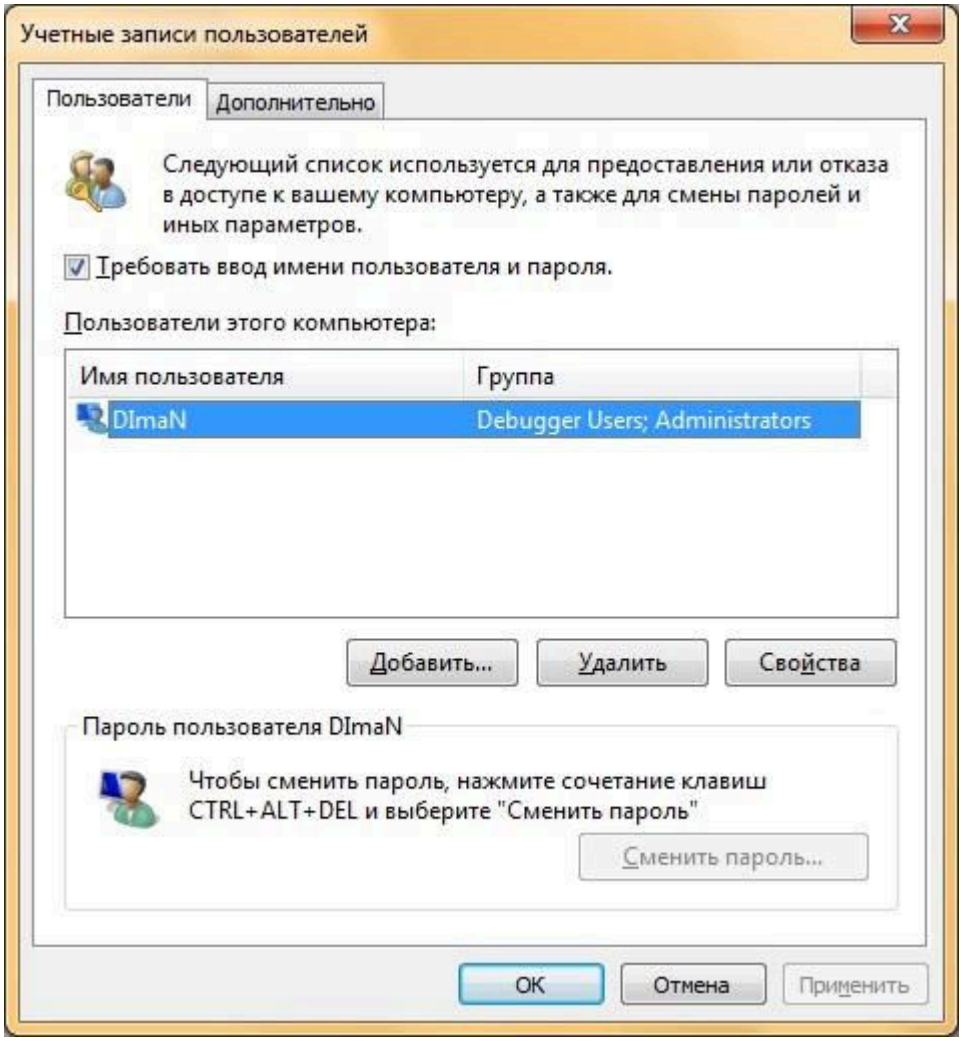
В этом диалоге, можно выбрать одну из двух типов учетных записей: **"обычные учетные записи пользователей"**, которые предназначены для повседневной работы или **"учетные записи администратора"**, которые предоставляют полный контроль над компьютером и применяются только в необходимых случаях.

Создание учетной записи при помощи диалога "Учетные записи пользователей"

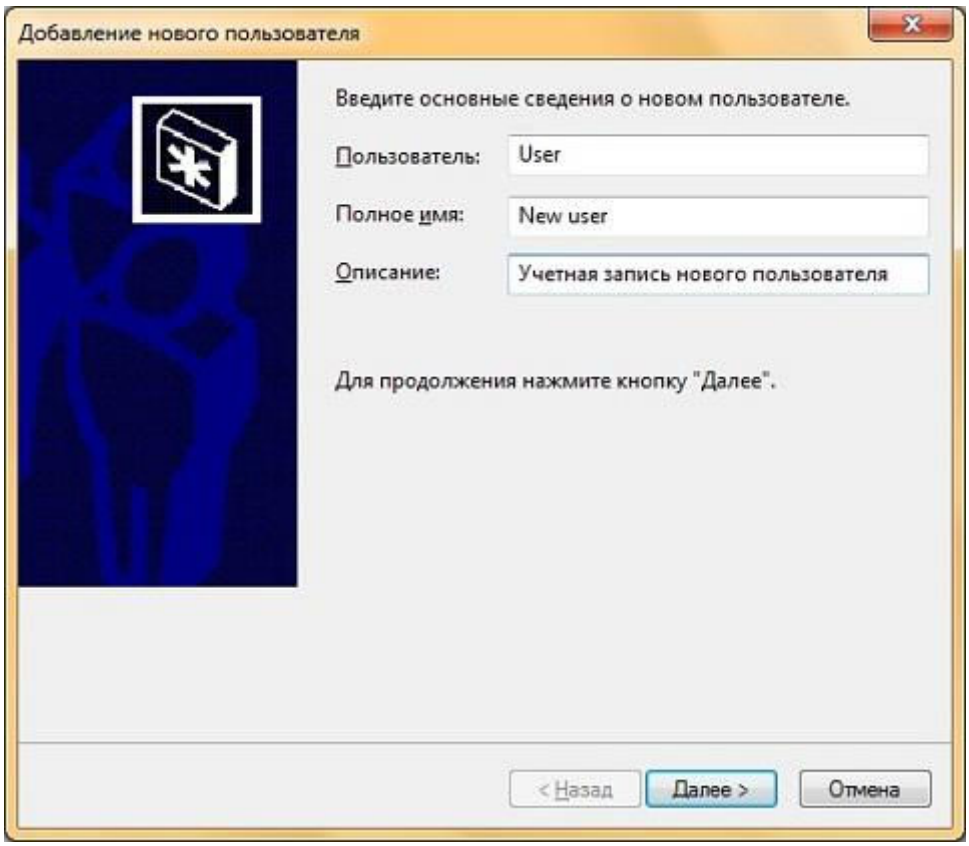
Доступный через панель управления диалог **"Управление учетными записями пользователей"** имеет очень серьезное ограничение: оно предлагает на выбор только учетные записи типа Обычный доступ или Администратор . Для того чтобы при создании нового пользователя его можно было поместить в какую-либо определенную группу, нужно сделать следующее:

1. Воспользоваться комбинацией клавиш +R для открытия диалога **"Выполнить"**;

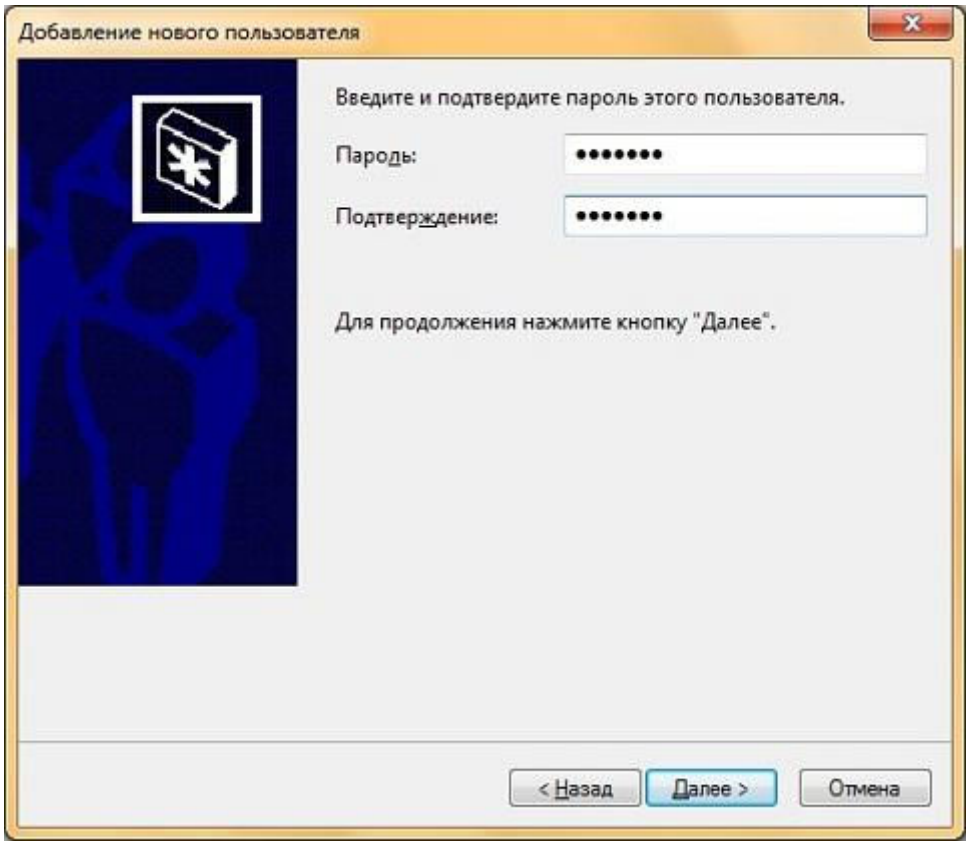
2. В диалоговом окне **"Выполнить"**, в поле **"Открыть"** введите control userpasswords2 и нажмите на кнопку **"ОК"**;



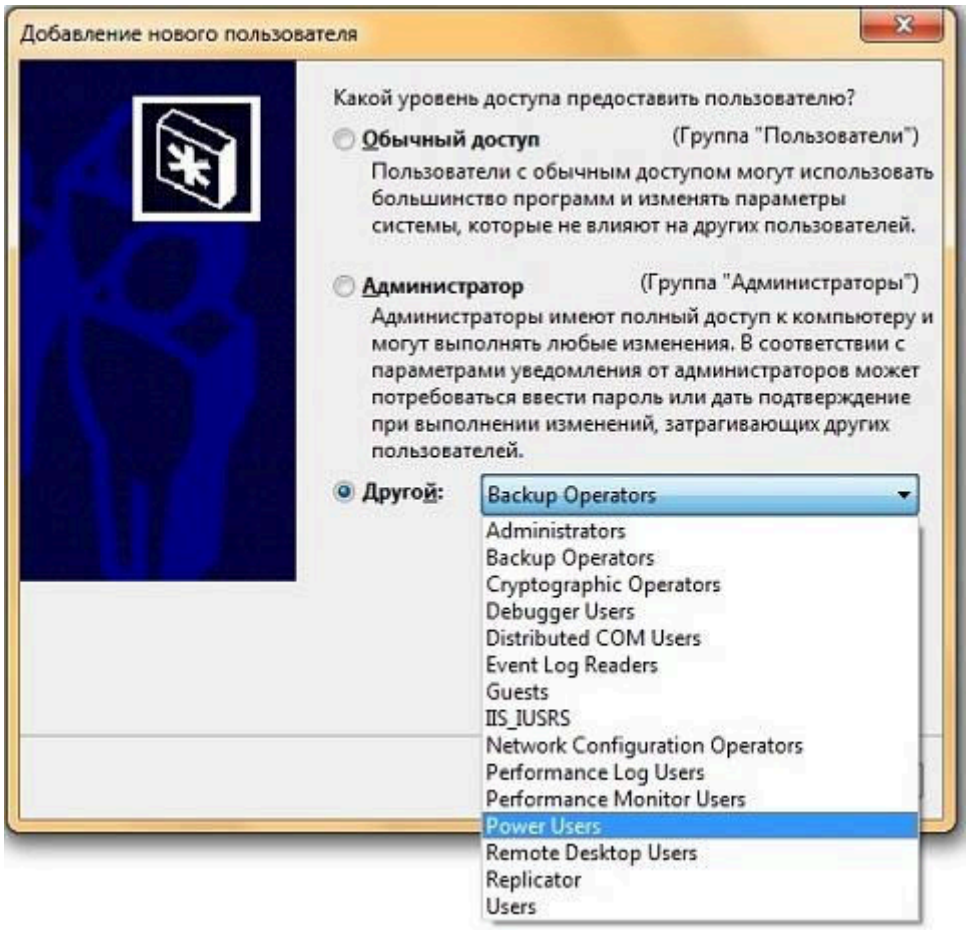
3. В диалоговом окне **"Учетные записи пользователей"** нажмите на кнопку **"Добавить"** для запуска мастера добавления нового пользователя;



4. В появившемся диалоговом окне **"Добавление нового пользователя"** введите имя пользователя. Поля **"Полное имя"** и **"Описание"** не являются обязательными, то есть их можно заполнять при желании. Нажимаем на кнопку **"Далее"**;



5. В диалоге **"Введите и подтвердите пароль этого пользователя"** введите пароль для данной учетной записи, а затем продублируйте его в поле **"Подтверждение"**, после чего нажмите на кнопку **"Далее"**;



6. Это последний диалог мастера добавления нового пользователя. Здесь необходимо установить переключатель, определяющий группу безопасности, к которой должна относиться данная учетная запись пользователя. Можно выбрать одну из следующих групп: Обычный доступ, Администратор или Другой. Последний переключатель стоит использовать в том случае, если нужно отнести пользователя к какой-то другой группе, созданной по умолчанию в операционной системе Windows 7.

В следующем списке перечислены 15 встроенных групп операционной системы Windows 7. Эти права назначаются в рамках локальных политик безопасности:

- **Administrators (Администраторы)**. Пользователи, входящие в эту группу, имеют полный доступ на управление компьютером и могут при необходимости назначать пользователям права пользователей и разрешения на управление доступом. По умолчанию членом этой группы является учетная запись администратора. Если компьютер подключен к домену, группа **"Администраторы домена"** автоматически добавляется в группу **"Администраторы"**. Эта группа имеет полный доступ к управлению компьютером, поэтому необходимо проявлять осторожность при добавлении пользователей в данную группу;
- **Backup Operators (Операторы архива)**. Пользователи, входящие в эту группу, могут архивировать и восстанавливать файлы на компьютере независимо от любых разрешений, которыми защищены эти файлы. Это обусловлено тем, что право

выполнения архивации получает приоритет над всеми разрешениями. Члены этой группы не могут изменять параметры безопасности.

- **Cryptographic Operators (Операторы криптографии)**. Членам этой группы разрешено выполнение операций криптографии.
- **Debugger Users (Группа удаленных помощников)**. Члены этой группы могут предлагать удаленную помощь пользователям данного компьютера.
- **Distributed COM Users (Пользователи DCOM).** Членам этой группы разрешено запускать, активировать и использовать объекты DCOM на компьютере.
- **Event Log Readers (Читатели журнала событий)**. Членам этой группы разрешается запускать журнал событий Windows.
- **Guests (Гости)**. Пользователи, входящие в эту группу, получают временный профиль, который создается при входе пользователя в систему и удаляется при выходе из нее. Учетная запись **"Гость"** (отключенная по умолчанию) также является членом данной встроенной группы.
- **IIS_IUSRS**. Это встроенная группа, используемая службами IIS.
- **Network Configuration Operators (Операторы настройки сети).** Пользователи, входящие в эту группу, могут изменять параметры TCP/IP, а также обновлять и освобождать адреса TCP/IP. Эта группа не имеет членов по умолчанию.
- **Performance Log Users (Пользователи журналов производительности).** Пользователи, входящие в эту группу, могут управлять счетчиками производительности, журналами и оповещениями на локальном или удаленном компьютере, не являясь при этом членами группы **"Администраторы"**.
- **Performance Monitor Users (Пользователи системного монитора)**. Пользователи, входящие в эту группу, могут наблюдать за счетчиками производительности на локальном или удаленном компьютере, не являясь при этом участниками групп **"Администраторы"** или **"Пользователи журналов производительности"**.
- **Power Users (Опытные пользователи)**. По умолчанию, члены этой группы имеют те же права пользователя и разрешения, что и учетные записи обычных пользователей. В предыдущих версиях операционной системы Windows эта группа была создана для того, чтобы назначать пользователям особые административные права и разрешения для выполнения распространенных системных задач. В этой версии операционной системы Windows учетные записи обычных пользователей предусматривают возможность выполнения большинства типовых задач настройки, таких как смена часовых поясов. Для старых приложений, требующих тех же прав опытных пользователей, которые имелись в предыдущих версиях операционной системы Windows, администраторы могут применять шаблон безопасности, который позволяет группе **"Опытные пользователи"** присваивать эти права и разрешения, как это было в предыдущих версиях операционной системы Windows.
- **Remote Desktop Users (Пользователи удаленного рабочего стола).** Пользователи, входящие в эту группу, имеют право удаленного входа на компьютер.
- **Replicator (Репликатор)**. Эта группа поддерживает функции репликации. Единственный член этой группы должен иметь учетную запись пользователя домена, которая используется для входа в систему службы репликации контроллера домена. Не добавляйте в эту группу учетные записи реальных пользователей.
- **Users (Пользователи)**. Пользователи, входящие в эту группу, могут выполнять типовые задачи, такие как запуск приложений, использование локальных и сетевых принтеров и блокировку компьютера. Члены этой группы не могут предоставлять общий доступ к папкам или создавать локальные принтеры. По умолчанию членами этой группы являются группы **"Пользователи домена"**, **"Проверенные**

пользователи" и **"Интерактивные"**. Таким образом, любая учетная запись пользователя, созданная в домене, становится членом этой группы.

Создание учетной записи при помощи оснастки "Локальные пользователи и группы"

Оснастка **"Локальные пользователи и группы"** расположена в компоненте **"Управление компьютером"**, представляющем собой набор средств администрирования, с помощью которых можно управлять одним компьютером, локальным или удаленным. Оснастка **"Локальные пользователи и группы"** служит для защиты и управления учетными записями пользователей и групп, размещенных локально на компьютере. Можно назначать разрешения и права для учетной записи локального пользователя или группы на определенном компьютере (и только на этом компьютере).

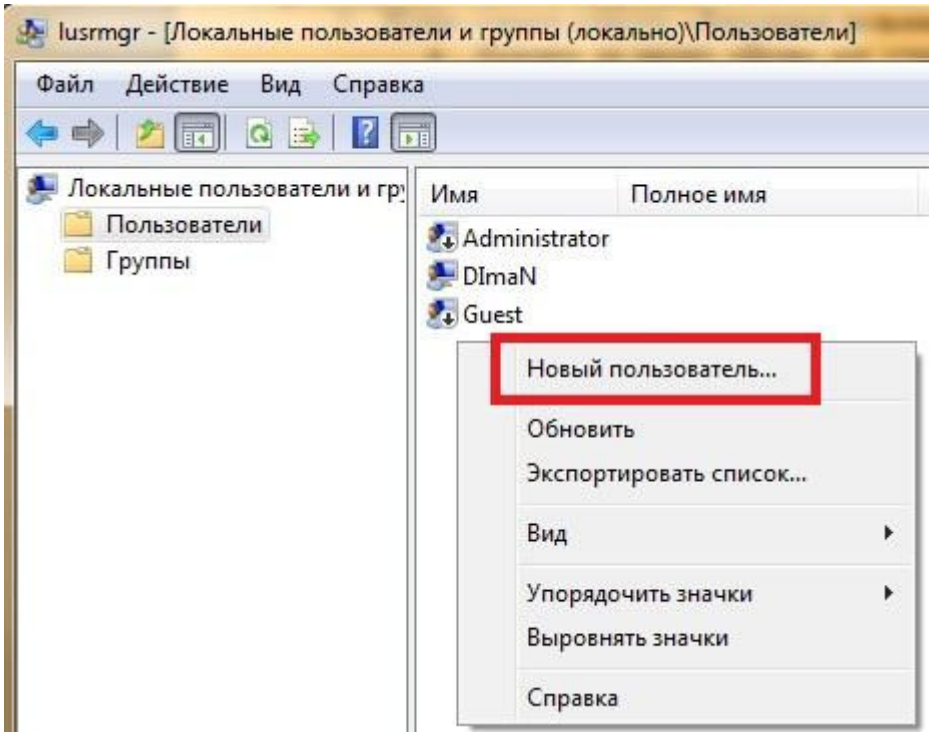
Использование оснастки **"Локальные пользователи и группы"** позволяет ограничить возможные действия пользователей и групп путем назначения им прав и разрешений. Право дает возможность пользователю выполнять на компьютере определенные действия, такие как архивирование файлов и папок или завершение работы компьютера. Разрешение представляет собой правило, связанное с объектом (обычно с файлом, папкой или принтером), которое определяет, каким пользователям, и какой доступ к объекту разрешен.

Для того чтобы создать локальную учетную запись пользователя при помощи оснастки **"Локальные пользователи и группы"**, нужно сделать следующее:

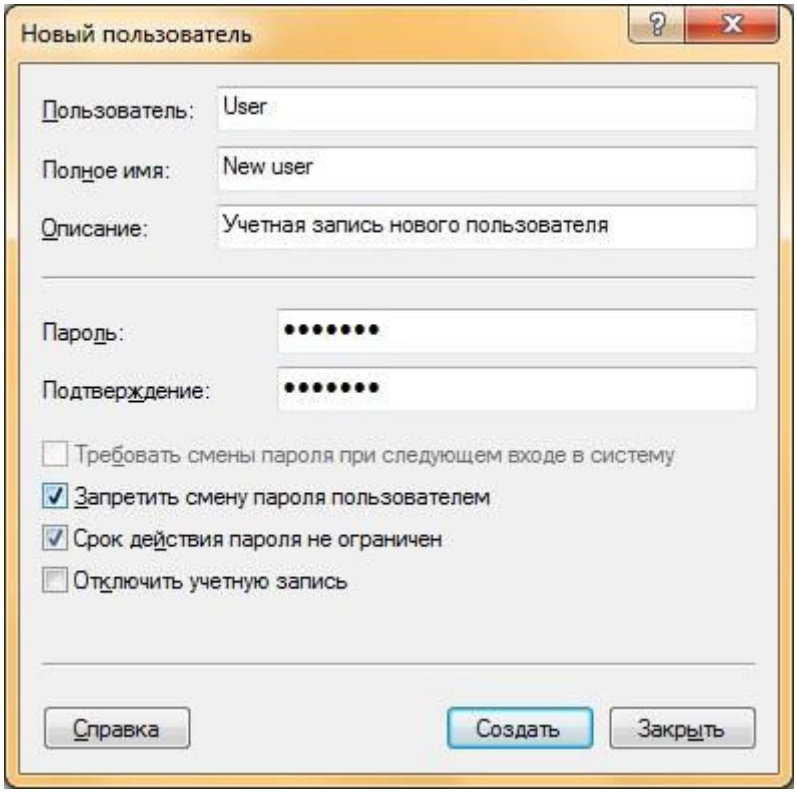
1. Откройте оснастку **"Локальные пользователи и группы"** одним из следующих способов:

- Нажмите на кнопку **"Пуск"** для открытия меню, откройте **"Панель управления"** и из списка компонентов панели управления выберите **"Администрирование"**, затем откройте компонент **"Управление компьютером"**. В **"Управлении компьютером"** откройте **"Локальные пользователи и группы"**;
- Открыть **"Консоль управления ММС"**. Для этого нажмите на кнопку **"Пуск"**, в поле поиска введите mmc, а затем нажмите на кнопку **"Enter"**. Откроется пустая консоль ММС. В меню **"Консоль"** выберите команду **"Добавить или удалить оснастку"** или воспользуйтесь комбинацией клавиш Ctrl+M. В диалоге **"Добавление и удаление оснасток"** выберите оснастку **"Локальные пользователи и группы"** и нажмите на кнопку **"Добавить"**. Затем нажмите на кнопку **"Готово"**, а после этого - кнопку **"ОК"**. В дереве консоли откройте узел **"Локальные пользователи и группы (локально)"**;
- Воспользоваться комбинацией клавиш +R для открытия диалога **"Выполнить"**. В диалоговом окне **"Выполнить"**, в поле **"Открыть"** введите lusrmgr.msc и нажмите на кнопку **"ОК"**;

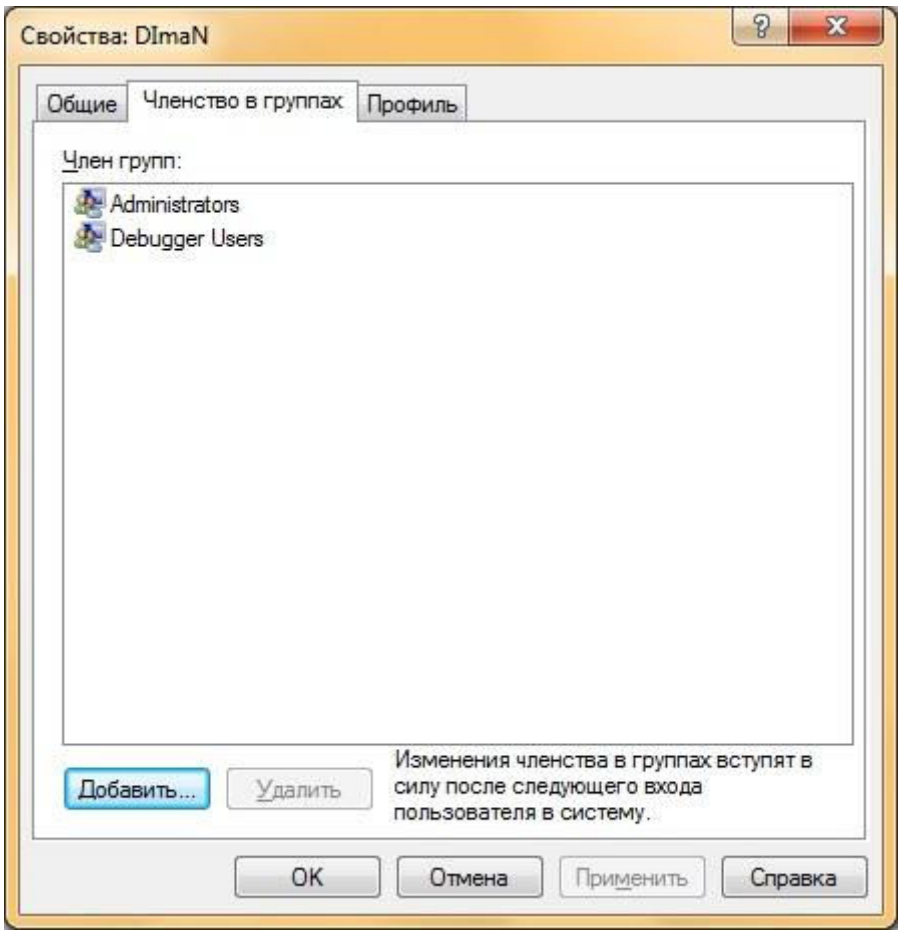
2. Откройте узел **"Пользователи"** и либо в меню **"Действие"**, либо из контекстного меню выбрать команду **"Новый пользователь"**;



3. В диалоговом окне **"Новый пользователь"** введите соответствующие сведения. Помимо указанных данных, можно воспользоваться следующими флажками: **Требовать смену пароля при следующем входе в систему**, **Запретить смену пароля пользователем**, **Срок действия пароля не ограничен**, **Отключить учетную запись** и нажать на кнопку **"Создать"**, а затем **"Закрыть"**.

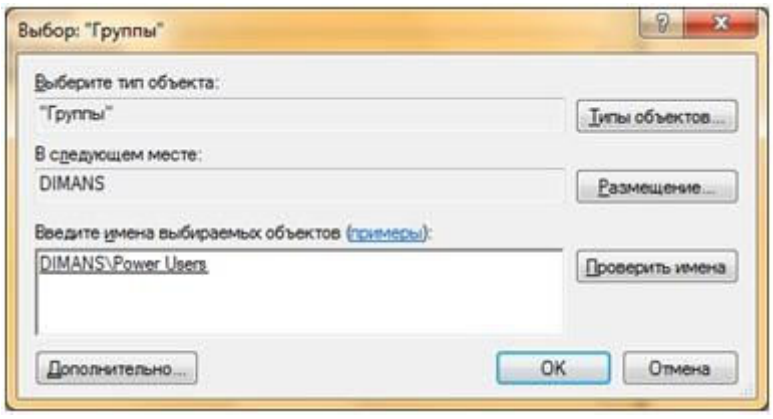


Для того чтобы добавить пользователя в группу, дважды щелкните имя пользователя для получения доступа к странице свойств пользователя. На вкладке **"Членство в группах"** нажмите на кнопку **"Добавить"**.

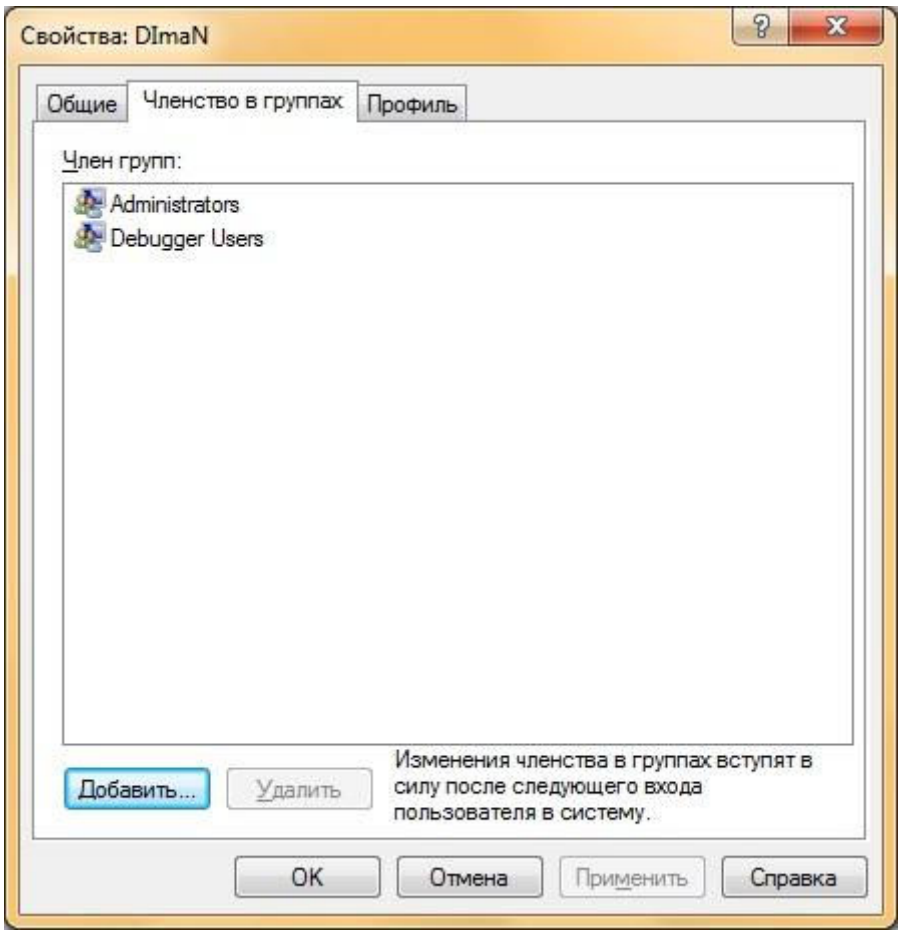


В диалоге **"Выбор группы"** можно выбрать группу для пользователя двумя способами:

1. В поле **"Введите имена выбираемых объектов"** введите имя группы и нажмите на кнопку **"Проверить имена"**, как показано на следующем скриншоте:



2. В диалоге **"Выбор группы"** нажмите на кнопку **"Дополнительно"**, чтобы открыть диалоговое окно **"Выбор группы"**. В этом окне нажмите на кнопку **"Поиск"**, чтобы отобразить список всех доступных групп, выберите подходящую группу и нажмите два раза на кнопку **"ОК"**.



Создание учетной записи при помощи командной строки

Помимо вышеперечисленных способов, учетные записи пользователей можно создавать, изменять и удалять при помощи командной строки. Для этого нужно выполнить следующие действия:

1. Запустите командную строку от имени администратора;

2. Для создания учетной записи при помощи командной строки используйте команду net user .

Команда net user используется для добавления пользователей, установки паролей, отключения учетных записей, установки параметров и удаления учетных записей. При выполнении команды без параметров командной строки отображается список учетных записей пользователей, присутствующих на компьютере. Информация об учетных записях пользователей хранится в базе данных учетных записей пользователей.

Пример команды:

net user User /add /passwordreq:yes /times:monday-friday, 9am-6pm/fullname:"New user"

Используемые параметры:

/add - этот параметр указывает, что необходимо создать новую учетную запись;

/passwordreq - этот параметр отвечает за то, чтобы при первом входе в систему пользователь сменил свой пароль;

/times - этот параметр определяет, сколько раз пользователю разрешено входить в систему. Здесь можно указывать как единичные дни, так и целые диапазоны (например Sa или M-F). Для указания времени допускается как 24-часовый формат, так и 12-часовый формат;

/fullname - этот параметр идентичен полю "Полное имя" при создании пользователя предыдущими способами.

Создание учетных записей пользователей для компьютеров, состоящих в домене

В серверной операционной системе Windows Server 2008 или Windows Server 2008 R2 в домене Active Directory учетные записи пользователей можно создавать шестью способами. Рассмотрим подробно каждый из них:

Создание пользователей при помощи оснастки "Active Directory - пользователи и компьютеры"

Для создания нового пользователя в домене при помощи оснастки **"Active Directory - пользователи и компьютеры"** нужно сделать следующее:

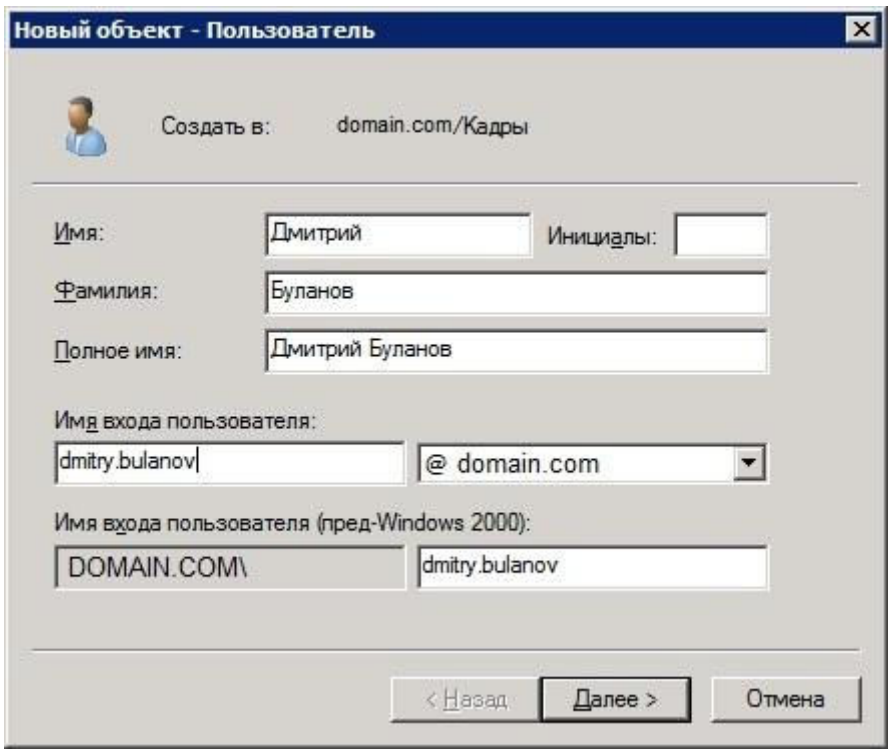
1. Открыть оснастку **"Active Directory - пользователи и компьютеры"**;

2. В дереве консоли разверните узел, предоставляющий домен и найдите контейнер, в котором нужно создать учетную запись пользователя;

3. Нажмите на подразделение или контейнер правой кнопкой мыши, выберите опцию **"Создать"** и примените команду **"Пользователь"**;

4. В диалоговом окне **"Новый объект - пользователь"** введите в поле **"Имя"** - имя пользователя, в поле **"Инициалы"** - его инициалы, в поле **"Фамилия"** - фамилию пользователя. Поле **"Полное имя"** должно заполниться автоматически, согласно CN пользователя. В поле **"Имя входа"** введите имя входа пользователя в систему и в раскрывающемся списке выберите суффикс основного имени пользователя, который будет прикреплен к имени входа с символом @. В поле **"Имя входа пользователя (пред-Windows 2000)"** введите имя входа для систем, предшествующих Windows 2000, так называемое низкоуровневым именем входа. Нажать на кнопку **"Далее"**.

5. В следующем диалоге введите пароль для данной учетной записи, а затем продублируйте его в поле **"Подтверждение"** и установите нужные флажки, после чего нажмите на кнопку **"Далее"**.



6. В последнем диалоге можно просмотреть введенные параметры и нажать на кнопку **"Готово"** для создания нового пользователя.

Создание пользователей с помощью командной строки

Для автоматизации создания любых объектов в домене Active Directory можно использовать команду DSADD USER UserDN, при помощи которой можно создавать объекты пользователей и принимать параметры, указывающие его свойства. Нового пользователя при помощи командной строки можно создать следующим образом:

dsadd user "CN=Дмитрий Буланов, OU=Кадры, DC=server, DC=com" -samid Dmitry.bulanov -pwd * -mustchpwd yes -profile \\server01\Profiles\dmitry.bulanov -fn "Дмитрий" -ln "Буланов" -display "Дмитрий Буланов" -upn Dmitry.bulanov@server.com

Определение используемых параметров:

Samid - указывает имя входа пользователя;

Pwd - этот параметр определяет пароль для учетной записи пользователя. Если указывать символ *, то будет предложено ввести пароль пользователя;

Mustchpwd - указывает, что пользователь должен изменить свой пароль при следующем входе в систему;

Profile - указывает путь к профилю учетной записи пользователя;

Fn - указывает имя пользователя;

Ln - указывает фамилию пользователя;

Display - указывает отображаемое имя пользователя;

Upn - указывает имя входа пользователя (пред-Windows 2000).

Импорт пользователей с помощью команды CSVDE

Утилита командной строки CSVDE позволяет импортировать и экспортировать объекты Active Directory в виде текстового файла с разделенными запятыми (Comma-Separated Values, *.csv). Эти файлы можно создавать и изменять при помощи таких программ, как Блокнот или, например, Microsoft Office Excel. Эта утилита - способ автоматизации создания учетных записей пользователя на основе информации пользователей из базы данных Excel и Microsoft Office Access. Команда импортирует текстовый файл, в котором строка определяет атрибуты импорта с помощью их имен LDAP. Синтаксис команды следующий:

Csvde -i -f имя_файла -k

Параметр i указывает режим импорта, а параметр k используется для игнорирования ошибок.

CSV файл должен выглядеть следующим образом:

DN, objectClass, sAMAccountName, sn, givenName, userPrincipalName

"cn=Дмитрий Буланов, ou=Пользователи, dc=server, dc=com", user, Dmitry.bulanov, Буланов, Дмитрий, dmitry.bulanov@server.com

Импортировать пароли при помощи команды CSVDE нельзя.

Импорт пользователей с помощью команды LDIFDE

При помощи команды LDIFDE также можно импортировать и экспортировать объекты Active Directory. В данном случае используется стандарт файлового формата LDIF (Lightweight Directory Access Protocol Data Interchange Format). Этот файловый формат состоит из блока строк, которые вместе образуют одну операцию. Разные операции разделяются пустой строкой. Каждая строка содержит имя атрибута, а после него должно стоять двоеточие со значением атрибута. Далее можно увидеть листинг LDIF файла:

DN: CN=Дмитрий Буланов, OU=пользователи, DC=server, DC=com

changeType: add

CN: Дмитрий Буланов

objectClass: user

sAMAccountName: Dmitry.bulanov

userPrincipalName: Dmitry.bulanov@server.com

givenName: Дмитрий

sn: Буланов

displayName: Дмитрий Буланов

Файл можно создавать в такой программе как Блокнот, но сохранять его нужно с расширением *.ldf. в командной строке введите следующее:

Ldifde -i -f имя_файла -k

Создание пользователей с помощью Windows PowerShell

При помощи Windows PowerShell для создания пользователя в Active Directory пользователя можно создать следующим образом:

1. Подключитесь к контейнеру, в котором будет создан объект;

2. Примените метод Create совместно с классом и отличительным именем RDN;

3. Заполните атрибуты при помощи метода Put;

4. Подтвердите изменения при помощи метода SetInfo.

Далее можно увидеть листинг скрипта PowerShell - *.ps1-файла:

```
$ObjOU=[ADSI]"LDAP://OU=Пользователи, DC=server, DC=com"

$ObjUser=$ObjOU.Create("user", "CN=Дмитрий Буланов")

$ObjUser.Put("sAMAccountName", "dmitry.bulanov")

$ObjUser.Put("userPrincipalName", "dmitry.bulanov @server.com")

$ObjUser.Put("displayName", "Дмитрий Буланов")

$ObjUser.Put("givenName", "Дмитрий")

$ObjUser.Put("sn", "Буланов")

$ObjUser.Put("description", "Тестировщик программного обеспечения")

$ObjUser.Put("company", "Company")

$ObjUser.Put("department", "Отдел тестирования")

$ObjUser.Put("title", "Тестировщик программного обеспечения")

$ObjUser.Put("mail", " dmitry.bulanov @server.com ")

$ObjUser.Put("c", "UA")
$ObjUser.Put("postalCode", "73003")
$ObjUser.Put("st", "Херсон")
$ObjUser.Put("l", "Херсон")

$ObjUser.Put("streetAddress", "Улица")
```

```
$ObjUser.Put("postOfficeBox", "Номер дома")

$ObjUser.SetInfo()

$ObjUser.SetPassword("P@ssword")

//$ObjUser.Put("pwdLastSet", 0) - для смены пароля при следующем старте

$ObjUser.psbase.InvokeSet("AccountDisabled", $false)

$ObjUser.SetInfo()
```

Можно вводить все строки вручную, а можно использовать *.ps1-файлы для автоматизации создания новых пользователей. Для того, чтобы разрешить Windows PowerShell открывать скрипты, введите следующую команду:

Set-ExecutionPolicy RemoteSigned

Политика выполнения указывает сценарии, которые можно запускать. После назначения политики выполнения можно запустить сценарий, но если указывать для запуска только имя сценария, то может возникнуть ошибка. Чаще всего нужно будет указывать еще и путь к самому сценарию.

Создание пользователей с помощью VBScript

В связи с тем, что VBScript также как и Windows PowerShell использует интерфейс ADSI для манипулирования объектами в Active Directory, процесс создания пользователя в VBScript идентичен созданию пользователя в Windows PowerShell. Прежде всего, сценарий подключается к контейнеру OU, в котором будет создан пользователь. После чего сценарий применит к объекту ADSI инструкцию GetObject. При присвоении объекта переменной, для создания объектной ссылки используется инструкция Set.

После этого активизируется метод Create для создания объекта конкретного класса так же, как и в PowerShell. Далее используется метод Put, но аргументы заключаются в круглые скобки. Последняя строка - идентична Windows PowerShell. Пример скрипта:

```
Set objOU=GetObject("LDAP: //OU=Пользователи, DC=server, DC=com")

Set objUser=objOU.Create("user", "CN=Дмитрий Буланов")

objUser.Put "sAMAccountName", " dmitry.bulanov"

objUser.Put "displayName", " Дмитрий Буланов"

objUser.Put "givenName", " Дмитрий"

objUser.Put "sn", " Буланов"

ObjUser.SetInfo()
```

Практическая работа № 6

Тема: Установка и удаление программ и оборудования в ос Windows

Цель: получение практических навыков в конфигурировании ОС Windows для ПК в соответствии с его функциональным назначением.

Теоретические сведения:

Установка и удаление программ.

С помощью данной утилиты, которая вызывается командой **Пуск – Панель управления – Установка и удаление программ**можно установить новую программу или корректно удалить ненужную программу. (Рис. 1). Обычно установочный файл имеет имя Setup.exe.

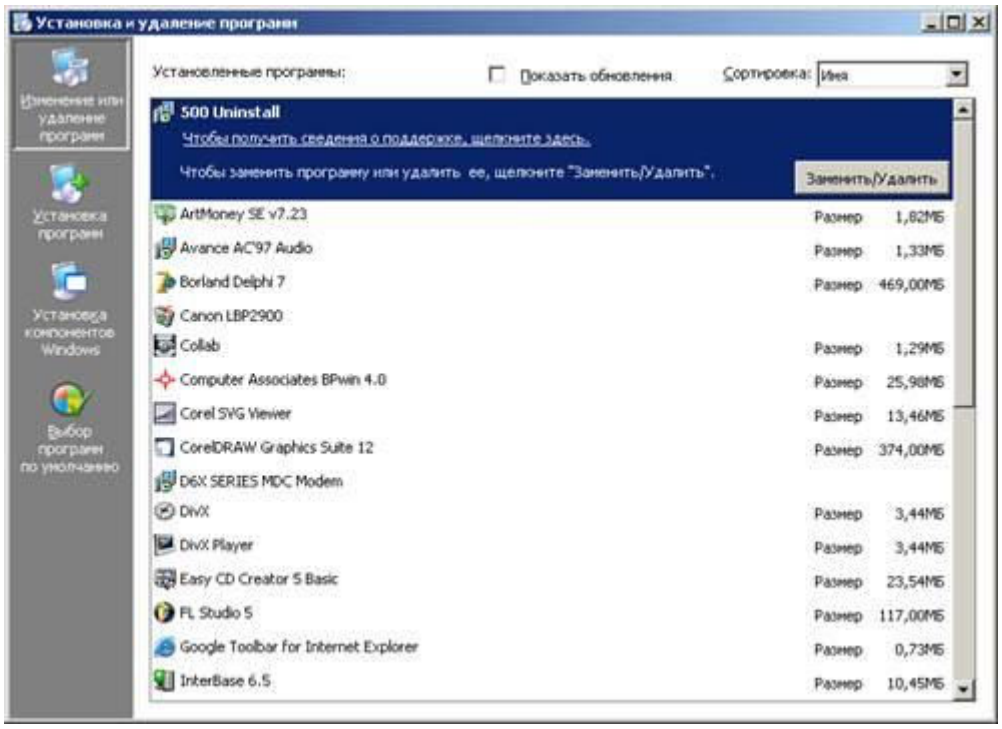


Рис. 1. Установка и удаление программ.

Установка и удаление компонентов Windows.

Для установки и удаления компонентов Windows выполните следующие действия:

1. В окне **Установка и удаление программ**щелкните **Установка компонентов Windows.**На экране появится окно с перечнем компонентов (Рис. 2). Дважды щелкните на название компонента, чтобы узнать, какие из входящих в него программ установлены, а какие нет.

2. Для установки компонента отметьте его флажком и выберите **Далее**.

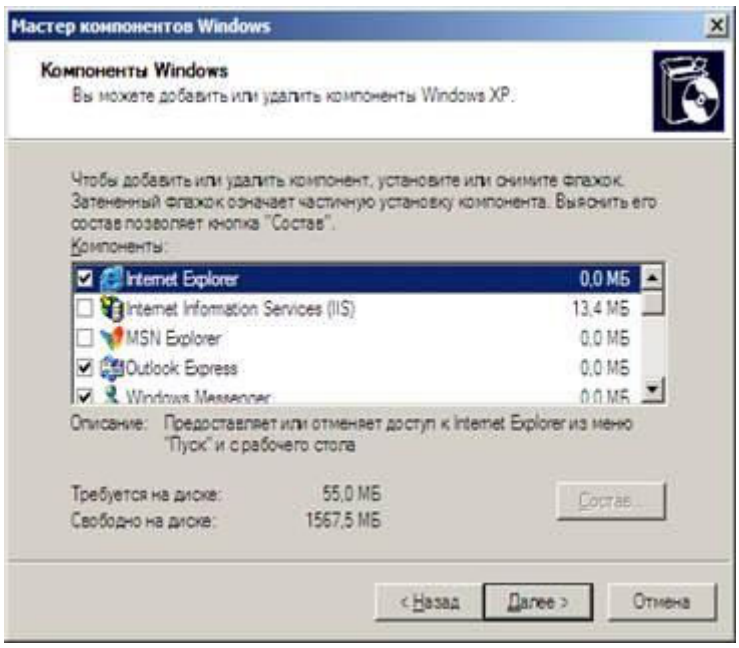


Рис. 2 Установка компонентов

В случае повреждения системного реестра операционная система Windows может не загрузиться. Единственным выходом кажется переустановка Windows, процесс долгий. Но этого легко избежать, используя универсальный способ - резервное копирование.

Чтобы не знать никаких проблем с восстановлением Windows после не удачных экспериментов с реестром, сделайте следующее.

· Скопируйте инсталянт Windows на жесткий диск и сделайте загрузочную дискету. В любом случае восстановление Windows придется вести из режима командной строки, а в таком случае доступ к CD-ROM будет невозможен, если не загрузить драйвера для него.

· Установите на жесткий диск в новую папку в корневом каталоге Norton Commander для MS DOS либо Volkov Commander либо Far manager.

· Найдите в каталоге с операционной системой файлы System.dat и User.dat. Скопируйте их, в корневой каталог или любой другой. По окончании процесса список резервов должен выглядеть так:

-папка WinXp в корневом каталоге с инсталянтом Windows;

-папка VC в корневом каталоге с Volkov Commander;

-файлы system.da_ и user.da_ в корневом каталоге;

-файлы system.da_ и user.da_ в каталоге VC.

Если ваш компьютер сконфигурирован для работы на нем нескольких пользователей, то алгоритм действий будет такой же, только к копируемым файлам еще придется прибавить свой User.dat - из каталога Profiles\Имя. Если катастрофа все же происходит, и при загрузке операционной системы вы видите сообщение: "Системный реестр поврежден",

или "Невозможно загрузить драйвер…", или что-нибудь подобное, то действия должны быть следующими:

1. Нажать кнопку ***Reset***.

2. При появлении строчки Starting Windows XP нажать кнопку ***F8***. В появившемся меню выберите пункт ***Command Prompt Only*** и дождитесь появления системного приглашения. Если у вас компьютер настроен так, что на F8 он не реагирует, то вставьте в дисковод загрузочную дискету и загрузитесь с нее.

3. С помощью команд "***cd имя_каталога***" (переход в нужный каталог) или "cd .." (переход на один уровень выше) доберитесь до директории с Norton и запустите оболочку.

4. Зайдите для начала в папку с операционной системой, сотрите там файлы ***System.dat*** и ***User.dat***, а затем переименуйте файлы ***System.da0*** и ***User.da0*** в ***System.dat*** и ***User.dat***. Для начала надо попробовать использовать автоматически зарезервированные копии - там сохранены самые последние настройки.

5. Перезагрузите компьютер кнопкой ***Reset*** (без дискеты). Если Windows загружается без проблем, то все нормально, продолжайте работать. Если же нет, то пришло время воспользоваться резервной копией реестра.

6. Повторите шаги 1-3. Но теперь уже скопируйте ваши резервные файлы реестра (в вышеприведенном примере - ***system.da_*** и ***user.da_***) из корневого каталога в папку с операционной системой, сотрите там файлы ***System.dat*** и ***User.dat***, а затем переименуйте файлы ***System.da_*** и ***User.da_*** в ***System.dat*** и ***User.dat***. Снова перезагрузитесь.

7. Если опять операционная система не загружается, попробуйте заменить, как описано в шаге 6, файлы ***System.dat*** и ***User.dat*** из каталога с операционной системой файлами из второй резервной копии, если она есть.

Практическая работа № 7

Тема: Работа с виртуальной машиной в ос Windows XP

Цель работы: Изучение основных понятий о виртуальных машинах для их практического применения.

Теоретические сведения:

1. Виртуальные машины в целом

1.1. Определение и понятие

Чтобы построить полный взгляд на виртуальные машины, разберем для начала, а что такое виртуальная машина?

Виртуальная машина — программная или аппаратная среда, исполняющая некоторый код (например, байт-код, шитый код, p-код или машинный код реального процессора), или спецификация такой системы (например: «виртуальная машина языка программирования Си»). [Википедия]

Для сравнения приведем несколько других определенней, а именно: Виртуальная машина — это полностью изолированный программный контейнер, способный выполнять собственную операционную систему и приложения, как физический компьютер. Виртуальная машина работает абсолютно так же, как физический компьютер, и содержит собственные виртуальные (т.е. программные) ЦП, ОЗУ, жесткий диск и сетевую интерфейсную карту (NIC).

Проще говоря, виртуальная машина – это программа, которую вы запускаете из своей операционной системы. Программа эмулирует реальную машину. На виртуальные машины, как и на реальные, можно ставить операционные системы. У неё есть BIOS, отведенное место на вашем жестком диске, сетевые адаптеры для соединения с реальной машиной, сетевыми ресурсами или другими виртуальными машинами.

1.2. Преимущества и недостатки виртуальных машин

1.2.1. Преимущества виртуальных машин

Приведу вам несколько преимуществ использования виртуальных машин:

5. Приведу самый просто пример. Нынче, как мы знаем, вышли новые операционные системы. Windows Vista и Windows 7. И как многие из вас убедились, некоторые приложения, в частности игры, на них не работают. Так в чём проблема? Когда можно установить виртуальную машину с, допустим, операционной системой Windows XP. И всё прекрасно будет работать.
6. Второй пункт можно отнести к злобным хакерам или просто к компьютерным хулиганам. Имеется в виду, что на виртуальной машине вы можете спокойно написать вирус или вредоносное программное обеспечение, которое сможет повредить вам лишь гостевую операционную систему виртуальной машины.
7. Третий пункт можно было отнести ко второму. А именно то, что на виртуальную машину вы можете ставить любое ПО, не опасаясь чего-либо. Вы можете экспериментировать с различными настройками и прочее.

8. Ну и одно из самых главных это то, что вы можете легко изучать новые операционные системы, не стирая свою старую.

Это конечно далеко не все преимущества виртуальных машин. Каждый пользователь может сам придумать, для чего ему нужна виртуальная машина.
Перед возможностью установки нескольких хостовых операционных систем на один компьютер с их раздельной загрузкой, виртуальные машины имеют следующие неоспоримые преимущества:

6. Возможность работать одновременно в нескольких системах, осуществлять сетевое взаимодействие между ними.
7. Возможность сделать «снимок» текущего состояния системы и содержимого дисков одним кликом мыши, а затем в течение очень короткого промежутка времени вернуться в исходное состояние.
8. Простота создания резервной копии операционной системы (не надо создавать никаких образов диска, всего лишь требуется скопировать папку с файлами виртуальной машины).
9. Возможность иметь на одном компьютере неограниченное число виртуальных машин с совершенно разными операционными системами и их состояниями.
10. Отсутствие необходимости перезагрузки для переключения в другую операционную систему.

1.2.2. Недостатки виртуальных машин

Тем не менее, несмотря на все преимущества, виртуальные машины также имеют и свои недостатки:

5. Потребность в наличии достаточных аппаратных ресурсов для функционирования нескольких операционных систем одновременно.
6. Операционная система работает несколько медленнее в виртуальной машине, нежели на «голом железе». Однако, в последнее время показатели производительности гостевых систем значительно приблизились к показателям физических ОС (в пределах одних и тех же ресурсов), и вскоре, за счет улучшения технологий реализации виртуальных машин, производительность гостевых систем практически будет равна реальным.
7. Существуют методы определения того, что программа запущена в виртуальной машине (в большинстве случаев, производители систем виртуализации сами предоставляют такую возможность). Вирусописатели и распространители вредоносного программного обеспечения, конечно же, в курсе этих методов и в последнее время включают в свои программы функции обнаружения факта запуска в виртуальной машине, при этом никакого ущерба вредоносное ПО гостевой системе не причиняет.
8. Различные платформы виртуализации пока не поддерживают полную виртуализацию всего аппаратного обеспечения и интерфейсов. В последнее время количество поддерживаемого аппаратного обеспечения стремительно растет у всех производителей платформ виртуализации. Помимо основных устройств компьютера, уже поддерживаются сетевые адаптеры, аудиоконтроллеры, интерфейс USB 2.0, котроллеры портов COM и LPT и приводы CD-ROM. Но хуже всего обстоят дела с виртуализацией видеоадаптеров и поддержкой функций аппаратного ускорения трехмерной графики.

Все недостатки в принципе можно решить, да и преимущества виртуальных машин перевешивают их недостатки. Именно поэтому виртуализация сейчас продвигается семимильными шагами вперёд. А пользователи находят всё больше и больше причин их использовать.

1.3. Архитектура виртуальных машин

Виртуализация один из важных инструментов разработки компьютерных систем, а сами виртуальные машины используются в самых разных областях.
Виртуальные машины разрабатываются большим количеством специалистов, преследующих самые разные цели, и в этой области существует не так уж много общепринятых концепций. Поэтому лучше всего будет рассмотреть понятие виртуализации и всё разнообразие архитектур виртуальных машин в единой перспективе.

1.3.1. Абстракция и виртуализация

Компьютерные системы разрабатываются по определенной иерархии и имеют хорошо определенные интерфейсы, из-за чего они и продолжают развиваться. Использование таких интерфейсов облегчает независимую разработку аппаратных и программных подсистем силами разных групп специалистов. Абстракции скрывают детали реализации нижнего уровня, уменьшая сложность процесса проектирования.

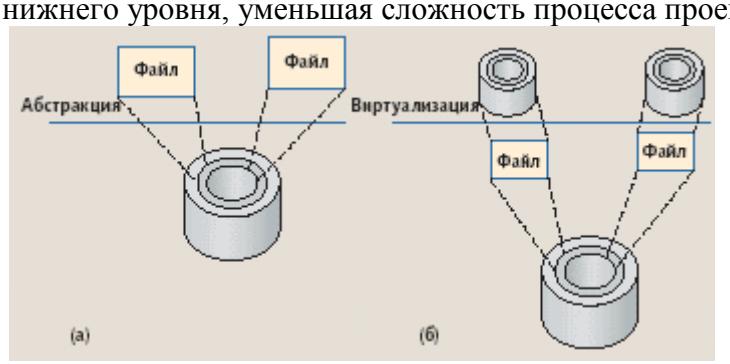


Рис. 1 Абстракция и виртуализация в применение к дисковой памяти.

На рис. 1 (а) приведен пример абстракции в применении к дисковой памяти. Операционная система абстрагируется от тонкостей адресации на жестком диске, от его секторов и дорожек, чтобы для прикладной программы диск выглядел как набор файлов переменного размера. Опираясь на эту абстракцию, «прикладные» программисты могут создавать файлы, записывать и читать данные, не зная устройства и физической организации жесткого диска.
Концепция архитектуры системы команд компьютера (instruction set architecture, ISA) наглядно иллюстрирует преимущества хорошо определенных интерфейсов. Они позволяют разрабатывать взаимодействующие компьютерные подсистемы не только в разных организациях, но и в разные периоды, иногда разделенные годами. Например, Intel и AMD создают микропроцессоры с системой команд IA-32 (x86), в то время как разработчики Microsoft пишут программное обеспечение, которое компилируется в эту систему команд. Поскольку обе стороны соблюдают спецификацию ISA, можно ожидать, что программное обеспечение будет правильно выполняться любым ПК на базе микропроцессора с архитектурой IA-32.
К сожалению, хорошо определенные интерфейсы имеют и недостатки. Подсистемы и компоненты, разработанные по спецификациям разных интерфейсов, не способны взаимодействовать друг с другом. Например, приложения, распространяемые в двоичных кодах, привязаны к определенной ISA и зависят от конкретного интерфейса к операционной системе. Несовместимость интерфейсов может стать сдерживающим

фактором, особенно в мире компьютерных сетей, в котором свободное перемещение программ столь же необходимо, как и перемещение данных.
Виртуализация позволяет обойти эту несовместимость. Виртуализация системы или компонента (например, процессора, памяти или устройства ввода/вывода) на конкретном уровне абстракции отображает его интерфейс и видимые ресурсы на интерфейс и ресурсы реальной системы. Следовательно, реальная система выступает в роли другой, виртуальной системы или даже нескольких виртуальных систем.
В отличие от абстракции, виртуализация не всегда нацелена на упрощение или сокрытие деталей. В примере на рис. 1(б) виртуализация позволяет преобразовать один большой диск в два меньших виртуальных диска, каждый из которых имеет собственные секторы и дорожки. При отображении виртуальных дисков на реальный программные средства виртуализации используют абстракцию файла как промежуточный шаг. Операция записи на виртуальный диск преобразуется в операцию записи в файл (и, следовательно, в операцию записи на реальный диск). Отметим, что в данном случае никакого абстрагирования не происходит — уровень детализации интерфейса виртуального диска (адресация секторов и дорожек) ничем не отличается от уровня детализации реального диска.

1.3.2. Процессные и системные виртуальные машины

Понятия пошли от того, что система и процесс видят машину по-разному, поэтому и виртуальные машины бывают процессные и системные.
Процессная виртуальная машина — это виртуальная платформа для выполнения отдельного процесса. Она предназначена для поддержки процесса, создаётся при его активации и «умирает» после его окончания. Системная виртуальная машина – полнофункциональная, постоянно действующая системная среда, служащая для поддержки операционной системы вместе с большим количеством её пользовательских процессов; она обеспечивает «гостевой» операционной системе доступ к виртуальным аппаратным средствам, в том числе к процессору и памяти, устройствам ввода/вывода, а иногда — и к графическому интерфейсу.
Процесс или система, которые выполняются на виртуальной машине, называются гостем, платформа, поддерживающая виртуальную машину, — хостом. Программное обеспечение, реализующее процессную виртуальную машину, называют рабочей средой, а программное обеспечение виртуализации системной виртуальной машины – монитором виртуальной машины.
Процессные виртуальные машины создают среды ABI и API для пользовательских приложений, что позволяет в многозадачном режиме осуществлять репликацию операционной среды, эмулировать систему команд, оптимизировать код или выполнять программы на языках высокого уровня.
Системная виртуальная машина обеспечивает полнофункциональную среду, в которой могут сосуществовать операционная система и несколько процессов, относящихся к разным пользователям. С помощью них одна аппаратная платформа может поддерживать несколько гостевых операционных систем одновременно.

1.3.3. Типы виртуализаций

Рассмотрим основные типы виртуализации различных компонент ИТ — инфраструктуры.

8. Виртуализация операционной системы.Является наиболее распространенной в данный момент формой виртуализации. Виртуальная операционная система (виртуальная машина) представляет собой, как правило, совмещение нескольких

операционных систем, функционирующих на одной аппаратной основе. Каждая из виртуальных машин управляется отдельно при помощи VMM (Virtual Machine Manager). Лидерами в области поставок решений для виртуализации информационных систем являются Microsoft, AMD, Intel и VMware.

9. Виртуализация серверов приложений. Под данным процессом виртуализации понимают процесс интеллектуальной балансировки нагрузки. Балансировщик нагрузки управляет несколькими веб — серверами и приложениями, как единой системой, пользователь, при этом, «видит» только один сервер, который, фактически, предоставляет функционал нескольких серверов.
10. Виртуализация приложений. Под виртуализацией приложений следует понимать использование программных решений в рамках изолированной виртуальной среды (более подробно виртуализация приложений будет рассмотрена в последующих лекциях).
11. Виртуализация сети. Представляет собой объединение аппаратных и программных ресурсов в единую виртуальную сеть. Выделяют внутреннюю виртуализацию сети — создающую виртуальную сеть между виртуальными машинами одной системы, и внешнюю — объединяющую несколько сетей в одну виртуальную.
12. Виртуализация аппаратного обеспечения. В данном случае виртуализация заключается в разбиении компонент аппаратного обеспечения на сегменты, управляемые отдельно друг от друга. В некоторых случаях, виртуализация операционных систем невозможна без виртуализации аппаратного обеспечения.
13. Виртуализация систем хранения. В свою очередь делится на два типа: виртуализацию блоков и виртуализацию файлов. Виртуализация файлов, как правило используется в системах хранения, при этом ведутся записи о том, какие файлы и каталоги находятся на определенных носителях. Виртуализация файлов отделяет статичный указатель нахождения виртуального файла (C:\, к примеру) от его физического местоположения. Т.е. при запросе пользователем файла C:\file.doc решение виртуализации файлов отправит запрос к месту реального размещения файла. Виртуализация блоков. Используется в сетях распределенного хранения данных. Сервера — хранилища данных используют RAID-технологию. iSCSI интерфейс также использует блочную виртуализацию, позволяя операционной системе распределить виртуальное блочное устройство. Более подробную информацию о виртуализации систем хранения см. в п.№4 списка источников для самостоятельного изучения.
14. Виртуализация сервисов. По своей сути, виртуализация сервисов является объединением всех вышеуказанных типов виртуализации. Решение виртуализации сервисов позволяет работать с приложением вне зависимости от физического расположения его частей, объединяя и управляя их взаимодействием.

Приведенная выше типология рассматривает виртуализацию, в зависимости от части ИТ — инфраструктуры, в которой она применяется. Подходы к созданию интерфейсов между виртуальными машинами и системами виртуализации ресурсов также можно разделить на следующие типы:

- Полная виртуализация — технология, которая обеспечивает полную симуляцию базового оборудования, гостевая операционная система остается в нетронутом виде.
- Аппаратная виртуализация — технология, позволяющая запускать на одном компьютере (хосте) несколько экземпляров операционных систем (гостевых операционных систем). При этом гостевые ОС независимы друг от друга и от аппаратной платформы. Аппаратная виртуализация представляет собой набор инструкций, облегчающих выполнение операций на аппаратном уровне, которое до

этого могли выполняться только программно, при этом затрачиваются дополнительные программные ресурсы.
- Паравиртуализация — техника виртуализации, при которой гостевые операционные системы подготавливаются для исполнения в виртуализированной среде, для этих целей в ядро ОС вносят незначительные изменения. Для взаимодействия с гостевой операционной системой используется API — интерфейс.

2. Различные виртуальные машины

Все отличия существующих виртуальных машин, по сути, сводятся лишь к перечню поддерживаемых ими **операционных систем**, а так же **стоимости**. Наиболее распространены сегодня системы VirtualBox, Windows Virtual PC и VMWare. Чем же они отличаются?

2.1. ORACLE VirtualBox — универсальная бесплатная виртуальная машина

VirtualBox — очень простой, мощный и бесплатный инструмент для виртуализации, развивающийся благодаря поддержжке знаменитой корпорации ORACLE. Он распространяется бесплатно, с открытым исходным кодом. VirtualBox

позволяет устанавливать в качестве «гостевой» практически любую современную операционную систему, будь то Windows, MacOS или любой из многочисленных представителей семейства Linux.

Преимуществом VirtualBox является простой и понятный пользовательский интерфейс. Хорошо сделан перевод на русский язык. Все основные функции вынесены в виде кнопок под меню. Создание виртуальных машин выполняется с помощью пошагового мастера.

VirtualBox поддерживает работу с сетями, поэтому ваша виртуальная ОС сможет легко выйти в Интернет. Очень полезной является функция «снимков» операционной системы. Виртуальная машина записывает на винчестер «точки восстановления», к которым вы в любой момент можете откатить гостевую систему в случае возникновения ошибок или сбоев.

2.2 Windows Visual PC — виртуальная машина от Microsoft

Windows Virtual PC — виртуальная машина для работы только и исключительно с Windows. Установить на Visual PC операционную систему Linux или MacOS просто невозможно.

Visual PC позволяет запускать несколько разных копий Windows на одном компьютере. Поддерживается работа с операционными системами Microsoft разных поколений, в том числе с 64-битными.

Плюсом Visual PC является возможность задать, какая из запущенных виртуальных машин будет более приоритетной по сравнению с другими. При этом «хостовый» компьютер сможет в автоматическом режиме выделять под ее нужды большее количество ресурсов за счёт других виртуальных систем, если «гостевой» системе это необходимо.

Моноплатформенность виртуальной машины Visual PC является её главным недостатком, впрочем, если требуется тестировать только разные версии Windows, это не актуально.

Некоторым недостатком можно считать менее функциональный и менее удобный чем в VirtualBox интерфейс. В остальном Visual PC вполне надёжный инструмент, позволяющий тестировать операционные системы Microsoft.

2.3 VMware Workstation — для серьёзных задач

VMware Workstation – мощная, платная, максимально-надёжная программа для виртуализации, которая поддерживает работу с Windows и Linux. Для виртуализации MacOS, данная машина не предназначена.

Благодаря высокой надёжности и широчайшей функциональности VMware Workstation часто используется не просто для тестирования, а даже для постоянной работы виртуальных машин в качестве серверов даже для бизнес-приложений, будь то фаервол, отеляющий сеть организации от Интернет или даже сервер какой-либо базы данных.

VMware Workstation можно очень гибко настраивать, включая множество параметров сетевых подключений для работы с интернетом. Система имеет собственный виртуальный 3D-ускоритель, который позволяет получить высокое качество графики.

Интерфейс VMware Workstation достаточно грамотно организован, поэтому освоиться со всем её богатым функционалом довольно легко. В программе полностью поддерживается русский язык.
Необходимо отметить, что у VMware Workstation есть бесплатный «младший брат» — VMWare Player. В отличие от версии Workstation, плеер не умеет создавать виртуальные машины, но позволяет запускать ранее созданные. Эта программа будет полезна в случаях тестирования, когда, к примеру, разработчик какой-либо автоматизированной системы передат её на ознакомление именно в виде образа виртуальной машины. Эта практика получает всё большее распространение, поскольку избавляет пользователя от необходимости разворачивать незнакомую программу самостоятельно.

Практическая часть

9. Установить **ORACLE VirtualBox.**
10. Запустить программу на исполнение.
11. Создать виртуальную машину для установки ОС Windiws XP.
12. Укажите объем оперативной памяти 343МБ.
13. Создайте новый виртуальный жесткий диск (тип VDI).
14. Укажите формат хранения «Фиксированный виртуальный жесткий диск».
15. Размер жесткого диска должен быть 11ГБ.
16. Покажите результат преподавателю.

Вопросы для самоконтроля

7. Что называется виртуальной машиной?
8. Какие преимущества у виртуальной машины? Какие недостатки?
9. Чем отличается системная виртуальная машина от процессорной?
10. Перечислите основные типы виртуализаций.
11. Какие существуют подходы к созданию интерфейсов между виртуальными машинами и системами виртуализации ресурсов?
12. Какие существуют виртуальные машины? В чем их отличие друг от друга?

Практическая работа № 8

Тема: Linux. монтируемые файловые системы.

Монтирование файловых систем

С одной стороны, файловая система Linux – это одно большое дерево с корневой директорией /, с другой стороны, мы говорим о файловых системах на различных устройствах и разделах. Как разрешить это кажущееся противоречие? Монтирование корневой файловой системы (/) является частью процесса инициализации. Все остальные созданные файловые системы невозможно использовать в Linux до тех пор, пока они не будут *смонтированы* в определенных *точках монтирования*.

В текущем наборе смонтированных файловых систем точка монтирования является обычной директорией дерева каталогов, в которую файловая система устройства добавляется при монтировании. Монтирование – это процесс, благодаря которому файловая система устройства становится доступной для использования. Например, можно смонтировать файловые системы разделов жесткого диска в точках монтирования /boot, /tmp или /home, файловую систему дискеты в точке монтирования /mnt/floppy, а файловую систему компакт-диска в точке монтирования /media/cdrom1. Как видите, точки монтирования могут располагаться как в корневой директории, так и в поддиректориях дерева каталогов любого уровня вложенности.

Помимо файловых систем на разделах, дискетах и компакт-дисках, существуют и другие типы файловых систем. Например, файловая система tmpfs является файловой системой в виртуальной памяти. Сетевые файловые системы, такие как NFS или AFS, позволяют монтировать на локальном компьютере файловые системы, расположенные на других компьютерах. Можно даже создать файл в существующей файловой системе, отформатировать его в качестве отдельной файловой системы (возможно, другого типа), после чего смонтировать эту новую файловую систему. Этот способ часто используется при работе с образами оптических носителей, когда загруженный ISO-образ CD- или DVD-диска монтируется вместо реального физического носителя для последующего копирования. Другим таким примером может являться пространство подкачки, размещенное не на выделенном разделе, а в файле.

Несмотря на то, что в процессе монтирования фактически монтируется *файловая система* какого либо устройства (или другого ресурса), принято говорить просто о "монтировании устройства", хотя на самом деле подразумевается, что речь идет о "монтировании файловой системы устройства".

Для монтирования и демонтирования файловых систем обычно требуются привилегии пользователя root. Если вы вошли в систему под учетной записью обычного пользователя, используйте для получения прав суперпользователя команду `su -` или `su`. Если в наших примерах вы видите, что приглашение командной строки начинается с символа #, как в листинге 1, значит, для их выполнения необходимо обладать привилегиями пользователя root.

Базовая форма команды `mount` принимает на вход два параметра: имя устройства (или другого ресурса), содержащего монтируемую файловую систему, и точку монтирования. В листинге 1 приведен пример, в котором мы смонтировали FAT32-раздел /dev/sda9 в точке монтирования /dos.

Листинг 1. Монтирование в точку монтирования /dos

```
[root@echidna ~]# mount /dev/sda9 /dos
```

Точка монтирования, в которую монтируется файловая система, должна существовать. В противном случае возникнет ошибка, и нужно будет либо создать указанную точку монтирования, либо использовать другую точку монтирования, как показано в листинге 2.

Листинг 2. Ошибка монтирования

```
[root@echidna ~]# mount /dev/sda9 /dos
mount: mount point /dos does not exist
[root@echidna ~]# mkdir /dos
[root@echidna ~]# mount /dev/sda9 /dos
```

Когда файловая система монтируется в существующую директорию, все файлы и поддиректории этой файловой системы становятся файлами и поддиректориями точки монтирования. Если директория точки монтирования уже содержала какие-либо файлы или поддиректории, то они не теряются, но становятся невидимыми до тех пор, пока смонтированная файловая система не будет демонтирована. Во избежание таких проблем рекомендуем использовать в качестве точек монтирования только пустые директории.

После того как файловая система смонтирована, все файлы и директории, создаваемые или копируемые в точку монтирования или ее поддиректории, будут созданы в смонтированной файловой системе. Если в нашем примере мы создадим файл /dos/sampdir/file.txt, то он будет создан в файловой системе FAT32, которая была смонтирована в точку монтирования /dos.

Обычно команда `mount` автоматически определяет тип монтируемой файловой системы. Тем не менее, иногда вам может потребоваться явно указать тип файловой системы, для чего можно использовать опцию `-t`, как показано в листинге 3.

Листинг 3. Монтирование с явным указанием типа файловой системы

```
[root@echidna ~]# mount -t vfat /dev/sda9 /dos
```

Чтобы просмотреть список смонтированных файловых систем, используйте команду `mount` без каких-либо параметров. В листинге 4 показан пример нашей тестовой системы. Заметьте, что для просмотра списка файловых систем привилегии суперпользователя не требуются.

Листинг 4. Просмотр смонтированных файловых систем

```
[ian@echidna ~]$ mount
/dev/sda6 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda2 on /grubfile type ext3 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
gvfs-fuse-daemon on /home/ian/.gvfs type fuse.gvfs-fuse-daemon
(rw,nosuid,nodev,user=ian)
dw.raleigh.ibm.com:/vol/vol1/dwcontent on /mnt/dwcontent type nfs
(rw,addr=9.42.155.6)
```

```
/dev/sdb9 on /mnt/sdb9 type ext3 (rw)
/dev/sda9 on /dos type vfat (rw)
/dev/sr0 on /media/KNOPPIX type iso9660
(ro,nosuid,nodev,uhelper=udisks,uid=1000,gid=1000
,iocharset=utf8,mode=0400,dmode=0500)
```

Эту же информацию можно получить, просмотрев содержимое файла /proc/mounts или /etc/mtab; оба этих файла содержат информацию о смонтированных файловых системах

Опции монтирования

Команда `mount` имеет ряд опций, которые изменяют ее поведение по умолчанию. Например, с помощью опции `-o ro` можно смонтировать файловую систему в режиме "только для чтения". Если файловая система уже была смонтирована, добавьте опцию `remount`, как показано в листинге 5.

Листинг 5. Повторное монтирование в режиме только для чтения

```
[root@echidna ~]# mount -o remount,ro /dos
```

Примечания.

- Если используется несколько опций (например, `remount` и `ro`), необходимо разделять их запятыми.
- Если выполняется повторное монтирование уже смонтированной файловой системы, достаточно указать либо точку монтирования, либо имя устройства. Нет необходимости указывать и то, и другое.
- Невозможно смонтировать файловую систему, предназначенную "только для чтения", в режиме чтения и записи. Носители, файловые системы которых изменить невозможно (например, CD- и DVD-диски), автоматически монтируются в режиме "только для чтения".
- Для повторного монтирования допускающего возможность записи устройства в режиме чтения и записи, используйте опцию `-o remount,rw`.

Команды повторного монтирования не будут успешно завершены, если какой-либо процесс использует файлы или директории повторно монтируемой файловой системы. Дополнительная информация содержится в разделе этой статье.

Метки, идентификаторы и ссылки

В UNIX и ранних системах Linux директория /dev обычно содержала записи всех возможных устройств, которые могли быть подключены к системе. Любое подключаемое устройство всегда располагалось в одном и том же месте дерева /dev, поэтому использование имен устройств (например, /dev/sda6) было естественным. Современные устройства, поддерживающие технологию горячей замены, такие как устройства с интерфейсом USB или Firewire (IEEE 1394), могут подключаться каждый раз к разным портам. В таких случаях может возникнуть желание всегда монтировать определенный USB-накопитель в одну и ту же директорию (например, /media/myusbstick) независимо от того, к какому порту он будет подключен. В другой статье нашей серии (тема 102) упоминалось, что вместо имен устройств для идентификации разделов можно использовать метки и универсальные идентификаторы UUID (Universally Unique ID). Если файловая система раздела поддерживает метки или UUID, то их можно использовать и

для работы с командой `mount`. Чтобы узнать, какой идентификатор и метка (если она есть) назначены устройству, используйте команду `blkid`. В листинге 6 показано, как использовать `blkid` для определения метки и UUID нашего корневого раздела, создать две дополнительные точки монтирования и смонтировать в них корневой раздел. Этот пример приведен лишь для демонстрации, и его не стоит использовать его на практике.

Листинг 6. Монтирование с использованием меток и идентификаторов UUID

```
[root@echidna ~]# blkid /dev/sda6
/dev/sda6: LABEL="Fedora-13-x86_64" UUID="082fb0d5-a5db-41d1-ae04-
6e9af3ba15f7"
 TYPE="ext4"
[root@echidna ~]# mkdir /mnt/sda6label
[root@echidna ~]# mkdir /mnt/sda6uuid
[root@echidna ~]# mount LABEL="Fedora-13-x86_64" /mnt/sda6label
[root@echidna ~]# mount UUID="082fb0d5-a5db-41d1-ae04-6e9af3ba15f7"
/mnt/sda6uui
```

Благодаря менеджеру устройств udev, в директории /dev можно найти дополнительные символические ссылки на такие устройства, как, например, жесткие диски. В листинге 7 показаны ссылки на устройство /dev/sda6 в моей операционной системе Fedora 13.

Листинг 7. Символические ссылки на устройство /dev/sda6

```
[ian@echidna ~]$ find /dev -lname "*sda6"
/dev/disk/by-label/Fedora-13-x86_64
/dev/disk/by-uuid/082fb0d5-a5db-41d1-ae04-6e9af3ba15f7
/dev/disk/by-path/pci-0000:00:1f.2-scsi-0:0:0:0-part6
/dev/disk/by-id/wwn-0x50014ee001a8d027-part6
/dev/disk/by-id/scsi-SATA_WDC_WD1001FALS-_WD-WMATV3772868-part6
/dev/disk/by-id/ata-WDC_WD1001FALS-00J7B1_WD-WMATV3772868-part6
/dev/block/8:6
```

Символические ссылки можно использовать как еще один способ указания имени устройства при его монтировании.

Процесс загрузки и файл fstab

В другой статье нашей серии упоминалось, что параметр `root=` говорит менеджерам загрузки LILO и GRUB о том, что файловая система должна быть смонтирована как корневая. После того, как файловая система смонтирована, процесс инициализации запускает команду `mount` с опцией `-a` для автоматического монтирования ряда файловых систем, перечисленных в файле /etc/fstab. В листинге 8 показано содержимое файла /etc/fstab в тестовой операционной системе Fedora 13. В этом примере все разделы диска идентифицируются с помощью UUID. В листинге 9 показан другой пример, на этот раз из операционной системы Ubuntu 9.10.

Листинг 8. Пример файла fstab в Fedora 13

```
#
# /etc/fstab
# Created by anaconda on Fri May 28 12:37:05 2010
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
```

```
UUID=082fb0d5-a5db-41d1-ae04-6e9af3ba15f7 /                    ext4    defaults
1 1
UUID=488edd62-6614-4127-812d-cbf58eca85e9 /grubfile            ext3    defaults
1 2
UUID=2d4f10a6-be57-4e1d-92ef-424355bd4b39 swap                 swap    defaults
0 0
UUID=ba38c08d-a9e7-46b2-8890-0acda004c510 swap                 swap    defaults
0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
```

Листинг 9. Пример файла fstab в Ubuntu 9.10

```
# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
# / was on /dev/sda7 during installation
UUID=8954fa66-e11f-42dc-91f0-b4aa480fa103 /               ext3
errors=remount-ro 0  1
# /grubfile was on /dev/sda2 during installation
UUID=3a965842-b6dd-4d52-8830-2d0fdb4284a2 /grubfile       ext3    defaults
0  2
/dev/sda5       none            swap    sw              0       0
/dev/scd0       /media/cdrom0   udf,iso9660 user,noauto,exec,utf8 0       0
/dev/fd0        /media/floppy0  auto    rw,user,noauto,exec,utf8 0       0
```

Строки, начинающиеся с # – это комментарии. Остальные строки содержат по шесть полей. Поскольку эти поля позиционные, то все они должны быть не пустыми.

file system

Это поле может содержать имя устройства, например, /dev/sda1, метку (LABEL=) или уникальный идентификатор (UUID=). Для корневой файловой системы Fedora 13 из нашего примера значением этого поля может быть /dev/sda6, LABEL="Fedora-13-x86_64" или UUID="082fb0d5-a5db-41d1-ae04-6e9af3ba15f7". Использование меток или UUID обеспечивает дополнительную гибкость в работе при добавлении или извлечении устройств.

mount point

Это поле содержит точку монтирования, о которой уже упоминалось в разделе. Для пространства подкачки это поле должно содержать значение 'none' или 'swap'. В более ранних системах значением этого поля обычно было 'none'.

type

Это поле определяет тип файловой системы. CD- и DVD-приводы часто поддерживают файловые системы ISO9660 или UDF, поэтому можно указать через запятую все возможные значения, как это сделано в листинге 9. Если вы хотите, чтобы команда `mount` автоматически определяла тип, то укажите значение `auto`, как в последней строке листинга 9 (опция для флоппи-дисковода).

option

Это поле содержит опции монтирования. Если вы хотите использовать опции по умолчанию, укажите значение `defaults`. Перечислим несколько полезных опций, о которых следует знать.

- `rw` и `ro` – определяют, должна ли файловая система быть смонтирована для чтения и записи, либо в режиме "только для чтения".
- `noauto` – запрещает автоматическое монтирование файловой системы во время загрузки или при выполнении команды `mount -a`. В нашем примере эта опция используется для сменных носителей.
- `user` – разрешает монтировать и демонтировать файловую систему обычному (не root) пользователю. Эта опция особенно полезна для использования со сменными носителями. В более ранних системах эта опция указывалась в файле /etc/fstab, а не в команде `mount`. В современных системах эта опция может быть указана в правилах udev, расположенных в файлах правил /lib/udev/rules.d или /etc/udev/rules.d. Опции для DVD-привода моего компьютера под управлением Fedora 13 указаны в правилах udev, поэтому в файле /etc/fstab запись для оптического привода отсутствует.
- `exec` и `noexec` – разрешают или запрещают выполнение файлов, расположенных в смонтированной файловой системе. По умолчанию для монтируемых пользователем файловых систем используется опция `noexec`, если только опция `exec` не указана **после** опции `user`.
- `noatime` – запрещает запись времени доступа к файлам, что позволяет повысить быстродействие.

dump

Это поле определяет, должна ли команда `dump` учитывать файловые системы ext2 или ext3 при создании резервных копий. Значение 0 указывает команде `dump` игнорировать эти файловые системы.

pass

Значение этого поля, отличное от нуля, определяет очередность проверки файловых систем во время загрузки

При монтировании файловых систем, перечисленных в файле /etc/fstab, можно указать либо имя устройства, либо точку монтирования. Нет необходимости указывать и то, и другое.

В некоторых системах, например SUSE 11.2, файл `fstab`, сгенерированный в процессе инсталляции системы, может содержать символические ссылки на устройства. Таким образом, вместо имени /dev/sda6 вы можете увидеть /dev/disk/by-id/ata-WDC_WD1001FALS-00J7B1_WD-WMATV3772868-part6.

Для получения дополнительной информации, включая не рассмотренные в этой статье опции, обратитесь к man-страницам `fstab`, `mount` и `udev`.

Демонтирование файловых систем

Обычно операционная система выполняет демонтирование всех смонтированных файловых систем автоматически при перезагрузке или выключении компьютера. При демонтировании файловой системы все ее кэшированные данные, находящиеся в оперативной памяти, сбрасываются на диск.

Файловые системы можно демонтировать и вручную. Более того, вы **должны** делать это при извлечении доступных для записи съемных устройств, таких как дискеты или USB-накопители.

Для демонтирования файловой системы выполните команду `umount`, указав в качестве аргумента *либо* имя устройства, *либо* точку монтирования. В листинге 10 показано, как демонтировать файловую систему, указав точку монтирования /dos, а затем снова смонтировать ее и вновь демонтировать, указав на этот раз имя устройства.

Листинг 10. Демонтирование файловых систем

```
[root@echidna ~]# umount /dos
[root@echidna ~]# mount /dev/sda9 /dos
[root@echidna ~]# umount /dev/sda9
```

После того, как файловая система будет демонтирована, все файлы директории, использовавшейся в качестве точки монтирования, снова окажутся видимыми.

Если вы попытаетесь демонтировать файловую систему, содержащую файлы, открытые каким-либо процессом, то вы получите сообщение об ошибке. Прежде чем начинать демонтирование файловой системы, следует проверить, что ее файлы не используются работающими процессами. Чтобы посмотреть, какие файлы открыты определенным процессом или какой процесс использует те или иные файлы, используйте команды `lsof` и `fuser`. Чтобы не выдавались предупреждения, относящиеся к Gnome Virtual File system (gvfs), может потребоваться использовать опцию `-w` команды `lsof`. О дополнительных опциях монтирования и команде `lsof` можно узнать из соответствующих man-страниц. Если вы проверяете все устройство целиком, то можно указать имя устройства или точку монтирования. Также можно проверить использование отдельного файла.

Для демонстрации работы этих команд я создал копию файла /etc/fstab в директории /dos и написал небольшой сценарий, который каждые 10 секунд считывает строки с устройства стандартного ввода и выводит их в устройство стандартного вывода. В листинге 11 показано сообщение об ошибке, выводимое командой `umount` при использовании файлов демонтируемой файловой системы, а также приведен пример использования команд `lsof` и `fuser` для проверки файлов в директории /dos, т. е. на устройстве /dev/sda9.

Листинг 11. Проверка на наличие открытых файлов

```
[root@echidna ~]# umount /dos
umount: /dos: device is busy.
        (In some cases useful info about processes that use
         the device is found by lsof(8) or fuser(1))
[root@echidna ~]# lsof -w /dos
COMMAND    PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
slowread. 2560  ian    0r   REG    8,9      899  123 /dos/fstab
sleep     2580  ian    0r   REG    8,9      899  123 /dos/fstab
[root@echidna ~]# lsof -w /dev/sda9
COMMAND    PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
slowread. 2560  ian    0r   REG    8,9      899  123 /dos/fstab
sleep     2580  ian    0r   REG    8,9      899  123 /dos/fstab
[root@echidna ~]# fuser -m /dos
/dos:                 2560  2586
[root@echidna ~]# fuser -m /dev/sda9
/dev/sda9:            2560  2588
```

На этом этапе вы можете либо подождать, пока все открытые файлы не освободятся, либо выполнить *"ленивое" демонтирование*, указав опцию `-l`. Такой способ демонтирования немедленно удаляет файловую систему из дерева каталогов, а также все ссылки на нее по мере того, как она освобождается.

Файловые системы на съемных носителях

В этой статье я упоминал о съемных носителях, таких как устройства с интерфейсом USB или Firewire (IEEE 1394). Зачастую неудобно каждый раз переключаться в режим суперпользователя для монтирования или демонтирования таких устройств. То же самое касается флоппи-дисководов, CD- и DVD-приводов, когда каждый раз для смены диска или дискеты нужно демонтировать устройство. Далее, рассматривая команду `fstab`, я упомянул об опции `user`, которая позволяет монтировать и демонтировать устройства обычным пользователям. В листинге 9 показан один из способов настройки записей в `fstab` для флоппи-дисководов, CD- и DVD-приводов.

Обратите внимание на то, что для оптических дисков указаны типы файловых систем `udf,iso9660`, а для флоппи-дисков тип файловой системы выбирается автоматически (опция `auto`). В случае оптических дисков процесс монтирования сначала проверяет наличие файловой системы udf (обычно используется на DVD-дисках), а затем iso9660 (обычно используется на CD-дисках). При использовании дискет тип файловой системы определяется в результате проверок. Вы можете создать или отредактировать файл /etc/filesystems и изменить порядок выполнения различных проверок файловых систем.

Примечание. Всегда следует выполнять демонтирование съемных устройств или носителей, прежде чем отключать или извлекать их. В противном случае вы можете потерять данные, которые еще не были записаны на устройство из кэша.

Если вы работаете с графическим рабочим столом, например, с Nautilus, то, как правило, эта среда содержит опции, позволяющие автоматически монтировать съемные устройства и носители. Например, если я вставлю DVD-диск с Knoppix в DVD-привод моего компьютера, то выдается запись о монтировании, подобную той, что показана в листинге 12. Запись 'uid=1000' означает, что диск может быть демонтирован пользователем с идентификатором 1000. Команда `id` показывает, что 1000 – это значение идентификатора uid пользователя ian, поэтому Ян может демонтировать этот диск.

Листинг 12. Монтирование DVD в режиме графического рабочего стола

```
[ian@echidna ~]$ mount | grep sr0
/dev/sr0 on /media/KNOPPIX type iso9660 (ro,nosuid,nodev,uhelper=udisks,
uid=1000,gid=1000,iocharset=utf8,mode=0400,dmode=0500)
[ian@echidna ~]$ id ian
uid=1000(ian) gid=1000(ian) groups=1000(ian)
```

Если такая возможность поддерживается дисководом (например, она поддерживается большинством приводов CD или DVD), то для извлечения съемного носителя можно использовать команду `eject`. Если перед этим устройство не было демонтировано, команда `eject` демонтирует его и извлечет носите.

Пространство подкачки

При обсуждении `fstab` вы могли обратить внимание на то, что пространство подкачки не имеет точки монтирования. В процессе монтирования обычно активируется пространство подкачки, определенное в файле /etc/fstab (если не используется опция `noauto`). Чтобы вручную управлять пространством подкачки в работающей системе (например, если добавляется новый раздел подкачки), используйте команды `swapon` и `swapoff`. Для получения подробной информации обратитесь к man-страницам.

Для просмотра пространств подкачки, доступных в текущий момент, выполните команду `cat /proc/swaps` или `swapon -s`, как показано в листинге 13.

Листинг 13. Просмотр пространства подкачки

```
[ian@echidna ~]$ swapon -s
Filename                                Type            Size    Used    Priority
/dev/sdb1                               partition       514044  0       -1
/dev/sdb5                               partition       4192928 0       -2
[ian@echidna ~]$ cat /proc/swaps
Filename                                Type            Size    Used    Priority
/dev/sdb1                               partition       514044  0       -1
/dev/sdb5                               partition       4192928 0       -2
```

Практическая работа № 9

Тема: Планирование заданий в ос Windows XP

В процессе эксплуатации системы часто требуется выполнять периодические регламентные задачи обслуживания в автоматическом режиме без вмешательства человека, например, проверку наличия обновлений, создание резервных копий критически важных файлов, выполнения сценариев администрирования и т.п. Для планирования и автоматического выполнения задач в среде Windows предусмотрено стандартное средство **Планировщик заданий**. Планировщик заданий обеспечивает выполнение заранее подготовленных задач в определенное время, или при возникновении определенных событий, однократно или с периодичностью, в контексте системных или пользовательских учетных записей. Задачи могут обеспечивать исполнение сценариев командной строки, скриптов WSH, командлетов PowerShell или приложений. В отличие от предыдущих версий Windows, Windows Vista, 7, 8 содержат обширную библиотеку предварительно настроенных заданий. Эти задания выполняют широкий круг операций, обеспечивающих обслуживание системы и поддержание ее в рабочем состоянии. Кроме того, Планировщик заданий в данных операционных системах, стал необходимым для нормальной работы компонентом, который нельзя отключить стандартными средствами.

Выполнение заданий по расписанию обеспечивается службой Планировщика заданий. Эта служба выполняется под локальной системной учетной записью, но позволяет настраивать задания для выполнения в контексте учетных записей определенных пользователей, указав при создании задания имя пользователя и пароль. Задачи можно создавать не только на локальном компьютере, но и на удаленном, по отношению к которому имеются соответствующие права. Процесс создания заданий планировщика упрощается при использовании соответствующего мастера.

В Windows существуют два основных типа планируемых заданий

Стандартные задания. Задания этого типа используются для автоматизации повседневных задач обслуживания компьютерной системы. Пользователь может видеть эти задания и модифицировать их в случае необходимости.

Скрытые задания. Задания этого типа используются для автоматизации выполнения специальных системных задач. Эти задания в стандартном режиме просмотра скрыты от пользователей и их изменение нежелательно.

Каждое задание планировщика имеет определенные **свойства**:

Триггеры - задают условия начала и завершения выполнения задания. Выполнение задания можно начинать по расписанию, а также по событиям - при входе пользователя в систему, при запуске компьютера, при подключении/отключении терминальной сессии или при записи в журналы системы событий с определенным кодом. **Действия** - определяют операцию, которую должно выполнять запущенное на выполнение задание, в качестве которого может использоваться приложение или сценарий.

Условия - определяют условия, при которых активированное задание запускается или останавливается. Например, условия можно использовать, чтобы выполнение задания было возможным только в случае наличия определенного сетевого подключения. С помощью условий можно запускать, останавливать и перезапускать задания. .

Параметры - определяют дополнительные параметры, влияющие на выполнение задания – перезапуск после сбоя, выполнение просроченного задания, поведение при наличии уже выполняющегося задания.

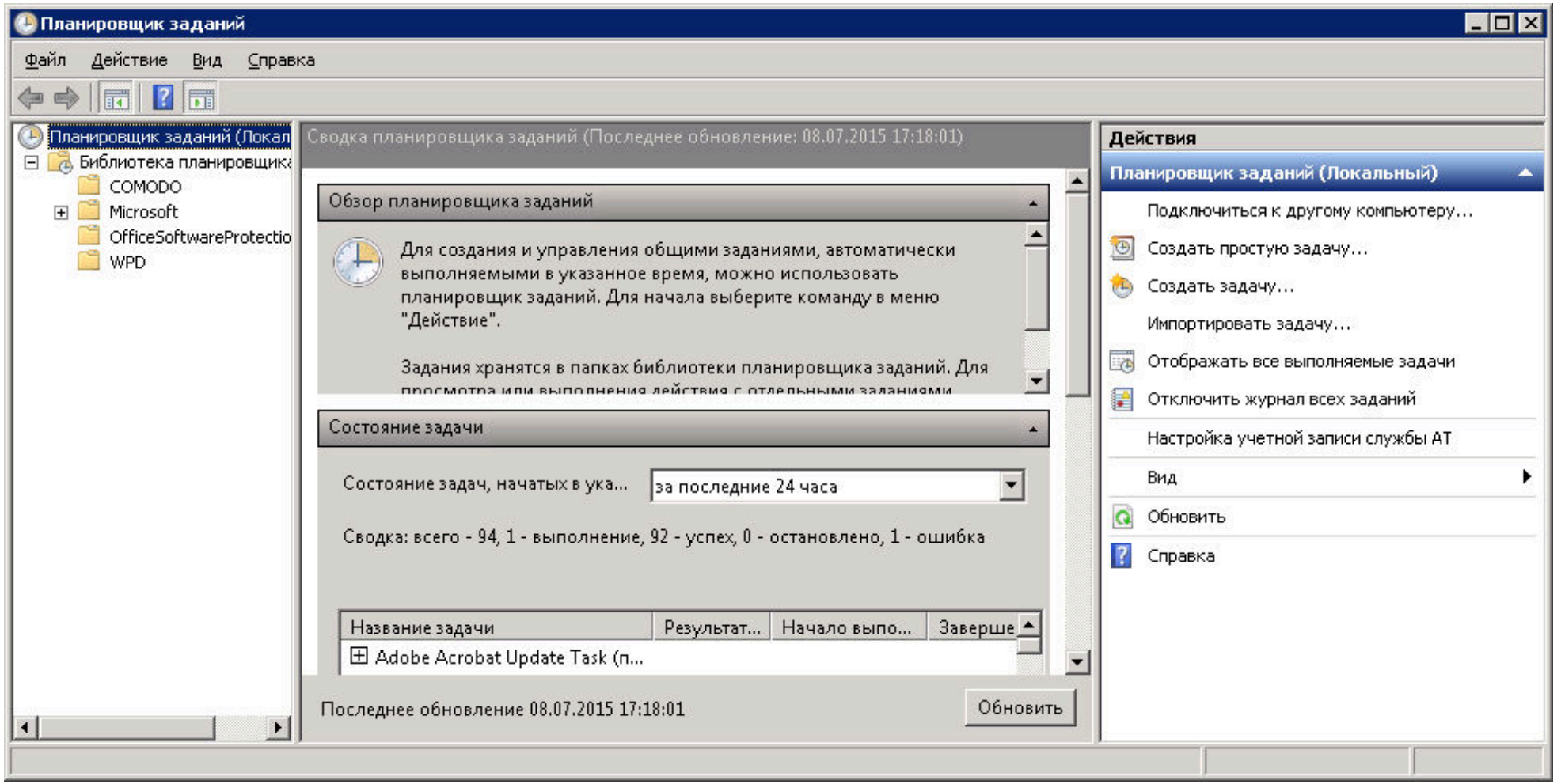
Настройки планировщика и параметры заданий хранятся в разделе реестра

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule

Результаты работы планировщика заданий могут фиксироваться в журнале, если это определено его настройками (В консоли управления планировщиком заданий - **Действие** - **Отключить (Включить) журнал всех заданий**).

Управление заданиями на локальных и удаленных системах.

Управление заданиями Windows выполняется с помощью специальной оснастки консоли управления Microsoft (mmc.exe), для запуска которой можно воспользоваться главным меню **Панель управления - Администрирование - Управление компьютером - Планировщик заданий**, или **Выполнить**(комбинация Win+R) - taskschd.msc.


Планировщик заданий
Файл Действие Вид Справка
Планировщик заданий (Локал
Библиотека планировщика
COMODO
Microsoft
OfficeSoftwareProtectio
WPD
Сводка планировщика заданий (Последнее обновление: 08.07.2015 17:18:01)
Обзор планировщика заданий
Для создания и управления общими заданиями, автоматически выполняемыми в указанное время, можно использовать планировщик заданий. Для начала выберите команду в меню "Действие".
Задания хранятся в папках библиотеки планировщика заданий. Для
Состояние задачи
Состояние задач, начатых в ука...
за последние 24 часа
Сводка: всего - 94, 1 - выполнение, 92 - успех, 0 - остановлено, 1 - ошибка
Название задачи
Результат...
Начало выпо...
Заверше
Adobe Acrobat Update Task (п...
Последнее обновление 08.07.2015 17:18:01
Обновить
Действия
Планировщик заданий (Локальный)
Подключиться к другому компьютеру...
Создать простую задачу...
Создать задачу...
Импортировать задачу...
Отображать все выполняемые задачи
Отключить журнал всех заданий
Настройка учетной записи службы AT
Вид
Обновить
Справка

В левой части основного окна отображаются списки заданий, упорядоченные в соответствии с назначением, в виде структуры папок. В средней части отображается информация о состоянии заданий и их свойствах. В правой части отображается меню действий, допустимых по отношению к заданиям. По умолчанию отображаются задания, относящиеся к локальному компьютеру. Для работы с заданиями удаленных компьютеров щелкните правой кнопкой мыши по элементу верхнего уровня "Планировщик заданий (Локальный)" и выберите пункт меню "Подключиться к другому компьютеру". В открывшемся диалоговом окне установите переключатель "Другой компьютер" и введите имя или IP-адрес требуемого компьютера. Для доступа к управлению заданиями на удаленном компьютере требуются права администратора системы, и, если текущий пользователь таковыми не обладает, то подключение нужно выполнить с использованием другой учетной записи, включив режим "Подключаться как другой пользователь". Естественно, удаленные подключения должны быть разрешены правилами брандмауэра и, кроме того, нужно учитывать, что структура данных заданий планировщика отличается в разных версиях Windows и подключение из среды, например Windows XP не выполнится при подключении к компьютеру с Windows 8.

Для работы с заданием щелкните по нему правой кнопкой мышки в основной панели и в контекстном меню выберите одну из следующих команд:

- **Удалить** (Delete) — полностью удалить задание;
- **Отключить** (Disable) — временно отключить задание;
- **Свойства** (Properties) — просмотреть и/или редактировать свойства задания.
- **Экспортировать** (Export) — экспортировать задание в файл, данные из которого можно импортировать на другом компьютере. Операционные системы Windows 8 и Windows Server 2012 имеют такую же архитектуру заданий, как и Windows 7 и Windows Server 2008 R2, но архитектура заданий более ранних версий Windows другая. При экспорте заданий можно указать операционную систему, с которой это задание допустимо использовать, с помощью параметра "Настроить для" (Configure for) на вкладке "Общие" окна свойств задания.
- **Выполнить** (Run) — запустить задачу на выполнение;
- **Завершить** (End) — если задача выполняется, остановить ее выполнение.

Задания, созданные пользователями и прикладными программами можно изменять или удалять без каких-либо особых проблем, но большинство заданий, созданных самой операционной системой, трогать не стоит, поскольку это может иметь неприятные последствия. Для отображения системных задач в меню "Вид" необходимо установить флажок "Отобразить скрытые задачи" (Show hidden tasks).

Просмотр списка заданий, исполняющихся на компьютере в настоящее время.

Для получения перечня заданий, выполняемых в данный момент времени, можно выбрать верхний уровень структуры "Планировщик заданий" и в контекстном меню, вызываемом правой кнопкой мышки, выбрать пункт "Отображать все выполняемые задачи". Можно также использовать пункт основного меню "Действие" - "Отображать все выполняемые

задачи"

Создание заданий для планировщика.

Библиотека планировщика заданий, отображаемая в левой части окна оснастки планировщика, имеет довольно непростую иерархическую структуру, поэтому, для освоения приемов работы с запланированными задачами, можно создать отдельную папку, с использованием контекстного меню, вызываемого правой кнопкой мышки и пункта **Создать папку**, ввести имя папки, и в дальнейшем, именно в ней создавать свои тестовые или рабочие задания.

В качестве задания планировщика будем использовать задачу, выполняющую командный файл, который определяет с какими привилегиями он выполняется, и выводит на экран сообщение об этом текущему пользователю. Использование такой задачи позволит наглядно продемонстрировать некоторые настройки свойств задания.

Для проверки уровня привилегий используется стандартная команда **whoami** в цепочке с командой **find**, для вывода сообщений - **msg**. В результате выполнения командного файла, на экран выводится сообщение "Задача выполняется с правами администратора" или " Задача выполняется с правами пользователя - имя ". Пусть данный командный файл имеет имя **who-admin.bat**. При желании, описание используемых команд и принцип определения привилегий, с которыми выполняется командный файл, можно найти в разделе Список команд CMD Windows

Содержимое файла с именем **who-admin.bat**:

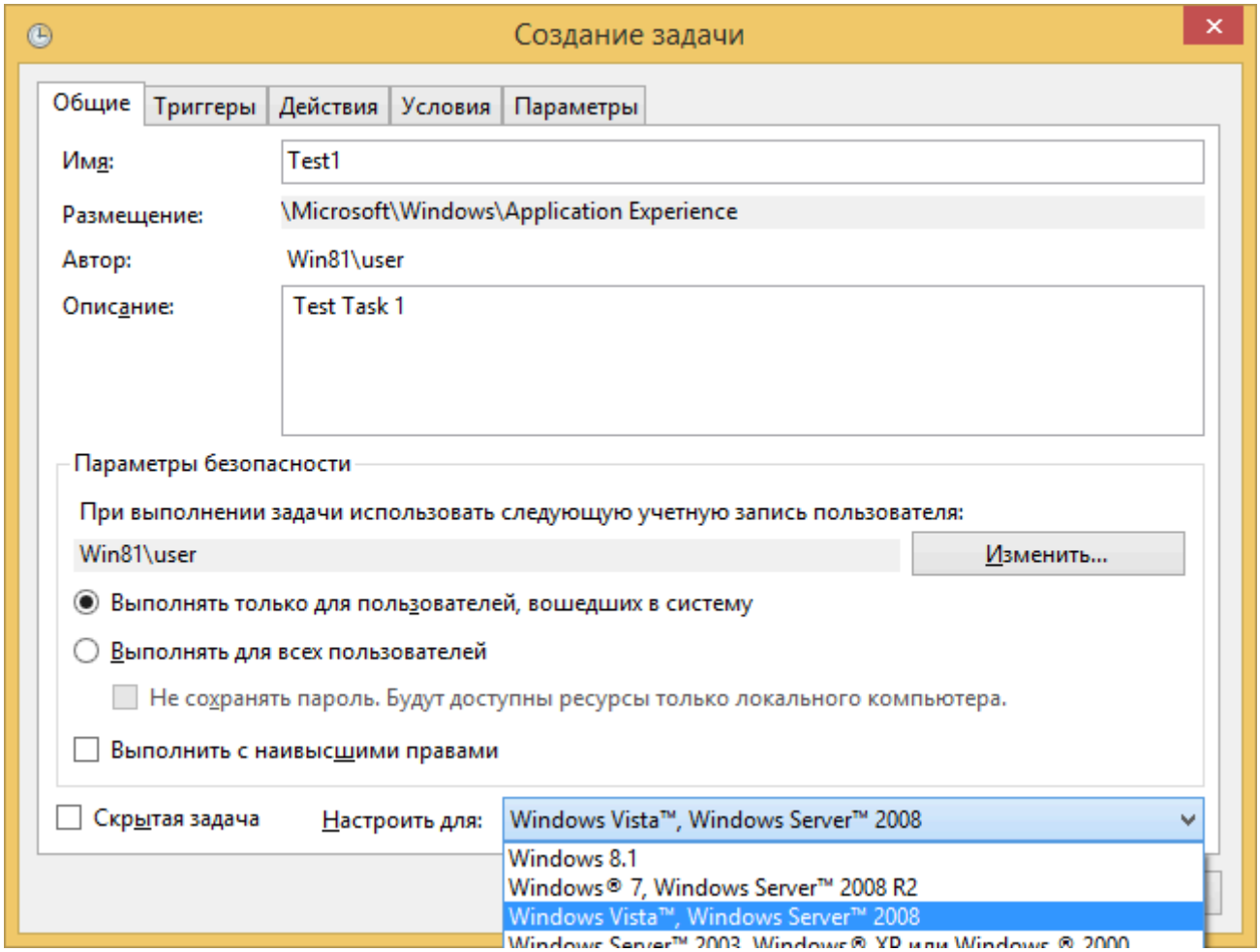
```
@echo OFF
WHOAMI /PRIV | find /i "SeRemoteShutdownPrivilege"
if %ERRORLEVEL% == 0 goto admin
msg * Задача выполняется с правами пользователя - %USERNAME%
exit
:admin
msg * Задача выполняется с правами администратора.
```

Файл можно создать в обычном текстовом редакторе, скопировав приведенный выше текст, выделенный зеленым цветом и присвоив ему имя who-admin.bat . Для правильного отображения кириллицы необходимо использовать DOS-кодировку (CP866, кодовую страницу 866) или изменить текст сообщений для вывода на английском языке. Если текстовый редактор не поддерживает кириллицу в DOS-кодировке, можно воспользоваться сторонними программами (например - **iconv**) или онлайн-сервисами наподобие Универсального декодера-конвертера кириллицы. Можете просто скачать архив who-admin.zip с командным файлом в правильной кодировке.

Для создания задач планировщика могут использоваться два мастера, вызываемые в режимах **Создать простую задачу** и **Создать задачу**. При создании простой задачи используется минимальный набор параметров, не предусматривающий наличие множественных условий выполнения и множественных действий.

Новую задачу можно создать следующим способом:

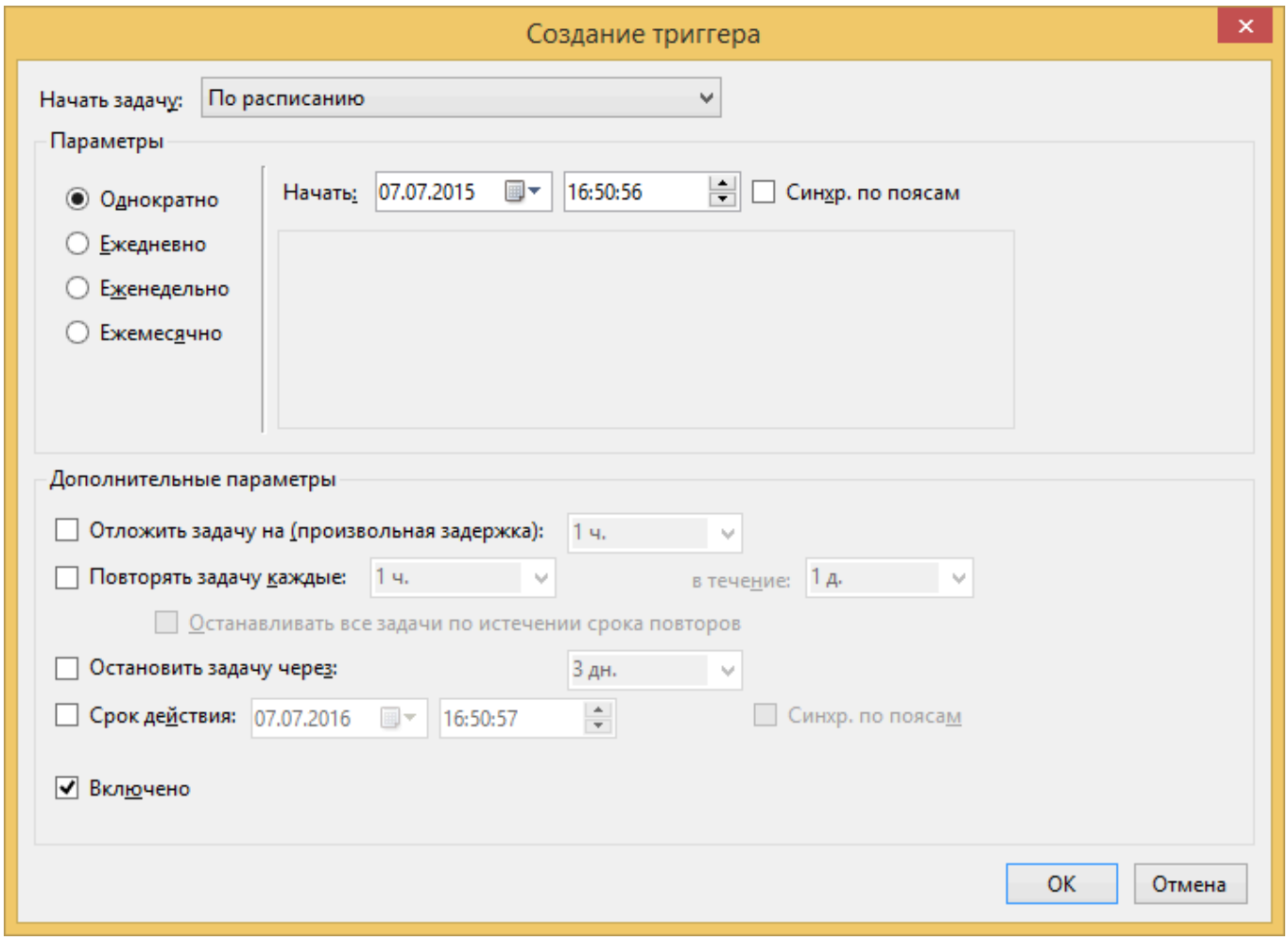
С использованием пункта меню **Действие - Создать задачу** или через контекстное меню, вызываемое правой кнопкой мышки на уровне "Планировщик заданий". После чего запускается мастер создания задачи:



На вкладке **Общие** окна мастера введите имя задания и его описание. Имя, для примера - **Test1**, а описание "Test Task 1". Описание может быть произвольным текстом, но желательно, чтобы оно отражало суть создаваемой задачи. Остальные настройки можно не выполнять, оставив значения по умолчанию. Эти настройки будем менять перед выполнением созданной задачи, чтобы понять их смысл и назначение.

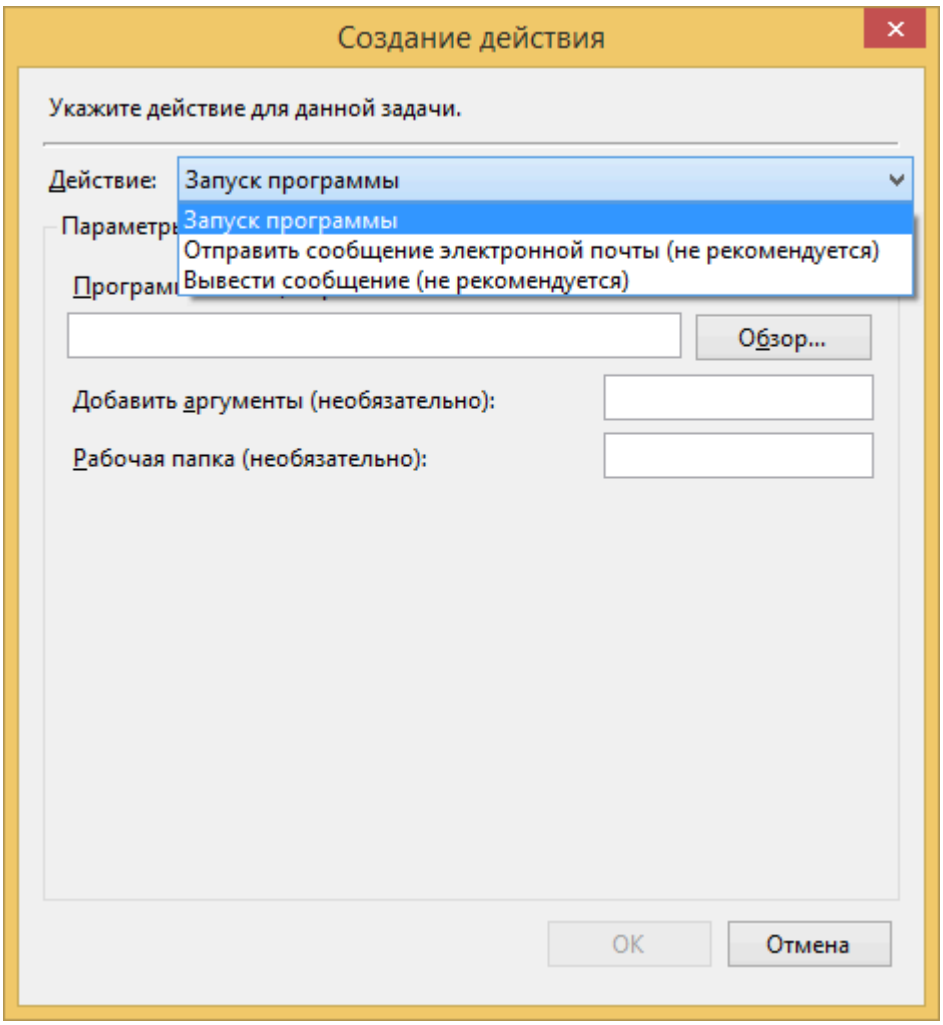
По умолчанию задания исполняются, только если пользователь выполнил вход в систему. Чтобы выполнять задачу независимо от того, работает пользователь в системе или нет, используется переключатель **Выполнять вне зависимости от регистрации пользователя** (Run whether user is logged on or not). Также можно задать выполнение задания с наивысшими полномочиями и/или настроить его для определенных версий Windows.

На вкладке **Триггеры** нажмите кнопку Создать, в открывшемся окне **Создание триггера** и задайте условие при возникновении которого будет выполняться создаваемая задача.

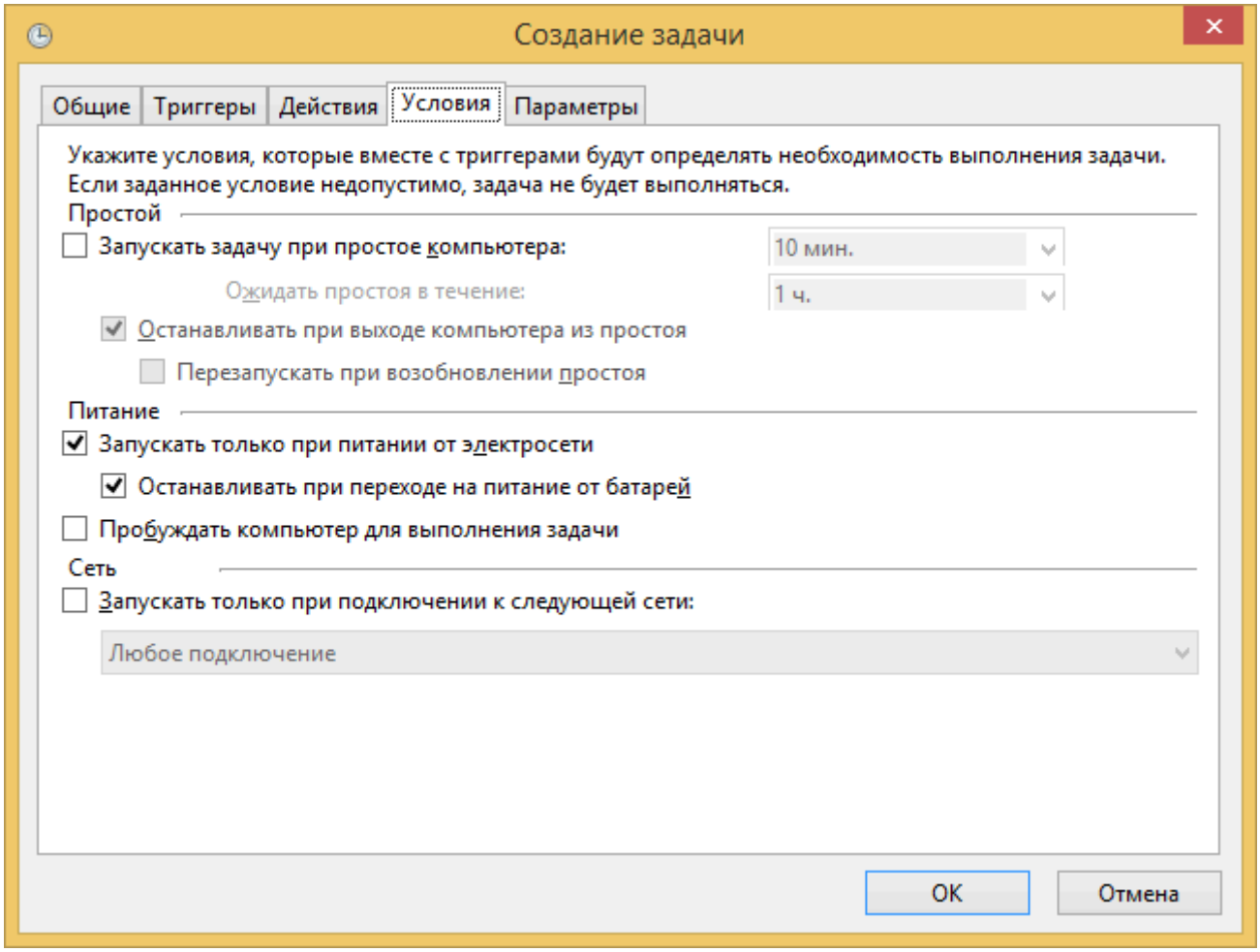


Для целей обучения, лучше определить триггер таким образом, чтобы задача не выполнялась автоматически, а только вручную, например, выбрав режим однократного запуска на уже прошедшую или будущую дату. Это позволит менять настройки задания и выполнять его в нужный момент времени вручную.

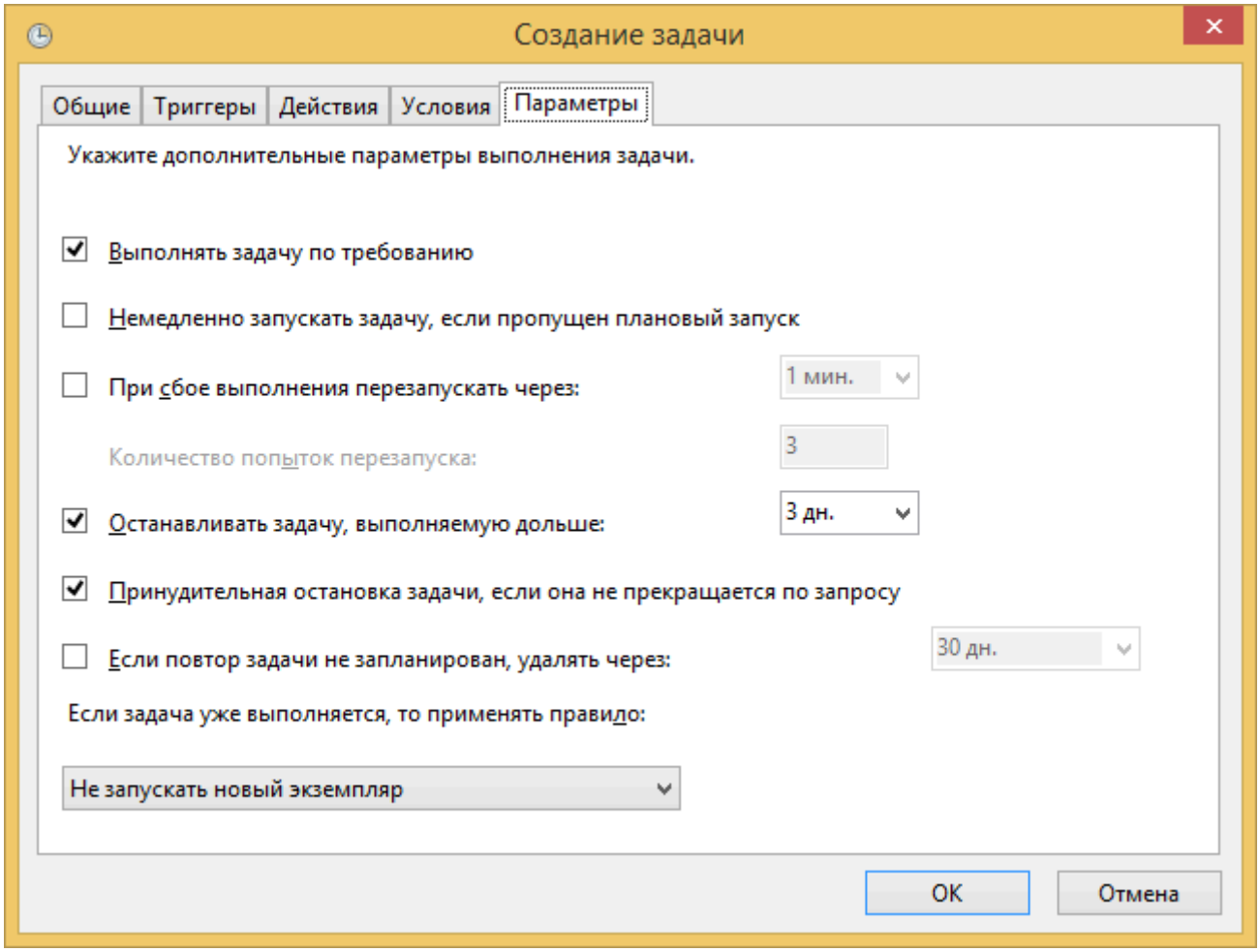
На вкладке **Действия** в качестве действия выбираем **Запуск программы** в качестве которой будет выступать созданный ранее командный файл **who-admin.bat**, который нужно выбрать с использованием кнопки **Обзор…**.



На вкладке **Условия** оставьте настройки по умолчанию. Изменения настроек можно выполнить для определения их влияния на поведение создаваемого задания позже.



На вкладке **Параметры** дополнительные параметры для задания также оставим для последующих экспериментов.



Выполнив все требуемые настройки, нажмите кнопку **OK** и задание будет создано. Практически любые настройки созданного задания можно изменять при наличии у пользователя соответствующих прав.

Диагностика состояния заданий планировщика.

В процессе настройки заданий можно столкнуться с несколькими типами проблем:

- Задание не запускается;

- Задание не завершается;

- Задание завершается с ошибкой;

Для определения статуса задания, выберите требуемое задание в Планировщике заданий и просмотрите сведения, такие как состояние, время прошлого запуска, результат последнего запуска и т. п. Состояния:

Отключено (Disabled) означает, что задание существует, но его выполнение запрещено настройками триггера.

Поставлено в очередь (Queued) означает, что задание ожидает выполнения в

запланированное время.

Готово означает, что задание готово к следующему выполнению.

Также, отображается время следующего запуска, время прошлого запуска, результат выполнения, автор задания и дата его создания.

Для длительного контроля за выполнением заданий планировщиком, можно включить режим ведения журнала, в котором будет сохраняться история выполнения заданий.

Состояние **Работает**, может в действительности означать, что задание зависло, ожидает ввод пользовательских данных, или, например, из-за ошибок программирования, ушло на бесконечный цикл. Узнать, действительно ли задание зависло, можно, проверив столбец **Время прошлого запуска**, в котором указано время запуска задания. Если задание выполняется более суток, то, за редким исключением, это не нормально. Чтобы остановить задание, щелкните на нем правой кнопкой мышки и в контекстном меню выберите команду **Завершить**. Для заданий, время выполнения которых известно, можно установить предельное время выполнения в **Свойства-Параметры-Остановить задачу, выполняемую дольше:**. В случае превышения отведенного на выполнение времени, задание будет принудительно завершено планировщиком.

Имея тестовое задание, попробуйте установить скрытый режим его выполнения ("Скрытая задача"), выполнить с правами администратора ("Выполнить с наивысшими правами"), перенести его на другой компьютер, используя режим экспорта и импорта. Полезно будет поэкспериментировать над условиями и параметрами задания, например, добавив в командный файл команду **pause** при выполнении в скрытом режиме, и тем самым вызвав имитацию зависания задания. Желательно также отработать вариант выполнения просроченного задания, когда запланированное время соответствует выключенному состоянию компьютера, и включен параметр "Немедленно запускать задачу, если пропущен плановый запуск".

Перечень некоторых стандартных заданий Windows 7,8:

Application Experience\AitAgent - Сбор и передача данных дистанционного отслеживания приложений (при явном согласии участвовать в программе улучшения качества программного обеспечения). Даже если время выполнения установлено на 2-3 часа ночи, в настройках присутствует признак выполнения просроченного задания и передача данных в Microsoft выполняется.

Application Experience\ProgramDataUpdater - Сбор телеметрических данных программы при участии в программе улучшения качества ПО

Autochk\Proxy - собирает и загружает данные SQM (при участии в программе улучшения качества программного обеспечения).

Customer Experience Improvement Program\BthSQM - Задача программы улучшения качества Bluetooth собирает статистику по Bluetooth, а также сведения о вашем компьютере, и отправляет их в корпорацию Майкрософт. Полученные сведения используются для повышения надежности, стабильности и общей функциональности Bluetooth в Windows. При отсутствии согласия пользователя на участие в программе улучшения программного обеспечения Windows эта задача не выполняет никаких действий. Задача присутствует в Windows 8.

Customer Experience Improvement Program\Consolidator При выполнении задачи программы улучшения качества ПО, выполняющейся в режиме ядра (Kernel CEIP), осуществляется сбор дополнительных данных о системе, которые затем передаются в корпорацию Майкрософт. Если пользователь не дал своего согласия на участие в данной программе, то эта задача не выполняет никаких действий.

Customer Experience Improvement Program\UsbCeip - При выполнении задачи программы улучшения качества ПО шины USB (USB CEIP) осуществляется сбор статистических данных об использовании универсальной последовательной шины USB и сведений о компьютере, которые направляются инженерной группе Майкрософт по вопросам подключения устройств в Windows. Полученные сведения используются для повышения надежности, стабильности и общей производительности шины USB в Windows. При отсутствии согласия пользователя на участие в программе улучшения программного обеспечения Windows задача не выполняет никаких действий.

RAC\RacTask - Задача средства анализа стабильности системы, предназначенная для обработки данных о надежности компьютера.

Одним из малоизвестных заданий является **Registry\RegIdleBackup** - Задание архивации реестра, выполняющееся раз в 10 дней и обеспечивающее сохранение файлов реестра в папке **C:\Windows\System32\config\RegBack**. В некоторых случаях, когда нет точек восстановления или они некондиционны, есть шанс восстановить работоспособное состояние системы, загрузившись в другой ОС и скопировав файл(ы) реестра в каталог **\Windows\System32\config**. Можно также использовать ручное выполнение данного задания для получения копии файлов реестра, нужных в данный момент времени.

В Windows 8 присутствует папка **\Microsoft\Windows\WS** с несколькими задачами использующими обмен данными с магазином Windows.

Это далеко не полный перечень системных заданий, присутствующих в библиотеке планировщика Windows 7 и Windows 8. Полезность многих из них сомнительна, и даже более того, отрицательная, поскольку может приводить к снижению времени работы ноутбуков от батарей, расходу недешевого мобильного трафика, дополнительных затрат электроэнергии, снижению реальной производительности и т.п. В заключение напомню, что отключение системных заданий требует осторожности и понимания того, что вы делаете, в противном случае, можно получить неработоспособную или частично работоспособную систему.

В библиотеке планировщика, как правило, присутствуют также задания созданные антивирусами для обновлений программ и баз данных, задания, созданные прочими, не всегда нужными, программами, проверяющими наличие новых версий (продукты Adobe, Google, бесплатное ПО и пр.) и даже обновляющихся без ведома пользователя.

Управление заданиями планировщика из командной строки CMD Windows.

Для управления заданиями из командной строки Windows используется утилита **schtasks.exe**.Примеры использования:

schtasks /Query - вывести список всех заданий;

SCHTASKS /Run /I /TN "System Backup" - выполнить на локальном компьютере задание "System Backup" немедленно, игнорируя любые ограничения.

SCHTASKS /end /TN "System Backup" - снять задачу "System Backup"

schtasks /change /tn "\Microsoft\Windows\Application Experience\ProgramDataUpdater" /disable - отключить задание \Microsoft\Windows\Application Experience\ProgramDataUpdater

В заключение, приведу текст командного файла, позволяющего отключить или включить системные задания планировщика, полезность которых сомнительна. В комментариях, которые предшествуют команде **schtasks** представлено краткое описание задания, взятое из общих свойств задания, отображаемое в консоли управления. Для выполнения данного командного файла требуется запуск от имени администратора. Действие, выполняемое по отношению к заданиям планировщика определяется значением переменной **task_action**. Команда **set task_action=disable** приведет к отключению заданий, **set task_action=enable** - к включению. При выполнении команды **schtasks**, ее вывод сохраняется в файле журнала, имя которого соответствует имени командного файла, с расширением **txt**.

```
@echo off
Rem action - установка значение disable или enable
set task_action=disable
REM Проверка наличия прав администратора
@WHOAMI /PRIV | find /i "SeRemoteShutdownPrivilege" > nul
if %ERRORLEVEL% == 0 goto admin
echo Execution aborted. Run this batch file AS Administrator  !!!
pause
exit
:admin
@echo try %task_action% scheduled tasks ...

rem Сбор и передача данных дистанционного отслеживания приложений (при явном
согласии участвовать в
rem программе улучшения качества программного обеспечения).
schtasks /change /tn "\Microsoft\Windows\Application Experience\AitAgent"
/%task_action% >> %~dn0.log

rem Сбор телеметрических данных программы при участии в программе улучшения
качества ПО
schtasks /change /tn "\Microsoft\Windows\Application
Experience\ProgramDataUpdater" /%task_action% >> %~dn0.log

rem Эта задача собирает и загружает данные SQM при участии в программе
улучшения качества
rem программного обеспечения.
schtasks /change /tn "\Microsoft\Windows\Autochk\Proxy" /%task_action% >>
%~dn0.log

rem Если пользователь изъявил желание участвовать в программе по улучшению
качества программного
rem обеспечения Windows, эта задача будет собирать и отправлять сведения о
работе программного
rem обеспечения в Майкрософт.
schtasks /change /tn "\Microsoft\Windows\Customer Experience Improvement
Program\Consolidator" /%task_action% >> %~dn0.log

rem При выполнении задачи программы улучшения качества ПО, выполняющейся в
режиме ядра (Kernel
rem CEIP), осуществляется сбор дополнительных данных о системе, которые затем
передаются в
```

```
rem корпорацию Майкрософт. Если пользователь не дал своего согласия на
участие в данной программе,
rem то эта задача не выполняет никаких действий.
schtasks /change /tn "\Microsoft\Windows\Customer Experience Improvement
Program\KernelCeipTask" /%task_action% >> %~dn0.log

rem При выполнении задачи программы улучшения качества ПО шины USB (USB CEIP)
осуществляется сбор
rem статистических данных об использовании универсальной последовательной
шины USB и сведений о
rem компьютере, которые направляются инженерной группе Майкрософт по вопросам
подключения устройств
rem в Windows. Полученные сведения используются для повышения надежности,
стабильности и общей
rem производительности шины USB в Windows. При отсутствии согласия
пользователя на участие в
rem программе улучшения программного обеспечения Windows задача не выполняет
никаких действий.
schtasks /change /tn "\Microsoft\Windows\Customer Experience Improvement
Program\UsbCeip" /%task_action% >> %~dn0.log

rem Эта задача выполняет дефрагментацию жестких дисков компьютера.
schtasks /change /tn "\Microsoft\Windows\Defrag\ScheduledDefrag"
/%task_action% >> %~dn0.log

rem Измеряет быстродействие и возможности системы
schtasks /change /tn "\Microsoft\Windows\Maintenance\WinSAT" /%task_action%
>> %~dn0.log

REM Запуск приложений, настроенных для Windows HotStart
schtasks /change /tn "\Microsoft\Windows\MobilePC\HotStart" /%task_action% >>
%~dn0.log

rem Это задание предназначено для анализа состояния системы и поиска условий,
которые могут привести
rem к повышенному энергопотреблению.
schtasks /change /tn "\Microsoft\Windows\Power Efficiency
Diagnostics\AnalyzeSystem" /%task_action% >> %~dn0.log

rem Задача средства анализа стабильности системы, предназначенная для
обработки данных о надежности
rem компьютера.
schtasks /change /tn "\Microsoft\Windows\RAC\RacTask" /%task_action% >>
%~dn0.log

REM (Windows Live Social Object Extractor Engine) After installing Windows
Live Essentials 2011,
REM it creates a task called "Extractor Definitions Update Task". The purpose
of this task is to
REM update the definition file for the Windows Live Social Object Extractor
Engine. The task
REM can be found under, Task Scheduler Library\Microsoft\Windows Live\SOXE. -
REM Read more at http://www.shouldiremoveit.com/windows-live-soxe-5304-
program.aspx
schtasks /change /tn "\Microsoft\Windows Live\SOXE\Extractor Definitions
Update Task" /%task_action% >> %~dn0.log

rem Эта запланированная задача предназначена для уведомления пользователя о
том, что архивация
rem Windows не настроена.
schtasks /change /tn "\Microsoft\Windows\WindowsBackup\ConfigNotification"
/%task_action% >> %~dn0.log
```

Учтите, что отключение системных заданий - небезопасная операция, и ее последствия целиком на вашей совести. Так, например, командный файл отключает задание на дефрагментацию файлов, которое не имеет смысла при использовании SSD-диска, но может быть полезным при использовании механических HDD. Кроме того, если вы посмотрите действия, выполняемые заданием дефрагментации, то увидите, что выполняется запуск программы **%windir%\system32\defrag.exe –c**, где **-c** - параметр, задающий выполнение дефрагментации на всех дисках. Может быть, имеет смысл создать несколько заданий с разными триггерами для разных дисков, исключив SSD. Можно также изменить парвметры существующей задачи, исключив дефрагментацию SSD - дисков:

defrag /E C: - выполнить дефрагментацию на всех локальных дисках, кроме диска **C:**

Практическая работа №10

Тема: Процессы в ос linux

Цель работы – изучение вопросов порождения и взаимодействия процессов в ОС LINUX.

Теоретическая часть

В ОС Linuxдля создания процессов используется системный вызов***fork():***

#include <sys/types.h>

#include <unistd.h>

pid_t fork (void);

В результате успешного вызова ***fork()***ядро создаёт новый процесс, который является почти точной копией вызывающего процесса. Другими словами, новый процесс выполняет копию той же программы, что и создавший его процесс, при этом все его объекты данных имеют те же самые значения, что и в вызывающем процессе. Созданный процесс называется***дочерним процессом***, а процесс, осуществивший вызов***fork()***, называется***родительским***. После вызова родительский процесс и его вновь созданный потомок выполняются одновременно, при этом оба процесса продолжают выполнение с оператора, который следует сразу же за вызовом***fork()***. Процессы выполняются в разных адресных пространствах, поэтому прямой доступ к переменным одного процесса из другого процесса невозможен.

Следующая короткая программа более наглядно показывает работу вызова ***fork()*** и использование процесса:

#include <stdio.h>

#include <unistd.h>

Int main ()

{

pid_t pid; /* идентификатор процесса */

printf (“Пока всего один процесс\n”);

pid = fork (); /* Создание нового процесса */

printf (“Уже два процесса\n”);

If (pid = = 0){

printf (“Это Дочерний процесс его pid=%d\n”, getpid());

printf (“А pid его Родительского процесса=%d\n”, getppid());

}

else if (pid > 0)

printf (“Это Родительский процесс pid=%d\n”, getpid());

else

printf (“Ошибка вызова fork, потомок не создан\n”);

}

Для корректного завершения дочернего процесса в родительском процессе необходимо использовать функцию ***wait()*** или ***waitpid()***:

pid_t wait(int *status); pid_t waitpid(pid_t pid, int *status, int options);

Функция ***wait***приостанавливает выполнение родительского процесса до тех пор, пока дочерний процесс не прекратит выполнение или до появления сигнала, который либо завершает текущий процесс, либо требует вызвать функцию-обработчик. Если дочерний процесс к моменту вызова функции уже завершился (так называемый***«зомби»***), то функция немедленно возвращается. Системные ресурсы, связанные с дочерним процессом, освобождаются.

Функция ***waitpid ()***приостанавливает выполнение родительского процесса до тех пор, пока дочерний процесс, указанный в параметре***pid***, не завершит выполнение, или пока не появится сигнал, который либо завершает родительский процесс, либо требует вызвать функцию-обработчик. Если указанный дочерний процесс к моменту вызова функции уже завершился (так называемый***«зомби»***), то функция немедленно возвращается. Системные ресурсы, связанные с дочерним процессом, освобождаются. Параметр***pid***может принимать несколько значений:

pid< -1 означает, что нужно ждать любого дочернего процесса, чей идентификатор группы процессов равен абсолютному значению***pid***.

pid= -1 означает ожидать любого дочернего процесса; функция**wait**ведет себя точно так же.

pid = 0означает ожидать любого дочернего процесса, чей идентификатор группы процессов равен таковому у текущего процесса.

pid> 0означает ожидать дочернего процесса, чем идентификатор равен***pid***.

Значение ***options***создается путем битовой операции***ИЛИ***над следующими константами:

WNOHANG- означает вернуть управление немедленно, если ни один дочерний процесс не завершил выполнение.

WUNTRACED - означает возвращать управление также для остановленных дочерних процессов, о чьем статусе еще не было сообщено.

Каждый дочерний процесс при завершении работы посылает своему процессу-родителю специальный сигнал ***SIGCHLD***, на который у всех процессов по умолчанию установлена реакция "игнорировать сигнал". Наличие такого сигнала совместно с системным вызовом***waitpid()***позволяет организовать асинхронный сбор информации о статусе завершившихся порожденных процессов процессом-родителем.

Для перегрузки исполняемой программы можно использовать функции семейства ***exec***. Основное отличие между разными функциями в семействе состоит в способе передачи параметров.

Int execl(char *pathname, char *arg0, arg1, ..., argn, null);

int execle(char *pathname, char *arg0, arg1, ..., argn, NULL, char **envp);

int execlp(char *pathname, char *arg0, arg1, ..., argn, NULL);

int execlpe(char *pathname, char *arg0, arg1, ..., argn, NULL, char **envp);

int execv(char *pathname, char *argv[]);

int execve(char *pathname, char *argv[],char **envp);

int execvp(char *pathname, char *argv[]);

int execvpe(char *pathname, char *argv[],char **envp);

Основное отличие между разными функциями в семействе состоит в способе передачи параметров. Как видно из рис. 1, все эти функции выполняют один системный вызов ***execve***.

execl

execle

execve

execvp

execv

execlp

Рис. 1. Дерево семейства вызовов ***exec***

Порядок выполнения работы

3. Изучить теоретическую часть лабораторной работы.
4. Написать программу, создающую два дочерних процесса с использованием двух вызовов ***fork()***. Родительский и два дочерних процесса должны выводить на экран свой***pid***и***pid***родительского процесса и текущее время в формате:***часы: минуты: секунды: миллисекунды***. Используя вызов***system (),***выполнить команду***ps -x***в родительском процессе. Найти свои процессы в списке запущенных процессов.

Варианты индивидуальных заданий

8. Написать программу нахождения массива ***N***последовательных значений функции***y[i]=sin(2*PI*i/N) (i=0,1,2…N-1)***с использованием ряда Тейлора. Пользователь задаёт значения***N*** и количество***n***членов ряда Тейлора. Для расчета каждого членаряда Тейлоразапускается отдельный процесс. Каждый процесс выводит на экран и в файл свой***pid***и рассчитанное значениеряда. Головной процесс считывает из файла значения всех рассчитанных членов ряда Тейлора, суммирует ихи полученное значение ***y[i]*** записывает в файл. Проверить работу программы для***N=256 n=5; N=1024 n=10.***
9. Написать программу синхронизации двух каталогов, например, ***Dir1***и***Dir2***. Пользователь задаёт имена***Dir1***и***Dir2***. В результате работы программы файлы, имеющиеся в***Dir1***, но отсутствующие в***Dir2***, должны скопироваться в***Dir2***вместе с правами доступа. Процедуры копирования должны запускаться в отдельном процессе для каждого копируемого файла. Каждый процесс выводит на экран свой***pid,***полный путь к копируемому файлу и число скопированных байт. Число одновременно работающих процессов не должно превышать***N***(вводится пользователем). Скопировать несколько файлов из каталога***/etc*** в свой домашний каталог. Проверить работу программы для каталога***/etc и***домашнего каталога.
10. Написать программу поиска одинаковых по содержимому файлов в двух каталогов, например, ***Dir1***и***Dir2***. Пользователь задаёт имена***Dir1***и***Dir2***. В результате работы программы файлы, имеющиеся в***Dir1***, сравниваются с файлами в***Dir2*** по их содержимому. Процедуры сравнения должны запускаться в отдельном процессе для каждой пары сравниваемых файлов. Каждый процесс выводит на экран свой***pid***, имя файла, общее число просмотренных байт и результаты сравнения. Число одновременно работающих процессов не должно превышать***N***(вводится пользователем). Скопировать несколько файлов из каталога***/etc*** в свой домашний каталог. Проверить работу программы для каталога***/etc и***домашнего каталога.
11. Написать программу поиска заданной пользователем комбинации из ***m***байт (***m <255***) во всех файлах текущего каталога. Пользователь задаёт имя каталога. Главный процесс открывает каталог и запускает для каждого файла каталога

отдельный процесс поиска заданной комбинации из***m***байт. Каждый процесс выводит на экран свой***pid,*** полный путь к файлу, общее число просмотренных байт и результаты (сколько раз найдена комбинация) поиска. Число одновременно работающих процессов не должно превышать***N***(вводится пользователем). Проверить работу программы для каталога***/etc u***строки ***ifconfig***.

12. Разработать программу «***интерпретатор команд***», которая воспринимает команды, вводимые с клавиатуры, (например,***ls -l /bin/bash***) и осуществляет их корректное выполнение. Для этого каждая вводимая команда должна выполняться в отдельном процессе с использованием вызова***exec()***. Предусмотреть возможность перенаправления (ввода\вывода ’>> ’, ’> ’, ’ <’).
13. Написать программу подсчета количества слов в файлах заданного каталога его подкаталогов. Пользователь задаёт имя каталога. Главный процесс открывает каталоги и запускает для каждого файла каталога отдельный процесс подсчета количества слов. Каждый процесс выводит на экран свой ***pid,*** полный путь к файлу, общее число просмотренных байт и количество слов. Число одновременно работающих процессов не должно превышать***N***(вводится пользователем). Проверить работу программы для каталога***/etc u***строки ***ifconfig***.
14. Написать программу подсчета частоты встречающихся символов в файлах заданного каталога его подкаталогов. Пользователь задаёт имя каталога. Главный процесс открывает каталоги и запускает для каждого файла каталога и отдельный процесс подсчета количества слов. Каждый процесс выводит на экран свой ***pid,*** полный путь к файлу, общее число просмотренных байт и количество слов. Число одновременно работающих процессов не должно превышать***N***(вводится пользователем). Проверить работу программы для каталога***/etc u***строки ***ifconfig***.