

Департамент внутренней и кадровой политики Белгородской области
Областное государственное автономное профессиональное образовательное
учреждение
«Белгородский индустриальный колледж»

Рассмотрено
ЦК «Информатики и ПОВТ»
Протокол заседания № 1
от «30» августа 2019 г.
Председатель цикловой
комиссии
_____ Третьяк И.Ю.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ПРАКТИЧЕСКИХ РАБОТ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ**

ОП.01 Операционные системы и среды

по специальности

09.02.06 Сетевое и системное администрирование

Разработчик: Ченская И.Б.,
преподаватель
ОГАПОУ «Белгородский
индустриальный колледж

Белгород 2019

СОДЕРЖАНИЕ

1. Пояснительная записка.....	3
2. Тематическое планирование лабораторных работ.....	6
3. Содержание лабораторных работ.....	7
4. Литература	66

1. Пояснительная записка

При изучении дисциплины “Операционные системы и среды” необходимо закреплять полученные знания практическими навыками. Современные информационные технологии способны заменить целые лаборатории, мастерские и другие оборудованные помещения. Например, студентам помогут виртуальные машины (Oracle VM VirtualBox, Microsoft Virtual PC), программно имитирующие работу отдельного компьютера, защищающие реальное оборудование компьютера, позволяющие выполнять функции администратора, а также различного рода офисное программное обеспечение (Microsoft Office, Apache OpenOffice), графические редакторы (PaintNET, Inkscape).

В данном пособии предложены работы, направленные на закрепление дидактического материала и приобретение навыков в сфере информационных технологий. Выполняя задания и решая задачи, обучающиеся могут проверять знания и правильность своих рассуждений по рассмотренным темам.

Лабораторные работы предназначены для усвоения материала теоретических занятий, изучения работы в средах различных операционных систем, принципов управления ресурсами, решения основных задач администрирования.

Выполнение лабораторной работы включает этапы:

1. Сбор данных и выполнение списка действий согласно инструкции.
2. Оформление отчета.
3. Защита лабораторной работы.

Сбор данных.

Инструкция по выполнению работы включает следующие разделы: № лабораторной работы, название, цель, оборудование, порядок выполнения.

Сбор данных выполняется в следующем порядке:

- изучается инструкция по выполнению работы;
- уясняется цель работы и последовательность действий;

- уточняются у преподавателя непонятные моменты;
- подготавливаются необходимые таблицы;
- выполняются действия согласно пунктам раздела «Ход работы».

Основные действия и выводы конспектируются.

Данные конспектируются и затем заносятся в отчет в соответствии с п.2.

Оформление отчета.

Отчет оформляется индивидуально каждым студентом на листах формата А4.

Отчет по каждой работе должен включать разделы:

1. № лабораторной работы (см. инструкцию по выполнению работ);
2. Название работы (см. инструкцию по выполнению работ);
3. Цель работы (см. инструкцию по выполнению работ);
4. Ход работы (упорядоченное изложение хода выполнения работы, выводы и данные по пунктам, заполненные таблицы).

Защита лабораторной работы.

Для защиты лабораторной работы студент должен:

- изучить практический и теоретический материал согласно вопросам к защите;
- представить результат выполнения работы;
- предъявить полный отчет о проделанной работе;
- ответить на вопросы к защите и дополнительные вопросы по данной теме.

Отчет защищенной лабораторной работы подписывается преподавателем с указанием даты защиты работы.

Выполненные в полном объеме лабораторные работы являются допуском к зачету. Студенты, не защитившие всех лабораторных работ, к зачету не допускаются!

Требования к оформлению отчетов по лабораторным работам

Общие требования

Отчет предоставляется в печатном виде на стандартных листах белой бумаги формата А4 (29.7х21.0 см, предназначенной для ксерокопирования). Содержит стандартную таблицу для внесения необходимых данных: код группы (подгруппы или персональный шифр для заочников), фамилию автора и инициалы, номер лабораторной работы, номер варианта индивидуального задания, если таковое имеется, дату создания и подпись автора.

Требования к содержанию отчета

Отчет должен содержать название лабораторной работы, цель ее выполнения, подробное описание всех действий и перечень вводимых команд в ходе ее выполнения.

Пример.

Задание: перейдите в папку Мои документы. В отчете должен быть описан способ, который Вы использовали для этого:

ввел команду и написал путь к папке (cd "c:\Documents and settings\user216\Мои документы")

В конце отчета нужно подвести итоги о проделанной Вами работе и написать вывод с указанием того, достигнута ли поставленная цель.

Основные показатели оценки результата

Оценивается правильность выполнения лабораторной работы, качество и скорость её выполнения, верные ответы на контрольные вопросы, а также выполнение дополнительных заданий преподавателя

2. Тематическое планирование лабораторных работ

Название лабораторной работы	Количество часов
Лабораторная работа №1. Изучение структуры операционной системы Windows	2
Лабораторная работа №2. Изучение структуры операционной системы Linux	2
Лабораторная работа №3. Изучение работы с командами в операционной системе Windows	2
Лабораторная работа №4. Изучение работы с командами в операционной системе Linux	2
Лабораторная работа №5. Работа с файлами и каталогами в операционных системах MS DOS и Windows	2
Лабораторная работа №6. Работа с файлами и каталогами в операционной системе LINUX	2
Лабораторная работа №7. Создание файлов пакетной обработки данных	2
Лабораторная работа №8. Файлы пакетной обработки данных. Операторы пакетных файлов	2
Лабораторная работа №9. Управление процессами в операционной системе Windows	2
Лабораторная работа №10. Управление процессами в операционной системе Linux	2
Лабораторная работа №11. Работа с текстовыми файлами в операционных системах MS DOS и Windows	2
Лабораторная работа №12. Работа с текстовыми файлами в операционной системе LINUX	2
Лабораторная работа №13. Работа с операционными оболочками (Total Commander)	2
Лабораторная работа №14. Работа с архиваторами в операционных системах MS DOS и Windows	2
Лабораторная работа №15. Работа с архиваторами в операционной системе Linux	2
Лабораторная работа №16. Изучение эмуляторов операционных систем MS DOS, Windows, LINUX	2
Лабораторная работа №17. Сетевое администрирование в виртуальной машине	2
Лабораторная работа №18. Работа с системными командами. Реестр операционной системы	2
Лабораторная работа №19. Основы работы с BIOS Setup Utility	2
Лабораторная работа №20. Установка и анализ функциональности различных операционных систем	2
Всего:	40

3. Содержание лабораторных работ

Лабораторная работа № 1

Изучение структуры операционной системы Windows.

Цель работы: обобщить сведения о структуре интерфейса операционной системы Windows; отработать навыки работы с графической системой.

Изучение теоретического материала.

Выполнение последовательно настроек

- стартового меню;
- панели задач;
- меню “Пуск”.

Изучение содержимого директорий соответственно заданию и описание в отчете их назначения.

Просмотреть содержимое диска C:. Зайдите в каталог C:\WINDOWS, ознакомьтесь с его содержимым и с содержимым всех подкаталогов.

Ознакомиться с исполняемыми файлами, находящимися в каталоге WINDOWS, в отчете укажите их назначение.

Изучите содержимое следующих директорий и опишите в отчете их назначение:

- Documents and Settings,
- Windows\ Driver Cache,
- Windows\Media,
- Windows\msagent,
- Windows\Shell New,
- Windows\WinSxS,
- Windows\Minidump,
- Windows\Fonts,
- Windows \Resources,
- Windows\Tasks,
- Windows\system32\dllcache,
- Windows \system32\Restore
- Windows\ Temp

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Ответы на вопросы преподавателя.

Лабораторная работа № 2

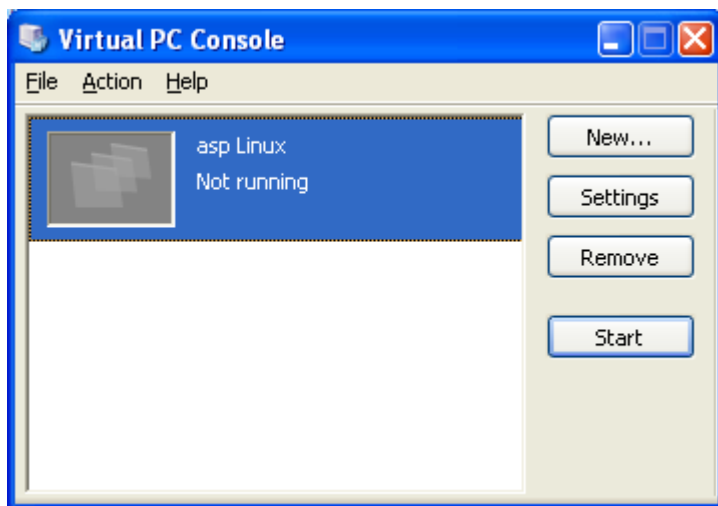
Изучение структуры операционной системы Linux

Цель работы: изучить ряд структурных элементов и элементов интерфейса операционной системы Linux.

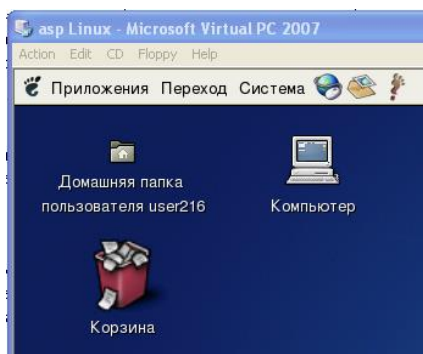
Изучение теоретических сведений.

Загрузка виртуальной операционной системы Linux.

- дважды кликните по ярлыку на рабочем столе Microsoft Virtual PC
- в появившемся окне дважды кликните по выделению asp Linux или нажмите кнопку Start:



- Введите имя пользователя, нажмите Enter
- Перепишите названия всех значков на рабочем столе:



- Удалите все посторонние ярлыки и каталоги
- Изучите свойства оставшихся каталогов и их содержимое.

- Изучите основные элементы KDE (Иконки, кнопки быстрого запуска приложений, панель задач, меню)

Выполнение необходимых настроек согласно методическим указаниям

Восстановление прежних настроек.

Завершение работы с операционной системой Linux

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Ответы на вопросы преподавателя.

Лабораторная работа № 3

Изучение работы с командами в операционной системе Windows

Цель работы: учиться применять основные команды Windows, отрабатывать навыки работы с операционной системой Windows.

Изучение теоретических сведений.

Выполните задания согласно указаниям с помощью консольных команд:

из любой заданной папки удалить все временные и резервные копии файлов;

Создать файл с использованием команды `at`, `if`, `for`. Файл должен использовать пакетные параметры командной строки с модификаторами, фильтры, команды сортировки и перенаправления, переменные среды,

Вывести в файл список всех команд.

Вывести в файл с именем `ИмяКоманды.txt` помощь по выбранной команде.

Создать файл с именем, содержащим текущую дату или дату создания какого-либо файла (например – `Дата_15_03_2020`), используя команду `for`.

Создать файл, в котором используется команда `call`. Написать в отчете разницу между простым вызовом исполняемого файла и запуском файла с использованием команды `call`.

Перенесите в отчет названия действий и необходимую запись в командной строке для его выполнения

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Ответы на вопросы преподавателя.

Лабораторная работа № 4

Изучение работы с командами в операционной системе Linux

Цель работы: получить основные навыки работы с файлами и каталогами в командной строке ОС Linux.

Изучение теоретических сведений.

Загрузка виртуальной операционной системы Linux.

Выполнение практических заданий согласно методическим указаниям

Восстановление прежних настроек.

Завершение работы с операционной системой Linux

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Ответы на вопросы преподавателя.

Лабораторная работа №5

Работа с файлами и каталогами в операционных системах MS DOS и Windows

Цель работы: отработать навыки работы с объектами Windows, изучить их свойства и настройки, изучить основные клавиатурные сочетания.

Изучение теоретических сведений.

Изучение основных сочетаний клавиш.

Выполнение дополнительного задания согласно теме варианта

Темы:

1. Глоссарий Windows.
2. Общие сведения о сочетаниях клавиш в Windows.
3. Общие сведения о диспетчере служебных программ.
4. Игры Windows.

5. Установка сетевой службы.
 6. Использование средств администрирования на консоли управления Microsoft.
 7. Использование средства «Производительность».
 8. Краткое руководство по совместному использованию ПК.
 9. Общее представление о параметрах безопасности и конфиденциальности.
 10. Основные сведения о сертификатах.
 11. Включение и отключение брандмауэра Windows.
 12. Общие сведения о брандмауэре Windows.
 13. Общие сведения о протоколе TCP/IP.
- Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.
- Ответы на вопросы преподавателя.

Лабораторная работа №6

Работа с файлами и директориями в операционной системе Linux.

Цель работы: получить основные навыки работы с файлами и каталогами в командной строке ОС Linux.

Изучение теоретических сведений.

Загрузка виртуальной операционной системы Linux

Выполнение практических заданий согласно методическим указаниям

Изучение основных команд для работы с файлами и директориями в консольном режиме

Восстановление прежних настроек

Завершение работы с операционной системой Linux

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Ответ на контрольные вопросы, выполнение дополнительного задания.

Лабораторная работа №7.

Создание файлов пакетной обработки данных

Цель работы: приобрести практические навыки создания файлов пакетной обработки.

Краткие теоретические сведения

Параметры, значения которых будут заданы при обращении к командному файлу, внутри файла могут иметь обозначения %1, %2, ..., %N (где N<10).

Любой командный пакетный файл имеет расширение bat и может вызываться с параметрами, указанными через пробел (например: class.bat Иванов Петров Сидоров), доступ к параметрам в командном файле осуществляется следующим образом: %номер_параметра (например, %1 - Иванов, %2 - Петров)

Основные команды пакетных командных файлов.

Синтаксис команды	Название команды
REM <любые символы>	Комментарии в командном файле
ECHO <сообщение>	Вывод сообщения на экран
ECHO OFF	Выключение вывода команд на экран
PAUSE <сообщение>	Приостановка выполнения командного файла
GOTO <метка> ... :<метка>	Переходы в командном файле
IF <условие> <команда> <ul style="list-style-type: none">• IF ERRORLEVEL < команда >• IF EXIST <имя или шаблон файла> <команда>• IF %==<текстовая строка> <команда>	Проверка условий в командном файле <ul style="list-style-type: none">• кода завершения программы, перед IF• наличия файла (NOTотсутствие файла)• сравнение двух строк
FOR %переменная IN (множество) DO команда [параметры]	Выполняет заданную команду для каждого файла или набора файлов

Команды системы

Команда CALL

call [имя_командного_файла] [параметры]

Команда CALL вызывает выполнение командного файла из другого командного файла. Если указаны параметры, то эти параметры передаются

вызываемому файлу. Они доступны там, как значения символов %1 - %9. По окончании выполнения вызванного файла, выполнение исходного файла продолжается со следующей строки. В команде call не допускается перенаправление ввода/вывода (использование символов < > |).

Команда COMMAND –пошаговое выполнение командных файлов
COMMAND /Y /C имя_командного_файла [параметры]

Команда CHOICE (внешняя команда!)

CHOICE [/C[:]клавиши] [/N] [/S] [/T[:]с,nn] [текст]

Команда CHOICE выводит пользователю заданную подсказку для выбора командного файла. Ждет, пока пользователь выберет из указанного набора клавиш. Эту команду можно использовать только в командных файлах.

CHOICE [/C[:]клавиши] [/N] [/S] [/T[:]с,nn] [текст]

Параметр "текст" задает текст, который вы хотите выводить перед подсказкой.

Параметр /C[:]клавиши задает допустимые в подсказке клавиши. При выводе на экран клавиши будут разделяться запятыми, заключаться в квадратные скобки ([]) и сопровождаться вопросительным знаком. Если вы не указываете параметр /C, CHOICE использует по умолчанию YN. Двоеточие (:) не обязательно.

Параметр /N приводит к тому, что CHOICE не выводит подсказку. Однако текст перед подсказкой выводится. При задании /N указанные клавиши все равно будут доступны.

/S приводит к различимости в CHOICE регистра символов. Если параметр /S не задан, то для любых заданных пользователем клавиш будет восприниматься как верхний, так и нижний регистр.

Например, если в командном файле вы указываете команду:

choice /c:unc

то пользователь видит следующее:

[Y,N,C]?

Если вы добавите:

choice /c:unc Yes, No, or Cancel

то пользователю выводится:

Yes, No, or Cancel [Y,N,C]?

Если вы зададите команду:

choice /c:unc /t:n,7

то пользователь видит на экране:

[Y,N,C]?

Если по истечении 7 секунд пользователь не нажмет клавишу, CHOICE выбирает N и возвращает значение ERRORLEVEL 2. При нажатии соответствующей клавиши до истечения 7 секунд CHOICE возвращает значение, соответствующее выбору пользователя.

Команда DEBUG запускает программу-отладчик Debug, которую вы можете использовать для проверки и отладки выполняемых файлов.

DEBUG [[диск:][маршрут]имя_файла [параметры]]

Команда СНСР без параметров выводит номер активного набора символов (кодовой страницы). Эту команду можно также использовать для изменения активного набора символов для всех устройств, поддерживающих переключение наборов (для смены кодировки можно использовать параметр **1251**).

Параметр	Описание
/? или /H	Выводят справочный текст.
nnn	Номер кодовой страницы.

Параметр nnn задает системный набор символов. DOS предусматривает наборы символов для следующих стран и языков:

Страна	Кодовая страница	Код страны
Великобритания	437 850	044
Германия	437 850	049
Испания	437 850	034
Россия	866 850	007
США	437 850	001
Франция	437 850	033
...

Изучение теоретических сведений.

Создать следующие пакетные файлы, которые выполняют следующие действия:

1.bat в папке D:\129 создает подпапку 1, в которой создает текстовые файлы 1.txt, 2.txt. Затем переименовывает файл 1.txt в 3.txt.

2. bat удаляет D:\129\1.

Создать два файла 31.bat и 32.bat, которые демонстрируют работу оператора call.

4.bat копирует указанный файл в подготовленную заранее папку.

5.bat сообщает, существует ли указанный файл.

6.bat. Если в качестве параметра указать 1.txt и он есть, то скопировать его в папку D:\129\1. Иначе сообщить, что не выполнено.

7.bat создает папку D:\129\7 и копирует в нее файлы с расширениями txt и bat из текущей папки (использовать команду for).

8.bat удаляет папку D:\129\7.

Изучите подробно команды set, color, title, pause, start, cmd, if, prompt. Приведите примеры пакетных файлов с использованием указанных команд. Для этого в режиме командной строки наберите название команды и «/?». Например, title/?. Создайте текстовый файл с описанием указанных команд.

Реализуйте калькулятор с использованием bat-файлов. После вычисления значения реализуйте возможность запроса на повторное вычисление. Для этого используйте конструкцию вида set /p var0=»Фраза» с последующей проверкой.

Создайте 2 bat-файла — первый настраивает под Ваши предпочтения командную строку и цветовую гамму, второй восстанавливает исходное состояние.

Реализуйте 2 bat-файла для открытия одного и того же html-документа средствами Internet Explorer и Mozilla FireFox.

Изучить команду Path, добавить несколько записей. Привести примеры запуска различных программ. Организуйте запуск нескольких программ с ожиданием, т.е следующая программа ждет пока закончит работу предыдущая.

Предложить использование команды start с другими ключами.

Создайте bat-файл с использованием if, в котором есть команды вида NEQ. Например, запрашиваются 3 числа, выводится наибольшее и

наименьшее.

Изучить возможности `for` для создания рекурсий. Вывести содержимое всех подкаталогов каталога `D:\129`.

Создать «большой» пакетный файл, производящий очистку определенных директорий.

Создать пакетный файл, который бы заменял некий файл на диске.

Пакетный файл для резервного копирования. Завершение работы с операционной системой

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Ответы на вопросы преподавателя.

Лабораторная работа №8.

Файлы пакетной обработки данных. Операторы пакетных файлов

Цель работы: отрабатывать практические навыки создания файлов пакетной обработки.

Краткие теоретические сведения

Параметры, значения которых будут заданы при обращении к командному файлу, внутри файла могут иметь обозначения `%1`, `%2`, ..., `%N` (где $N < 10$).

Любой командный пакетный файл имеет расширение `bat` и может вызываться с параметрами, указанными через пробел (например: `class.bat Иванов Петров Сидоров`), доступ к параметрам в командном файле осуществляется следующим образом: `%номер_параметра` (например, `%1` - Иванов, `%2` - Петров)

Основные команды пакетных командных файлов.

Синтаксис команды	Название команды
<code>REM <любые символы></code>	Комментарии в командном файле

ECHO <сообщение>	Вывод сообщения на экран
ECHO OFF	Выключение вывода команд на экран
PAUSE <сообщение>	Приостановка выполнения командного файла
GOTO <метка> ... :<метка>	Переходы в командном файле
IF <условие> <команда> <ul style="list-style-type: none"> • IF ERRORLEVEL < команда > • IF EXIST <имя или шаблон файла> <команда> • IF %==<текстовая строка> <команда> 	Проверка условий в командном файле <ul style="list-style-type: none"> • кода завершения программы, перед IF • наличия файла • сравнение двух строк

Изучение теоретических сведений.

Загрузка операционной системы.

Выполнение практических заданий согласно методическим указаниям

Работа в эмуляторе ОС с правами администратора

Создание пакетных файлов в режиме командной строки и в текстовом редакторе в соответствии с заданиями.

Задание 1.

Работая в командной строке, создайте пакетный файл на рабочем столе, в который должны войти следующие команды:

- вывести дерево каталогов для Рабочего стола;
- переместить получившийся список дерева каталогов в файл *desktop.doc* на Рабочем столе;
- запустить Qbasic;
- просмотреть содержимое директории Windows в кратком виде;
- вывести на экран информацию обо всех текстовых файлах на Рабочем столе;
- создать на Рабочем столе папку TU;
- скопировать все файлы с расширением .dll из каталога Windows в TU;
- вывести на экран подробные сведения о конфигурации компьютера и операционной системы.

Задание 2.

Работая в командной строке или в приложении Блокнот, создайте командный файл lab.bat, реализующий выполнение следующего алгоритма: при обращении к данному файлу с указанием фамилии студента (<имя файла>_<соответствующий параметр>) на экране должна появиться краткая характеристика на этого студента.

Порядок выполнения

1. **Заблокировать вывод команд на экран.**
2. **Вывести на экран строку: CHARACTERISTIC OF THE STUDENT <фамилия нужного студента> (или HAKAKTERISTIKA STUDENTA <фамилия нужного студента>, так как в связи с проблемой кодировки лучше осуществлять набор английскими символами).**
3. **Добавить комментарий:** Проверяем введенный параметр - это Петров? Если да, переходим к метке **P**
4. **В соответствии с комментарием ввести строку для проверки условия: IF %1==Petrov GOTO P**
5. **Добавить комментарий:** Проверяем введенный параметр - это Иванов? если да, переходим к метке **I**
6. **В соответствии с комментарием ввести строку для проверки условия: (самостоятельно)**
7. **Добавить комментарий:** Если ни Иванов, ни Петров, то переходим к метке **E**
8. **GOTO E**
9. **Добавить комментарий:** ЗАДАНИЕ МЕТОК
10. **Добавить комментарий:** Выводим характеристику Петрова
- 11.:**P**
- 12.**ECHO Huligan**
- 13.**GOTO exit**
- 14.**Добавить комментарий:** Выводим характеристику Иванова
- 15.:**I**
- 16.**ECHO Otlichnik**
- 17.**GOTO exit**
- 18.**Добавить комментарий:** Выводим сообщение об отсутствии информации о студенте
- 19.:**E**
- 20.**ECHO neizvesten**

21.GOTO exit

22.Добавить комментарий: Конец командного файла

23.:exit

24. Добавьте характеристики еще для трех студентов

Работа с файлом autoexec.bat (наполнение командами, перезагрузка ОС)

Восстановление прежних настроек.

Завершение работы с операционной системой

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Ответы на вопросы преподавателя.

Лабораторная работа №9.

Управление процессами в операционной системе Windows

Цель работы: научиться использовать сервисы и средства управления процессами в операционной среде Windows.

Краткие теоретические сведения

Schtasks

Настраивает выполнение команд и программ через заданные интервалы или в указанное время. Добавляет и удаляет задания из расписания, запускает и останавливает задания по требованию, отображает и изменяет назначенные задания.

- Schtasks create
- Schtasks change
- Schtasks run
- Schtasks end
- Schtasks delete
- Schtasks query

Schtasks create- Создает новое назначенное задание.

schtasks /create /tn *имя_задания* /tr *выполнение_задания* /sc *расписание* [/mo *модификатор*] [/d *день*] [/m *месяц*[,*месяц*...]] [/i *время_простоя*] [/st *время_запуска*] [/sd *дата_запуска*] [/ed *дата_окончания*] [/s *компьютер*] [/u [*домен*]*пользователь* /r *пароль*]] [/ru {[*домен*]*пользователь* | "System"}] [/rp *пароль*]] /?

Параметры

/tn имя_задания

Определяет имя для задания.

/tr выполнение_задания

Указывает программу или команду, которая выполняет задание. Введите полный путь и имя исполняемого файла, файла сценария или пакетного файла. Если пропущен путь, программа SchTasks.exe предполагает, что файл находится в *системном_корневом_каталоге\System32*.

/sc расписание

Задаёт тип расписания. Допустимыми значениями являются ЕЖЕМИНУТНО, ЕЖЕЧАСНО, ЕЖЕДНЕВНО, ЕЖЕНЕДЕЛЬНО, ЕЖЕМЕСЯЧНО, ОДНОКРАТНО, ПРИ ЗАПУСКЕ, ПРИ ВХОДЕ В СИСТЕМУ, ПРИ ПРОСТОЕ.

Значение	Описание
ЕЖЕМИНУТНО, ЕЖЕЧАСНО, ЕЖЕДНЕВНО, ЕЖЕНЕДЕЛЬНО, ЕЖЕМЕСЯЧНО	Определяют единицу времени для расписания
ОДНОКРАТНО	Задание выполняется один раз в указанное время и дату
ПРИ ЗАПУСКЕ	Задание выполняется каждый раз при запуске системы. Можно указать дату запуска или выполнить задание в следующий раз при запуске системы
ПРИ ВХОДЕ В СИСТЕМУ	Задание выполняется каждый раз при входе пользователя (любого) в систему. Можно указать дату запуска или выполнить задание в следующий раз при входе пользователя в систему
ПРИ ПРОСТОЕ	Задание выполняется каждый раз при простое компьютера в течение заданного промежутка времени. Можно указать дату запуска или выполнить задание в следующий раз при простое компьютера

/то модификатор

Определяет частоту выполнения команды в соответствии с его типом расписания. Этот параметр обязателен для расписания ЕЖЕМЕСЯЧНО. Параметр является допустимым, но необязательным для расписаний

ЕЖЕМИНУТНО, ЕЖЕЧАСНО, ЕЖЕДНЕВНО или ЕЖЕНЕДЕЛЬНО. По умолчанию значение равно 1.

Тип расписания	Модификатор	Описание
ЕЖЕМИНУТНО	1 - 1439	Задание выполняется через каждые n минут.
ЕЖЕЧАСНО	1 - 23	Задание выполняется через каждые n часов.
ЕЖЕДНЕВНО	1 - 365	Задание выполняется через каждые n дней.
ЕЖЕНЕДЕЛЬНО	1 - 52	Задание выполняется через каждые n недель.
	1 - 12	Задание выполняется через каждые n месяцев.
	ПОСЛЕДНИЙ	Задание выполняется в последний день месяца.
ЕЖЕМЕСЯЧНО	ПЕРВЫЙ, ВТОРОЙ, ТРЕТИЙ, ЧЕТВЕРТЫЙ, ПОСЛЕДНИЙ	При использовании с параметром /d <i>день</i> задание выполняется в определенный день недели. Например, в третью среду месяца.

/d *день*

Указывает день недели и день месяца. Параметр допустим только с расписаниями ЕЖЕНЕДЕЛЬНО или ЕЖЕМЕСЯЧНО.

Тип расписания	Значение дня
ЕЖЕНЕДЕЛЬНО	Необязательно. Допустимые значения — с ПН по ВС и *(каждый день). По умолчанию установлено значение ПН
ЕЖЕМЕСЯЧНО	Значения ПН — ВС требуются, когда используется модификатор (/mo) ПЕРВЫЙ, ВТОРОЙ, ТРЕТИЙ, ЧЕТВЕРТЫЙ ИЛИ ПОСЛЕДНИЙ. Значения 1 — 31 являются необязательными и допустимы, только если модификатора нет или тип его 1 — 12. По умолчанию установлено значение 1 (первый день месяца).

/m *месяц*[,*месяц*...]

Указывает месяц года. Допустимые значения — с ЯНВ по ДЕК и * (каждый месяц). Параметр /m действителен только для расписания ЕЖЕМЕСЯЧНО. Параметр требуется при использовании модификатора ПОСЛЕДНИЙ. В

других случаях он является необязательным. Значение по умолчанию *(каждый месяц).

/i время_простоя

Указывает интервал времени простоя компьютера (в минутах) до начала выполнения задания. Введите любое число от 1 до 999. Параметр допустим только с расписанием ПРИ ПРОСТОЕ, в этом случае он обязателен.

/st время_запуска

Задаёт время начала выполнения задания в 24-часовом формате в виде чч:мм:сс. По умолчанию установлено локальное время, когда команда завершена. Параметр /st является действительным для расписаний типа ЕЖЕМИНУТНО, ЕЖЕДНЕВНО, ЕЖЕНЕДЕЛЬНО, ЕЖЕМЕСЯЧНО и ОДНОКРАТНО . Требуется для расписания ОДНОКРАТНО.

/sd дата_запуска

Задаёт дату начала выполнения задания в формате мм/дд/гггг. По умолчанию установлено значение текущей даты. Параметр /sd можно использовать со всеми расписаниями. Он требуется для расписания ОДНОКРАТНО.

/ed дата_окончания

Определяет последнюю дату, на которую назначается выполнение команды. Параметр является необязательным. Недопустимо использование с расписаниями типа ОДНОКРАТНО, ПРИ ЗАПУСКЕ, ПРИ ВХОДЕ В СИСТЕМУ, ПРИ ПРОСТОЕ. По умолчанию у расписаний нет даты окончания.

/s компьютер

Задаёт имя или IP-адрес удаленного компьютера (с обратными косыми чертами или без них). По умолчанию используется локальный компьютер.

/u [домен\]пользователь

Выполняет команду с разрешения указанной учетной записи пользователя. По умолчанию, команда запускается с разрешения пользователя, вошедшего в систему компьютера, на котором выполняется SchTasks.

/p пароль

Задаёт пароль учетной записи пользователя, указанной параметром /u. Параметр требуется при использовании параметра /u.

/ru {[домен\]пользователь | "System"}

Выполняет задания с разрешения указанной учетной записи пользователя. По умолчанию, начинается выполнение задание с разрешения пользователя, вошедшего в систему компьютера, на котором выполняется SchTasks.

Значение

Описание

[домен\]пользователь Определяет учетную запись пользователя.

"System" или "" Указывает учетную запись NT Authority\System, которую использует операционная система.

/rp пароль

Определяет пароль учетной записи пользователя, заданной параметром /u. Если пропустить этот параметр при указании учетной записи пользователя, программ SchTasks.exe предлагает ввести пароль и скрывает введенный текст. Для заданий, выполняемых с разрешениями учетной записи NT Authority\System, не требуется пароль и программа SchTasks.exe не предлагает его ввести.

/?

Отображает справку в командной строке.

Примечания

Использование параметров командной строки /s, /u и /p

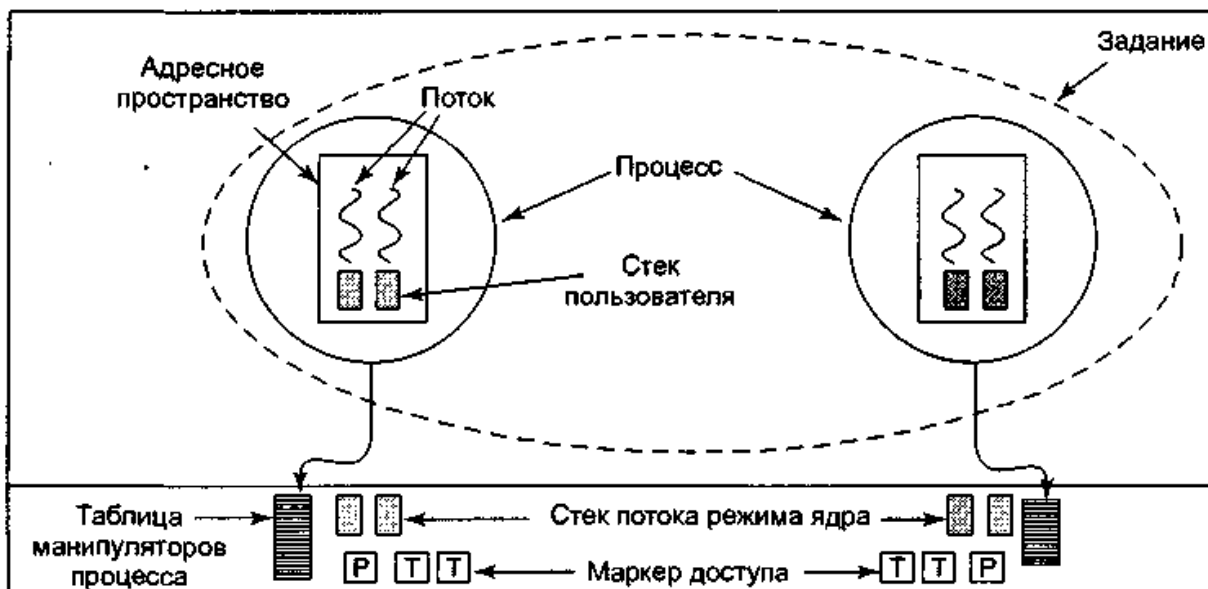
Параметры командной строки /u и /p доступны только при использовании параметра /s. Для ввода пароля пользователя необходимо использовать параметры /p и /u.

Пример:

```
C:\Documents and Setting\Бася>SCHTASKS /create /sc minute /tn  
"калькулятор" /tr C:\WINDOWS\system32\calc.exe /st "00:42:05" /sd  
"17/12/2019"
```

Ход работы.

Изучить работу утилиты PerfMon.exe (Performance monitor - системный монитор), входящий в состав ОС Windows NT/2000/XP. Используя утилиту PerfMon, исследовать загрузку центрального процессора и подсистемы управления памятью (использование оперативной памяти и файла подкачки) при запуске ресурсоемких приложений. Зарисовать диаграмму загрузки процессор. Изучить назначение и работу диспетчера задач. Используя диспетчер задач, определить какие процессы запущены и сколько памяти они занимают. Записать протокол загрузки центрального процессора. Используя диспетчер задач, исследовать загрузку центрального процессора и использование оперативной памяти и файла подкачки при запуске ресурсоемких приложений.



Ход работы

1. Зарисуйте диаграмму загрузки процессора.
2. Запустите командную оболочку Windows (**cmd.exe**)
3. Перенесите в отчет информацию о командах Windows для работы с процессами

Команды Windows для работы с процессами:

- at** - запускает программы в заданное время;
- Schtasks** - настраивает выполнение команд по расписанию;
- Start** - запускает определенную программу или команду в отдельном окне;
- Taskkill** - завершает процесс;
- Tasklist** - выводит информацию о работающих процессах и памяти

Для получения более подробной информации, можно использовать центр справки и поддержки или команду **help** (например: **help at** или **at /?**)

Отобразите подробную информацию о запущенных процессах и памяти

4. Отобразите все процессы, которые загрузили модули dll
5. Просмотрите подробную информацию о командах для работы с процессами Windows.

Выполните примеры работы базовой команды с различными параметрами (в отчете укажите, что выполняют использованные

вами команды и их параметры)

Завершение работы с операционной системой

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Ответы на вопросы преподавателя.

Лабораторная работа №10.

Управление процессами в операционной системе Linux

Цель работы: научиться использовать сервисы и средства управления процессами в операционной среде Linux.

Краткие теоретические сведения

Информация о процессах

Если вы выведете список содержимого каталога /proc, вы увидите много каталогов, именами которых являются номера. Эти каталоги содержат информацию о всех процессах в системе, запущенных в данный момент:

```
$ ls -d /proc/[0-9]*  
/proc/1/  /proc/302/ /proc/451/ /proc/496/ /proc/556/ /proc/633/  
/proc/127/ /proc/317/ /proc/452/ /proc/497/ /proc/557/ /proc/718/  
/proc/2/  /proc/339/ /proc/453/ /proc/5/  /proc/558/ /proc/755/ .....
```

Обратите внимание, что как пользователь, вы можете вывести информацию только о своих собственных процессах, но не других пользователей. Поэтому, если имеете такую возможность, войдите в систему под root'ом и посмотрите, какая информация доступна для процесса 1, которым является процесс init и который отвечает за запуск всех остальных процессов:

```
$ su  
Password:  
# cd /proc/1  
# ls -l
```

```
total 0
-r----- 1 root root 0 Aug 15 18:14 auxv
-r--r--r-- 1 root root 0 Aug 15 18:14 cmdline
lrwxrwxrwx 1 root root 0 Aug 15 18:14 cwd -> //
-r----- 1 root root 0 Aug 15 18:14 environ
lrwxrwxrwx 1 root root 0 Aug 15 18:14 exe -> /sbin/init*
dr-x----- 2 root root 0 Aug 15 18:14 fd/
-rw-r--r-- 1 root root 0 Aug 15 18:14 loginuid
-r--r--r-- 1 root root 0 Aug 15 18:14 maps
```

Каждый из каталогов содержит одинаковые пункты:

- `cmdline`: этот (псевдо-) файл содержит полную командную строку, использованную для вызова процесса. Он не отформатирован: между программой и её аргументами нет пробелов, а в конце строки нет символа окончания строки. Чтобы просмотреть его, вы можете использовать: `perl -ple 's,\00, ,g' cmdline`.
- `cwd`: эта символическая ссылка указывает на текущий рабочий каталог процесса (следует из имени).
- `environ`: этот файл содержит все переменные окружения, определённые для этого процесса, в виде ПЕРЕМЕННАЯ=значение. Как и в `cmdline`, вывод вообще не отформатирован: нет разделителей строк для отделения различных переменных, и в конце нет символа окончания строки. Одно из решений для его просмотра: `perl -ple 's,\00,\n,g' environ`.
- `exe`: эта символическая ссылка указывает на исполняемый файл, соответствующий запущенному процессу.
- `fd`: этот подкаталог содержит список файловых дескрипторов, открытых в данный момент процессом.
- `maps`: когда вы выводите содержимое этого именованного канала (при помощи команды `cat`, например), вы можете увидеть части адресного пространства процесса, которые в текущий момент распределены для файла. Вот эти поля (слева направо): адресное пространство, связанное с этим распределением; права доступа, связанные с этим распределением; смещение от начала файла, где начинается распределение; старший и младший номера (в шестнадцатеричном виде) устройства, на котором находится распределенный файл; номер `inode` файла; и, наконец, имя самого файла. Если устройство обозначено как 0 и отсутствует номер `inode` или имя файла - это анонимное распределение.
- `root`: эта символическая ссылка указывает на корневой каталог, используемый процессом. Обычно это будет `«/»`.
- `status`: этот файл содержит разнообразную информацию о процессе: имя исполняемого файла, его текущее состояние, его PID и

PPID, его реальные и эффективные UID и GID, его использование памяти и другие данные. Обратите внимание, что файлы stat и statm теперь устарели. Информация, которая в них содержалась, теперь хранится в status.

Ход работы

Изучение теоретических сведений.

Загрузка виртуальной операционной системы Linux.

Выполнение практических заданий в терминале ОС Linux:

1. Войти в систему пользователь root / пароль 123456
2. Изучить возможности параллельной работы нескольких консолей Linux
3. Изучить команду ps
4. Получить список процессов для текущей сессии: ps
5. Провести исследование параметров запущенных процессов
7. Аутентифицироваться еще в одной консоли
8. Запустить еще один telnet подключиться к серверу (шаги 1-3).

Выполнить еще раз команду ps -a. Изучить список новых процессов

9. Принудительно завершить процесс KILL -s SIGKILL номер процесса.

Номер процесса определяется по имени нового командного интерпретатора sh

10. Запустить несколько процессов одновременно: man ps &ps и наоборот ps &man ps. Проанализировать отличия в работе процессов
11. Получить список процессов, находящихся в фоновом режиме (jobs)
12. Активировать фоновые процессы fg номер процесса
13. Запланировать команду на однократное выполнение: команда at время выполнения

Восстановление прежних настроек.

Завершение работы с операционной системой Linux

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Ответы на вопросы преподавателя.

Лабораторная работа №11.

Работа с текстовыми файлами в операционных системах MS DOS и Windows

Цель работы: учиться применять основные команды DOS/Windows в работе с текстовыми файлами

Изучение теоретических сведений.

Загрузка виртуальной операционной системы Windows.

Выполнение практических заданий по редактированию текста согласно методическим указаниям

Реализация выполнения команд и определение их назначения

Восстановление прежних настроек.

Завершение работы с операционной системой Windows

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Ответы на вопросы преподавателя.

Лабораторная работа №12.

Работа с текстовыми файлами в операционной системе LINUX

Цель работы: изучить основные команды ОС Linux для работы с текстовыми файлами, получить практические навыки работы с утилитами

Краткие теоретические сведения

Команды для работы с текстовыми файлами

Выполняемая функция	Наименование команды	Спецификация команды	Пример использования
Создать файл	cat	cat > имя_файла введите данные, по окончании ввода нажмите ^D	cat > /home/test/1.txt Примечание: ввод буферизован (накапливается 512 знаков, потом сохраняется)
Создать файл	cat	cat -u> имя_файла	cat -u> /home/test/1.txt Примечание: ввод небуферизован (unbuffers)

Выполняемая функция	Наименование команды	Спецификация команды	Пример использования
Сравнить файлы	diff cmp	diff имя_файла cmp имя_файла	diff 1.txt 2.txt
Искать pattern в files	grep pattern files	—	—
Искать рекурсивно pattern в dir	grep -r pattern dir	—	—
Искать pattern в выводе command	command grep pattern	—	—
Найти все файлы с именем file	locate file	—	—
Отсортировать данные в файле	sort	sort имя_файла	sort 1.txt Примечание: сортировка осуществляется по первому символу
Отсортировать данные в файле с указанием столбца при сортировке	sort	sort +n имя_файла, где n – количество столбцов, которое необходимо проигнорировать при сортировке	sort +1 1.txt Примечание: сортировка будет производиться по второму столбцу (столбцы отделены табулятором)
Отсортировать данные в файле с игнорированием лидирующих пробелов	sort	sort -b имя_файла,	sort +1 -b 1.txt Примечание: сортировка будет производиться по второму столбцу с игнорированием лидирующих пробелов
Отсортировать данные в файле с сохранением отсортированных данных в файле	sort	sort имя_файла_для_сортировки -o имя_отсортированного_файла	sort +2 -b 1.txt -o sort1.txt Примечание: сортировка будет производиться по третьему столбцу с игнорированием лидирующих пробелов с сохранением результатов сортировки в файле

Выполняемая функция	Наименование команды	Спецификация команды	Пример использования
			sort 1.txt
Разрезать файл на части	split	split имя_файла	split 1.txt Примечание: по умолчанию в каждой части будет содержаться 1000 строк
Разрезать файл на части с указанием количества строк	split	split -m имя_файла, где m - количество строк	split -10 1.txt
Сборка файла	cat	cat [name1]>[name2]	cat split?>new
Поиск файла	find	find [path] [key] [name]	find / -name 1.txt Найти от корневого каталога файл с именем 1.txt
Печать файла	lpr lp	lpr [ключи] имя_файла	lpr -w72 -l25 -h "My programm" 1.txt Печать файла с указанием длины, ширины, заголовка на каждой странице
Считать количество слов (строк)	wc	wc [ключи] имя_файла	wc 1.txt

Ход работы

Изучение теоретических сведений.

Загрузка виртуальной операционной системы Linux.

Выполнение практических заданий по редактированию и форматированию текста согласно методическим указаниям

Выполнение следующих действий с текстовыми файлами с помощью команд:

1. Перенаправить информацию о содержании каталога /root в текстовый

файл /root/ test /dir.txt.

2. Скопировать с переименованием файл /root/test/dir.txt в каталог /home под именем view1.txt. Предварительно создайте каталог test/

3. С помощью текстового редактора vi (используя методы копирования и вставки любого содержания) создать файл /root/view2.txt с количеством строк 50.

4. Просмотреть содержимое файла /root/view2.txt и определить количество строк в этом файле.

5. Разрезать файл /root/view2.txt на части размером 10 строк с именами частей, начинающихся на split_test. Части файла поместить в каталог /root/test.

6. Собрать разбитый на части файл в один с именем /root/test/new_test.txt.

7. Сравнить файлы /root/view2.txt и /root/test/new_test.txt до разрезания и после сборки.

8. Создать файл /root/view3.txt.

9. Данные разместить в виде столбцов (например, Иванов Иван Иванович 1975), в качестве разделителя используйте табулятор).

10. Найти в указанных файлах слово (например, Иванов).

11. Отсортировать файл /root/view3.txt сначала по первому, потом по второму, третьему столбцам.

Завершение работы с операционной системой Linux

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Ответы на вопросы преподавателя.

Лабораторная работа №13.

Работа с операционными оболочками (Total Commander)

Цель работы: научиться работать с файловым менеджером Total Commander; изучить структуру и команды используемые в нем.

Краткие теоретические сведения

Total Commander позволяет выполнять команды DOS, программы и командные файлы. Несколько команд, введенных последними, запоминаются в особом буфере, откуда, их можно извлечь, при необходимости отредактировать и выполнить вновь.

Одновременно с перемещением по каталогам и действиями над файлами (копированием, удалением и т. п.) можно вводить очередную команду). Выполнена она будет в тот момент, когда будет нажата клавиша «Enter» или ЛКМ (Левая кнопка мыши).

Если нужно выполнить без параметров файл текущего каталога с расширением .COM, .EXE или .BAT, достаточно подвести курсор к его имени и нажать клавишу «Enter» или ЛКМ .

Можно не набирать команду заново, а отредактировать ранее введенную. При первом нажатии клавиши «CTRL+E» вплотную к подсказке появляется команда, выполненная последней, при следующем - предшествующая и т. д. Возвращаться от более ранних команд к более поздним позволяют клавиши «Ctrl+X».

Важно знать, что Total Commander (Far, WC и др.). – резидентная программа для облегчения работы пользователя с операционной системой. Она позволяет:

- постоянно видеть содержимое одного или двух каталогов;
- получать информацию об оперативной памяти и памяти на магнитном диске;
- переходить из каталога в каталог;
- создавать, переименовывать и удалять каталоги;
- копировать, перемещать из каталога в каталог без копирования, переименовывать и удалять файлы (поодиночке и группами);
- просматривать произвольные и редактировать текстовые файлы;

- редактировать командную строку и выполнять любые команды DOS.

Ход работы

Изучение теоретических сведений.

Изучение меню и рабочей области Total Commander.

Выполнение практических заданий согласно методическим указаниям:

Запустите Windows Commander. Откройте в левом файловом окне содержимое диска C:, для чего в поле списка дисков левого окна выберите диск C:

Задайте вид отображения папок и файлов в левом окне Windows Commander в виде таблицы с показом имени, типа файла, его размера, даты изменения и атрибутов, для чего в меню Вид выберите опцию Подробный.

Упорядочите размещение значков папок и файлов в левом окне по дате последнего изменения, для чего выберите в меню Вид команду По дате. Отсортируйте значки папок и файлов в правом файловом окне по алфавиту имен, для чего выберите в меню Вид команду По имени.

Создайте в корневом каталоге диска C: папку с именем 1111, для чего, щелкнув кнопку F7, в окне Создать новый каталог задайте имя каталога 1111. Откройте в левом окне каталог 1111.

Установите активным правое файловое окно. Задайте отображение на правой панели каталога C:\Wincmd. Скопируйте из каталога C:\Wincmd файлы с расширением *.txt. Для выделения группы файлов в меню Выделение выберите команду Выделить группу, затем в окне Добавить выделение файлов укажите шаблон *.txt и щелкните кнопку ОК. Для копирования выделенных файлов щелкните кнопку F5 Копия и подтвердите операцию, щелкнув ОК.

Упакуйте файлы в папке C:\1111 в архив, для чего, выделив все файлы в папке C:\1111 в левом окне, выберите в меню Файл команду Упаковать. Затем в окне Упаковка файлов задайте имя каталога для размещения архива, включите опции Сохранять пути, Учитывать подкаталоги и выберите

архиватор ZIP. Имя файла архива оставьте по умолчанию 1111.

Выполнение дополнительного задания, полученного от преподавателя

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Лабораторная работа №14.

Работа с архиваторами в операционных системах MS DOS и Windows

Цель работы: отработать навыки работы с архиваторами ОС Windows (WinRAR, 7-Zip, PowerArchiver): создание самораспаковывающихся и защищенных архивов, тестирование, установка пароля.

Краткие теоретические сведения

Сжатие данных похоже на производство сухого молока или сухофруктов. То есть, это – процесс удаления воды, которую затем можно добавить, чтобы предать продукту первоначальный вид.

А какая в данных может быть вода? Это вода информационная. В данных встречается много повторов. Это можно использовать для сжатия данных.

Например, сжатие текстовых файлов происходит приблизительно так. Составляется таблица встречающихся в тексте слов и выражений. Затем всем словам и выражениям в этой таблице даются номера. И весь текст в файле заменяется номерами из таблицы слов и выражений. Такой способ позволяет уменьшить размер текстового файла в 2-3 раза. Иногда текст сжимается и в 10 раз, если в нём много повторов.

Программа, переводящая текстовый файл в "сжатый" вид, называется упаковщиком. А полученный в результате сжатия файл называется упакованным или сжатым файлом.

Очень часто сжатые файлы называют архивами или архивными файлами, что, если подходить строго к терминологии, неверно. Изначально архивами назывались файлы, специально созданные во время процессов резервного копирования. В ходе такого процесса создавался один файл, который содержал в себе несколько исходных файлов и папок. Это и был архив.

Никакого сжатия при этом не производилось. Подобная ситуация до сих пор существует в операционной системе Linux, где архивация данных и сжатие данных являются двумя независимыми процессами. В операционной системе MS-DOS, а затем и в MS Windows программы сжатия данных с самых первых своих версий стали поддерживать как сжатие, так и архивацию данных, то есть создавали сжатый файл, содержащий не один, а несколько исходных (архивируемых) файлов и папок. С той поры в этих операционных системах понятие "архивация" означает и архивацию (сбор в один файл-архив), и одновременное сжатие данных.

Поскольку архивный файл записан не в текстовом формате, с ним не могут работать редакторы текстов. Перед открытием архивного файла редактором текстов этот файл необходимо разархивировать. Разархивацию производит та же самая программа – архиватор. После разархивации текстовый файл приобретает точно такой же вид и размер, как и до архивации.

Архиваторы текстов могут архивировать также файлы программ. Только программы гораздо меньше сжимаются, чем текст.

Упаковщики, применяемые для сжатия текстов и программ, не могут эффективно сжимать звуковые, графические или видео файлы. Для их сжатия были разработаны другие, более сложные, алгоритмы. Правда, после распаковки полученные файлы немного отличаются от оригиналов (такое сжатие называется сжатием с потерями). Но этого не улавливает обычное человеческое ухо и не замечает обычный глаз на экране монитора.

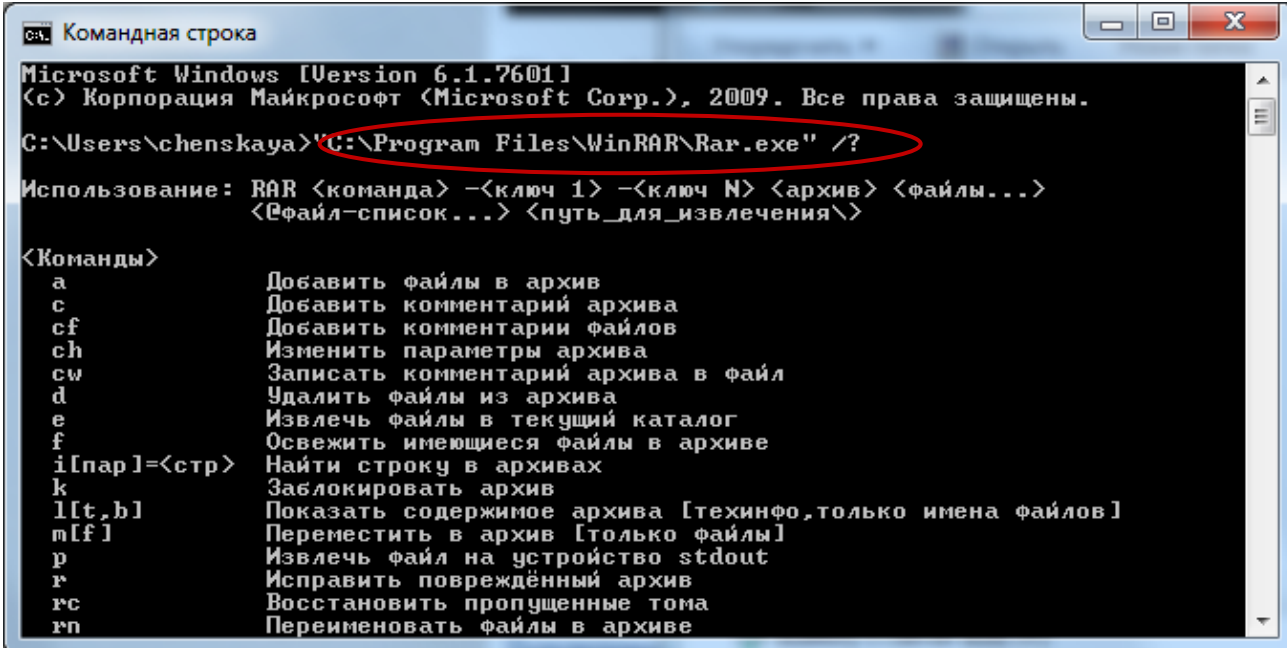
Ход работы

Изучение теоретических сведений.

Выполнение практических заданий согласно методическим указаниям

Выполнение дополнительного задания, полученного от преподавателя

1. Создайте архив в командной строке, используя утилиту WinRAR с необходимыми ключами (см рисунок ниже).



The screenshot shows a Windows Command Prompt window titled "Командная строка". The text inside is as follows:

```
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.
C:\Users\chenskaya>"C:\Program Files\WinRAR\Rar.exe" /?
Использование: RAR <команда> -<ключ 1> -<ключ N> <архив> <файлы...>
                  <@файл-список...> <путь_для_извлечения\>

<Команды>
a              Добавить файлы в архив
c              Добавить комментарий архива
cf            Добавить комментарии файлов
ch            Изменить параметры архива
cw            Записать комментарий архива в файл
d            Удалить файлы из архива
e            Извлечь файлы в текущий каталог
f            Освежить имеющиеся файлы в архиве
i[пар]=<стр>  Найти строку в архивах
k            Заблокировать архив
l[t,b]       Показать содержимое архива [техинфо, только имена файлов]
m[f]         Переместить в архив [только файлы]
p            Извлечь файл на устройство stdout
r            Исправить поврежденный архив
rc           Восстановить пропущенные тома
rp           Переименовать файлы в архиве
```

2. Добавьте любой файл в архив (указать команду с параметрами).
3. Просмотрите информацию об архиве (указать команду с параметрами).
4. Протестируйте ваш архив (указать команду с параметрами).
5. Защитите архив, установив пароль (указать команду с параметрами).
6. Преобразуйте обычный архив в самораспаковывающийся архив (указать команду с параметрами).
7. Извлеките файл из архива (указать команду с параметрами).
8. Оформление отчета с подробным описанием ваших действий и напишите вывод о проделанной работе.

Лабораторная работа №15.

Работа с архиваторами в операционной системе Linux

Цель работы: получить основные навыки работы со сжатыми файлами и системной информацией ОС Linux.

При сжатии файлы обрабатывают с помощью специального алгоритма, чтобы они соответственно занимали меньше места, это удобно при

необходимости длительного хранения или передачи на носителях или по сети.

Для разных файлов процент сжатия отличается, он зависит от содержания файлов, из-за особенностей алгоритма архивации самым высоким процентом обладают те файлы, которые содержат одинаковые символы. Например, файл размером 1 Гбайт, заполненный нулями, при сжатии будет преобразован в файл размером в несколько килобайт. Одними из самых “сжимаемых” файлов являются текстовые файлы, а “несжимаемых” – архивы, т.е. уже сжатые, при попытке их сжатия размер может уменьшиться незначительно или вообще увеличиться.

Основным средством архивирования в UNIX (а, следовательно, и в Linux) является комплекс из двух программ — tar и gzip.

Программа tar

У программы tar имеется 8 опций, отличающихся от остальных тем, что при вызове программы должна обязательно задаваться одна из этих опций. Эти опции определяют основные функции программы.

Основные опции программы tar

Опция	Значение
-A, --catenate, --concatenate	Добавляет файлы в существующий архив
-c, --create	Создает новый архив
-d, --diff, --compare	Найти различия между архивом и файловой системой
--delete	Удалить из архива (не может использоваться с магнитной лентой!)
-r, --append	Дописывает файлы в конец архива
-t, --list	Выводит список файлов архива
-u, --update	Добавляет только файлы, которые новее, чем имеющаяся в архиве копия
-x, --extract, --get	Извлечь файлы из архива

Программа gzip

Хотя программа tar создает архивы, она не сжимает архивы, а просто соединяет отдельные файлы в единый архивный файл. Для сжатия этого файла часто применяют команду gzip. В простейшем случае она вызывается в следующем формате:

[user]\$ gzip файл

В командной строке можно указать сразу несколько имен файлов или шаблон имени файла. Но в этом случае каждый из указанных файлов будет заархивирован отдельно (общий архив не создается).

Для того, чтобы распаковать архив, используйте команду

```
[user]$ gzip -d файл_архива
```

или

```
[user]$ gunzip файл_архива
```

Исходные файлы после сжатия удаляются, остается только архивный файл (файлы перемещаются в архив), а при разархивации удаляется архив.

Основные опции программы gzip

Опция	Значение
-h, --help	Вызов краткой помощи по использованию программы
-l, --list	Выдает имя файла, содержащегося в архиве, его объем и степень сжатия
-L, --license	Отображает номер версии и лицензию на программу
-N, --name	Сохранять (или восстанавливать) исходное имя и время создания файла
-n, --no-name	Не сохранять (не восстанавливать) исходное имя и время создания файла
-q, --quiet	Подавляет выдачу на экран предупреждающих сообщений
-r, --recursive	Рекурсивно обрабатывать подкаталоги (используется в случае, когда задан шаблон имен обрабатываемых файлов)
-S .suf, --suffix .suf	Добавить суффикс .suf к имени сжатого файла (вместо добавляемого по умолчанию суффикса gz; но учтите, что при разархивации файлов с суффиксами, отличными от gz, программа вас не поймет)
-t, --test	Протестировать архивный файл
-v, --verbose	Выдача дополнительных сообщений в процессе работы программы

-V, -- version	Отобразить версию программы
-1, --fast	Быстрое сжатие
-9, --best	Более высокая степень сжатия

Поскольку программа `gzip` не умеет сохранять в одном архиве несколько файлов, то обычно ее применяют для сжатия архивов, созданных программой `tar`.

Некоторые команды Linux для выполнения архивации

tar cf file.tar files – создать tar-архив с именем *file.tar* содержащий *files*

tar xf file.tar – распаковать *file.tar*

tar czf file.tar.gz files – создать архив tar с сжатием Gzip

tar xzf file.tar.gz – распаковать tar с Gzip

tar cjf file.tar.bz2 – создать архив tar с сжатием Bzip2

tar xjf file.tar.bz2 – распаковать tar с Bzip2

gzip file – сжать *file* и переименовать в *file.gz*

gzip -d file.gz – разжать *file.gz* в *file*

Изучение теоретических сведений.

Загрузка виртуальной операционной системы Linux.

Выполнение практических заданий согласно методическим указаниям по варианту

Восстановление прежних настроек.

Завершение работы с операционной системой Linux

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Ответы на вопросы преподавателя.

Лабораторная работа №16.

Изучение эмуляторов операционных систем MS DOS, Windows, LINUX

Цель работы: научиться создавать виртуальный жесткий диск и виртуальную

машину с помощью MS Virtual PC, VirtualBox.

Изучение теоретического материала.

Работа с командами в среде MS VirtualPC и VirtualBox по установке и настройке различных ОС

Установить главный контроллер домена

Подсоединить компьютер к домену

Установить соединение с доменом в качестве пользователя

Изменить политику безопасности

Установка ОС Ubuntu на виртуальную машину.

При загрузке виртуальной машины появится первичный диалог (см. рис. 17), в нем выберите язык программы инсталляции и нажмите «Enter».

В следующем меню (см. рис. 18) выберите «Установить Ubuntu».

Установка проходит в 6 шагов:

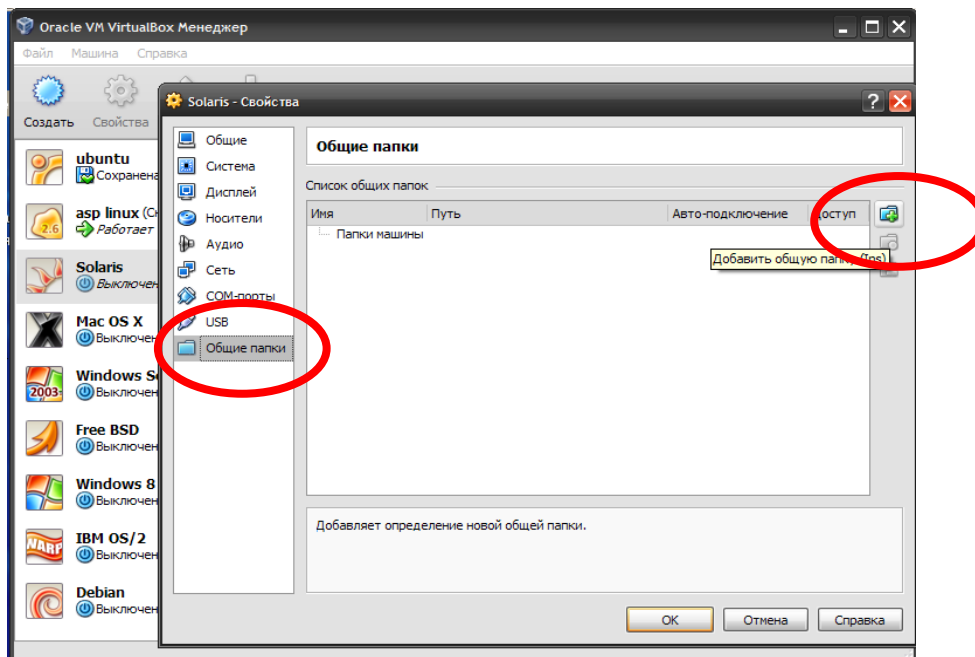
- выбор основного языка ОС;
- выбор местоположения;
- выбор раскладки клавиатуры;
- подготовка жесткого диска;
- указание параметров пользователя и компьютера, логина и пароля (указывать необходимо);
- проверка параметров установки.

При прохождении первых четырех шагов, параметры, указанные в них можно оставлять без изменений.

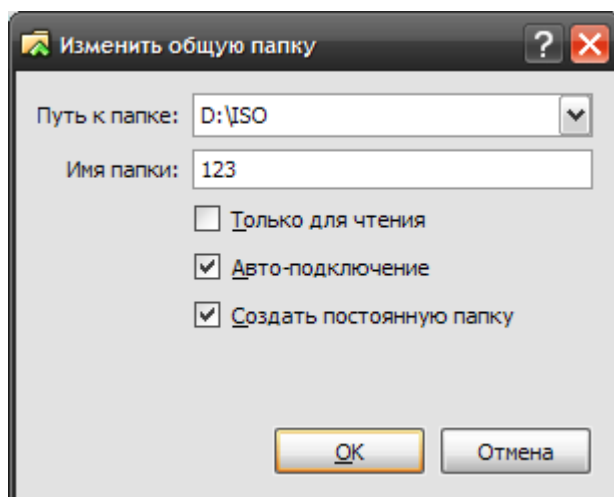
После окончания установки перезагрузите виртуальную машину.

Создание общей папки в VirtualBox.

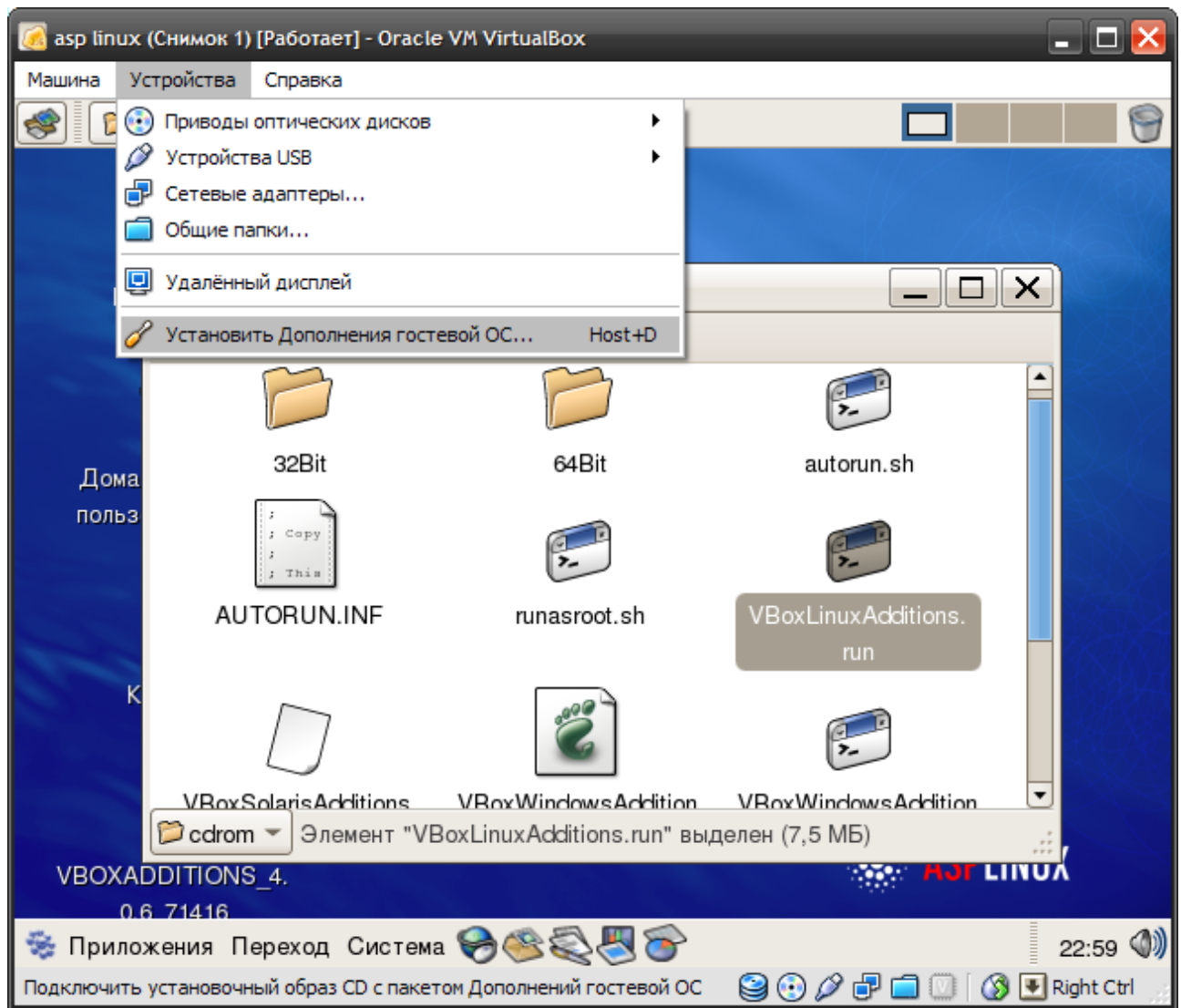
- 1. В VirtualBox** создайте виртуальную машину с ОС ASP LINUX, добавьте для неё общую папку



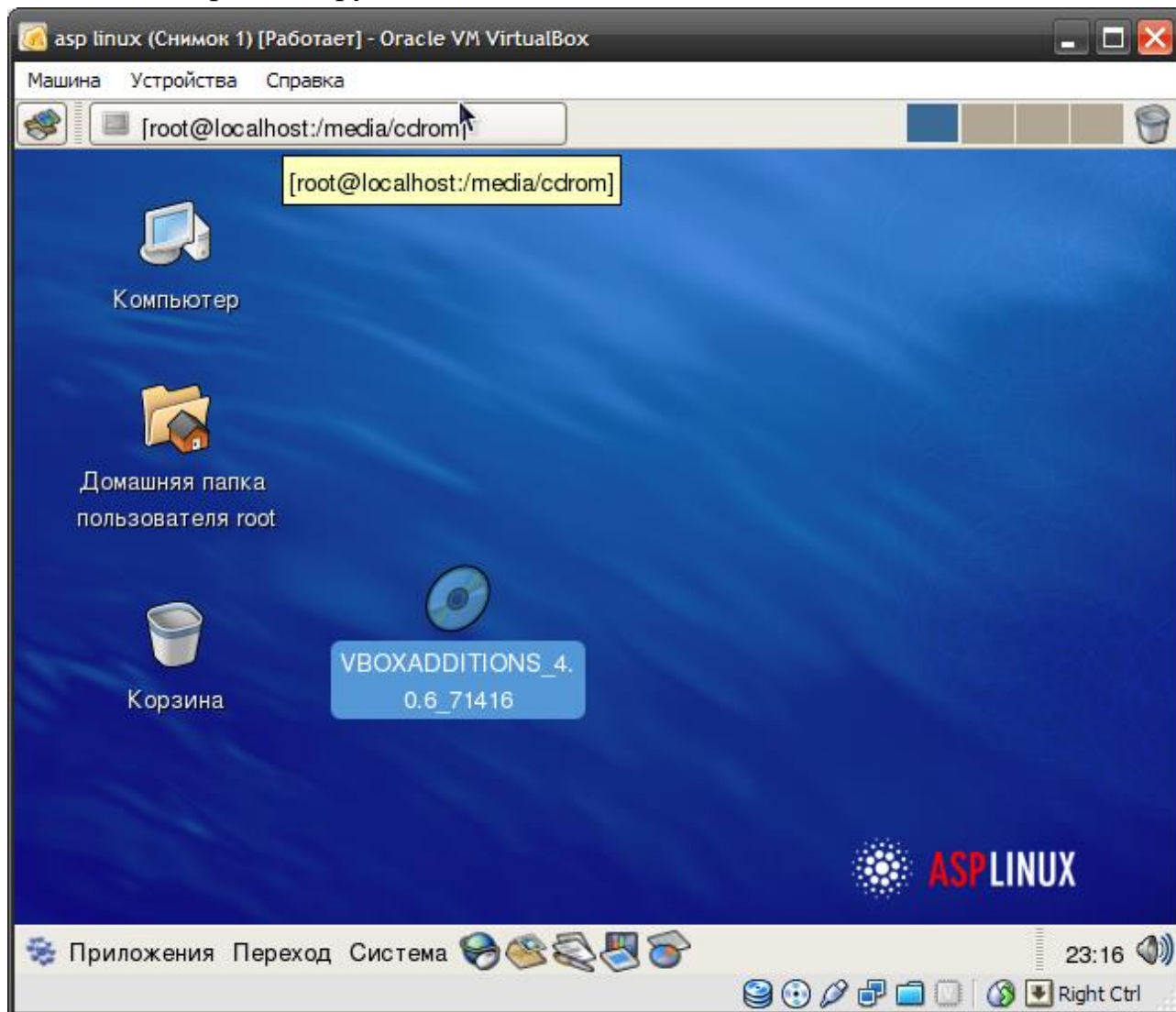
Укажите путь к любой папке Windows и дайте название вашей общей папке в Linux.



2. В **VirtualBox** запустите виртуальную машину с ОС ASP LINUX (войти в систему лучше под учетной записью root).
3. Установите дополнения гостевой ОС: в меню "Устройства" виртуальной машины выберите пункт "Установить дополнения гостевой ОС" (см. рис)



После этого примонтируется CD диск с дополнениями:

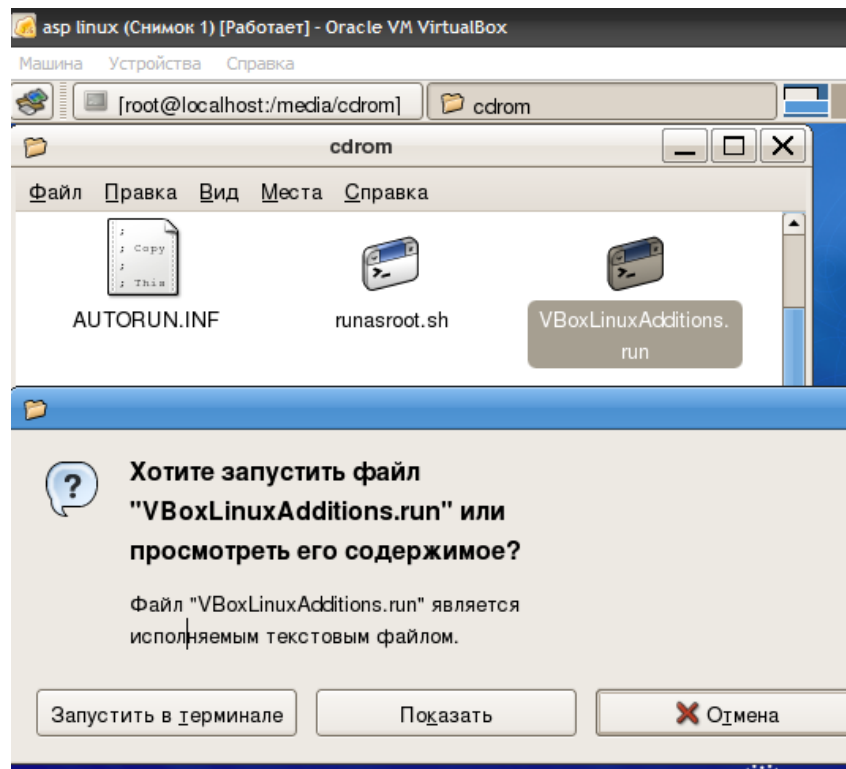


4. Запустите с CD диска VBoxLinuxAdditions.run из под root в терминале:
- откройте терминал;
 - в папку media в корневом каталоге монтируются устройства; отобразите её содержимое;
 - перейдите в папку cdrom, можете просмотреть её содержимое;
 - если вход выполнен не под учётной записью root, вы можете выполнять команды, требующие прав администратора, используя команду sudo. Введите в терминале:

```
sudo su
```

```
sh ./VBoxLinuxAdditions.run
```

- I. запуск соответствующего файла можно выполнить иначе, найдя его на рабочем столе, при условии открытия диалогового окна с выборкой запуска терминала, так как файлы типа .run не являются исполняемыми:



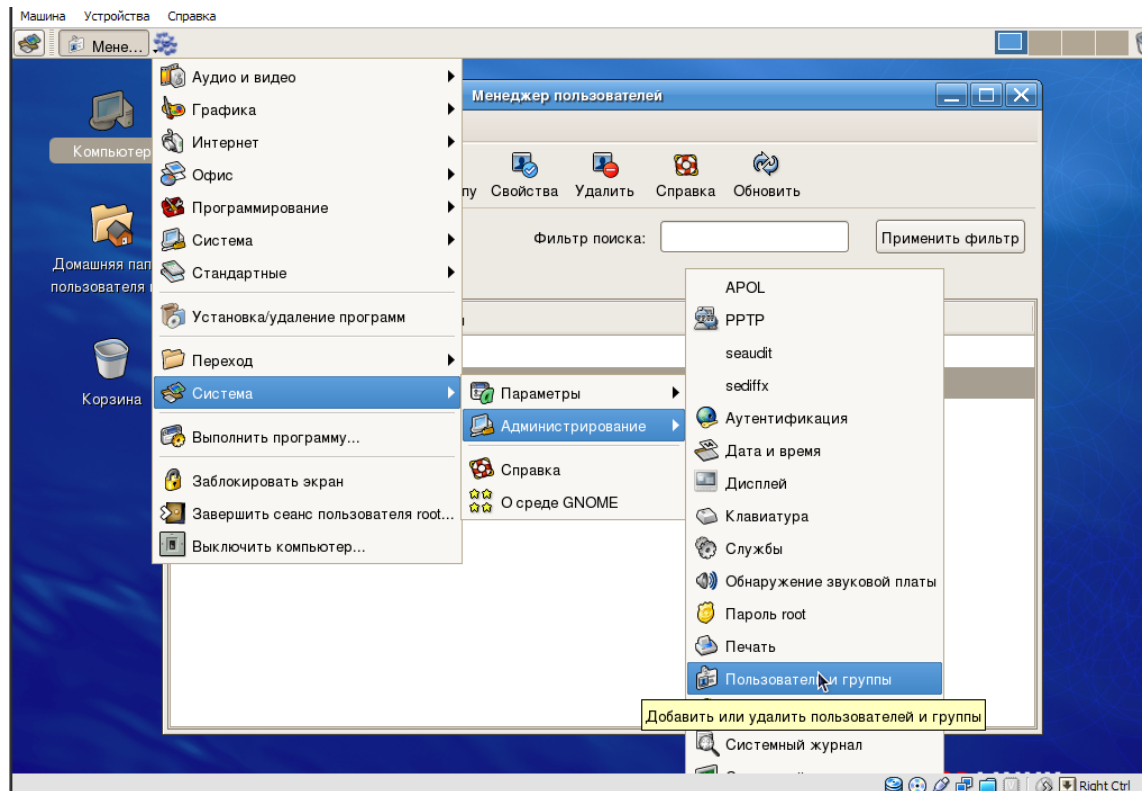
5. Добавьте вашего пользователя в группу vboxsf “вручную” с помощью меню либо через терминал:

```
sudo gpasswd -a $USER vboxsf
```

или

```
sudo usermod -a -G vboxsf $USER
```

ИЛИ



6. В гостевой системе создайте (удобнее всего в домашнем каталоге) папку, и назовите её 12, в которую будет смонтирована гостевая папка из основной ОС “Свяжите” ваши папки: `mount -t vboxsf Users /root/12`
7. Перезагрузите виртуальный компьютер.
8. Проверьте наличие общей папки, запишите путь к ней.

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Лабораторная работа №17.

Сетевое администрирование в виртуальной машине

Цель работы: закрепить навык работы с виртуальными машинами Microsoft Virtual PC; научиться настраивать сетевые параметры компьютера; изучить утилиты диагностики TCP/IP.

Краткие теоретические сведения

Настройки виртуальных машин

- **File Name** – здесь вы можете изменить имя файла настроек виртуальной машины (соответственно, изменится и имя самой машины).
- **Memory** – здесь вы можете изменить количество физической памяти хоста, выделяемой гостевой системе.
- **Hard Disk** – здесь вы можете изменить уже имеющийся или добавить новый виртуальный жесткий или Floppy-диск. При нажатии на кнопку «Virtual Disk Wizard» откроется мастер создания и изменения дисков. При создании нового диска этот мастер позволяет выбрать один из четырех типов дисков:
- **Undo Disks** – если поставить галку напротив этого пункта, то все изменения, которые произошли на дисках во время работы с виртуальной машиной, в конце сеанса работы с ней можно будет либо подтвердить, либо отменить. Включение таких дисков потребует наличие дополнительного места на физическом жестком диске.
- **CD/DVD Drive** - позволяет смонтировать физические CD и DVD приводы в виртуальную машину. Отметьте чекбоксы у тех приводов, которые вы хотите подключить к виртуальной машине.
- **Floppy Disk** – отключите автообнаружение флоппи-дисков, если вам не требуется их использование.
- **COM и LPT**. Эти пункты позволяют подключить порты COM и LPT в виртуальную машину и пользоваться устройствами, подключенными к ним.
- **Sound**. Позволяет проигрывать звук в виртуальной машине, используя физический аудиоконтроллер хоста. Уберите галку напротив этого пункта, если вам не требуется такая возможность.
- **Hardware Virtualization**. Позволяет виртуальной машине использовать инструкции аппаратной виртуализации вашего физического

процессора. Рекомендуется оставлять эту галку включенной, если такая возможность аппаратной виртуализации поддерживается процессором.

- **Mouse.** Интеграция указателя мыши с гостевой системой. Такая возможность появляется с установкой Virtual Machine Additions.
- **Shared Folders.** Эта опция позволяет создавать общие папки для обмена файлами между гостевой и хостовой ОС и доступна только при установленных Virtual Machine Additions. Создавать папки можно только при включенной гостевой системе. Для создания папки нажмите кнопку Share Folder и выберите папку в хостовой ОС, которую вы хотите подключить к гостевой системе. После создания, папка будет доступна как сетевой диск в гостевой системе.
- **Display.** Здесь вы можете выбрать поведения экранных настроек гостевой системы и, в частности, возможность старта в полноэкранном режиме, настройки разрешения и окна консоли гостевой системы. Кстати, комбинация клавиш Alt-Enter, во время работы с гостевой системой, позволяет переходить в полноэкранный режим и обратно.
- **Close.** Здесь можно выбрать действие, автоматически совершаемое при закрытии гостевой системы, а также отредактировать меню, возникающее при выборе пункта «Close».

Утилиты для диагностики TCP/IP

- **Arp** — утилита преобразовывает IP-адрес в физический адрес сетевого адаптера, используемый протоколом ARP (Address Resolution Protocol).
- **Finger** — отображение информации о пользователе на указанном компьютере, поддерживающем службу finger.
- **Hostname** — отображение имени локальной системы.

- **Ipconfig** — вывод информации о текущей конфигурации протокола TCP/IP, включая подробную информацию о сервера DNS и т.д. Кроме того, утилита может применяться для получения и освобождения арендованных адресов, предоставляемых службой DHCP.
- **Nbstat** — отображение конфигурационных данных протокола TCP/IP и списка текущих NBT-соединений (NetBIOS через TCP/IP).
- **Netstat** — отображение информации о протоколе TCP/IP и текущих сетевых соединений системы.
- **Ping** — используется для проверки способности удаленного компьютера принимать пакеты по протоколу TCP/IP.
- **Route** — применяется для поддержки и отображения таблиц маршрутизации.
- **Tracert** — используется для просмотра пути, по которому пакеты данных передаются удаленной системе.
- **Pathping** — утилита, которая входит в состав Windows и являет собой нечто среднее между утилитами ping и tracert.

Для получения дополнительной информации введите в командной строке интересующую команду с параметром *-?*, например, *netstat -?*.

Общие сведения

При настройке подключения компьютера к сети нужно задать следующую информацию:

- IP адрес вашего компьютера;
- IP адрес сети;
- широковещательный IP-адрес;
- имя домена, в который будет включен ваш компьютер;
- маску подсети;
- IP адрес маршрутизатора (router);
- IP адрес сервера имен (DNS-сервера).

Ход работы

Изучение теоретического материала.

Работа с системными командами в гостевой операционной системе в среде MS VirtualPC

Изучение утилит диагностики TCP/IP

Консоль Active Directory Users and Computers (Active Directory - пользователи и компьютеры) предназначена для создания, удаления или временного отключения глобальных учетных записей пользователей на основном контроллере домена и с локальными учетными записями на любом компьютере домена.

Зарегистрировать пользователей и составить «Шаблон планирования учетных записей».

Копировать шаблон пользовательского профиля на сервер.

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Лабораторная работа №18.

Работа с системными командами. Реестр операционной системы

Цель работы: ознакомиться с главным средством системного администрирования ОС Windows – системным реестром, возможностями управления записями реестра с помощью редактора системного реестра (regedit.exe), командного файла реестра (*.reg) и утилиты системного администратора Poledit

Краткие теоретические сведения

База данных реестра состоит из файлов типа .DAT. Файл SYSTEM.DAT описывает индивидуальную конфигурацию компьютера и содержит информацию о приложениях, установленных на компьютере. Файл USER.DAT предназначен для хранения пользовательских установок, таких, как конфигурация рабочего стола.

Файл USER.DAT может находиться как собственно на компьютере, так и на файловом сервере. Файл SYSTEM.DAT, записи которого соответствуют аппаратному обеспечению компьютера, хранится только на системном диске ПК.

Примечание. Файлы реестра имеют набор файловых атрибутов «Только чтение», «скрытый», «системный». Эти файлы можно увидеть в проводнике или DOS-приложениях Norton Commander или FAR с установленной опцией «Показывать скрытые и системные файлы».

Редактор базы данных регистрации – утилита *regedit.exe* – находится в главном каталоге Windows. Запуск утилиты приводит к появлению окна «Редактор реестра», которое содержит пять корневых записей реестра, называемых ключами (HKEY). Каждая запись хранит некоторую область данных реестра.

1. HKEY_CLASSES_ROOT специфицирует имя типа файла, основанное на расширении каждого установленного на ПК приложения. Например, если в Windows инсталлирован редактор Word for Windows, то файлы с расширением DOC ассоциируются с редактором Word.

2. HKEY_CURRENT_USER содержит информацию о пользователе, который в данный момент зарегистрирован в системе. Этот ключ включает идентификатор пользователя и пароль. Когда пользователь регистрируется на рабочей станции Windows, то информация о нем копируется в ключ HKEY_CURRENT_USER из записей HKEY_USERS.

3. HKEY_LOCAL_MACHINE хранит данные об аппаратном обеспечении данного компьютера.

4. HKEY_USERS поддерживает список всех пользователей и их пароли, которые содержатся в списке паролей локального компьютера.

Информация об индивидуальных установках пользователей также хранится в этом ключе.

5. HKEY_CURRENT_CONFIG хранит данные, которые определяют текущую конфигурацию рабочего стола каждого пользователя, последние используемые документы и другую информацию о текущих установках пользователей.

Редактор системного реестра. Утилита regedit.exe

Утилита редактора системного реестра – regedit.exe – является графическим инструментом, предназначенным для изменения параметров (переменных) системного реестра. Окно редактора поделено на две части и по форме напоминает проводник. В левой части окна отображаются пять, как правило, свернутых корневых записей, которые можно развернуть точно так же, как дерево каталогов в проводнике. Каталог в терминах редактора реестра называют разделом. Листьями этого дерева являются ключи. Ключи отображаются в правой части окна редактора. Ключ имеет атрибуты «имя», «тип» и «значение».

Как и в проводнике, *имя* ключа состоит из последовательности разделов, отделенных символом «\».

Ключ может иметь *тип «строковый параметр»* (REG_SZ), «двоичный параметр» (REG_BINARY) или «параметр DWORD» (REG_DWORD).

В зависимости от типа ключа его значения могут быть различны. В лабораторной работе мы будем использовать «параметр DWORD» и два его значения – 0x0000 0001(1) и 0x0000 0000(0), которые соответственно разрешают (включают) и отменяют (выключают) действие ключа.

Изучение теоретического материала.

Работа с редактором реестра regedit.exe в гостевой операционной системе в среде MS VirtualPC

Задание №1.

С помощью команды «найти» кнопки «пуск» найдите и запустите утилиту regedit.exe.

Возможно, после запуска редактора появится сообщение «Редактирование реестра запрещено администратором системы». В этом случае

работу нужно выполнить в гостевой ОС посредством виртуальной машины.

После запуска утилиты должно появиться окно с заголовком «*Редактор реестра*», в котором в свернутом виде будут представлены все пять корневых записей. Информация, хранящаяся в этих записях, чрезвычайно обширна, и изучить ее за время выполнения лабораторной работы не представляется возможным. Поэтому редактирование реестра мы разберем на примере ограничения настроек экрана.

В окне редактора системного реестра откройте раздел:
HKEY_CURRENT_USERS\Software\Microsoft\Windows\Current
Version\Policies\System

Если после открытия раздела *Policies* Вы не увидите раздел *System*, то создайте его по следующему правилу: поместите курсор на раздел *Policies*, отметьте его, а затем в меню «Правка» выполните директивы «Создать», «Раздел». Система создаст раздел с именем *Новый раздел #1*. Поменяйте его имя на *System*.

В разделе *System* создайте нижеописанные ключи (если они отсутствуют) с помощью меню «Правка» и директив «Создать», «Параметр DWORD».

NODISPSettingsPage
NODISPBackgroundPage
NODISPAppearancePage
NODISPScrSavPage
NODISPCPL

Перечисленные ключи предназначены для ограничения доступа к органам настроек экрана. Эти ключи, установленные с параметром *dword:00000001*, закрывают соответствующие им вкладки окна «Свойства: экран».

Задание №3

Создайте файлы `nodisp.reg` и `disp.reg` и проверьте правильность их действия на практике. Проверку проведите с открытым окном редактора `regedit`. Запуск командного файла должен привести к изменению значения параметра *NoDispCPL* в окне редактора реестра.

Задание №4.

1. Создайте и выполните командные файлы системного реестра, закрывающие и открывающие доступ к настройкам органов управления Windows согласно номеру варианта, заданному в табл. 1. После внесения изменений убедитесь, что соответствующий параметр появился в нужном месте реестра и его значение соответствует заданному.

В приложении приведен список ключей и необходимые комментарии к ним. Эти ключи могут понадобиться Вам при выполнении практической работы. Дополнительную информацию по настройкам системного реестра Windows можно найти в файле **reghelp.hlp** (или **справочник_по_реестру.chm** в папке **Операционные системы**).

2. Скройте изображение стрелочки в углу значка ярлыка (**reghelp.hlp** → Ярлыки)

Отключите отправку отчетов в средства удаления вредоносных программ в корпорацию Майкрософт (**reghelp.hlp** или **справочник_по_реестру.chm** в папке **Операционные системы**) → Средство удаления вредоносных программ)

3. Измените порог выдачи предупреждения о недостатке свободного места на диске

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

Ответы на вопросы преподавателя.

Лабораторная работа №19.

Основы работы с BIOS Setup Utility

Цель работы: отработать навыки работы с BIOS Setup с помощью программы Microsoft Virtual PC.Ход работы

Краткие теоретические сведения

1. Что такое BIOS

BIOS (Basic Input/Output System — базовая система ввода/вывода) — это программа для первоначального запуска компьютера, настройки оборудования и обеспечения функции ввода/вывода.

Другими словами, можно сказать, что BIOS — основа работы вычислительной системы, так как отвечает за самые базовые функции компьютера (аналогично системе рефлексов у человека).

BIOS записывается в микросхему flash-памяти, которая расположена на системной плате. Именно она наиболее широко известна среди пользователей, но BIOS присутствуют практически у всех компонентов компьютера: у видеоадаптеров, сетевых адаптеров, модемов, дисковых контроллеров, принтеров. BIOS материнской платы отвечает за инициализацию (подготовку к работе), тестирование и запуск всех ее компонентов.

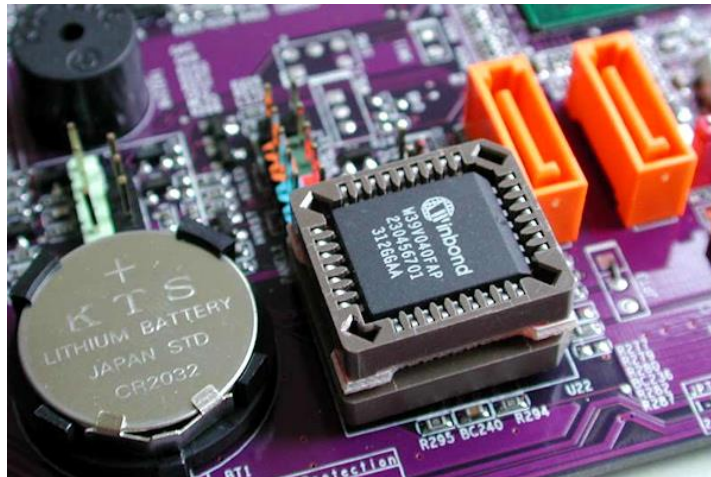
Операционная система и прикладные программы работают с аппаратным обеспечением компьютера посредством BIOS, которая переводит понятные пользователю команды операционной системы на язык, понятный компьютеру.

2. BIOS материнской платы

BIOS — это набор микросхем памяти, расположенных на материнской плате.

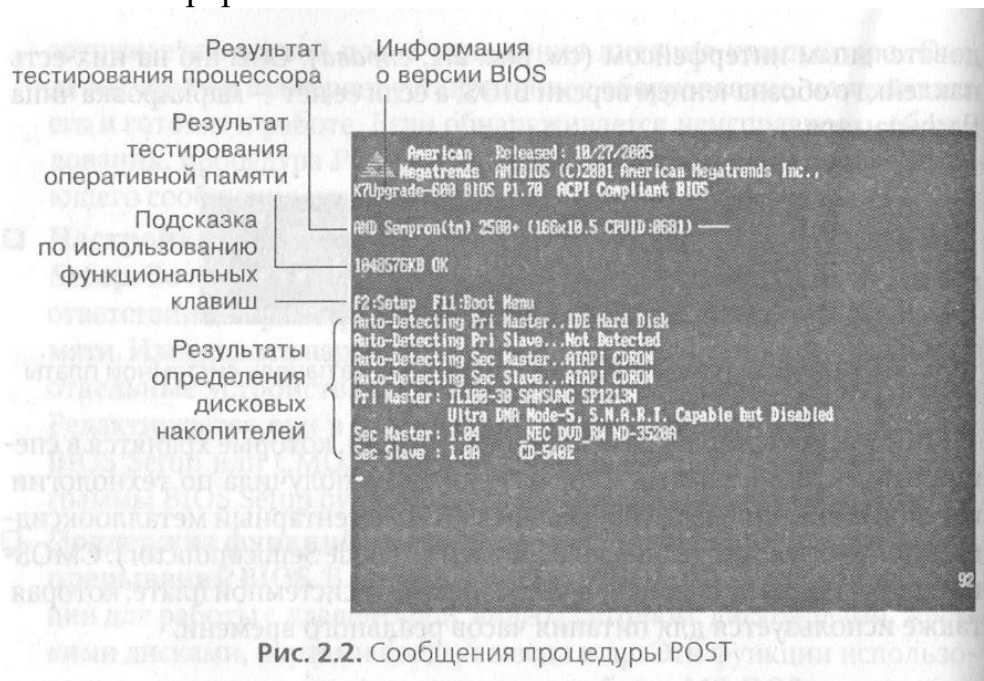
BIOS также называют аппаратно встроенным ПО.

Если заглянуть под крышку системного блока, то на материнской плате можно обнаружить микросхему с голографической наклейкой с надписью и логотипом, означающим производителя BIOS. Рядом обязательно будет круглый аккумулятор, питающий микросхему CMOS (Complementary Metal Oxide Semiconductor — энергозависимая память, применяемая для хранения установок BIOS).



3. *Функции BIOS:*

- запуск компьютера и процедура самотестирования **POST (Power-On SelfTest)**
- настройка параметров системы с помощью программы BIOS Setup
- поддержка функций ввода/вывода с помощью программных прерываний BIOS.



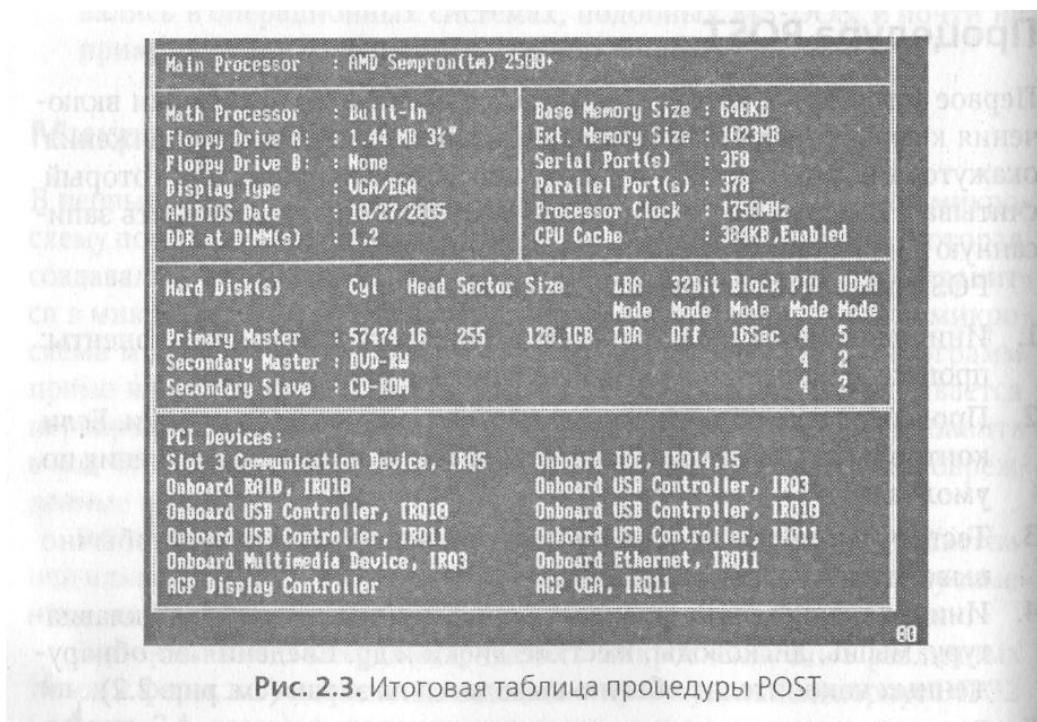


Рис. 2.3. Итоговая таблица процедуры POST

4. BIOS Setup Utility

Среди программ, содержащихся в BIOS, имеется **программа настройки параметров BIOS Setup Utility**, которая позволяет изменять данные, хранящиеся в памяти CMOS, с помощью системы меню.

Для обеспечения правильной работы операционной системы и прикладных программ с помощью BIOS Setup Utility вводятся параметры всех компонентов компьютера, начиная от оперативной памяти и рабочей частоты процессора и заканчивая режимом работы принтера и других периферийных устройств. Правильно настроив содержимое BIOS вашего компьютера, можно увеличить производительность его работы до 30%.

Замечание: неосторожные действия пользователя, как правило, не могут привести к физическому повреждению компьютера — он может лишь перестать загружаться. Это легко исправить. Современные BIOS имеют довольно обширные средства автоконфигурирования, поэтому роль пользователя в установке «правильных» параметров можно свести к минимуму. В последнее время в программе установки параметров появился пункт «Загрузить оптимизированные параметры». Выбор этого пункта позволяет пользователю установить «параметры по умолчанию» для имеющегося оборудования.

5. Как войти в BIOS Setup Utility

Программа установки параметров BIOS Setup Utility непосредственно недоступна пользователю во время работы компьютера (*существуют программы для ОС Windows, позволяющие работать с BIOS после загрузки ОС в оконном режиме*). Вход в BIOS Setup Utility обычно выполняется путем нажатия клавиши Del во время загрузки компьютера. Так же встречаются версии BIOS, вход в настройки которой выполняется с использованием других клавиш или их сочетаний.

В данной лабораторной работе для входа в BIOS будет использован наиболее распространенный вариант (клавиша Del).

6. Virtual PC и AMIBIOS

Для изучения основ работы в CMOS Setup Utility используйте MS Virtual PC. Выбор данной виртуальной машины не случаен: именно в ней эмуляция BIOS Setup реализована практически полноценно (версия AMIBIOS 2.10), хотя и не имеет широких возможностей настройки, реализованных для реального BIOS.

Программа Microsoft Virtual PC предназначена для создания полностью виртуального компьютера. Она позволяет выделить часть места на жёстком диске, создать в этой области так называемую виртуальную машину, установить на неё отдельную операционную систему и необходимые программы и безбоязненно экспериментировать с этим компьютером, не нанося вред основному реальному компьютеру.

!!! При подготовке к выполнению лабораторной работы добавьте в Virtual PC новую виртуальную машину, но не устанавливайте на нее операционную систему, так как для доступа к BIOS Setup Utility не требуется дополнительное программное обеспечение.

Подготовка к выполнению работ:

1. Запустить программу **Virtual PC** (*Пуск — Все программы — Microsoft Virtual PC*).

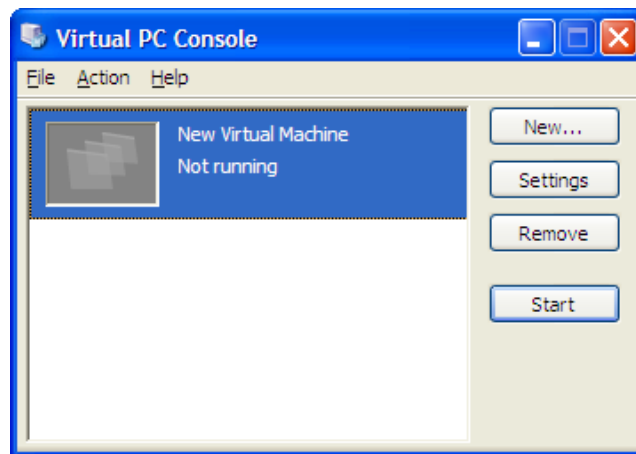


Рис. 1. Интерфейс виртуальной машины Virtual PC

2. Добавить новую виртуальную машину. Для этого нажать на кнопку «**New...**». Появится мастер создания виртуальной машины.

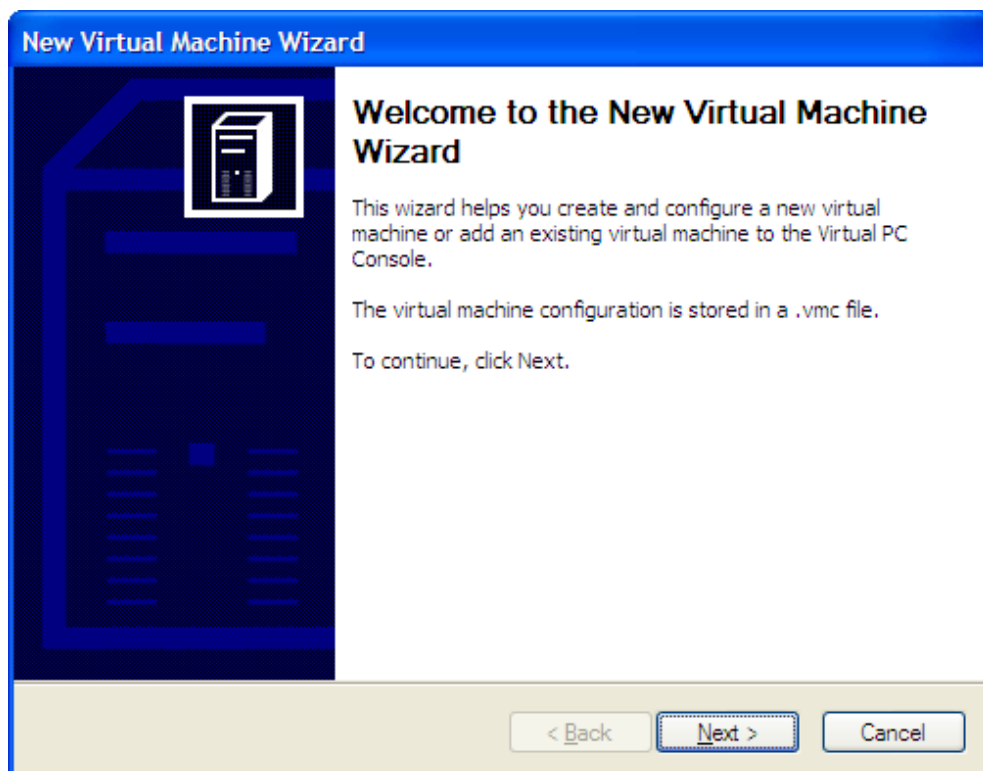


Рис. 2. Мастер создания и конфигурирования виртуальной машины

Нажать кнопку «**Next**» («Далее»).

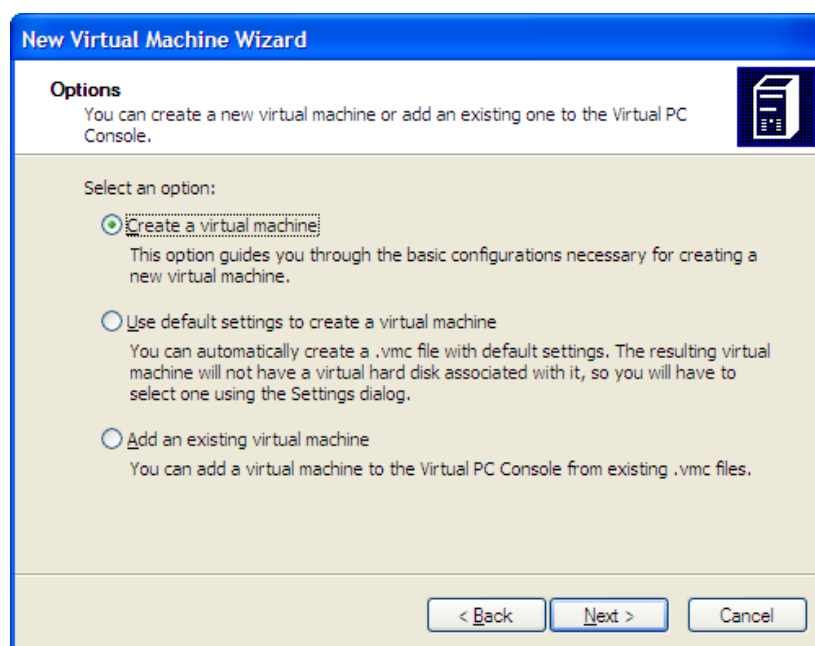


Рис. 3. Мастер создания и конфигурирования виртуальной машины

Выбрать «**Create a virtual machine**». Нажать «**Next**».

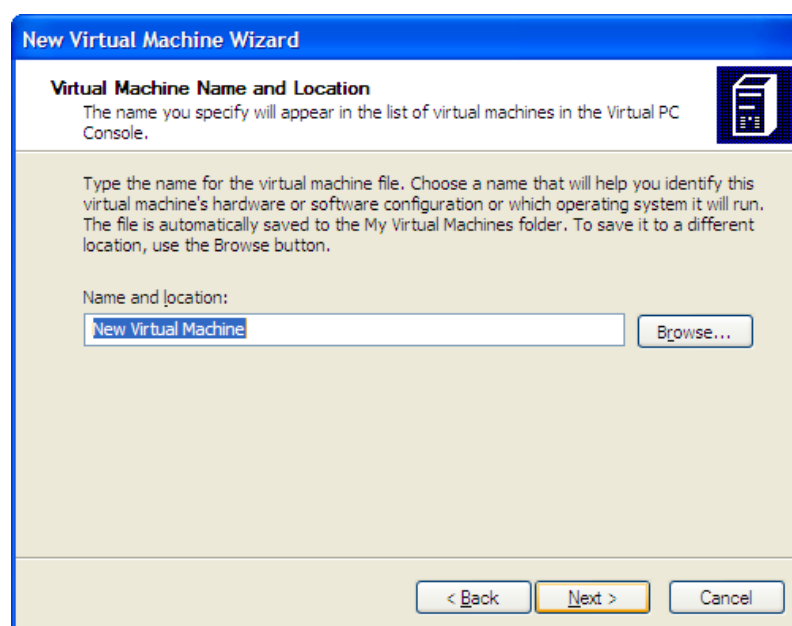


Рис. 4. Мастер создания и конфигурирования виртуальной машины

Ввести название виртуальной машины. Например, «**For BIOS ONLY**». Нажать кнопку «**Next**».

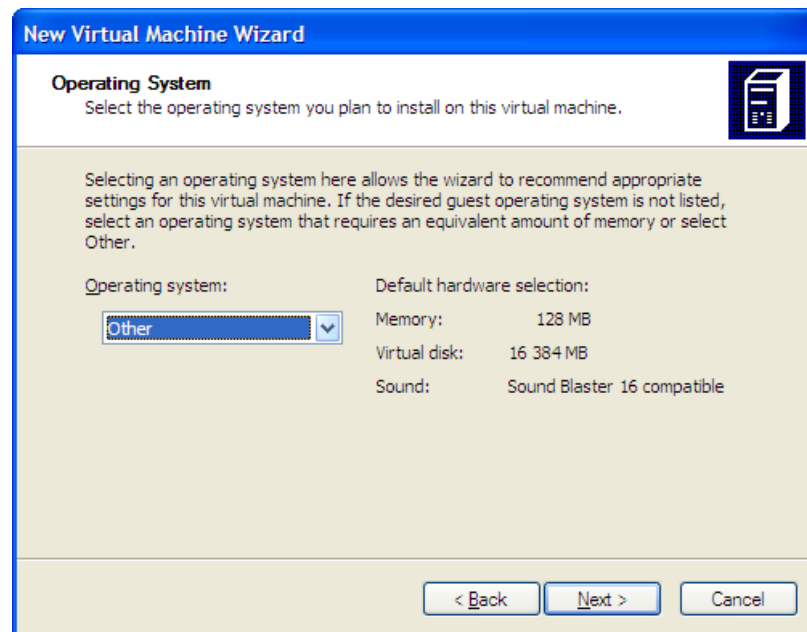


Рис. 5. Мастер создания и конфигурирования виртуальной машины

Тип операционной системы оставить «**Other**». Нажать кнопку «**Next**».

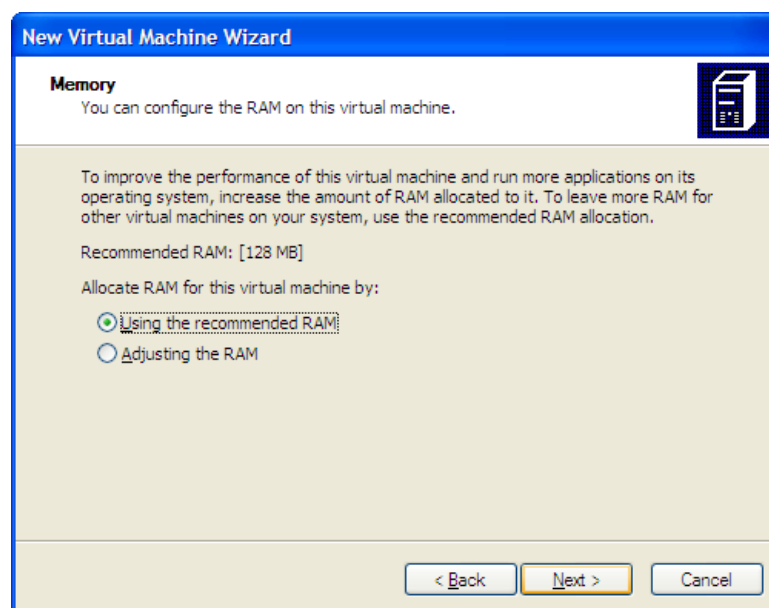


Рис. 6. Мастер создания и конфигурирования виртуальной машины

Выбрать «**Use the recommended RAM**». Нажать кнопку «**Next**».

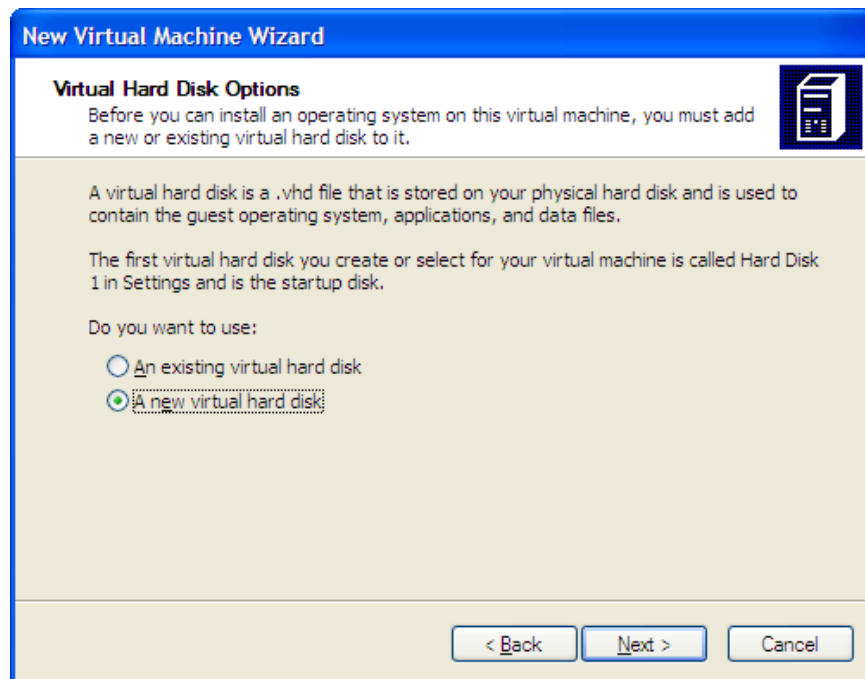


Рис. 7. Мастер создания и конфигурирования виртуальной машины

Выбрать «**A new virtual hard disk**», что означает создание нового виртуального жесткого диска. Нажать «**Next**».

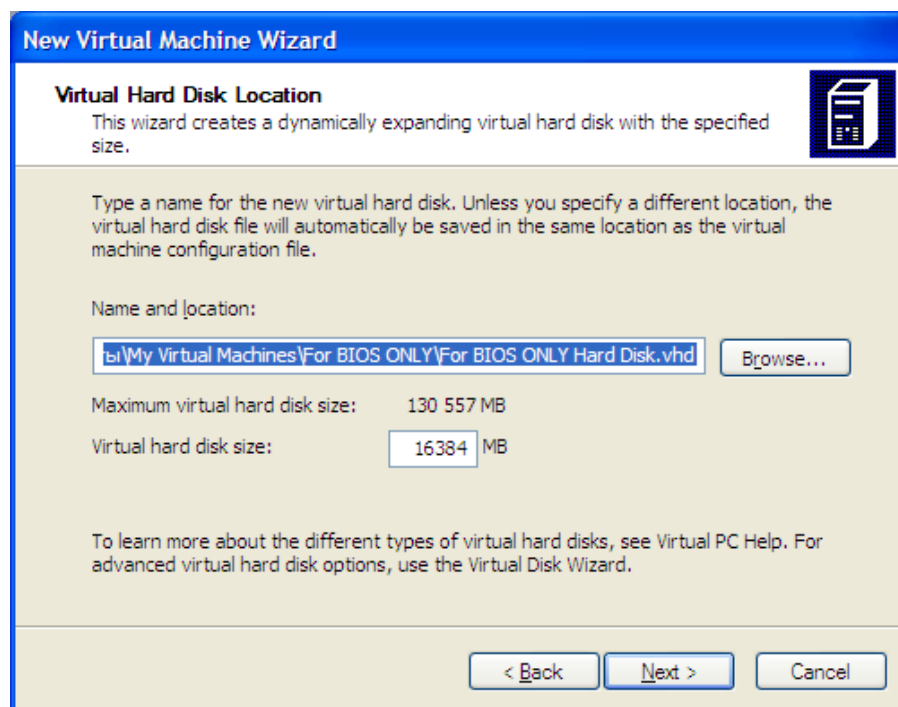


Рис. 8. Мастер создания и конфигурирования виртуальной машины

Оставить предложенные параметры, нажать «**Next**».

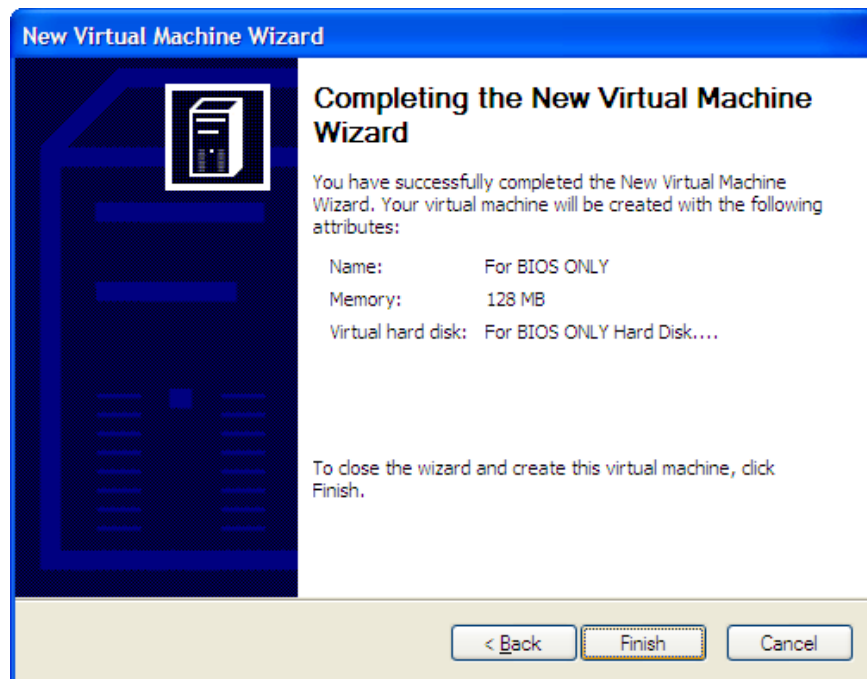


Рис. 9. Мастер создания и конфигурирования виртуальной машины

Нажать «**Finish**». Таким образом, в списке виртуальных машин появится новая — «**For BIOS ONLY**»

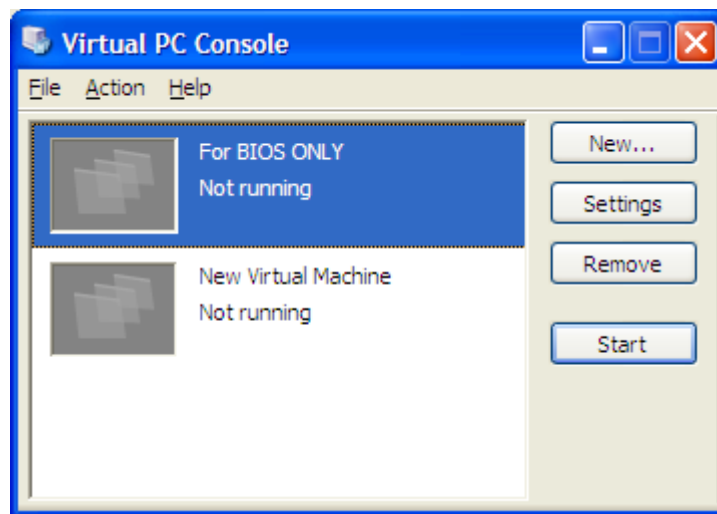


Рис. 10. Список виртуальных машин

3. Чтобы запустить виртуальную машину следует нажать на кнопку «**Start**». Как только начнется запуск, нажмите на клавишу **Delete**. Обычно при загрузке компьютера после нажатия именно этой клавиши происходит вход в настройки BIOS.
4. После загрузки BIOS Setup Utility экран примет следующий вид (Рис. 11):

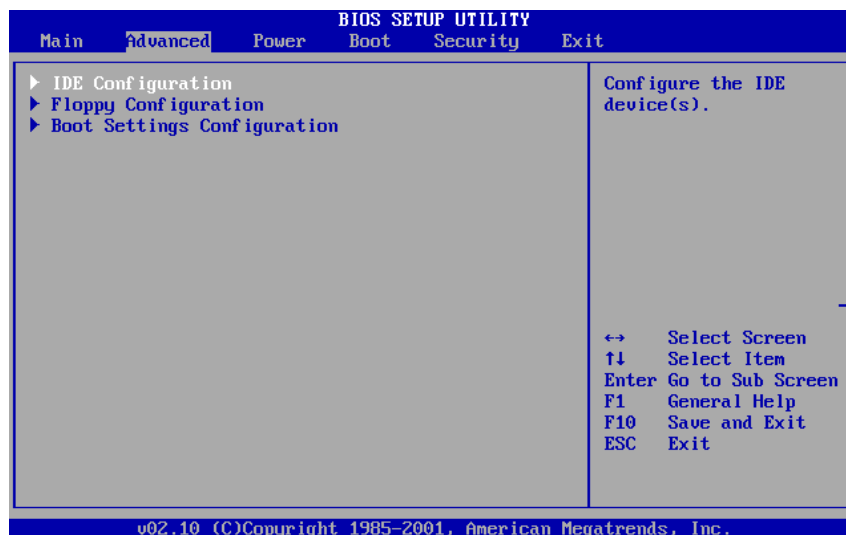


Рис. 11. Интерфейс BIOS Setup Utility после запуска

Чтобы скопировать содержимое окна, необходимо в главном меню виртуальной машины выбрать пункт *Edit — Select All*, а затем *Edit — Copy*.

Задания для самостоятельного выполнения

Внимание: после выполнения данной работы преподавателю необходимо предъявить отчет, в который следует включить снимки экрана при выполнении заданий и наименование соответствующего задания.

Не забывайте сохранять результаты выполнения работ на своем носителе информации!

1. Узнайте тип и версию BIOS.
2. Узнайте дату создания BIOS.
3. Установленный и максимально поддерживаемый размер памяти.
4. Определите параметры накопителей, подключенных к каналам стандартного IDE-контроллера.
5. Определите текущий порядок опроса накопителей при загрузке.
6. Измените порядок опроса накопителей при загрузке так, чтобы в первую очередь опрашивался CDROM, затем жесткий диск. Остальные носители не опрашиваются.
7. Установите пароль **qwaszx** к BIOS Setup Utility.
8. Оформите отчет с подробным описанием действий и формулирование вывода о проделанной работе.

Лабораторная работа №20.

Установка и анализ функциональности различных операционных систем

Цель работы: ознакомиться с возможностями настройки виртуальных машин в приложении VirtualBox, выполнить установку.

Изучение теоретического материала.

Выполнение последовательно настроек VirtualBox

Организация взаимодействия хостовой и гостевой операционных систем

Выполнение описания любого дистрибутива ОС Linux по образцу:

Системные требования: Intel/AMD не менее 1.4 ггц

Оперативной памяти: Минимум 1 гб . Желательно от 2.5 Гб

Жёсткий диск: минимум 11гб.

Видеокарта (GPU) для 3д и игр - Nvidia (из коробки будут работать от 7600 до GTX)

Для других видеокарт есть встроенный универсальный драйвер Gallium

Архитектура: amd64 (годится и для AMD и для Intel)

Таблэтка: Не требуется

Язык интерфейса: Мультиязычный (русский присутствует)

Описание: Ядро: 3.6.0, 3.5.0-17

Образ сочетает все известные плюсы Ubuntu 10.10 + некоторые особенности:

Компоненты системы и драйвера:

+Linux-kernel 3.6.0

(а значит больше драйверов для устройств, полную информацию о Linux kernel 3.6.0 читайте в интернете)

+MATE Desktop 1.4.0 (классический рабочий стол)

+драйвер nvidia 310.14 (бета версия)

+Compiz + Emerald 0.9.5(собиран вручную, настроен, но выключен по умолчанию)

+подключен репозиторий Quantal (!)

т.е. можно аккуратно устанавливать приложения из Ubuntu 12.10

его использование на данный момент не вызывает проблем.

поставлены все обновления на текущий момент.

+возвращено отображение Wine в меню - линк wine из прочих в mate-applications-merged (баг оригинального mint13) (спасибо за исправление Mindgames!)

Присутствуют следующие приложения:

Wine 1.5.15

добавлена gwenview для того чтобы можно было удобно работать с фото и смотреть эскизы

в настройках флеш плеера для firefox было включено автосогласие на запрос о видеозвонках и микрофоне для сайтов Вконтакте и одноклассники.

GIMP 2.8.2 (теперь на русском, исправлена комбинация клавиш для "сохранения как" обратно на CTRL-S)

Audacious 3 , Nero 4, VLC 2, Skype 4, Linux Multimedia Studio

Total commander 7, Winrar 4

FTP+Pureadmin (наиболее удобный FTP)

Nx client, server 3.5 (удалённое управление другими компьютерами)

Браузеры Firefox 16, Chrome 22 , Opera 12

Поддержка таппинга для ноутбуков теперь включается через меню.

Adblock 2 для блокировки рекламы

Vkontakte downloader для музыки, фото , видео

Youtube downloader для загрузки видео

PhysX, DirectX ,Xaudio (Xact) , K-lite codec pack

.NET Framework 1.0, 2.0 , 4.0 (4.0 вроде как замещает 3.0)

Оформление отчета с подробным описанием действий и формулирование вывода о проделанной работе.

4. Литература

Основные:

1. Таненбаум Э. Современные операционные системы. - 4-е изд. СПб.: Питер, 2015г.
2. Сафонов В.О. Основы современных операционных систем -2-е изд. СПб.: НОУ "Интуит", 2016 г.
3. Батаев А.В., Налютин Н.Ю., Сеницын С.В. Операционные системы и среды. М.: Академия", 2014 г.
4. Олифер В.Г., Олифер Н.А. Сетевые операционные системы СПб.: Питер, 2010г.
5. Симонович С.В. Информатика. Базовый курс - 3-е изд. СПб.: Питер, 2012г.
6. Адельштайн Г., Любанович Б. Системное администрирование в Linux СПб.: Питер, 2013 г.
7. Левин А. Самоучитель полезных программ. — СПб.: Питер, 2011.

Дополнительные:

1. Алехин А.Р., Операционные системы. СПб.: Питер, СПб.: Питер, 2014 г.
2. Бормотов С., Системное администрирование на 100% СПб.: Питер, 2009 г.
3. Фишер Г., Родригес К.З., Смолски С. Linux. Азбука ядра СПб.: Питер, 2012г.
4. Ватаманюк А. Установка, настройка и восстановление Windows СПб.: Питер, 2010г.

Интернет ресурсы:

<http://www.ossite.ru> – дайджест мира операционных систем

<http://www.linux.ru> – портал просвещения в области свободных систем: Linux, FreeBSD и т.д.

<http://education.aspu.ru> – образовательный портал

<http://www.ossite.ru> – дайджест о современных операционных системах

<http://www.linux.ru> – портал для пользователей и разработчиков Linux

<http://linuxgid.ru> – портал для пользователей и разработчиков Linux

<http://www.winall.ru> – портал для пользователей и разработчиков Windows

Периодические издания:

«CHIP»;

«Компьютер пресс».