Федеральное агентство по рыболовству
БГАРФ ФГБОУ ВО «КГТУ»
Калининградский морской рыбопромышленный колледж

Утверждаю
Заместитель начальника колледжа по учебно-методической работе
А.И.Колесниченко

Подписано цифровой подписью: Колесниченко Андрей Игоревич
Причина: Я утвердил этот документ
Дата: 23 Июнь 2025 г. 11:32:55
DN: E=uc_fk@roskazna.ru, S=77 Москва, ИНН ЮЛ=7710568760, ОГРН=1047797019830, STREET="Большой Златоустинский переулок, д. 6, строение 1", L=г. Москва, C=RU, O=Казначейство России, CN=Федеральное казначейство

ОП.01 ОПЕРАЦИОННЫЕ СИСТЕМЫ И СРЕДЫ

Методические указания для выполнения практических занятий
по специальности
09.02.07 Информационные системы и программирование

МО-09 02 07-ОП.01.ПЗ

РАЗРАБОТЧИКИ	Кондратьев П.С.
ЗАВЕДУЮЩИЙ ОТДЕЛЕНИЕМ	Судьбина Н.А
ГОД РАЗРАБОТКИ	2025

Содержание

Перечень практических занятий

№ п/п	Практическое занятие	Кол-во часов
Тема 1 История, назначение и функции операционных систем		
1	Практическое занятие № 1 Использование сервисных программ поддержки интерфейсов. Настройка рабочего стола. Настройка системы с помощью Панели управления. Работа со встроенными приложениями	2
Тема 2 Архитектура операционной системы		
2	Практическое занятие № 2 Установка и настройка системы. Установка параметров автоматического обновления системы. Установка новых устройств. Управление дисковыми ресурсами.	2
Тема 3 Общие сведения о процессах и потоках		
3	Практическое занятие № 3 Диагностика и коррекция ошибок операционной системы, контроль доступа к операционной системе	2
Тема 4 Взаимодействие и планирование процессов		
4	Практическое занятие № 4. Управление процессами с помощью команд операционной системы для работы с процессами.	2
5	Практическое занятие № 5 Конфигурирование файлов. Управление процессами в операционной системе. Резервное хранение, командные файлы	2
Тема 5 Управление памятью		
6	Практическое занятие № 6 Управление памятью	2
Тема 6 Файловая система и ввод и вывод информации		
7	Практическое занятие № 7 Работа с программой «Файл-менеджер Проводник». Работа с файловыми системами и дисками.	2
8	Практическое занятие № 8 Работа с командами в операционной системе. Использование команд работы с файлами и каталогами. Работа с дисками. Работа с текстовым редактором. Работа с архиватором. Работа с операционной оболочкой	2
Тема 7 Работа в операционных системах и средах		
9	Практическое занятие № 9 Выполнение порядка установки операционной системы на ПК.	2
Тема 9 История и общая характеристика семейства операционных систем LINUX		
10	Практическое занятие № 10 Установка ОС Linux	2
Тема 10 Концепции UNIX		
11	Практическое занятие № 11 Работа с терминалом ОС UNIX	2
ИТОГО		22

Тема 1 История, назначение и функции операционных систем

Практическое занятие № 1 Использование сервисных программ поддержки интерфейсов. Настройка рабочего стола. Настройка системы с помощью Панели управления. Работа со встроенными приложениями

Цель:

- рассмотреть сервисные программы поддержки интерфейсов ОС;

- рассмотреть основные настройки рабочего стола ОС;

Оборудование: ПК, ОС Windows, MS Word

Краткие теоретические сведения:

В общем случае, конечно, следует говорить о *связи с внешней средой,* поскольку, например, при использовании ЭВМ в системах управления технологическими комплексами (производство, летательные аппараты, корабли и пр.) человек может быть исключен (полностью или частично) из контура управления и внешними устройствами ЭВМ будут *датчики* (скорости, высоты, давления, температуры) и *эффекторы* (приводы рулей, манипуляторы, сервомоторы вентилей и пр.).

Связь с пользователем, сокращенно поименованная здесь как связь с оператором, — как говорят англичане, last but not least — последняя в списке, но не по важности функция ОС.

Связь с пользователем включает:

: командный (или иной) интерфейс по управлению системными процессами в вычислительной системе (собственно функции оператора ОС). Пользователь (привилегированный) осуществляет запуск-останов программ, подключение - отключение устройств и прочие релевантные операции;

: интерфейс по управлению пользовательскими процессами (контроль состояния процесса, ввод-вывод данных в процесс / из процесса).

В состав *пользователей* в общем случае включаются следующие группы лиц, контактирующих с системой:

: администратор системы лицо или группа, отвечающая за сопровождение данных, назначение уровней доступа, включение/исключение пользователей;

: оператор системы, осуществляющий сопровождение вычислительного процесса,

: прочие пользователи (не обладающие привилегиями доступа к данным), в том числе:

· операторы подготовки данных (ОПД) — персонал, осуществляющий ввод данных с рабочих листов или документов, на основе соответствующих инструкций, в среде специальных программных интерфейсов,

· интерактивные пользователи (ИП) — лица, имеющие доступ на ввод, коррекцию, обновление, уничтожение и чтение данных в рамках, как правило, ограниченной области БД,

· конечные пользователи (КП) — лица, использующие БД для получения справок и решения задач.

Очевидно, что именно *оператор ЭВМ является естественным пользователем ОС,* все же прочие пользователи становятся таковыми лишь вследствие расширения функций пользователя в связи с интеграцией (особенно в случае персональных ЭВМ) функции конечного пользователя, администратора системы и оператора.

Интерфейс— это способ общения пользователя с персональным компьютером, пользователя с прикладными программами и программ между собой. Интерфейс служит для удобства управления программным обеспечением компьютера. Интерфейсы бывают *однозадачные и многозадачные, однопользовательские и многопользовательские.* Интерфейсы отличаются между собой по удобству управления программным обеспечением, то есть по способу запуска программ.

Порядок выполнения практической работы:

1. Изучить теоретический материал;
2. Выполнить предложенные задания;

3. Составить отчет о выполнении практической работы;

4. Ответить на контрольные вопросы.

Задания для выполнения практической работы:

1. Включите ПК.

2. Ознакомьтесь с программой Панель управления ОС Windows

3. Настройка манипулятора мышь:

- в текстовом редакторе Word запишите свойства устройства Мышь и создайте скриншот окна Свойства;

- измените параметры назначения кнопок мыши;

- измените скорость выполнения двойного щелчка;

- включите залипание кнопки мыши;

- измените вид указателя мыши;

- измените скорость движения указателя;

- измените режим прокрутки колесика

ВЕРНИТЕ ВСЕ ИЗМЕНЕНИЯ В ИСХОДНОЕ ПОЛОЖЕНИЕ!

1. Настройка клавиатуры

- в текстовом редакторе Word запишите свойства устройства Клавиатура и создайте скриншот окна Свойства;

- измените скорость повтора вводимого символа;

- измените частоту мерцания курсора.

ВЕРНИТЕ ВСЕ ИЗМЕНЕНИЯ В ИСХОДНОЕ ПОЛОЖЕНИЕ!

1. Настройка элементов оформления экрана

- в текстовом редакторе Word запишите свойства вкладки Параметры экрана и создайте скриншот окна Параметры;

- измените тему рабочего стола;

- осуществляется выбор фонового рисунка;

- выберите расположение и цвет фона;

- настройте вид и параметры заставки;

- измените размер шрифта экрана.

ВЕРНИТЕ ВСЕ ИЗМЕНЕНИЯ В ИСХОДНОЕ ПОЛОЖЕНИЕ!

1. Настройка панели задач и меню "Пуск»

В текстовом редакторе Word опишите все настройки, которые можно применить к панели задач и меню «Пуск».

1. Настройка языка и региональных стандартов

- измените региональный стандарт языка;

- измените язык ввода по умолчанию.

ВЕРНИТЕ ВСЕ ИЗМЕНЕНИЯ В ИСХОДНОЕ ПОЛОЖЕНИЕ!

1. Настройка даты и время.

- измените текущую дату на 1 января 2020 г.

- измените часовой пояс и посмотрите какие изменения произошли, результат запишите в документе Word.

ВЕРНИТЕ ВСЕ ИЗМЕНЕНИЯ В ИСХОДНОЕ ПОЛОЖЕНИЕ!

Контрольные вопросы:

1. Для чего в ОС Microsoft Windows служит панель управления?

2. Как получить информацию об ОС, объеме памяти, типе процессора?

3. Для чего предназначена Панель задач и меню «Пуск» в ОС Windows?

4. Что такое папка, файл, диск?

5. Какие действия можно выполнить с папкой, файлом, диском?

6. Как закрепить значки на панели задач?

7. Как создать ярлык программы/файла?

Содержание отчета:

1. Фамилия, учебная группа, дата выполнения

2. Наименование практического занятия

3. Цель занятия

4. Отчет о поэтапном выполнении задания

5. Список использованных источников

6. Выводы о проделанной работе.

Тема 2 Архитектура операционной системы

Практическое занятие № 2 Установка и настройка системы. Установка параметров автоматического обновления системы. Установка новых устройств. Управление дисковыми ресурсами.

Цель: уметь настраивать операционную систему Windows; уметь проверять поверхность дисков, проводить дефрагментацию дисков; устанавливать параметры автоматического обновления системы; устанавливать новые устройства.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Настройка операционной системы Windows

Задание 1. Просмотр шрифтов.

– Дважды щелкнуть по значку Шрифты на Панели управления.

– Двойной щелчок по названию шрифта.

– Просмотреть 5–6 различных шрифтов.

– После просмотра шрифта окно закрыть.

Задание 2. Настройки фона рабочего стола

– Дважды щелкнуть по значку Экран на Панели управления.

– Щелкнуть по вкладке Фон.

– Выбрать рисунок из списка Рисунок (например, Облака). Щелкнуть по кнопке ОК.

– Переключатель Размножить позволяет размножать выбранный рисунок и покрыть

рабочий стол копиями рисунка. Переключатель По центру позволяет разместить рисунок

в центре рабочего стола.

– Если рисунок не задан или расположен в центре рабочего стола, то поверхность

рабочего стола можно заполнить узором, который выбирается в списке Фоновый узор. 6.

Вернуть фон Рабочего стола в исходное состояние.

Задание 3.Выбор и настройка экранной заставки

– Дважды щелкнуть по значку Экран на Панели управления,

– Щелкнуть по вкладке Заставка.

– В поле Заставка выбрать любую заставку. Для просмотра заставки щелкнуть по

кнопке Просмотр.

– По окончании просмотра выбрать тип заставки – Нет.

– Щелкнуть по кнопке ОК.

– Выберите в поле заставка Объемный текст. Нажмите кнопку Настройка Выберите

пункт Текст (черная точка должна стоять в круге рядом со словом Текст). В поле справа

введите номер своей группы Размер, разрешение, поверхность, скорость и стиль движения

настройте по своему усмотрению Нажмите ОК.

– Нажмите кнопку Просмотр. Просмотрите результат.

Задание 4. Настройка схемы оформления рабочего стола.

– Дважды щелкнуть по значку Экран на Панели управления.

– Щелкнуть по вкладке Оформление.

– Элемент оформления выбирается в списке Элемент. Выбрать

– Рабочий стол.

– Выбрать в списке схему оформления Дождливый день. Щелкнуть по кнопке ОК.

– Выбрать произвольную схему оформления.

– По окончании просмотра выбрать схему оформления Стандартная Windows.

Задание 5. Изменение размера и положения Панели задач.

– Изменить размер Панели задач: поместить указатель мыши на ее верхний край,

чтобы он принял вид двунаправленной стрелки. Нажать левую кнопку мыши и не отпуская ее перетащить верхний край Панели задач вверх. Максимальная ширина Панели задач

не может превышать половину экрана.

– Вернуть Панель задач в исходное состояние.

– Поместить Панель задач сбоку экрана: перетащить ее мышью.

– Щелкнуть правой кнопкой мыши на Панели задач. В появившемся меню выбрать

команду Свойства.

– Установить флажок Автоматически убирать с экрана. Щелкнуть мышью по кнопке ОК.

– Вернуть Панель задач в исходное состояние.

Задание 6. Настройка оформления Рабочего стола.

– Выберите в контекстном меню пункт Свойства – откроется диалоговое окно

Свойства: Экран. Откройте вкладку Рабочий стол.

– В списке Фоновый рисунок выберите рисунок Японский мотив. Щелкните на

кнопке ОК. Убедитесь в том, что фон Рабочего стола изменился.

– Повторите пункты 2–3, изменяя на вкладке Рабочий стол способ расположения

фонового рисунка с помощью раскрывающегося списка Расположение. Установите, как

влияют на оформление экрана способы По центру, Замостить и Растянуть,

– Повторите пункты 2–3, выбрав в качестве фонового рисунка объект Безмятежность и способ расположения Растянуть.

– Закройте все открытые окна.

Теоретические сведения

Программы обслуживания дисков

Форматирование дисков – это процесс формирования на рабочих поверхностях

дискеты дорожек и рабочих секторов. Кроме того, на дискете формируются необходимые

таблицы файловой системы: корневой каталог, FAT и т.д. В процессе форматирования вся

информация, которая находилась на дискете, будет уничтожена.

Способы форматирования:

– Быстрое форматирование – формируются новые таблицы файловой системы диска, физическая разметка рабочих поверхностей не производится;

– Полное форматирование – формируются новые таблицы файловой системы диска

и производится физическая разметка рабочих поверхностей;

Создание загрузочного диска – новые таблицы файловой системы не создаются,

физической разметки поверхностей не производится, обновляются только основные файлы операционной системы. В процессе эксплуатации магнитных дисков на их рабочих поверхностях могут возникать различные дефекты. В секторе, размещённом на дефектном

участке, информация может быть разрушена или недоступна.

Чтобы этого избежать, необходимо периодически контролировать качество рабочих поверхностей. Для этого в Windows есть средства проверки дисков.

Когда файл записывается на диск, ему выделяется группа кластеров, которые могут

располагаться последовательно или быть разбросаны по поверхности диска.

Дефрагментация диска – это процедура, при которой все файлы на диске записываются так, чтобы каждый файл занимал один сплошной участок диска, и, следовательно,

размещение файлов на диске окажется оптимальным для работы компьютера.

Задание 7. Проверка рабочих поверхностей дисков.

– Открыть меню Свойства контекстного меню диска А:

– Перейти на вкладку Сервис.

– Нажать кнопку Выполнить проверку.

– Просмотреть отчёт.

Задание 8. Дефрагментация диска.

– Открыть меню Свойства контекстного меню диска С:.

– Перейти на вкладку Сервис.

– Нажать кнопку Выполнить дефрагментацию.

Задание 9. Просмотр сведений о системе.

– Пуск – Все программы – Стандартные – Служебные – Сведения о системе.

– Записать в тетрадь, какие сведения о системе можно получить с помощью этой

программы.

– Просмотреть различные сведения. Установка новых устройств

Тема 3 Общие сведения о процессах и потоках

Практическое занятие № 3 Диагностика и коррекция ошибок операционной системы, контроль доступа к операционной системе

Цель: познакомиться со средствами диагностики и коррекция ошибок операционной системы; ознакомиться с возможностями Windows по ограничению доступа к объектам ОС, изучить основные инструменты управления доступом

Исходные материалы и данные:

- операционные системы Windows

Теоретический материал

Существуют встроенные средства устранения неполадок в ОС Windows 7 и более поздних версиях.

Устранение неполадок – это элемент панели управления Windows , предназначенный для автоматического решения самых распространенных проблем, с которыми пользователи обращаются в техподдержку Microsoft.

Если у вас возникла проблема с оборудованием, сетью, браузером Internet Explorer, Aero, либо неправильно работают программы, попробуйте решить ее встроенными средствами Windows. Структуры дисков.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание 1

- Откройте Пуск -- Поиск -- Устранение неполадок, либо введите в поиск control /name Microsoft.Troubleshooting и нажмите Enter.
- Чтобы отобразить все тесты, щелкните Просмотр всех категорий в левой панели. Запустите средство «Диагностика памяти Windows». Это можно сделать разными способами, в зависимости от конкретной ситуации. Его можно вызвать из меню «Параметры восстановления системы». Но если операционная система загружается нормально, а проблемы возникают лишь иногда, то все гораздо проще.
- Нажмите кнопку «Пуск» (Start), откройте Панель управления (Control Panel) и щелкните на значке «Система и безопасность» (System and Security).

– В открывшемся окне выбирите пункт «Администрирование» (Administrative Tools) и нажмите на значке «Диагностика памяти Windows». Можно также открыть меню «Пуск», ввести «память» (memory) в строке поиска и выбрать в результатах пункт «Диагностика проблем оперативной памяти компьютера» (Windows Memory Diagnostic).

– В появившемся окне «Средство проверки памяти Windows» (Windows Memory Diagnostic,) выберите опцию «Выполнить перезагрузку и проверку» (Restart Now and Check for Problems).

– Диалоговое окно закроется, и система будет автоматически перезагружена.

Задание 2. Запуск и проверка памяти.

– Вне зависимости от выбранного способа запуска, после перезагрузки появится экран средства диагностики памяти Windows и начнется проверка. Прогресс операции указывается в процентах и обозначается индикатором выполнения. В процессе диагностики утилита многократно записывает в память определенные значения, а затем считывает их, чтобы убедиться, что данные не изменились. По умолчанию, используется тест «Обычный» (Standard), но доступны и два других варианта.

– Чтобы выбрать один из них, нажмите кнопку [F1] для вызова экрана «Параметры» (Options). В разделе «Набор тестов» (Test Mix) можно выбрать тест «Базовый», который включает ограниченный набор проверок, или «Широкий», предлагающий расширенный спектр тестов – расширенный настолько, что проверка может затянуться на восемь и более часов. Каждый набор тестов имеет настройки кэша по умолчанию, оптимальные для данного варианта проверки. Но можно с помощью 14 клавиши [Tab] перейти в раздел «Кэш» (Cache) и задать собственные настройки.

– Под кэшем в данном случае имеется в виду кэш микропроцессора, который используется для хранения данных, полученных от модулей памяти. Некоторые тесты задействуют кэш, другие наоборот отключают, чтобы вынудить процессор обращаться непосредственно к модулям памяти.

– По мере выполнения в разделе «Состояние» (Status) появляется информация об обнаруженных неисправностях. Но вовсе не обязательно неотрывно следить за процессом, поскольку средство диагностики памяти Windows способно

идентифицировать проблемный сектор чипа и исключить его из использования. Благодаря этому Windows 7 будет запускаться нормально, без сбоев. После загрузки Windows и входа в систему сообщение о результатах проверки появится в области уведомлений.

– Посмотрите отчет с помощью средства «Просмотр событий» (Event Viewer). Для этого откройте журнал «Система» (System) и найдите «MemoryDiagnostics-Results» в списке «Источник» (Source). В графе «Код события» (Event ID) должно быть указано «1201».

Управление доступом. Теоретические сведения.

Файловые системы современных операционных систем при соответствующей настройке эффективно обеспечивают безопасность и надежность хранения данных на дисковых накопителях. Для операционных систем Windows стандартной является файловая система NTFS. Устанавливая для пользователей определенные разрешения для файлов и каталогов (папок), администраторы могут защитить информацию от несанкционированного доступа.

Каждый пользователь должен иметь определенный набор разрешений на доступ к конкретному объекту файловой системы. Кроме того, он может быть владельцем файла или папки, если сам их создает. Администратор может назначить себя владельцем любого объекта файловой системы, но обратная передача владения от администратора к пользователю невозможна. Назначение разрешений производится для пользователей или групп. Так как рекомендуется выполнять настройки безопасности для групп, то необходимо, чтобы пользователь был членом хотя бы одной группы на компьютере или в домене.

Разрешения могут быть установлены для различных объектов компьютерной системы, однако в настоящем издании рассмотрены разрешения для файлов и папок. Другие задачи, например разрешения для принтеров, решаются аналогичным образом. Для назначения разрешений для файла или папки администратор выбирает данный файл или папку и при нажатии правой кнопки мыши использует команду Свойства (Properties), в появившемся окне переходит на вкладку Безопасность (Security).

В зоне Имя (Name) имеется список групп и пользователей, которым уже назначены разрешения для данного файла или папки.

Для добавления пользователя или группы нажмите кнопку Добавить (Add) или Удалить (Remove).

При добавлении появится диалог Выбор: Пользователи, Компьютеры или Группы (Select Users, Computers or Groups). Добавив пользователя или группу, мы увидим этот объект в зоне Имя и, выделив его, можем задать необходимые разрешения с помощью установки флажков:

- Разрешить (Allow) или Запретить (Deny) в зоне Разрешения (Permissions).

Стандартные разрешения для файлов: ї полный доступ (Full Control);

- изменить (Modify);
- чтение и выполнение (Read&Execute); – чтение (Read); ї запись (Write).
- Стандартные разрешения для папок: ї полный доступ (Full Control); ї изменить (Modify);
- чтение и выполнение (Read&Execute); ї список содержимого папки;
- чтение (Read);
- запись (Write)

Разрешение Чтение позволяет просматривать файлы и папки и их атрибуты.

Разрешение Запись позволяет создавать новые файлы и папки внутри папок, изменять атрибуты и просматривать владельцев и разрешения.

Разрешение Список содержимого папки позволяет просматривать имена файлов и папок.

Разрешение Чтение и выполнение для папок позволяет перемещаться по структуре других папок и служит для того, чтобы разрешить пользователю открывать папку, даже если он не имеет прав доступа к ней, для поиска других файлов или вложенных папок. Разрешены все действия, 16 право на которые дают разрешения Чтение и Список содержимого папки.

Это же разрешение для файлов позволяет запускать файлы программ и выполнять действия, право на которые дает разрешение Чтение.

Разрешение Изменить позволяет удалять папки, файлы и выполнять все действия, право на которые дают разрешения Запись и Чтение и выполнение.

Разрешение Полный доступ позволяет изменять разрешения, менять владельца, удалять файлы и папки и выполнять все действия, на которые дают право все остальные разрешения NTFS.

Разрешения для папок распространяются на их содержимое: подпапки и файлы.

Задание 3

- Создайте папку, в которую поместите текстовый файл и приложение в виде файла с расширением exe, например одну из стандартных программ Windows, такую как notepad.exe (Блокнот).
- Установите для этой папки разрешения полного доступа для одного из пользователей группы Администраторы и ограниченные разрешения для пользователя с ограниченной учетной записью.
- Выполните различные действия с папкой и файлами для обеих учетных записей и установите, как действуют ограничения, связанные с назначением уровня доступа ниже, чем полный доступ.
- Установите разрешения общего доступа так, чтобы администратор не имел ограничений, а пользователь имел ограниченный уровень доступа. Экспериментально убедитесь в выполнении правил объединения разрешений NTFS и разрешений общего доступа.

Тема 4 Взаимодействие и планирование процессов

Практическое занятие № 4. Управление процессами с помощью команд операционной системы для работы с процессами.

Цель работы: получение практических навыков управления процессами и самостоятельной работы с документацией команд.

Теоретическая часть

Команды POSIX для работы с процессами (должны быть во всех операционных системах)

at - запускает программы в определенное время

crontab - файл содержащий таблицу расписаний запуска заданий

kill - прекращение выполнения процесса по PID процесса

nice - задает приоритет процесса перед его запуском

renice - изменяет приоритет работающего процесса

ps - выводит информацию о работающих процессах

fg - перевод процесса из фонового режима

bg - продолжение выполнения фонового процесса, если он приостановлен нажатием <Ctrl+Z>

Команды LINUX для работы с процессами

at - запускает программы в определенное время

atq - выводит список заданий, поставленных в очередь командой at

atrm - удаление задания из очереди команды at

/etc/crontab - файл содержащий таблицу расписаний запуска заданий

kill - прекращение выполнения процесса по PID процесса

killall - прекращение выполнения процесса по имени процесса

nice - задает приоритет процесса перед его запуском

renice - изменяет приоритет работающего процесса

ps - выводит информацию о работающих процессах

top - выводит динамическую информацию о процессах

fg - вывод процесса из фонового режима

bg - продолжение выполнения фонового процесса, если он приостановлен нажатием <Ctrl+Z>

ipcs - взаимодействие процессов (разделяемая память, семафоры, сообщения)

Для получения более подробной информации, можно использовать help (например: ps --help), или документацию (например: man ps, для выхода нажмите q).

Запуск фонового процесса осуществляется так:

ps -x &

При загрузке системы, необходимые процессы, загружаются в фоновый режим, их называют "демонами". Они находятся в каталоге /etc/rc.d/init.d/.

Некоторые комбинации клавиш:

<Ctrl+Z> - приостановить выполнение задания

<Ctrl+C> - завершить выполнение задания

Связывание процессов с помощью каналов. Запуск нескольких команд с передачей выходного потока следующей программе, "|" означает передачу выходного потока от первой программы ко второй.

ps -ax | more

запускается команда ps -ax, и передает выходной поток программ more которая запускается на выполнение.

Перенаправление ввода/вывода. Запуск команды с записью выходного потока в файл

ps -ax > test.txt

ps -ax > test.txt - добавит в коней файла

Группы команд

command-1;command-2;command-3

{command-1;command-2} > test.txt

Команды Windows для работы с процессами

Большую часть информации о процессах можно получить через диспетчер задач.

at - запуск программ 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

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

command.com - запуск командной оболочки MS-DOS

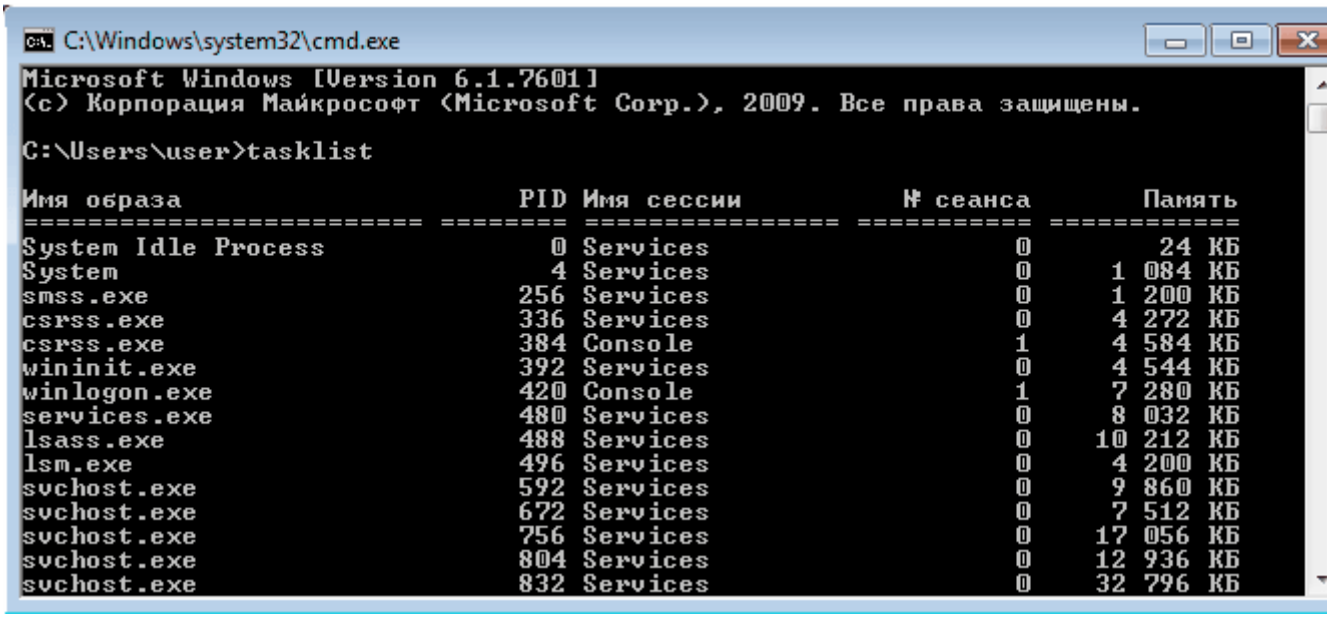
cmd.exe - запуск командной оболочки Windows

Практическая часть

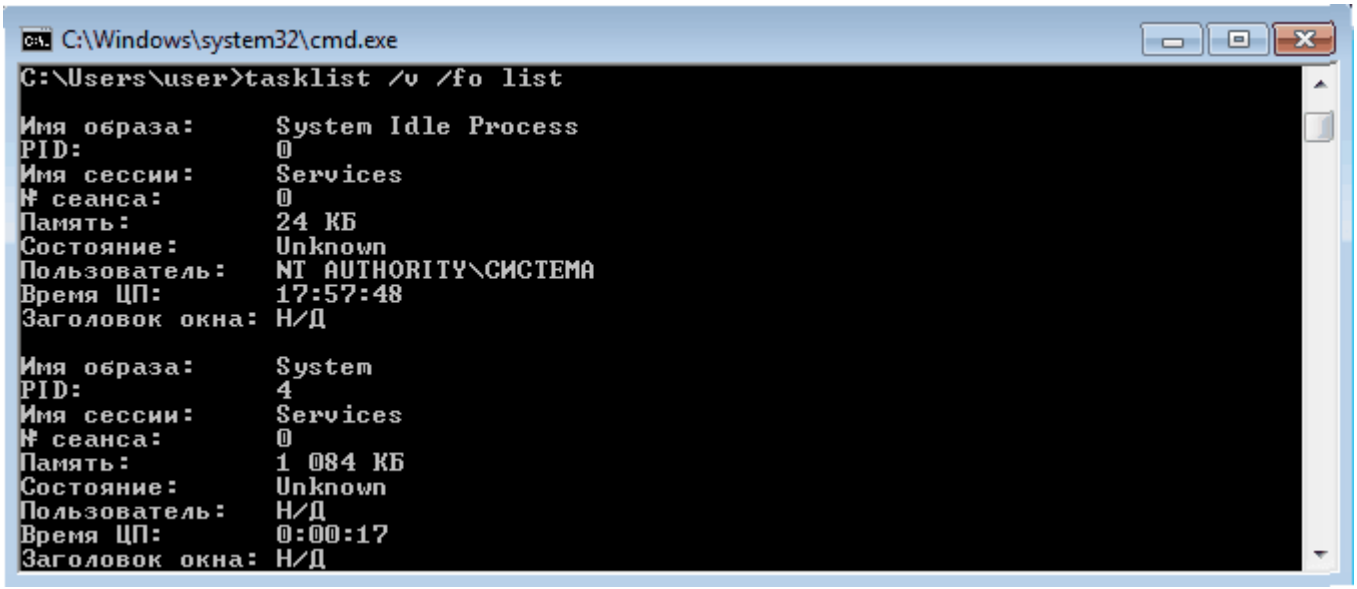
Для управления процессами Windows существует много разных утилит и приложений. Как правило, для этих целей командная строка используется очень редко. Но в некоторых случаях, это единственная возможность и другие средства недоступны. Такое может возникнуть при блокировке стандартных диспетчеров различными вирусами и другими, нехорошими программами. Поэтому, будет очень полезно знать и уметь работать с процессами через командную строку.

Чтобы управлять процессами, при помощи возможностей командной строки, в системе предусмотрено два специальных приложения: tasklist и taskkill. Из их названия можно догадаться и об их предназначении — первое может отображать список всех запущенных процессов на данном компьютере, или на удаленной машине, а вторая умеет их останавливать. Давайте глянем, как это работает на практике.

Введите, в окне терминала, tasklist и командная строка выдаст весь список рабочих процессов на данном компьютере.

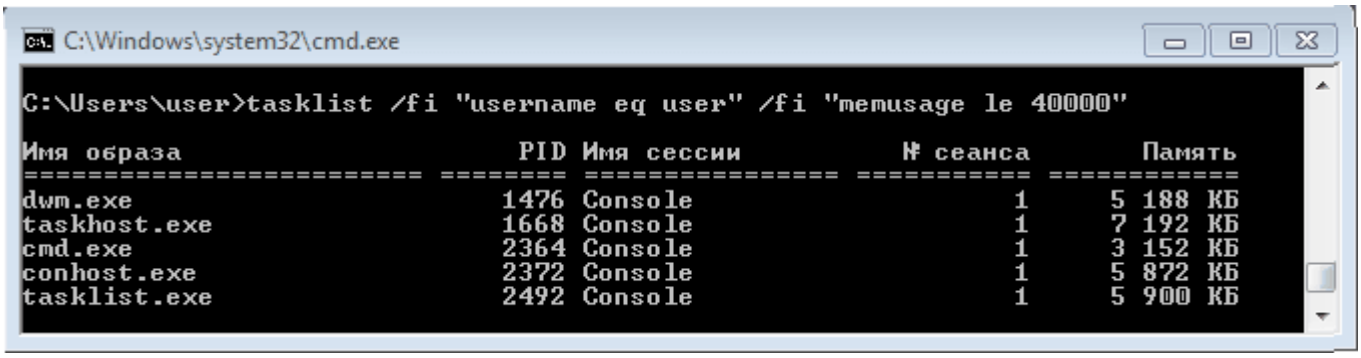


Все данные будут отображаться в табличной форме, но можно поменять вид данных в другой формат. Воспользуйтесь параметром /fo и все отобразится в виде списка (или как CSV), а если использовать параметр /v, то в этот список будут включены более подробные данные о каждом процессе. Введите команду taskliist /v /fo list (без кавычек естественно) и в окне должна появиться приблизительно такая картина.

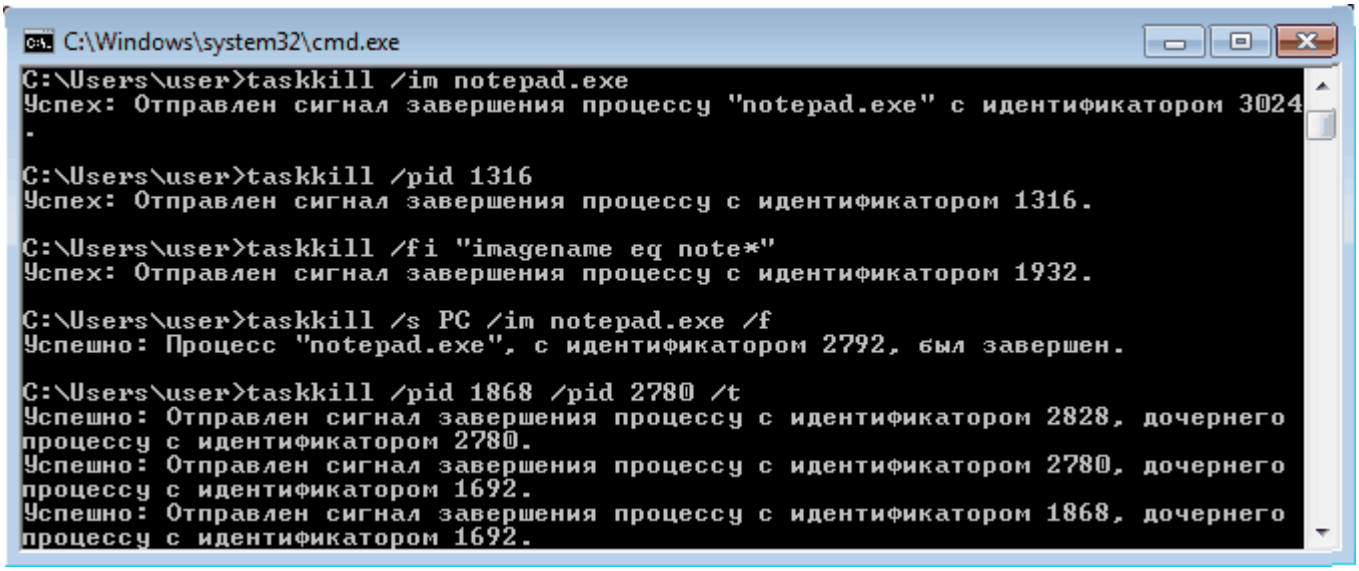


Обычно список выходит немалых размеров, и чтобы не листать его слишком долго, выполним более точный запрос. Для этих целей существует параметр /fi, способный уточнить поиск процессов используя фильтры. Для примера, введем команду, которая отобразит все процессы, запущенные от имени пользователя user и занимающие до 40 мегабайт в памяти компьютера. Команда выглядит следующим

образом: tasklist /fi »username eq user» /fi »memusage le 40000». А на рисунке ниже представлен результат работы команды.

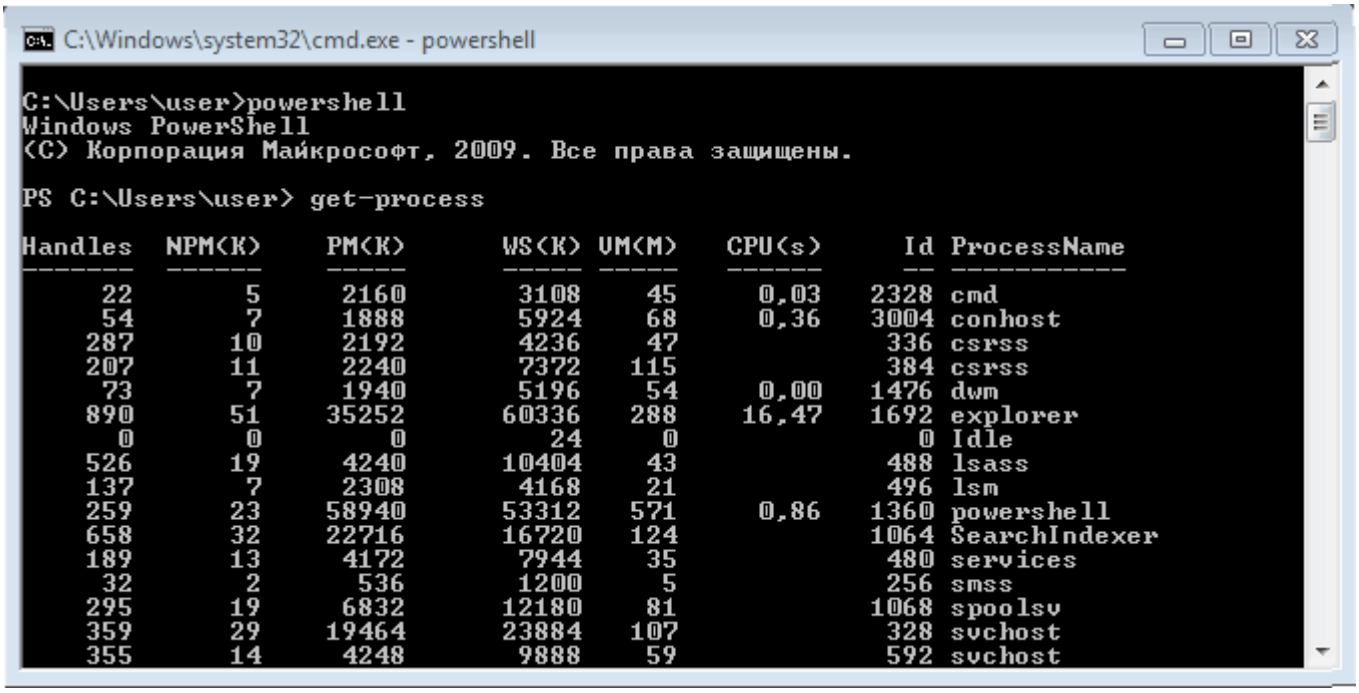


Допустим, мы нашили процессы, которые хотим завершить. Теперь настал черед утилиты «taskkill». Для прекращения работы процесса можно использовать его имя, или идентификатор PID, а можно прекращать процессы используя разные фильтры. Для эксперимента, мы запустим несколько окон программы Блокнот (notepad.exe) и на них испытаем разные способы.



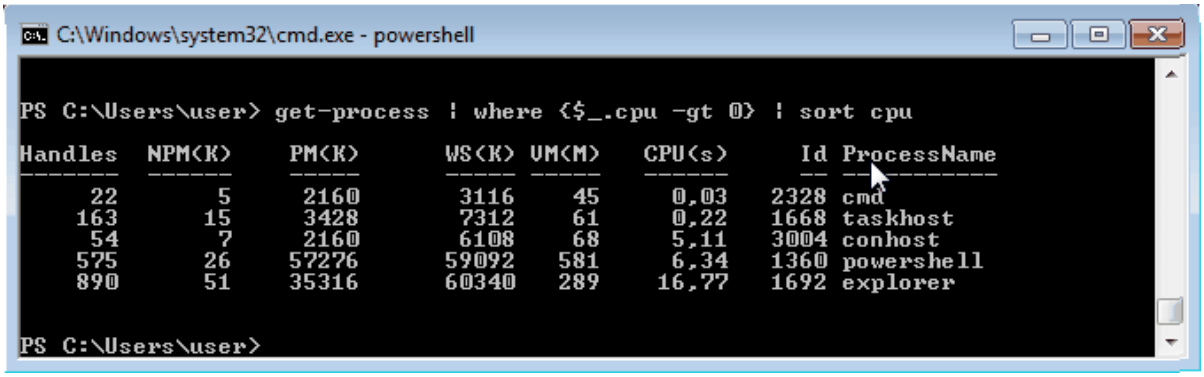
При помощи параметра /f процесс будет завершен в принудительном порядке, а если добавить еще ключ /t — завершатся все другие, которые были через него запущенны. Чтобы узнать больше возможностей, для команд tasklist и taskkill, добавьте к ним ключ /?. Он отобразит полную справку для этих утилит. Не будем забывать про такое мощное средство, как PowerShell. Мы можем им воспользоваться и прям в этом же окне командной строки. Чтобы просмотреть весь список процессов,

запустите команду Get-Process.



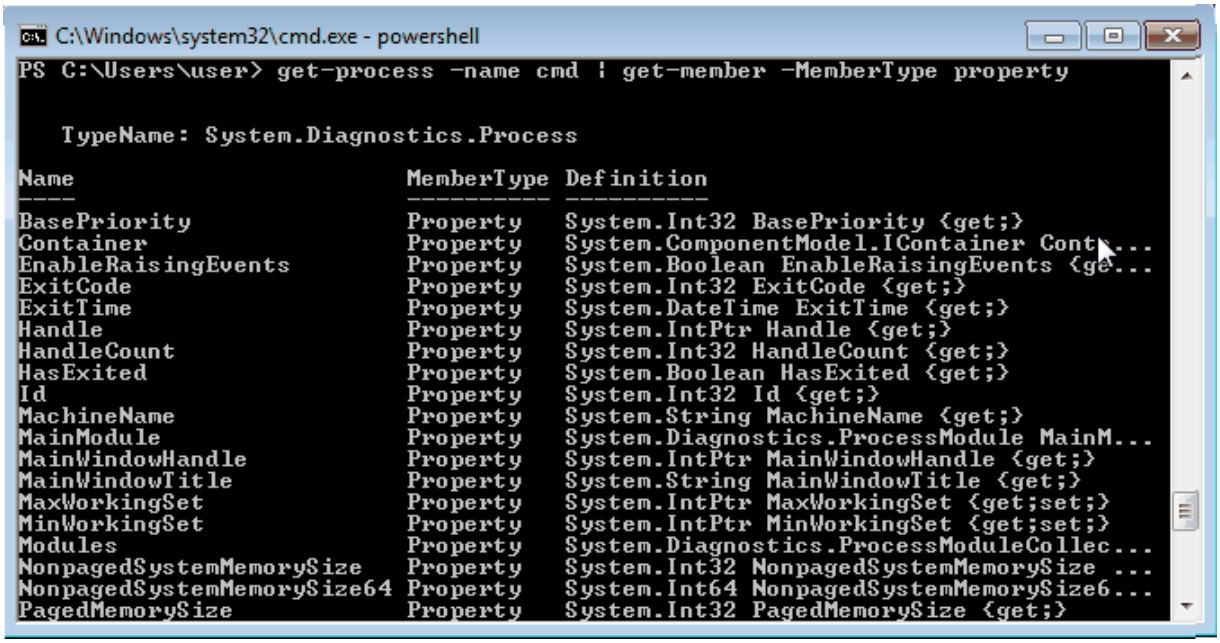
Чтобы опять не рыться во всех найденных процессах, можно задать фильтр для получения отдельных, соответствующих критериям, процессов. Для этого существует команду Where-Object. Давайте получим табличку с процессами, загружающими процессор на данном компьютере и выстроим их в порядке возрастания нагрузки. Воспользуемся командой следующего вида: Get-Process | where {$_.cpu -gt 0} | sort cpu

Результат будет иметь приблизительно такой вид:



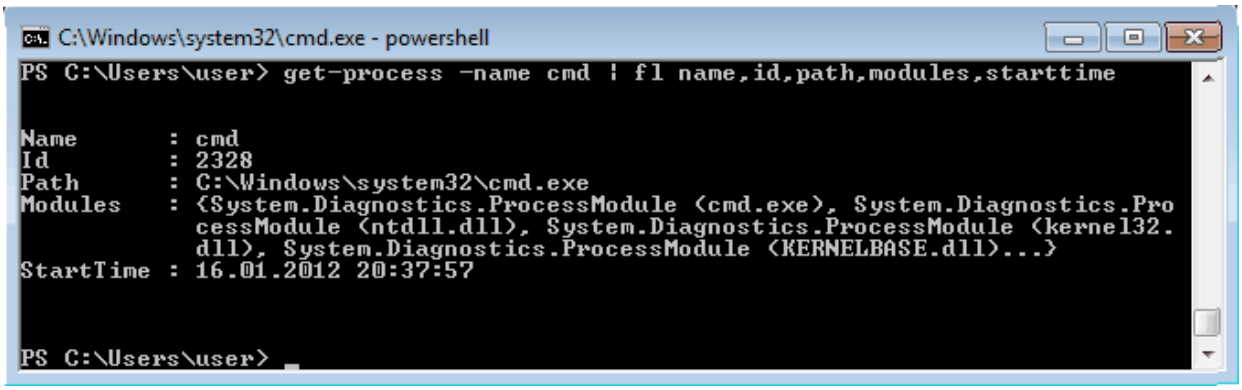
Здесь можно получить полную информацию о любом запущенном процессе. Сейчас мы узнаем список всех свойств процесса cmd. Это можно сделать при помощи такой команды:

Get-Process -Name cmd | Get-Member -Membertype property



Выведем только нужные свойства. Например, оставим только имя, идентификатор процесса, путь к его исполнительному файлу, подключенные модуле и время, когда процесс был запущен. Все это мы выводим списком, при помощи такой команды:

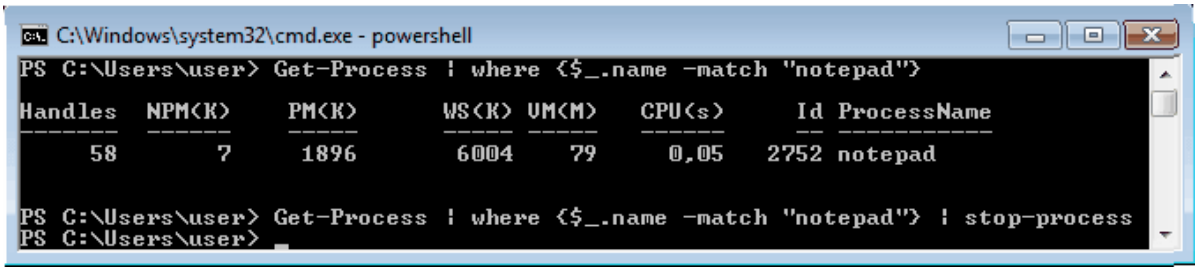
Get-Process -Name cmd | Format-List name, id, path, modules, starttime



Благодаря этому мы узнаем о том, кто запустил процесс, как сильно он грузит систему, где располагается его файл и еще кучу разной, полезной и не очень информации.

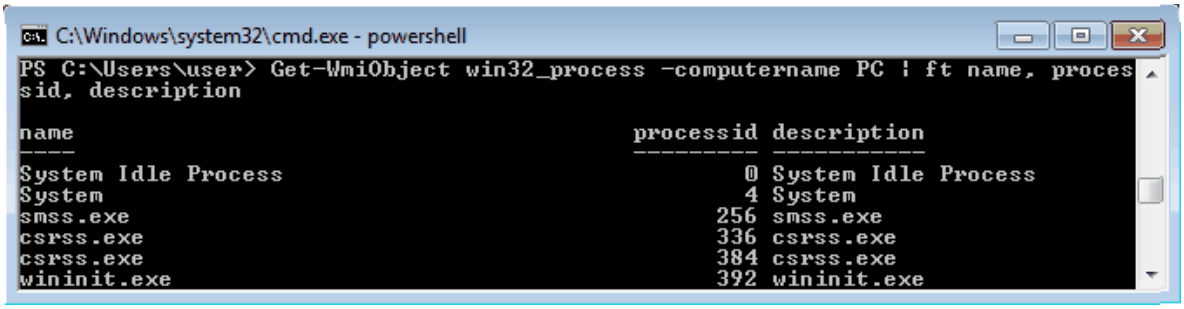
Чтобы остановить процесс через PowerShell используйте специальный командой Stop-Process. Для того, чтобы указать нужный процесс для завершения, используйте его имя или идентификатор. Например, можно по конвейеру прекратить работу блокнота:

Get-Process | where {$_.name -match ″notepad″} | Stop-Process



Для работы с процессами на другом, удаленном компьютере необходимо использовать совсем другой командой — Get-WmiObject. Сейчас посмотрим, какие процессы запущенны на компьютере с именем «PC»:

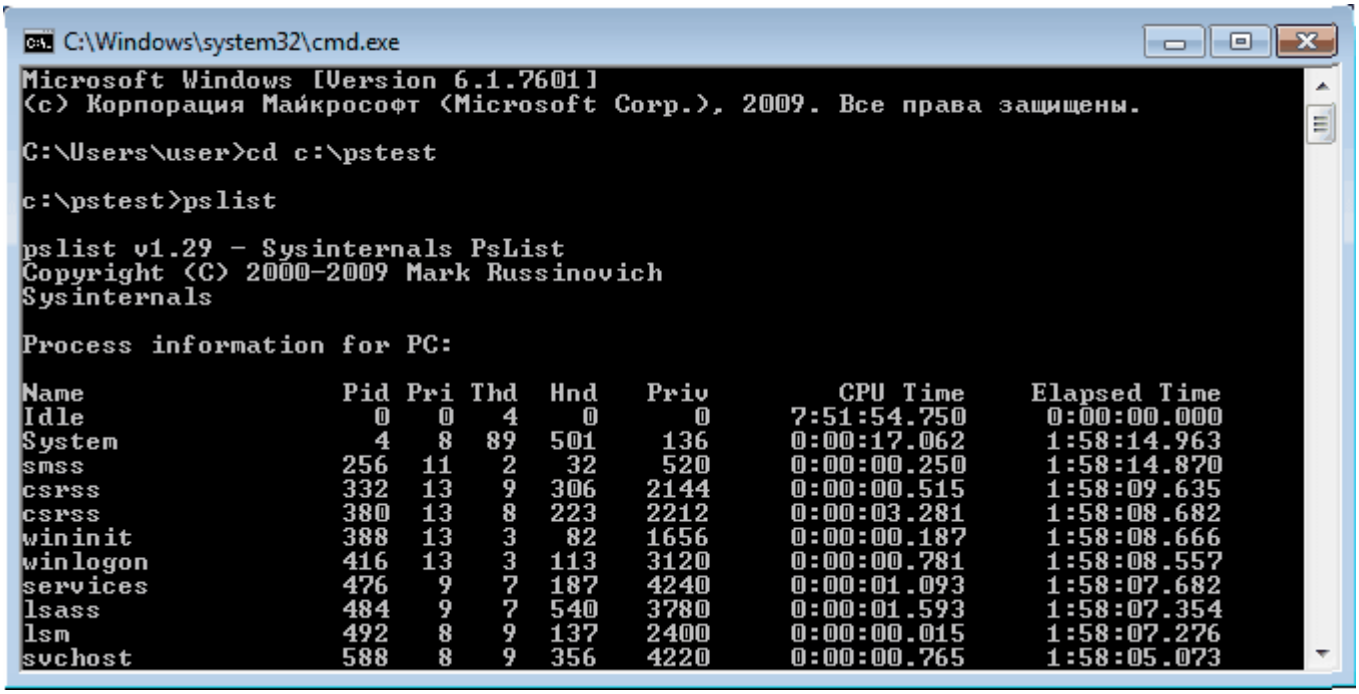
Get-WmiObject win32_process -computername PC | ft name, processid, description



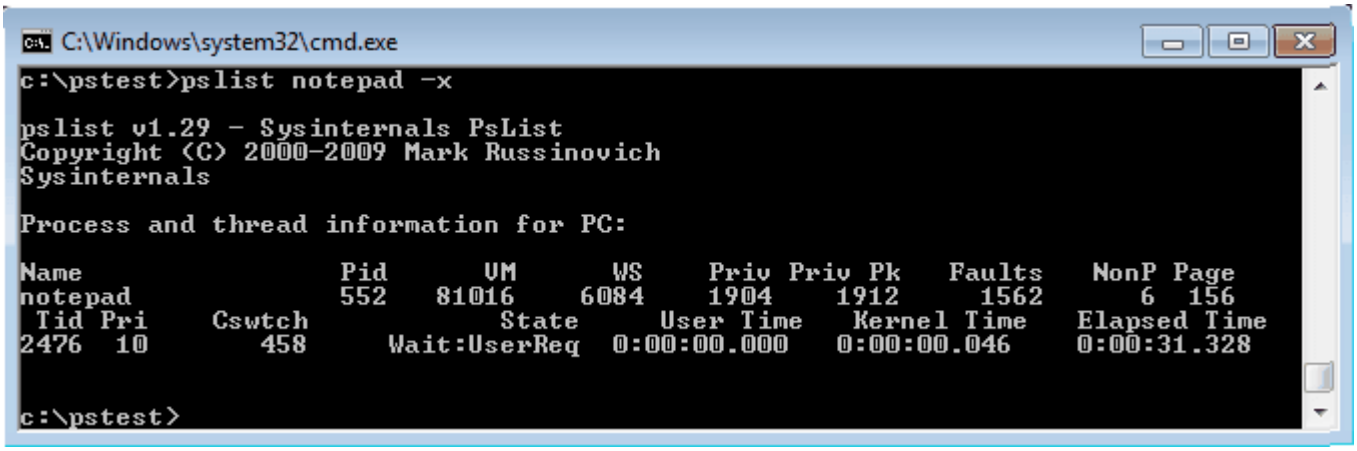
Чтобы узнать больше информации и различных ключей, для работы с процессами через PowerShell, используйте встроенную справку. Запустите команду Get-Help «имя команды» и появится справка по указанному параметру. Есть еще две дополнительные утилиты, способные выполнять операции с процессами. Ими являются Pslist и Pskill. Изначально их в системе нет, они входят в пакет PSTools.

Сами по себе эти приложения не нуждаются в установке. Их достаточно скопировать в любое место на своем жестком диске. Чтобы запустить, зайдите в

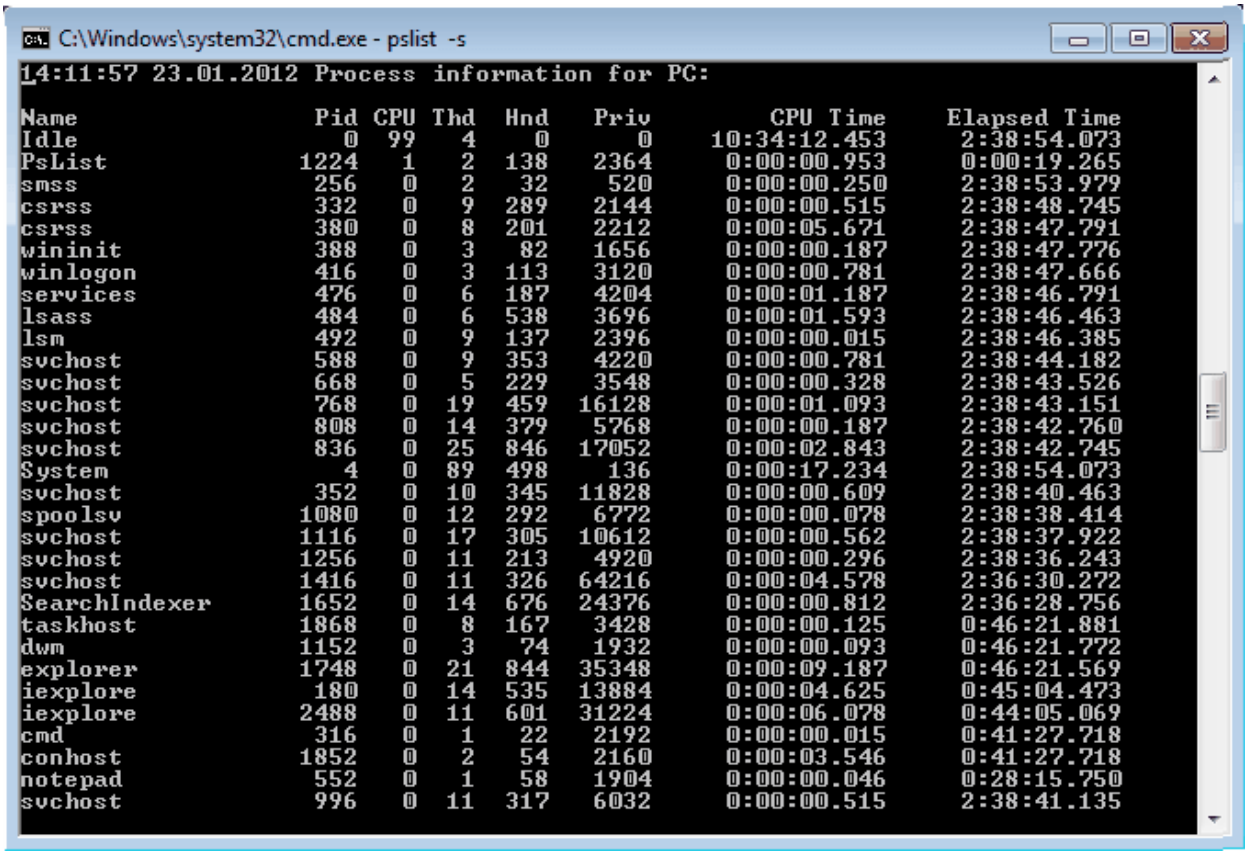
папку, где они хранятся, и активируйте нужной командой.



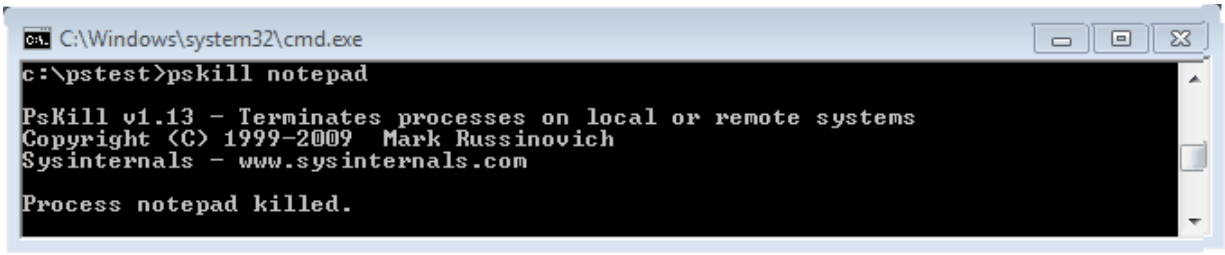
Соответственно, утилита Pslist способна вывести информацию о процессах, запущенных на компьютере. Можно выборочно находить отдельные процессы и выводить их на экран. Давайте продолжим издеваться над запущенным блокнотом и просмотрим о нем информацию командой pslist notepad -x.



Интересной функцией Pslist является возможность работы в режиме диспетчера процессов. Вся информация постоянно обновляется и можно задавать подходящий интервал получения актуальных данных. За этот режим ответствен ключ -s. Допустим, необходимо установить режим диспетчера с обновлением каждые десять секунд, эта команда будет выглядеть так: tasklist -s -r 10



Завершить любой процесс можно при помощи утилиты Pskill. Введите эту команду и идентификатор процесса, или его имя. На рисунке ниже можете посмотреть, как это будет выглядеть в случае с бедным блокнотом.



Дополнительную информацию, в виде справки, можно открыть дополнительным параметром /?

Практическое занятие № 5 Конфигурирование файлов. Управление процессами в операционной системе. Резервное хранение, командные файлы

Цель практической работы.

научиться создавать командные файлы; научиться выполнять архивирование данных и пользоваться службой восстановления конфигурационных файлов; научиться управлять процессами в операционной системе

Содержание и порядок выполнения задания.

Теоретические сведения

Командные файлы. Командный (пакетный) файл – это текстовый файл, который может содержать группу команд DOS и/или обращений к прикладным программам.

Командный файл имеет расширение bat и принадлежит к категории исполняемых файлов. Содержимое командного файла интерпретируется командным процессором и может включать:

- внешние или внутренние команды ОС;
- обращения к исполняемым программам *.COM или *.EXE, и вызовы других командных файлов;
- команды для управления выводом на экран, а также для организации ветвлений и циклов;
- метки, на которые совершается переход при выполнении заданного в файле условия.
- Каждая команда находится на отдельной строке.

При использовании стандартного приложения «Блокнот» (notepad.exe) для написания командного файла для правильного отображения символов русского алфавита нужно выбрать шрифт Terminal, с помощью меню Правка – Шрифт.

Работа с командным процессором предполагает использование двух устройств – устройства ввода (клавиатуры) и устройства вывода (дисплей). Для изменения стандартно используемых устройств ввода-вывода применяются специальные символы – символы перенаправления.

Для вывода справки не на экран а, в файл с именем help.txt, можно использовать команду help help.txt. При выполнении данной команды, в текущем каталоге будет создан файл с именем help.txt, содержимым которого будет результат вывода команды help.

Если файл help.txt существовал на момент выполнения команды, его содержимое будет перезаписано. Для того чтобы дописать данные в конец существующего файла, используют удвоение символа перенаправления вывода – «».

Часто используемые команды пакетной обработки: cls – очистка окна командной строки. rem [любая строка] – комментарий в тексте файла. Служит для каких-либо пояснений в содержимом командного файла или для временной блокировки команд.

Строка командного файла, начинающаяся со слова rem, игнорируется. echo [on или off или Сообщение] – вывод на экран сообщений. Параметры on и off включают и выключают выдачу на экран системных сообщений («эха»).

Команда echo off используется, чтобы не «засорять» экран при исполнении batфайла. Произвольная строка после echo понимается как сообщение и без изменений выводится на экран. pause – прерывает выполнения командного файла до тех пор, пока не будет нажата любая клавиша на клавиатуре, при этом на экран выводится сообщение

«Нажмите любую клавишу...»

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание № 1

- Каждый командный файл сохранять в отдельном файле на своем диске.
- Создать командный файл с именем hello.bat, который записывает в файл help.txt результат использования команды HELP.
- Создать командный файл с именем helpdir.bat, который записывает в файл helpdir.txt результат использования команды HELP DIR.
- Создать командный файл, который дописывает в файл helpdir.txt справку по использованию команды COLOR.
- Создать командный файл Ваша_фамилия.bat в каталоге Ваша_фамилия, описать следующую последовательность команд и прокомментировать каждую из них:
- очистить экран от служебных записей;

– вывести на экран поочередно информацию, хранящуюся во всех текстовых файлах в каталоге Ваша_фамилия;

– вывести на экран информацию о содержимом каталога Ваша_фамилия.

Задание 2 Написать bat-файл формирующий список всех файлов, расположенных на диске С:

– в каталоге Windows, выдать на экран и в файл Ваша_фамилия.txt.

– Написать bat-файл, который имена файлов, содержащих в расширении символ x, записывает в файл Ваше_имя.txt.

– Управление процессами Краткие теоретические сведения: Команды Windows для работы с процессами:

– at – запуск программ в заданное время;

– Schtasks – настраивает выполнение команд по расписанию;

– Start – запускает определенную программу или команду в отдельном окне; command.com – запуск командной оболочки MS-DOS;

– cmd.exe – запуск командной оболочки Windows. Для вывода списка процессов используется команда tasklist (англ. Task List – список задач).

– Для получения более подробной информации, можно использовать центр справки и поддержки или команду help (например: help at). Для остановки выполнения процесса используется команда taskkill (англ. Task KIll – «Убить» задачу).

– Для остановки процесса требуется указать идентификатор процесса PID (англ. Process IDentifier – Идентификатор процесса). PID – это «уникальное» целое число, назначенное каждому процессу. У всех процессов эти номера разные.

Теоретические сведения Создание резервных копий

Мастер архивации и восстановления (Backup or Restore Wizard) создает копию файлов и папок на указанном пользователем носителе информации. В случае потери или повреждения пользовательских данных их можно восстановить из файла резервной копии. Рекомендуется выполнять регулярное создание резервных копий важных файлов и папок. Частота архивации (резервного копирования) зависит от частоты изменений файлов, так как в случае потери данных придется повторно

создать то, что было сделано после последней архивации. По этой причине многие компании создают резервные копии важных файлов ежедневно.

Пользователь может выбирать различные типы архивации в зависимости от его требований.

Задание 3. Выполните резервное копирование системных конфигурационных файлов.

– Загрузите ОС Windows. Запустите Мастер Архивации (Пуск/Программы/ /Стандартные/Служебные/Архивация данных).

– Ознакомьтесь с информацией мастера и щелкните Далее.

– Выберите возможность мастера – Архивация файлов и параметров и щелкните Далее.

– Укажите выбор элементов архивирования в самостоятельном режиме – Предоставить возможность выбора объектов для архивации и щелкните Далее.

– Укажите элементы для архивации – папки Documents and Settings и Program Files и щелкните Далее.

– Укажите место хранения архива. **Задание 4. Восстановление системных конфигурационных файлов.**

– Запустите Мастер Архивации (Пуск/ Программы /Стандартные/ Служебные/ Архивация данных).

– Ознакомьтесь с информацией мастера и щелкните Далее.

– Выберите возможность мастера – Восстановление файлов и параметров и щелкните Далее.

– Выберите для восстановления в левом списке с содержимым архива, папку Мои рисунки (Далее).

– Ознакомьтесь с выбранными параметрами и активизируйте восстановление кнопкой Готово.

– Откройте отчет кнопкой Отчет и просмотрите его.

– Закройте диалоговое окно Ход восстановления кнопкой Закрыть

Тема 5 Управление памятью

Практическое занятие № 6 Управление памятью

Цель работы: Ознакомиться с распределением адресного пространства памяти. Исследовать обмен данными процессора с памятью

Теоретический материал

Распределение адресного пространства памяти

Все адресное пространство памяти компьютера разделяется на несколько областей, что связано, в первую очередь, с необходимостью обеспечения совместимости с первыми компьютерами семейства. В компьютере IBM PC XT на процессоре i8088 процессор мог адресовать 1 Мбайт памяти (20 адресных разрядов). Но все программные и аппаратные средства строились исходя из предположения, что доступное адресное пространство — только младшие 640 Кбайт (тогда это казалось вполне достаточным). Данная область памяти получила название стандартной памяти (Conventional memory). Именно в пределах этих 640 Кбайт (адреса 0…9FFFF) работает операционная система MS DOS и все ее прикладные программы.

Первые 1024 байта (адреса 0…3FF) хранят таблицу векторов прерывания (Interrupt Vectors) объемом 256 двойных слов, формируемую на этапе начальной загрузки. Однако если процессор работает в защищенном режиме, таблица векторов может располагаться в любом другом месте памяти.

Адреса 400…4FF отводятся под область переменных BIOS (BIOS Data Area). Подробнее о BIOS будет рассказано в следующем разделе.

Адреса 500…9FFFF включают в себя область операционной системы DOS (DOS Area) и память пользователя (User RAM).

Оставшиеся от 1 Мбайта памяти 384 Кбайта (адреса A0000…FFFFF), зарезервированные под другие системные нужды, называются UMA (Upper Memory Area) — область верхней памяти или UMB (Upper Memory Blocks) — блоки **верхней памяти** или High DOS Memory.

Пространство видеопамяти (адреса A0000…BFFFF) содержит области для хранения текстовой и графической информации видеоадаптера.

Пространство памяти с адресами E0000…FFFFF отведено под системную постоянную память компьютера ROM BIOS.

В этой же области выделено окно размером в 64 Кбайта (page frame) с адресами D0000…DFFFF, через которое программы могли получать доступ к **дополнительной (отображаемой) памяти** (Expanded memory) объемом до 32 Мбайт, оставаясь в пределах того же 1 Мбайта адресуемой памяти. Это достигается путем поочередного отображения четырех страниц по 16 Кбайт из дополнительной памяти в выделенное окно. При этом положение страниц в дополнительной памяти можно изменять программным путем. Понятно, что работать с дополнительной памятью менее удобно, чем с основной, так как в каждый момент компьютер «видит» только окно в 64 Кбайт. Поэтому сейчас она применяется довольно редко.

В настоящее время область памяти с адресами C0000…DFFFF чаще используется для оперативной и постоянной памяти , входящей в состав различных адаптеров и плат расширения компьютера.

В результате логическая организация адресного пространства в пределах 1 Мбайт получилась довольно сложной (рис.8). И такую же организацию должны поддерживать все персональные компьютеры семейства IBM PC для обеспечения совместимости с предшествующими моделями.

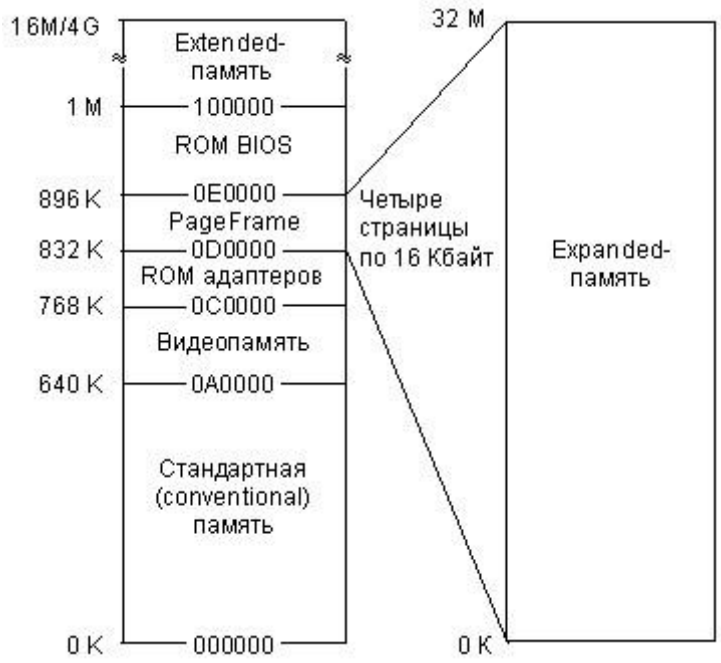


Рисунок 8

При дальнейшем расширении адресуемого пространства памяти в последующих моделях компьютеров вся память объемом свыше 1 Мбайт получила название **расширенной памяти** (Extended memory). Для доступа к ней

микропроцессор должен переходить из реального режима в защищенный и обратно. Общий объем памяти персонального компьютера (верхняя граница расширенной памяти) может доходить до 16 Мбайт (24 разряда адреса) или до 4 Гбайт (32 разряда адреса).

Особого упоминания заслуживает так называемая **теневая** память (Shadow RAM), представляющая собой часть оперативной памяти , в которую при запуске компьютера переписывается содержание постоянной памяти , и заменяющая эту постоянную память на время работы компьютера. Необходимость данной процедуры вызвана тем, что даже сравнительно медленная динамическая оперативная память оказывается все-таки быстрее, чем постоянная память . Постоянная память часто заметно сдерживает быстродействие компьютера. Поэтому было предложено выделять часть оперативной памяти для исполнения обязанностей как системной постоянной памяти ROM BIOS, так и постоянной памяти , входящей в состав дополнительных адаптеров, которые подключаются к компьютеру. Переписывание информации обычно предусмотрено в программе начального пуска.

В связи с особенностями работы динамической памяти для сокращения времени доступа к ней применяются специальные режимы работы оперативной памяти : режим расслоения (интерливинг) и страничный режим.

Использование режима интерливинга предполагает не совсем обычное разбиение памяти на банки (части). Если при обычном разбиении (последовательной адресации) адреса следующего банка начинаются после окончания адресов предыдущего, то при интерливинге адреса банков чередуются. То есть, например, после первого адреса первого банка следует первый адрес второго банка, затем второй адрес первого банка и второй адрес второго банка и т.д. Получается, что в одном банке четные слова, а в другом — нечетные. Таких чередующихся банков может быть не только два, а четыре, восемь, шестнадцать. Объемы банков при этом должны быть одинаковыми. В результате такого подхода появляется возможность начинать обращение к следующему слову еще до окончания процесса доступа к предыдущему.

Страничный режим предполагает постраничную работу микросхем памяти, когда выбор страницы производится один раз на всю страницу, а выбор ячейки внутри страницы может происходить гораздо быстрее. Для поддержки обоих режимов применяются специальные технологические решения.

Здесь же отметим, что при замене памяти компьютера или при установке дополнительных банков памяти надо строго следовать рекомендациям изготовителей системных плат, так как порядок заполнения банков может быть далеко не очевиден.

Каждому, кто изучал хотя бы минимальный курс информатики, известно, что все данные и программы их обработки хранятся в компьютере в **дискретном двоичном** виде. Согласно классическим принципам, минимальной допустимой информацией является 1 **бит** и именно бит служит основой компьютерной памяти, ее минимальным конструктивным элементом. В настоящее время благодаря успехам в технологии производства миниатюрных электронных схем сформулированный тезис не имеет столь очевидных (в прямом смысле этого слова!) доказательств, но, тем не менее, своей актуальности не утратил.

Бит слишком маленькая единица информации, чтобы быть достаточной для представления практически полезных данных. Известно, например, что для сохранения одного символа требуется 8 бит, стандартного целого числа – 16, а разрядность целочисленных данных в современных процессорах достигла 32 бит. Следовательно, обеспечивать доступ к каждому отдельному биту памяти едва ли нецелесообразно. Начиная с третьего поколения, в ЭВМ фактически сложился стандарт организации памяти, при котором минимальной считываемой порцией информации является 1 **байт**. Кроме того, для работы с более крупными данными современные процессоры способны одновременно считывать несколько байт, начиная с заданного (как правило, два или четыре).

Итак, минимальной единицей обмена информацией с памятью в современных компьютерах является 1 байт. Каждый байт имеет свой идентификационный номер, по которому к нему можно обращаться – его принято называть **адресом**. Адреса соседних байтов отличаются на единицу, зато для двух 32-разрядных чисел, хранящихся в памяти «друг за другом», эта разница по понятным причинам равняется четырем. Практически при обращении к памяти задается **адрес начального байта** и **их требуемое количество**

ПРАКТИЧЕСКАЯ ЧАСТЬ

- **Задание размера данных в команде**

В семействе процессоров Intel количество считываемых или записываемых байт определяется кодом машинной инструкции. Учитывая, что первые

представители этого семейства имели разрядность 16, и лишь начиная с модели 80386 перешли к 32 разрядам, система задания требуемого количества байт выглядит немного запутано. Так, коды команд обращения к байту или слову (вполне естественным образом) отличаются одним битом. Например, байтовая команда *MOV AL,1* имеет код *B0 01*, а двухбайтовая *MOV AX,1* кодируется *B8 01 00* (длина команды увеличилась из-за размера константы!); легко убедиться, что коды операций *B0* и *B8* действительно имеют отличие в единственном бите. Что касается четырехбайтовой команды *MOV EAX,1*, то код ее операции абсолютно такой же, как и у двухбайтовой команды (только константа 1 еще «длиннее» – 4 байта). Оказывается, что две эти одинаковые по кодам команды процессор различает по установленному режиму: при обработке 32-разрядного участка памяти константа заносится в полный регистр EAX, а 16-разрядного – в его младшую половину AX. Для изменения «режима по умолчанию» служит специальный префиксный код (так называемый *префикс переключения разрядности слова*, равный 66h), который для следующей за ним инструкции изменяет стандартный режим на противоположный. В результате при работе в 16-разрядном сегменте код *66 B8 01 00 00 00* реализует именно 32-разрядную операцию записи в EAX единицы.

Таким образом, мы видим, что разрядность команды определяется ее кодом (и, может быть, некоторыми «внешними» по отношению к программе факторами). Полученный вывод позволяет нам в дальнейшем ограничиться рассмотрением методов адресации для данных какой-либо фиксированной длины: для остальных достаточно будет просто написать другой код операции.

- **Задание адреса данных в команде**

В предыдущем разделе было показано, что для понимания методов адресации достаточно рассмотреть способы задания начального адреса при фиксированной длине данных. Поэтому в дальнейшем мы везде будем полагать, что данные имеют двухбайтовый размер. Для упрощения понимания идей адресации в большинстве примеров ограничимся также одной (самой простой!) инструкцией переписи данных, которая имеет мнемоническое обозначение MOV. Подчеркнем, что принятые допущения нисколько не ограничат общности наших знаний по описываемому вопросу.

- **Адресация простых данных**

Под простыми данными принято понимать такие, которые хранят в себе только одно значение, например, целое число. Сложные данные, напротив, включают в себя несколько значений, причем даже не обязательно одного типа; простейшим примером «однородных» сложных данных является массив, о котором мы будем говорить позднее. Не следует путать сложные структуры, состоящие из нескольких более простых значений, в частности из набора целых чисел, с простыми данными, занимающими в памяти нескольких байт, например, с отдельно взятым 32-разрядным целым числом.

Будем пока рассматривать адресацию простых данных. Как выбрано выше, это будут целые 16-разрядные числа.

Простейшие и очень часто используемые инструкции обработки данных, которые выполняются в регистрах микропроцессора, настолько естественны, что легко понимаются на интуитивном уровне. В частности, речь идет о случаях, когда в регистр заносится копия содержимого другого регистра (*MOV AX,BX*) или константа (*MOV AX,30*). Тем не менее, с теоретической точки зрения это уже простейшие методы адресации данных, которые для процессоров семейства Intel принято называть ***регистровой*** и ***непосредственной*** адресацией соответственно.

Чуть более сложным является случай, когда данные извлекаются из конкретной ячейки памяти, например, *MOV AX,[30]*; такой метод получил название ***прямой*** адресации. Приведенная в качестве примера команда считывает из памяти два байта начиная с адреса 30 и помещает их в 16-разрядный регистр *AX*. Обязательно обратите внимание на наличие в записи квадратных скобок, которые всегда появляются при ссылке на содержимое памяти.

Подобно тому, как заключение в квадратные скобки константы приводит нас к появлению прямой адресации, аналогичный прием для регистра (например, *MOV AX,[BX]*) также порождает новый (и очень важный!) метод адресации – ***косвенный регистровый***. Его суть заключается в том, что содержимое регистра рассматривается не как данные, а как адрес памяти, где эти данные расположены.

Для понимания сущности косвенной адресации можно рассмотреть следующую аналогию: при подготовке к экзамену ученик открывает свой конспект по требуемой теме, а там вместо текста для ответа видит запись: «материал по этому

вопросу прочитать в § 25». Иными словами, вместо готовой информации дается ссылка на нее.

Или еще одна аналогия из книги: «косвенная адресация похожа на операцию "для передачи (кому-то)", выполняемую почтовой службой США, когда указанный адрес не является реальным адресом получателя, а является адресом друга или родственника».

Чтобы закрепить четыре изученных метода адресации, рассмотрим несложный пример, который складывает два целых числа с адресами 200 и 204. Его выполнение зафиксировано в протоколе 1 и состоит из нескольких действий: ввод программы (команда ***a***) и ее вывод для контроля набора (***u***); ввод (***e200***) и вывод чисел (***d200***); контроль состояния регистров до выполнения программы (***r***); пошаговое выполнение четырех команд с выводом значений регистров после каждого шага (***t4***).

При анализе программы обязательно обратите внимание на то, какие методы использованы в каждой из команд. **Протокол 1**

-a

1423:0100 **mov ax,[204]** *1423*:0103 **mov bx,200** *1423*:0106 **mov cx,[bx]** *1423*:0108 **add ax,cx**

1423:010A

-u

1423:0100 A10402 MOV AX,[0204] *1423*:0103 BB0002 MOV BX,0200

1423:0106 8B0F MOV CX,[BX] *1423*:0108 01C8 ADD AX,CX *1423*:010A 0000 ADD [BX+SI],AL

...

-e200 5 0 0 0 3 0

-d200

1423:0200 05 00 00 00 03 00 *00 00-00 00 00 00 00 00 00 00* .....

... **-r**

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=*1423* ES=*1423* SS=*1423* CS=*1423* IP=0100 NV UP EI PL NZ NA PO NC *1423*:0100

A10402 MOV AX,[0204] DS:0204=0003 **-t4**

AX=0003 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=*1423* ES=*1423* SS=*1423* CS=*1423* IP=0103 NV UP EI PL NZ NA PO NC

1423:0103 BB0002 MOV BX,0200

AX=0003 BX=0200 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=*1423* ES=*1423* SS=*1423* CS=*1423* IP=0106 NV UP EI PL NZ NA PO NC *1423*:0106

8B0F MOV CX,[BX] DS:0200=0005

```
AX=0003  BX=0200 CX=0005  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=1423 ES=1423  SS=1423  CS=1423  IP=0108   NV UP EI PL NZ NA PO NC
1423:0108                01C8                          ADD                          AX,CX

AX=0008 BX=0200  CX=0005  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=1423 ES=1423  SS=1423  CS=1423  IP=010A   NV UP EI PL NZ NA PO NC
1423:010A 0000          ADD     [BX+SI],AL                  DS:0200=05
```

Тема 6 Файловая система и ввод и вывод информации

Практическое занятие № 7 Работа с программой «Файл-менеджер Проводник». Работа с файловыми системами и дисками.

Цель: Изучить технологию работы с файловой структурой с использованием программы «Файл-менеджер Проводник»

Теоретический материал

Файл (от англ. file – картотека, архив)– это именованная область диска для постоянного хранения информации (программ, данных для их работы, текстов, рисунков и т.

д.). Каждый файл имеет имя и тип (расширение), которые записываются через точку: имя.тип.

Для каждого файла, кроме имени и расширения, ОС хранит информацию о размере файла, дате и времени его создания или последней модификации, и несколько величин, называемых атрибутами. Атрибуты – дополнительные параметры, определяющие свойства файлов: Read Only (Только для чтения); Hidden (Скрытый); System (Системный); Archive (Архивный).

Файлы по любому общему признаку, выбранному пользователем, объединяются в каталоги (папки). Каталог (папка, folder) – место на диске, в котором хранятся сведения о файлах: их имена и их атрибуты. На логическом уровне каталоги – это элементы иерархической структуры, необходимые для обеспечения удобного доступа к файлам, особенно, если файлов на диске слишком много. Каждый каталог имеет свое имя, задаваемое пользователем при его создании. Каталог может быть вложенным, т. е. находиться в каталоге более высокого уровня. Корневой каталог является самым верхним уровнем вложенности иерархической структуры и организуется на диске ОС при форматировании диска. Корневой каталог обозначается обратной косой чертой (backslash) C:\.

В отличие от имен файлов, в именах папок (директорий, каталогов) расширение обычно не ставится, так как в этом нет особой необходимости.

В пределах одной папки могут находиться сколько угодно файлов, но имена файлов вместе с расширениями должны быть уникальными, то есть не должны повторяться, но не запрещено иметь в одной папке несколько файлов с одинаковыми именами, но разными типами: письмо.txt, письмо.doc

В процессе работы с файлами возникает необходимость создавать новые файлы, заменять одни файлы другими, перемещать их с одного места на другое, переименовывать, удалять.

Windows является наиболее популярной операционной системой с графическим интерфейсом и обеспечивает возможность многозадачности - одновременной работы нескольких приложений.

ХОД РАБОТЫ

Управление файлами и папками

Для управления файлами и папками в Windows используют объект «Проводник» (на примере системного обекта «Компьютер»

1. Откройте объект «Компьютер» через кнопку «Пуск». Появится окно, которое имеет стандартный внешний вид. Внутри окна располагаются объекты (рис.11). С помощью стандартного набора инструментальных средств (строка меню, лента кнопок, контекстное меню) выполняются различные операции над этими объектами, причем, как всегда, конкретная операция применяется либо ко всем, либо к выделенным объектам.

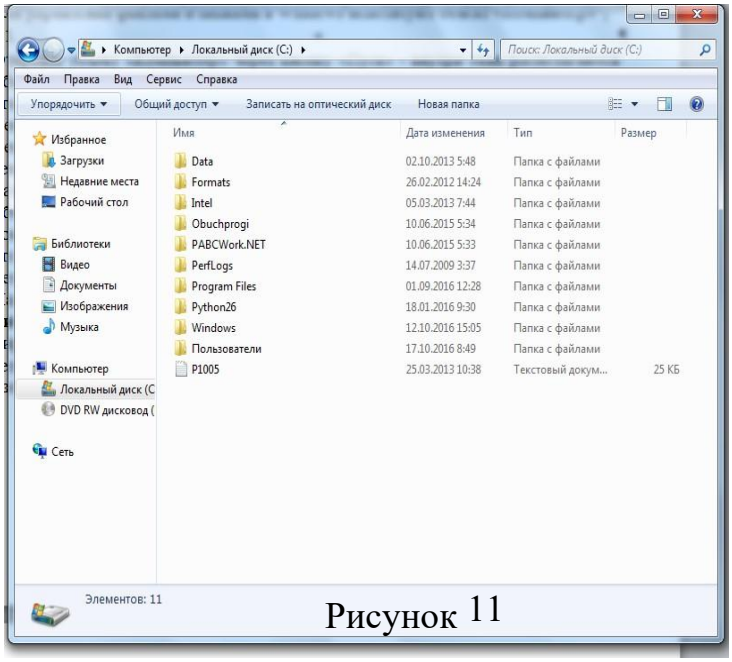


Рисунок 11

2. Самостоятельно изучите ленту кнопок «Упорядочить, Общий доступ, Запись, Новая папка, вид значков».

3. Работа с объектами: **Создание папки.**

Дважды щелкните по значку **Диск С (Локальный диск):** - откроется окно папки **Диск С:**, дважды щелкните по значку папки **«Документы»** – откроется окно папки **«Документы»** на свободном месте щелкните правой кнопкой мыши. В открывшемся контекстном меню выберите команду «**Создать – Папку».** Появится значок папки, введите название папки – Отчет_1. Аналогично можно создать файл: на свободном месте щелкните правой кнопкой мыши, из контекстного меню выберите команду «**Создать – Документ MS Word»,** укажите имя документа «Карточка предприятия.doc». Также на ленте кнопок есть команда «Новая папка», которая так же позволяет создать папку. Команда **«Файл»** в строке меню позволяет создавать и файлы и папки.

– Любым способом создайте 2 папки (Отчет 2015_2, Отчет 2015_3) и 2 файла (Список сотрудников.xls, Типовой договор.doc). **Переименование файлов и папок**

– Щелкните правой кнопкой мыши на значке Отчет-1. в открывшемся контекстном меню выберите пункт **«Переименовать**». Дайте папке имя Отчет _2015.

– Выделите значок Отчет_2015 в строке меню выберите команду «**Файл - Переименовать»** укажите папке имя Отчет_2015_1.

Перемещение файлов и папок

– Выделите папку Отчет_2015_1 в окне папки **«Документы»**. В строке меню выберите команду «**Правка- Вырезать»**. (Таким способом объект помещается в буфер обмена с удалением из текущей папки).

– Откройте «Локальный диск С: и в строке меню выберите команду «**Правка – Вставить»** (объект будет вставлен из буфера обмена).

– Команды **«Вырезать (Ctrl+X) – Вставить (Ctrl+C)»** также доступны при вызове контекстного меню.

Копирование файлов и папок

– Выделите файл «Карточка предприятия.doc» строке меню выберите команду **«Правка – Копировать**». (Таким способом объект помещается в буфер обмена без удаления). Откройте папку Отчет_2015_1, в ней в строке меню выберите команду **«Правка – Вставить»**. (Будет вставлен объект из буфера обмена).

– Скопируйте файл: Список сотрудников.xls, Типовой договор.doc в папки Отчет 2015_2, Отчет 2015_3

Удаление файлов и папок

– В окне папки Отчет 2015_3 удалите файл - «Карточка предприятия.doc». Для этого щелкните правой кнопкой мыши на значке папки и в открывшемся контекстном меню выберите пункт **«Удалить»**. В открывшемся диалоговом окне подтвердите необходимость удаления объекта. – Удалите все созданные вами папки, и файлы.

Практическое занятие № 8 Работа с командами в операционной системе. Использование команд работы с файлами и каталогами. Работа с дисками. Работа с текстовым редактором. Работа с архиватором. Работа с операционной оболочкой

Цель: изучить встроенные утилиты операционной системы Microsoft Windows для работы с файловой системой и диагностики сетевых подключений.

Теоретический материал

Командная среда – это программный продукт Microsoft, который обеспечивает связь между пользователем компьютера и операционной системой.

Командная оболочка Windows использует интерпретатор команд cmd.exe и присутствует во всех версиях операционных систем Windows. Многие возможности и функции управления операционной системой недоступны из графического интерфейса и поэтому cmd является единственным средством доступа к этим инструментам. Отличием работы из командной строки является полное отсутствие больших и громоздких графических утилит.

Пользовательский интерфейс текстовой строки предоставляет среду, в которой выполняются приложения и служебные программы. Среда, эмулирующая DOS имеет множество названий, таких как командная строка, окно, среда и т.д. С помощью cmd возможно создание сценариев автоматизации и пакетных файлов, т.е. выполнение одной или нескольких команд без вмешательства пользователя. Это отличный инструмент для создания сценариев, а также вы сможете в полной мере использовать команды для управления реестром.

Это значит, что одна или несколько команд будут выполняться без какого-либо вмешательства пользователя. Одним из примеров автоматической работы программного обеспечения служит настройка на автоматическое открытие необходимых вам программ при включении компьютера.

Управление данными и файлами. Преимущества cmd становятся очевидны, когда требуется выполнять однотипные операции над множеством объектов. Одним из важных преимуществ командной строки является непосредственная возможность командной строки управлять файлами и данными. К данным возможностям относятся:

копирование, удаление, перемещение и т.д. При этом, не забывайте, что вы можете автоматизировать данный процесс.

– Администрирование компьютера. Быстрое получение текущей информации сокращает время диагностики компьютера.

– Администрирование сети. Многие команды администрирования сети не имеют графических эквивалентов (например – команда ping, pathping, tracert). Командная строка очень удобна для контроля сетевой активности. Вы можете создавать службы, запускающиеся при старте оперативной системы, можете использовать команды администрирования сети, не имеющие графических эквивалентов.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание 1

– Создайте на рабочем столе папку iNDEX.

– Запустите командную строку cmd.exe (Пуск – ввод с клавиатуры «cmd» без кавычек).

– Проверьте системные дату и время с помощью команд date и time. Для этого в командной строке наберите нужную команду и нажмите Enter.

– С помощью утилиты cd измените текущий каталог на каталог Test, (созданный Вами заранее).

– С помощью команды md создайте каталог с именем Cat. Используя команду copy con, создайте файл с именем File.txt. (Команда copy con означает копирование с консоли, т.е. с клавиатуры).

– После данной команды введите следующий текст: Ваши Ф.И.О., группа и название ПРАКТИЧЕСКОЙработы. Закройте файл сочетанием клавиш Ctrl+Z.

– С помощью команды dir просмотрите список созданных объектов в папке Test. Команда в общей сложности фиксирует 3 каталога (папки), т.к. первая метка указывает на текущий каталог, обозначенный точкой, вторая – на предыдущий каталог (две точки).

– В каталоге Cat с помощью команды copy создайте две копии файла File.txt –

File1.txt и File2.txt.

– Объедините файлы File1.txt и File2.txt в файл oneFile.txt с помощью команды copy.

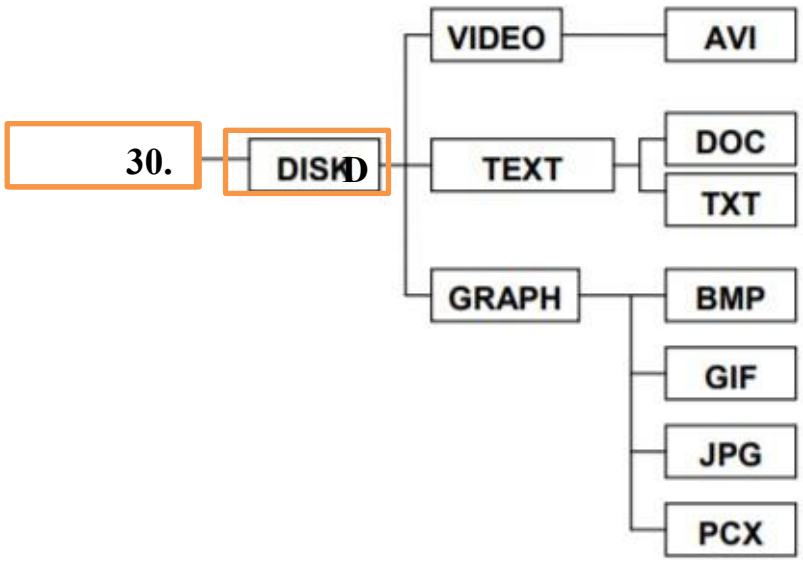
– Просмотрите полученный файл oneFile с помощью утилиты copy. С помощью команды move переместите файл oneFile.txt в папку Test. С помощью команды rename переименуйте файл oneFile.txt в newFile.txt.

– С помощью той же команды смените расширения у всех файлов в каталоге trat.

Задание для самостоятельной работы

– Загрузить операционную систему (нажать кнопку **«Поиск»** на панели задач и ввести команду **«cmd»**, Enter;

– Создать дерево каталогов;



– Отобразите дерево каталогов сравните с образцом.

– В каталогах 4 уровня создайте текстовые файлы: 1.txt, 2.txt, 3.txt, 4.txt, 5.txt, 6.txt, 7.txt. В каждом файле напишите по несколько разных команд по работе с DOS с их расшифровкой на русском языке).

– Скопируйте файл 7.txt в каталог DISK.

– Переместите файлы 3.txt, 4.txt, 5.txt в каталог VIDEO

– В каталоге DISK переименуйте файл добавив к их имени текущую дату (например, 26 янв7.txt).

- Удалите каталоги DOC и TXT.
- Отобразите системное время.
- Отобразите системную дату.
- Выведите справку по всем командам DOS.
- Отобразите дерево каталогов.
- Сделайте текущей папку GRAPH.
- Удалите содержимое папки GRAPH.
- Отобразите дерево каталогов, покажите результаты преподавателю.
- Удалите каталог 30.01.2020

Теоретический материал Утилиты

Утилита ipconfig Утилита ipconfig служит для отображения параметров текущих сетевых подключений, а также для управления клиентскими сервисами DHCP и DNS.

Синтаксис ввода: ipconfig [/all] [/renew [адаптер]] [/release [адаптер]].

При вводе команды могут использоваться ключи (дополнительные параметры). При вводе команды ipconfig без параметров выводится только IP-адрес, маска подсети и основной шлюз для каждого сетевого адаптера.

Результаты ввода команды ipconfig с ключом /all. можно условно разделить на три группы:

К первой группе (значения, выделенные синим цветом) относится общая информация о сетевом подключении:

- Подключение по локальной сети“ – Имя сетевого подключения;
- Ethernet адаптер“ – Тип адаптера;
- Attansic L1 Gigabit Ethernet 10/100/1000Base-T Controller“ – Описание адаптера;
- 00-1D-60-74-26-01“ – Физический адрес (MAC-адрес) адаптера.

Ко второй группе (значения, выделенные зеленым цветом) относится информация о сетевых настройках подключения:

- 10.144.39.224” – Сетевой адрес (IP-адрес) подключения;
- 255.255.248.0” – Маска подсети;
- 10.144.32.1” – Адрес шлюза;
- 85.21.192.3”, “213.234.192.8” – Адреса серверов DNS.

К третьей группе (значения, выделенные оранжевым цветом) относится информация о деталях аренды адреса у DHCP сервера:

- Dhcp включен: да” – Функция получения параметров у DHCP сервера включена;
- Автонастройка включена: да” – Функция авто-настройки подключения включена;
- 83.102.233.202”- Адрес DHCP сервера, у которого получены параметры;
- 10 января 2010 г. 14:51:58” – Дата получения параметров;
- 17 января 2010 г. 14:51:58” – Дата истечения аренды сетевого адрес.

Утилита ping предназначена для проверки работоспособности соединения между двумя устройствами на уровне протокола IP (сетевом уровне).

Утилита выполняет проверку, посылая на 30 указанный сетевой адрес эхо-запросы

(ICMP Echo-Request) протокола ICMP и фиксирует получение эхо-ответов (ICMP Echo-

Reply).

Синтаксис ввода: ping [-t] [-a] [-n счетчик] [имя_конечного_устройства] При вводе команды могут использоваться ключи (дополнительные параметры).

Утилита traceroute определяет путь, по которому проходят пакеты между локальным и удалённым устройствами. Полученный путь представляет собой это список ближайших интерфейсов устройств, работающих на сетевом уровне, находящихся на пути между устройствами.

Утилита используется для локализации проблем, или для сбора информации о наличии устройств в сети. Работа утилиты, так же как и работа утилиты ping основана на отправке эхо-запросов.

Синтаксис ввода: tracert [-d] [-h число] [имя_конечного_устройства]

При вводе команды могут использоваться ключи (дополнительные параметры). Команда arp Служит для вывода и изменения записей кэша протокола ARP, который содержит одну или несколько таблиц, использующихся для хранения IP-адресов и соответствующих им физических адресов Ethernet. Для каждого сетевого адаптера Ethernet, установленного в компьютере, используется отдельная таблица.

Синтаксис ввода: arp [-a [IP_адрес] [-N MAC_адрес]] При вводе команды могут использоваться ключи (дополнительные параметры).

Задание С помощью утилиты ipconfig определите и запишите в отчет следующую информацию:

- название сетевого подключения;
- тип используемого адаптера; – MAC-адрес адаптера;
- IP-адрес сетевого подключения;
- сетевую маску; – основной шлюз; – IP-адрес DNS-сервера;
- IP-адрес DHCP-сервера. **Задание 3 С помощью утилиты ping проверьте доступность следующих устройств:**
- сервер DHSP; – сервер DNS;
- информационный ресурс www.ystu.ru.
- Используя дополнительные ключи, сделайте так, чтобы количество посылаемых эхо-запросов равнялось номеру компьютера (последние 2 цифры в имени компьютера) + . Для каждого устройства и информационного ресурса запишите в отчёт следующую информацию:
- процент потерь;
- среднее время приёма передачи.

Задание 4 С помощью утилиты tracert проверьте доступность следующих устройств: – информационный ресурс www.ystu.ru;

- информационный ресурс www.ya.ru. Используя дополнительные ключи, сделать так, чтобы утилита не определяла DNS имена промежуточных устройств. Запишите в отчёт следующую информацию:

– количество промежуточных устройств; – IP-адрес всех промежуточных устройств.

Задание 5 С помощью команды arp определите и запишите в отчет MAC-адреса следующих устройств:

– ї основной шлюз; – ї три любых компьютера.

Тема 7 Работа в операционных системах и средах

Практическое занятие № 9 Выполнение порядка установки операционной системы на ПК.

Цель: получение практического опыта установки операционной системы.

Ход работы

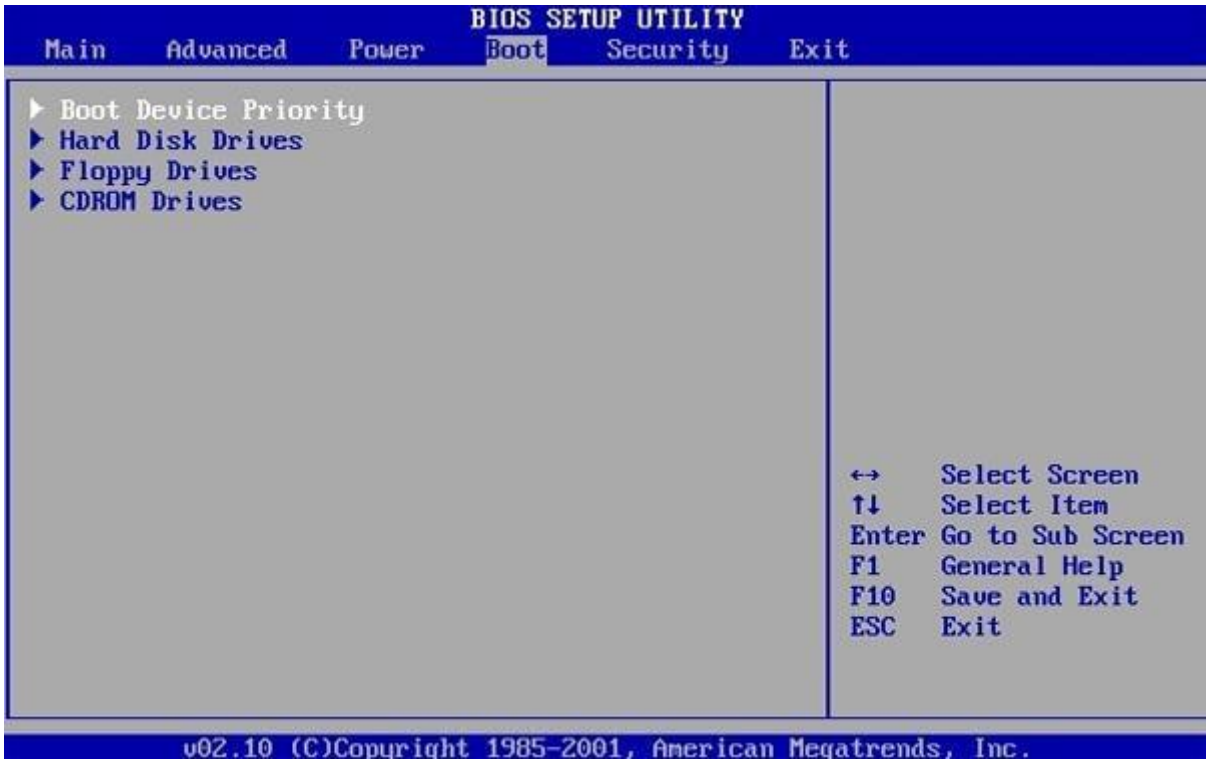
Помните: во время переустановки или установки Windows с нуля будут удалены накопленные на системном HDD/SSD пользовательские и программные данные, а также настройки встроенных и сторонних приложений. Чтобы исключить утрату важной информации, скопируйте представляющие ценность файлы на другой жёсткий диск или флэшку. Кроме того, можете перекопировать данные на облачный сервер: это бесплатно. Пропустите этот шаг, если ставите ОС на новый компьютер или ноутбук.

Для начала работы вам понадобится USB-накопитель (флешка, съёмный HDD) или диск с образом Windows. Образ чаще всего представлен в формате ISO. Скачайте для инсталляции ОС нужной разрядности и запишите её на устройство. Для компьютеров с оперативной памятью менее 4 Гб подойдёт ОС x86, более 4 Гб — x64. Копируйте образ с помощью специального приложения: Rufus, UltraISO или другой программы. Когда носитель будет записан, приступайте к следующему шагу — настройке BIOS.

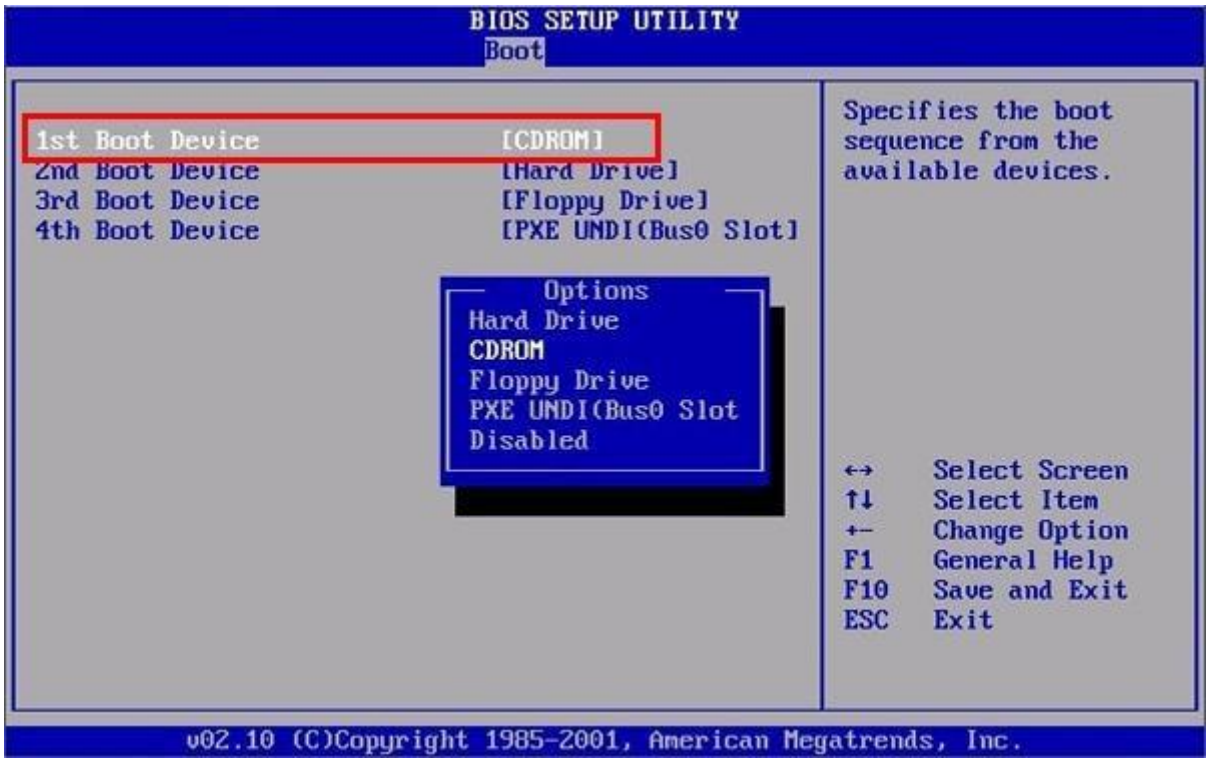
Настраиваем БИОС

Оформление BIOS или продвинутого варианта, UEFI, различается в зависимости от производителя и версии прошивки. Дать универсальную руководство для всех моделей ПК и ноутбуков невозможно — поэтому, отвечая, как установить Windows на ПК, приведём общий способ действий:

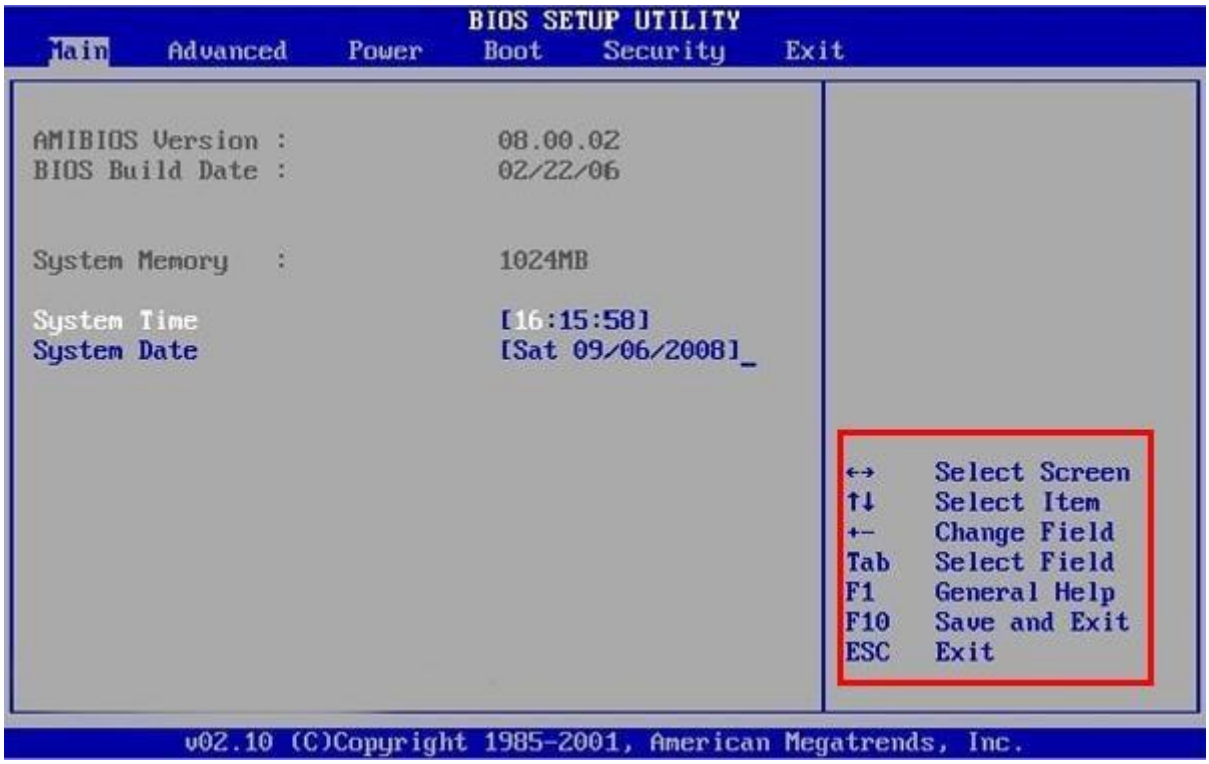
- Выключите компьютер или ноутбук.
- При новом включении нажмите несколько раз клавишу для входа в BIOS. В зависимости от производителя это может быть Delete, F2, F8, F12 или другая кнопка.
- Когда увидите на мониторе окно BIOS, перейдите на вкладку Boot.
- Откройте, нажав Enter, раздел Boot Device Priority. Он может называться и по-другому — а заголовке должно присутствовать слово Boot или «Загрузка».



- Откройте подраздел 1st Boot Device. Это первый носитель, к которому во время загрузки обращается ПК. По умолчанию это системный жёсткий диск — но вам нужно сделать так, чтобы на время приоритетным накопителем стала только что созданная загрузочная флешка. Выберите в списке нужный вариант — и нажмите Enter.



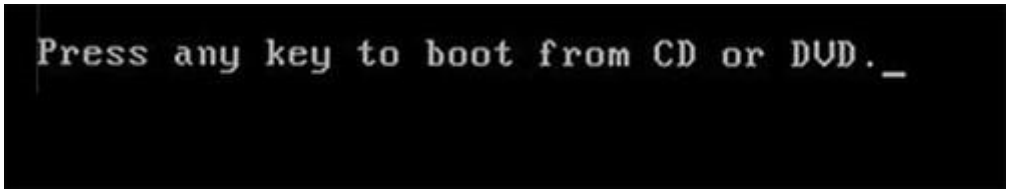
Чтобы перейти к процессу установки, нажмите клавишу F10 и подтвердите желание выйти из BIOS с сохранением внесённых изменений. Чтобы поменять порядок устройств в UEFI, достаточно просто перетащить флеш-карту на первое место в списке при помощи мыши.



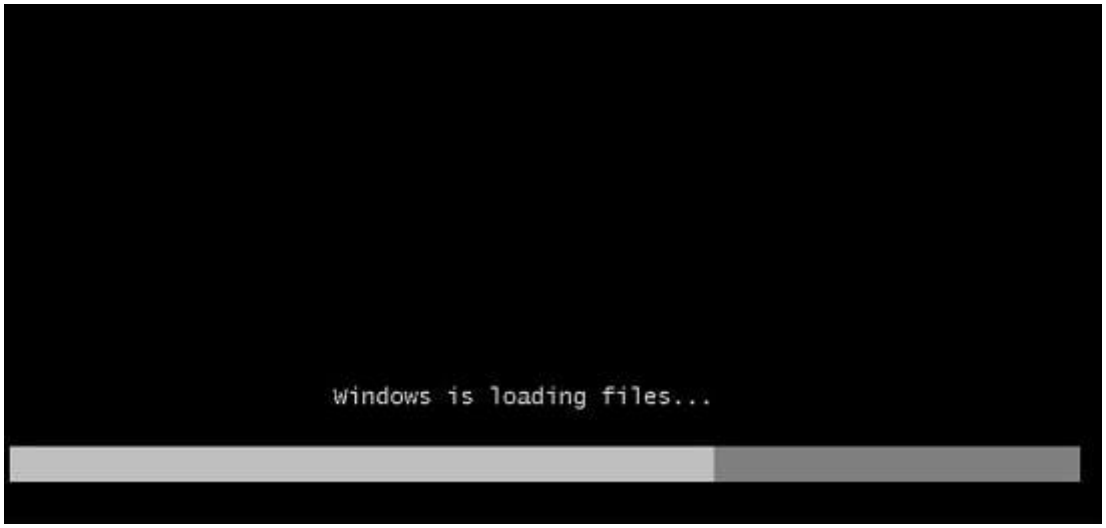
Устанавливаем «Виндовс»

Порядок переустановки или установки операционной системы с нуля определяется версией «Виндовс» и особенностями разметки HDD. Общая пошаговая инструкция по установке Windows включает такие шаги:

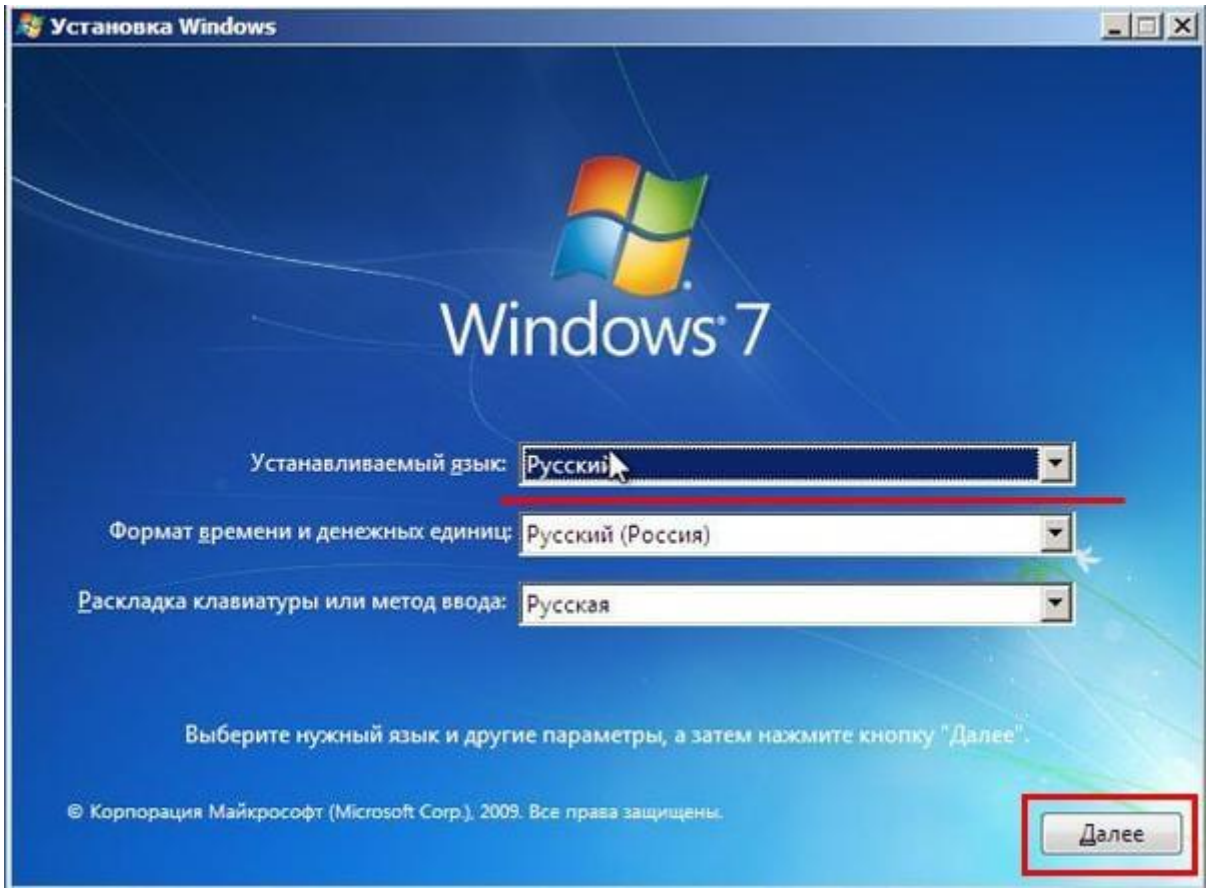
- После нового включения и появления на экране надписи Press any key быстро нажмите на любую кнопку — например, на пробел. Это позволит запустить инсталляцию именно с подготовленного носителя. Если никакой надписи нет и загрузка идёт в обычном порядке, установочная флешка записана неправильно.



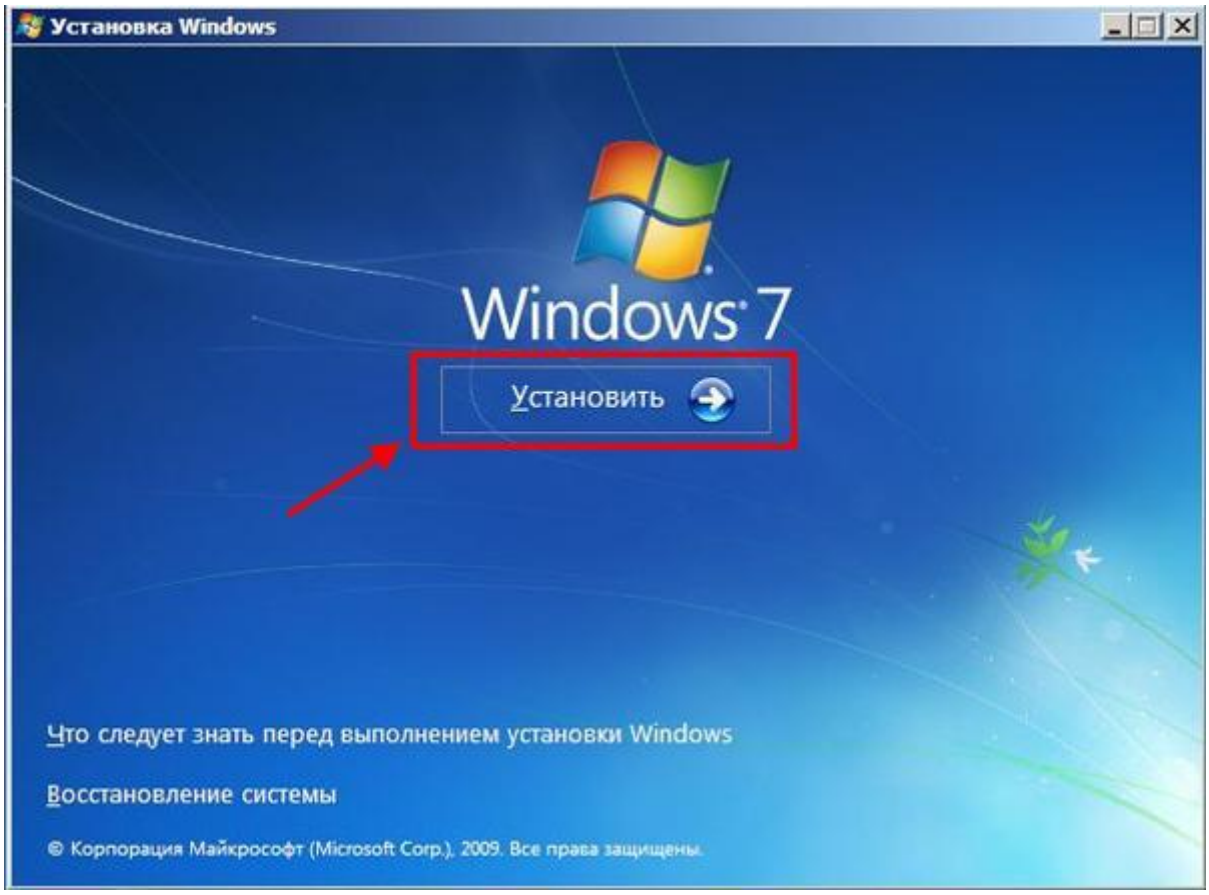
- Подождите, пока временные файлы будут распакованы на жёсткий диск.



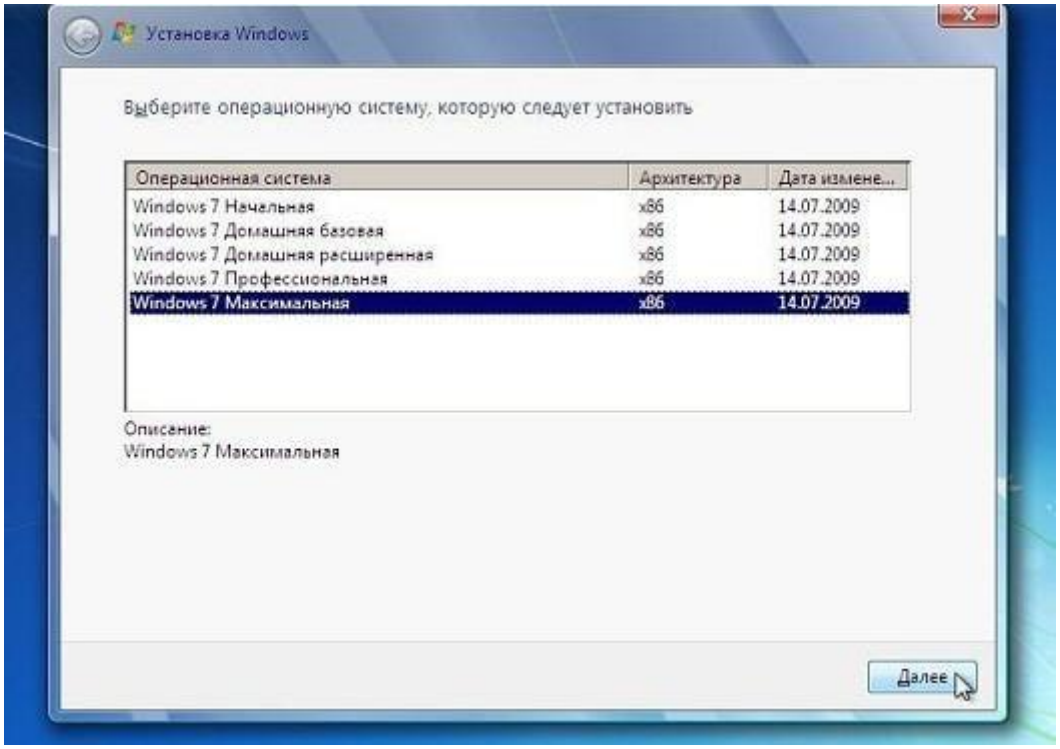
- Выберите язык интерфейса ОС, формат времени, раскладку клавиатуры, а затем нажмите «Далее».



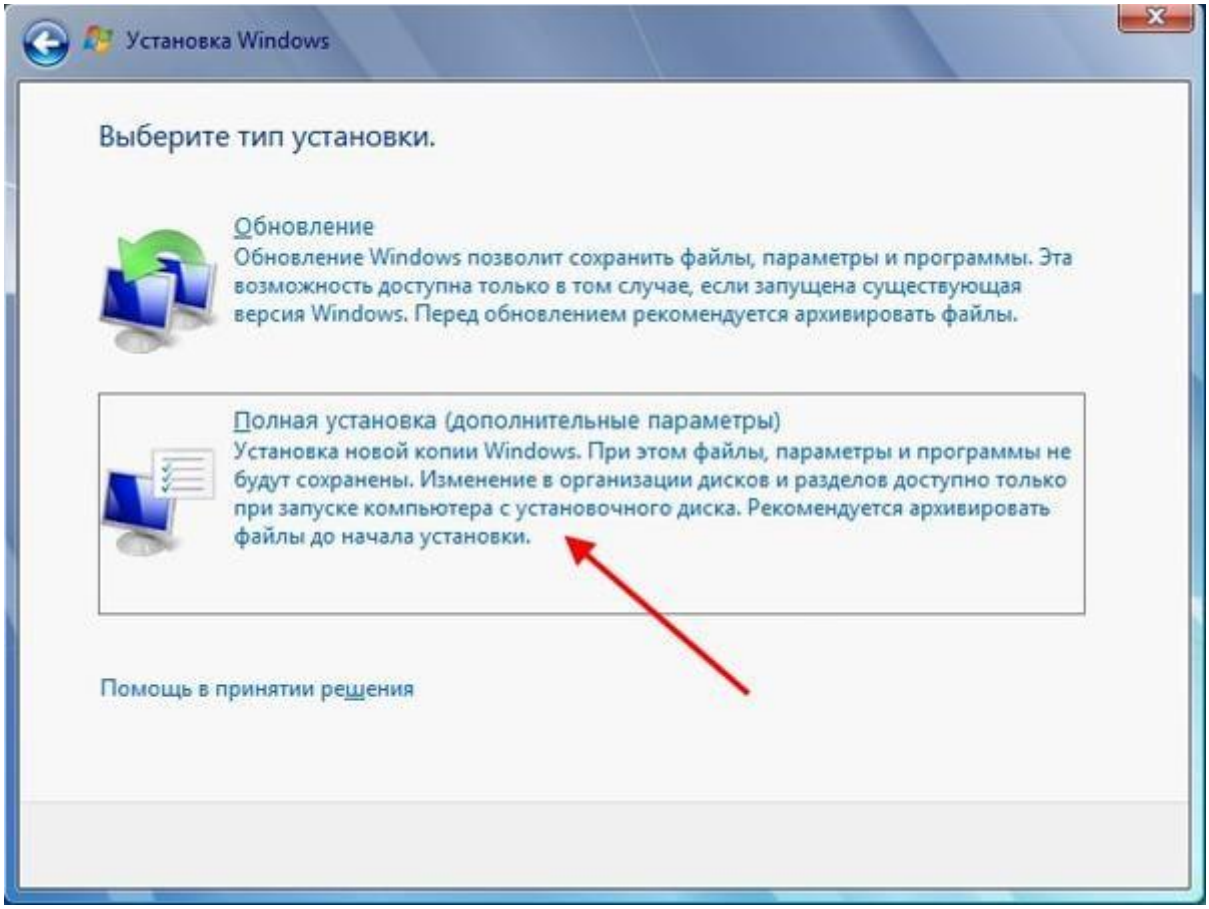
- Щёлкните по расположенной в центре новой страницы кнопке «Установить».



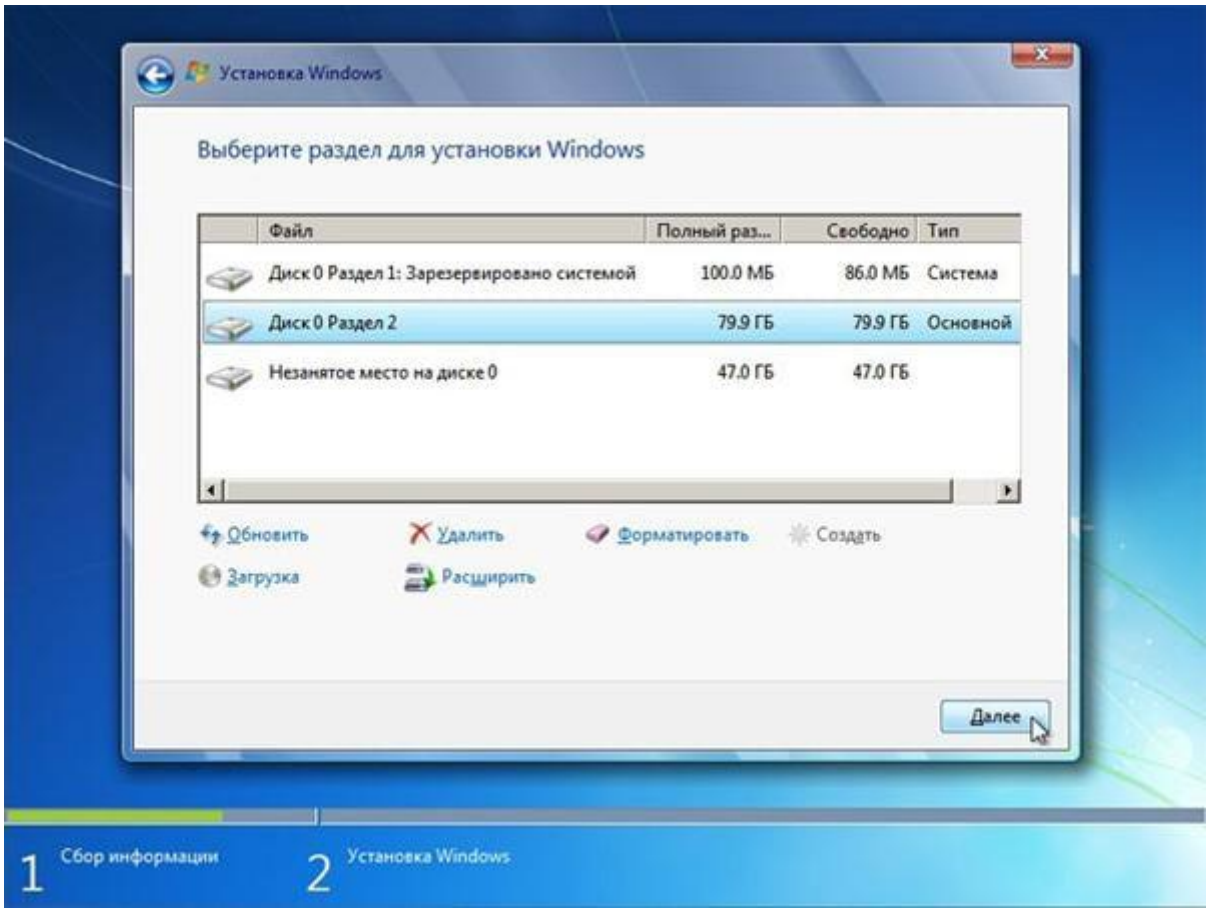
- Если образ включает несколько версий ОС, выберите подходящую. За один раз можно инсталлировать только один вариант.



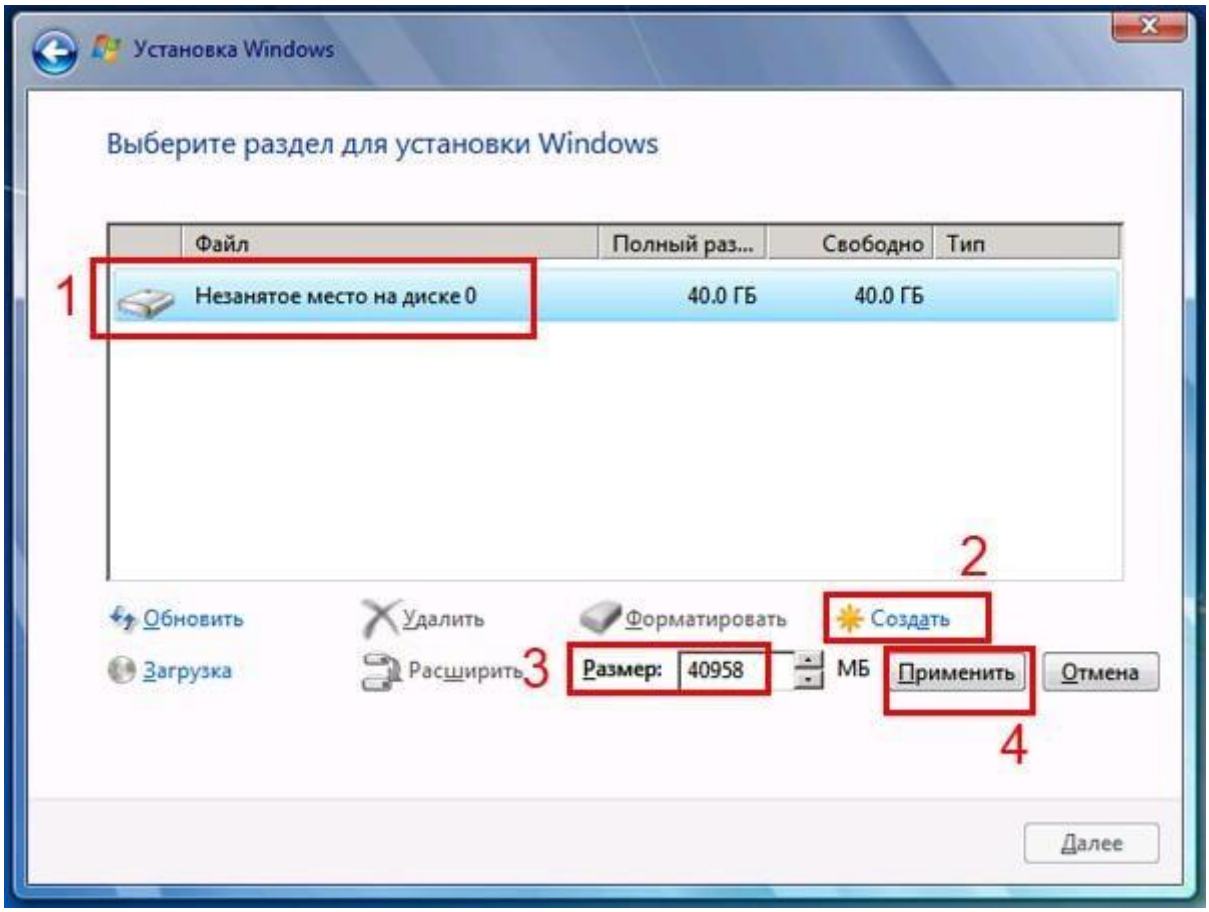
- Примите условия лицензионного соглашения и выберите вариант «Полная установка» вместо «Обновления».



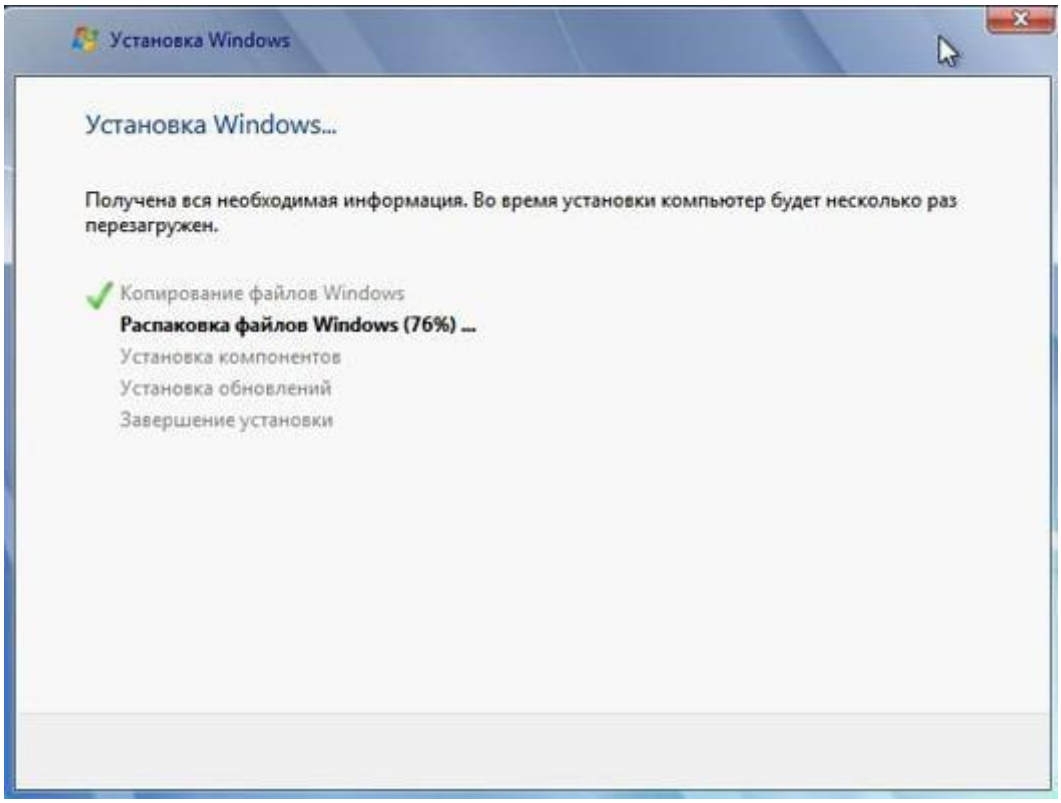
- Определите, в какой из разделов HDD будет установлена операционная система.



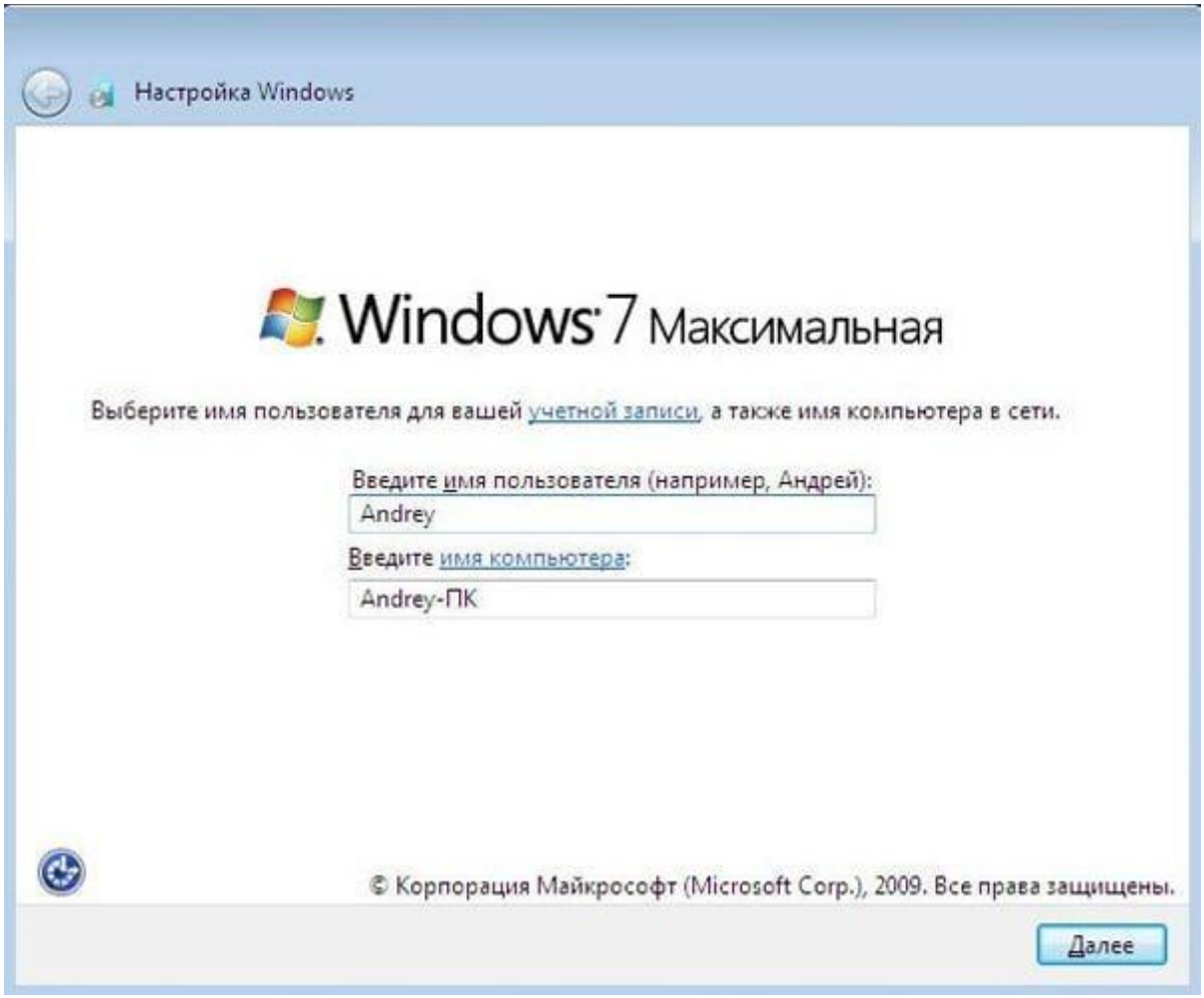
- Если ни одного раздела нет, предварительно создайте новый.



- Подождите, пока система будет распаковывать файлы, настраивать компоненты и обновления. Процесс займёт довольно много времени.



После нескольких перезагрузок «Виндовс» предложит вам придумать имя пользователя и пароль. На этом процесс установки будет завершён.



Ставим драйверы

Установка Windows с флешки или диска заканчивается инсталляцией драйверов — служебных программ, необходимых для корректной работы сетевой и звуковой карты, дисплея, периферийных устройств. Вы можете скачать драйверы по отдельности из интернета, использовать любую из авторских сборок — или установить утилиты с диска, прилагаемого производителем к вашему ПК или ноутбуку.

Тема 9 История и общая характеристика семейства операционных систем LINUX

Практическое занятие № 10 Установка ОС Linux

Цель: Научиться устанавливать ОС Linux

Ход работы

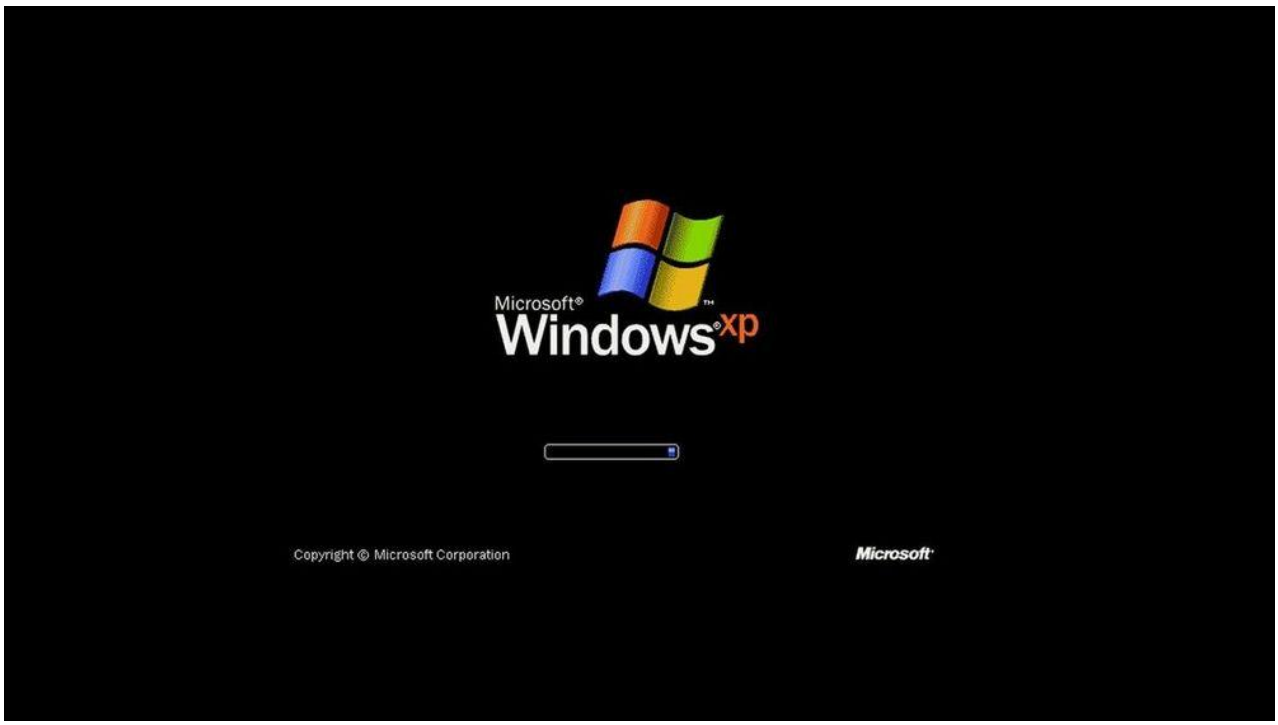
Установка Linux на компьютер

В первую очередь необходимо настроить компьютер для альтернативной загрузки через BIOS/UEFI.

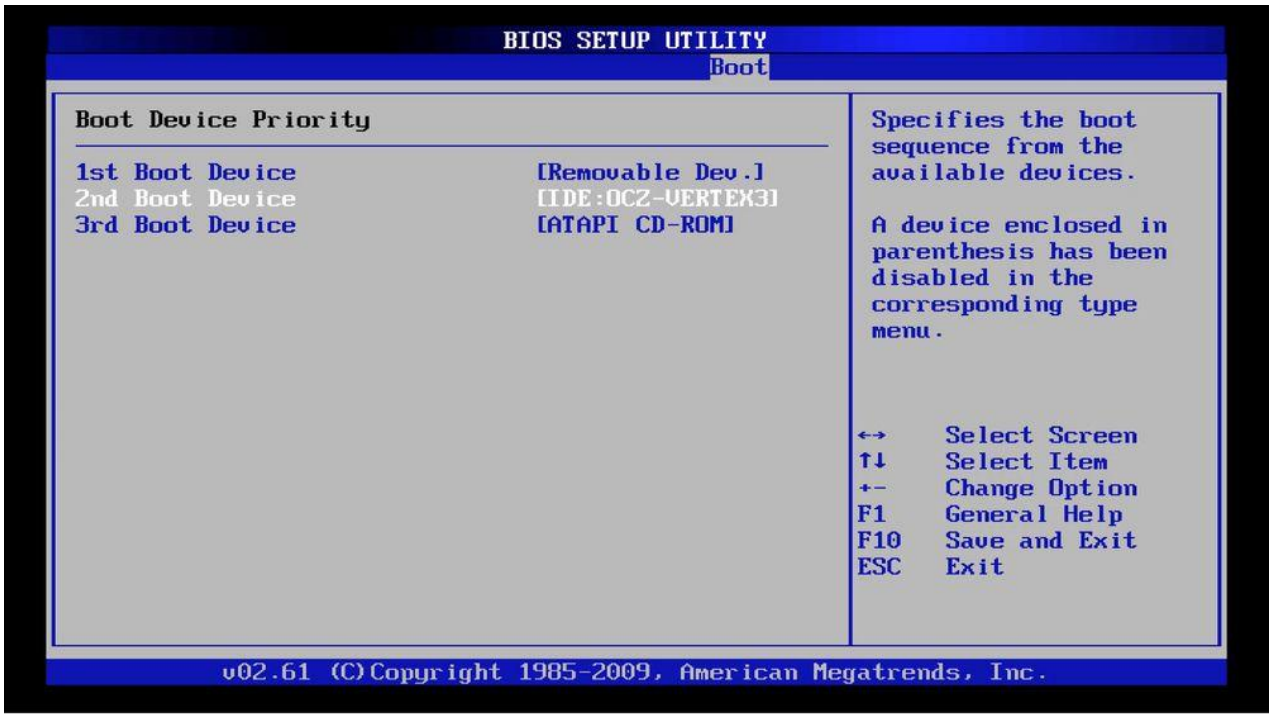
Процесс загрузки ПК останавливается до того, как запустится Windows, чтобы открыть меню настроек. Какие клавиши использовать – зависит от марки и модели устройства. Обычно подходит кнопка F1, F2 F9, F10, F12 или различные комбинации. Лучше всего узнать это, выполнив поиск в интернете по марке своего ноутбука.

В Windows 10 по умолчанию включена опция быстрой загрузки, поэтому необязательно знать кнопку перехода в BIOS. Достаточно зайти в параметры системы (Win+I), а оттуда в «Обновление» – «Восстановление» – «Особые варианты загрузки».

1. Выключить ПК и включить его снова. Сразу же после нажатия кнопки включения кликать по нужной клавише (примерно 5-10 раз).



2. Найти меню загрузки – может называться как «Boot List Option», «Boot Mode», «OS Mode Selection» и т.п.

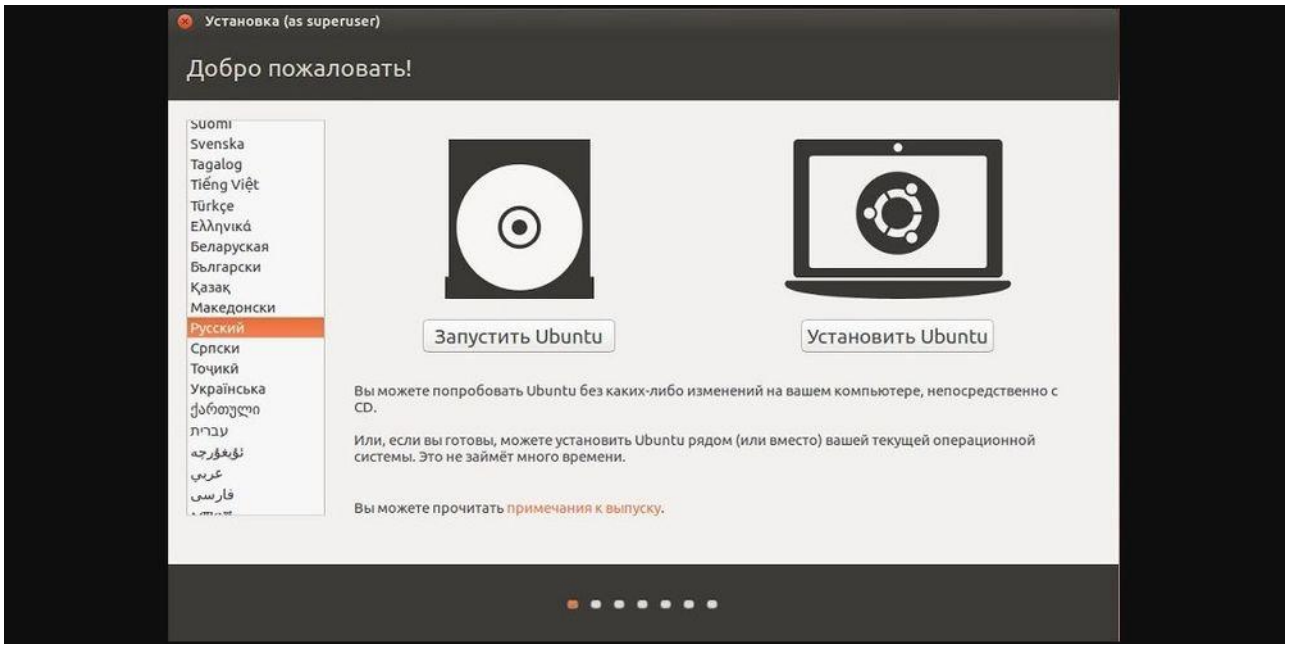


3. Установить порядок, в котором компьютер будет искать загрузочное устройство, начиная сверху вниз. Чтобы изменить первую запись, просто выбрать ее с помощью клавиш курсора и нажать Enter, а затем вместо жесткого диска указать USB.

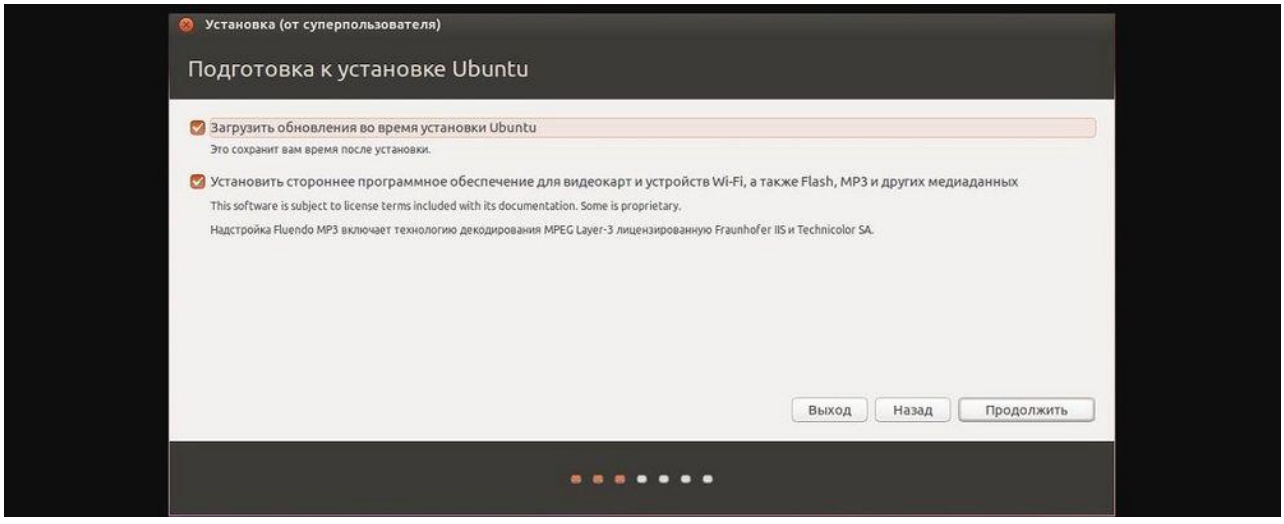
4. Выйти из меню BIOS.

Теперь следует вставить флешку в USB-разъем и перезагрузить компьютер. Он должен загрузить установщик Linux, а далее:

1. На экране появится новое окно, где выбирается язык интерфейса и режим загрузки, то есть «Установить».

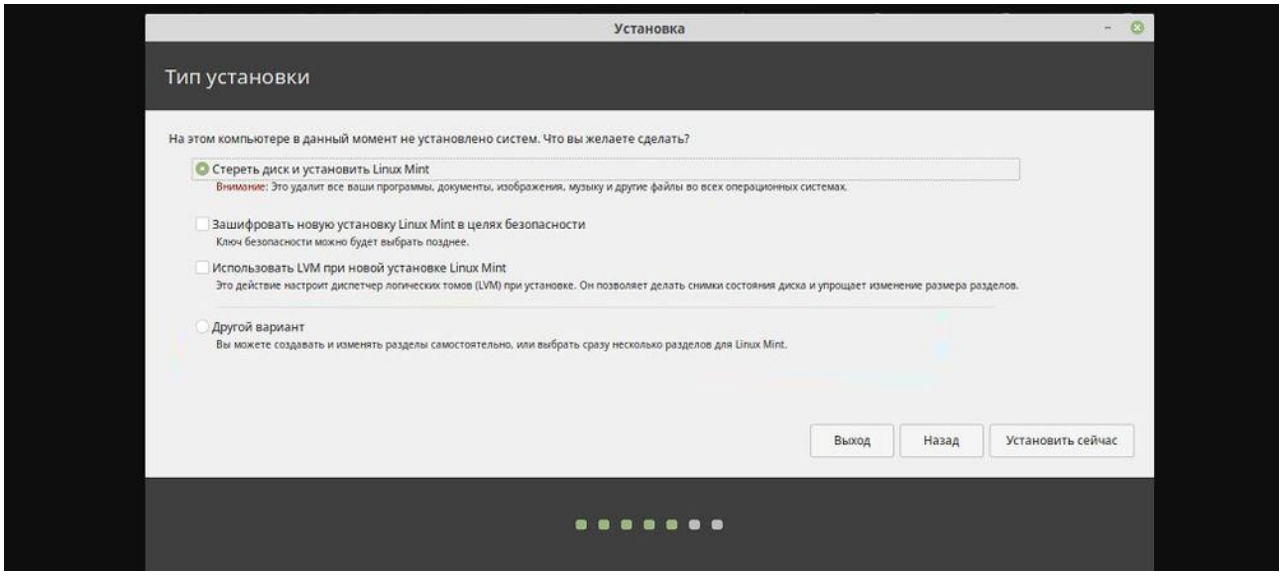


2. Отметить два поля на следующей странице перед тем, как нажать «Продолжить».

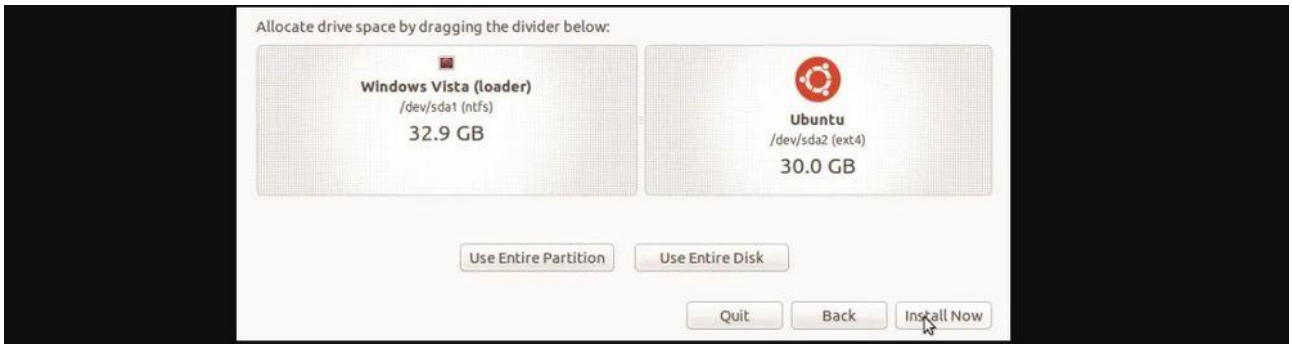


3. На этом шаге выбирается вариант установки. Если вместе с существующей установкой Windows будет устанавливаться дистрибутив, отметить пункт «Установить вместе с другими операционными системами». Если нужно стереть

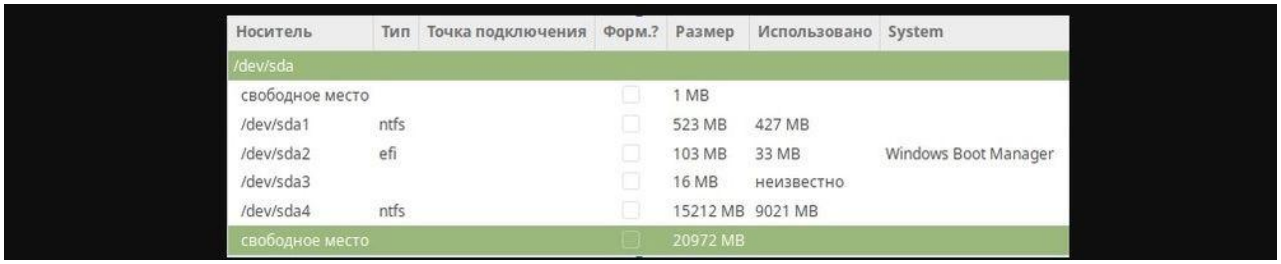
диск, отметить пункт «Стереть диск и установить».



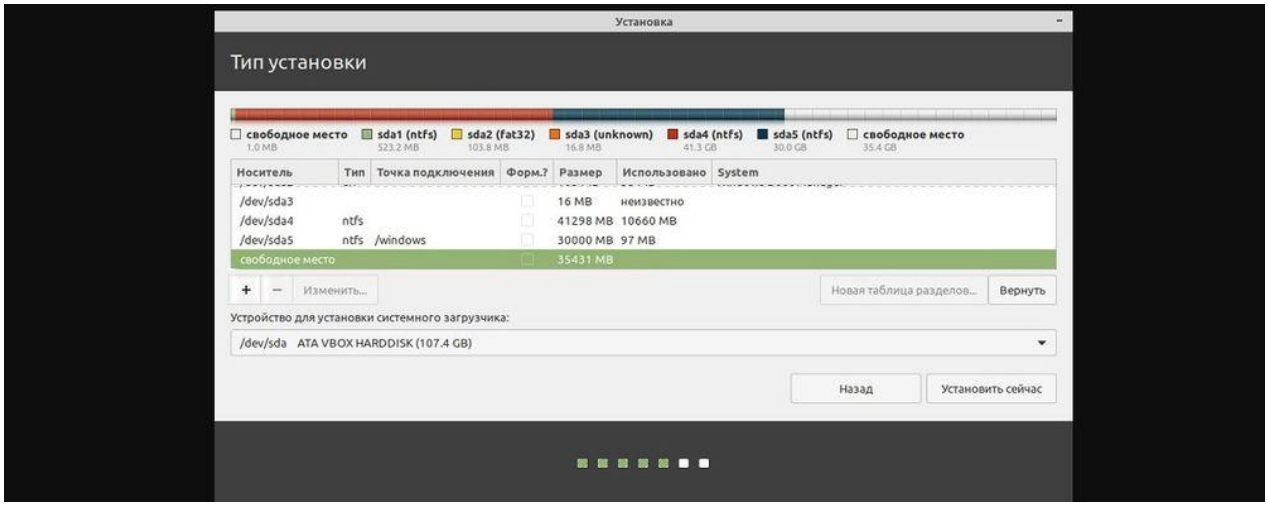
4. Отобразятся существующие разделы и то, как они будут разделены после установки. Чтобы изменить общий ресурс для дистрибутива или Windows, необходимо просто перетащить разделительную линию влево или вправо. Когда будет готово, нажать «Установить сейчас».



5. Если установка дистрибутива производится вместе с Windows, на следующей странице внизу в поле «Свободное место» будет указано заданное для Linux значение.

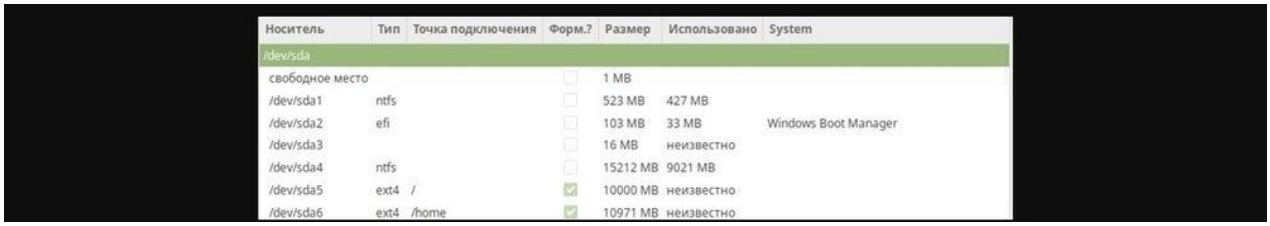


6. Для создания корневого и домашнего разделов кликнуть по кнопке «+».

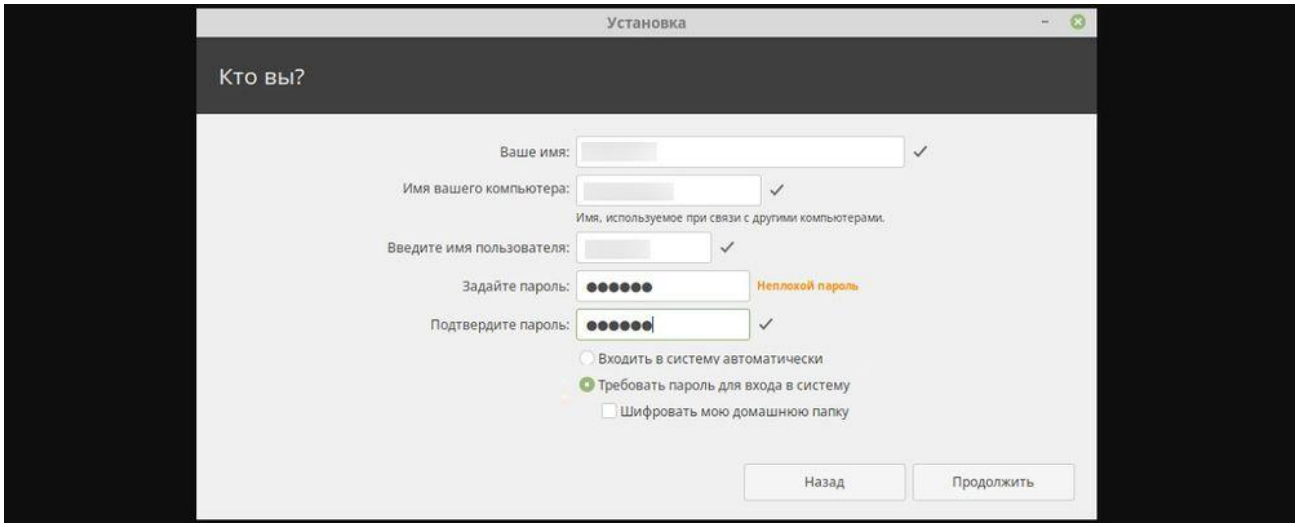


7. Задать необходимые параметры.

8. В итоге список обновится.



9. После установка продолжится выбором местоположения, раскладки клавиатуры и вводом своих данных в качестве исходного пользователя.



Когда установка завершится, Linux перезапустится, и можно начинать изучение системы.

Практическое занятие № 11 Работа с терминалом ОС UNIX

Цель: Научиться работать с терминалом

1.1 Вход с системной консоли

Вход в систему осуществляется с системной консоли, которая представляет собой монитор и клавиатуру, связанные непосредственно с системой. Как многопользовательская система UNIX предоставляет возможность работы в нескольких виртуальных символьных терминалах (виртуальных консолях), которые позволяют запускать программы в разных терминалах и от имени разных пользователей работать одновременно под несколькими именами или под одним именем и т.п.

Максимально возможное количество виртуальных терминалов - 12, по умолчанию установленная система представляет 6 виртуальных символьных терминалов и один графический. Переключение между терминалами осуществляется комбинацией клавиш - -первый терминал, - - второй терминал и т.д. Переключение из графического терминала в символьный осуществляется сочетанием трех функциональных клавиш - - *,* где # **-** номер символьного терминала.

При входе в систему на конкретном терминале пользователь видит приглашение **hostname login,** где **hostname -** имя машины, на которой регистрируется пользователь.

После успешного ввода имени пользователя и пароля система выводит приглашение к вводу команды.

- для суперпользователя **root;**

$ - для всех остальных пользователей.

Система готова к вводу команды, и пользователь может запустить утилиту тс, которая является удобной оболочкой работы с файловой системой.

$ mc

Часто при первом входе в систему пользователя требуется поменять пароль, назначенный пользователю администратором, - используйте команду **passwd.**

$ passwd

Выход из терминала осуществляется по команде **exit**

$ exit

1.2 Понятия login и password

Операционная система UNIX является многопользовательской операционной системой. Для обеспечения безопасной работы пользователей и целостности системы доступ к ней должен быть санкционирован. Для каждого пользователя, которому разрешен вход в систему, заводится специальное регистрационное имя - ***username*** или ***login*** **-** и сохраняется специальный пароль **-** ***password,*** соответствующий этому имени. Как правило, при регистрации нового пользователя начальное значение пароля для него задает системный администратор. После первого входа в систему пользователь должен изменить начальное значение пароля с помощью специальной команды. В дальнейшем он может в любой момент изменить пароль по своему желанию.

1.3 Упрощенное понятие об устройстве файловой системы в UNIX. Полные и относительные имена файлов

Понятие «файл» характеризует статическую сторону вычислительной системы. Все файлы, доступные в операционной системе UNIX, как и в уже известных вам операционных системах, объединяются в древовидную логическую структуру. Файлы могут объединяться в каталоги или директории. Не существует файлов, которые не входили бы в состав какой-либо директории. Директории, в свою очередь, могут входить в состав других директорий. Допускается существование пустых директорий, в которые не входит ни один файл, и ни одна другая директория (рис. 1.1). Среди всех директорий существует только одна директория, которая не входит в состав других директорий, - ее принято называть корневой. На настоящем уровне недостаточного знания UNIX можно заключить, что в файловой системе UNIX присутствует, по крайней мере, два типа файлов: обычные файлы, которые могут содержать тексты программ, исполняемый код, данные и т.д. (их принято называть регулярными файлами), и директории.

Рис. 1.1. ***Пример структуры файловой системы***

Каждому файлу (регулярному или директории) должно быть присвоено имя. В различных версиях операционной системы UNIX существуют те или иные ограничения на построение имени файла. В стандарте POSIX на интерфейс системных вызовов для операционной системы UNIX содержится лишь три явных ограничения:

• Нельзя создавать имена большей длины, чем это предусмотрено операционной системой (для Linux - 255 символов).

- • Нельзя использовать символ NUL (не путать с указателем **NULL!) -** он же символ с нулевым кодом, он же признак конца строки в языке Си.
- • Нельзя использовать символ 7'.

Единственным исключением является корневая директория, которая всегда имеет имя Эта же директория представляет собой единственный файл, который должен иметь уникальное имя во всей файловой системе. Для всех остальных файлов имена должны быть уникальными только в рамках той директории, в которую они непосредственно входят. Каким же образом отличить два файла с именами «ааа.с», входящими в директории «Ь» и «б» на рисунке 1.1, что было понятно о каком из них идет речь? Здесь на помощь приходит понятие полного имени файла. Мысленно построим путь от корневой вершины дерева файлов к интересующему нас файлу и выпишем все имена файлов (т.е. узлов дерева), встречающиеся на нашем пути, например, 7 usr b aaa.c". В этой последовательности первым будет всегда стоять имя корневой директории, а последним -имя интересующего нас файла. Отделим

имена узлов друг от друга в этой записи не пробелами, а символами "/", за исключением имени корневой директории и следующего за ним имени ("/usr/b/aaa.c"). Полученная запись однозначно идентифицирует файл во всей логической конструкции файловой системы. Такая запись и получила название полного имени файла.

1.4 Понятие текущей директории. Команда pwd. Относительные имена файлов

Для каждой работающей программы в операционной системе, включая командный интерпретатор (shell), который обрабатывает вводимые команды и высвечивает приглашение к их вводу, одна из директорий в логической структуре файловой системы назначается текущей или рабочей для данной программы. Узнать, какая директория является текущей для вашего командного интерпретатора, можно с помощью команды операционной системы ***pwd.***

Домашняя директория пользователя и ее определение. Для каждого нового пользователя в системе заводится специальная директория, которая становится текущей сразу после его входа в систему. Эта директория получила название домашней директории пользователя. Воспользуйтесь командой ***pwd*** для определения своей домашней директории.

1.5 Команда man - универсальный справочник

По ходу изучения операционной системы UNIX вам часто будет требоваться информация о том, что делает та или иная команда или системный вызов, какие у них параметры и опции, для чего предназначены некоторые системные файлы, каков их формат и т. д. Большая часть информации в UNIX Manual доступна в интерактивном режиме с помощью утилиты man.

Пользоваться утилитой man достаточно просто: наберите команду

$ man имя, где «имя» - это имя интересующей вас команды, утилиты, системного вызова, библиотечной функции или файла. Посмотрите с ее помощью информацию о команде pwd.

Чтобы пролистать страницу полученного описания, если оно не поместилось на экране полностью, следует нажать клавишу ***<пробел>.*** Для прокрутки одной строки воспользуйтесь клавишей **.** Вернуться на страницу назад позволит

одновременное нажатие клавиш ***и <Ь>.*** Выйти из режима просмотра информации можно с помощью клавиши ***.***

1.6 Команды cd для смены текущей директории и ls для просмотра состава директории

Для смены текущей директории командного интерпретатора можно воспользоваться командой cd (change directory). Для этого необходимо набрать команду в виде

$ cd имядиректории, где «имя_директории» - полное или относительное имя директории, которую вы хотите сделать текущей. Команда cd без параметров сделает текущей директорией вашу домашнюю директорию.

Просмотреть содержимое текущей или любой другой директории можно, воспользовавшись командой ls (от list). Если ввести ее без параметров, эта команда распечатает вам список файлов, находящихся в текущей директории. Если же в качестве параметра задать полное или относительное имя директории:

$ ls имядиректории, - то она распечатает список файлов в указанной директории. Надо отметить, что в полученный список не войдут файлы, имена которых начинаются с символа «точка» - «.». Такие файлы обычно создаются различными системными программами для своих целей (например, для настройки).

Посмотреть полный список файлов можно, дополнительно указав команде ls опцию -a, т.е. набрав ее в виде

$ ls -a или $ ***ls -a имя директории***

У команды ls существует и много других опций.

Команда ls с опциями -al. Позволяет получить подробную информацию о файлах в некоторой директории, включая имена хозяина, группы хозяев и права доступа, можно с помощью уже известной нам команды ls с опциями -al.

$ ls -al

В выдаче этой команды третья колонка слева содержит имена пользователей хозяев файлов, а четвертая колонка слева - имена групп хозяев файла. Крайняя левая колонка содержит типы файлов и права доступа к ним. Тип файла определяет первый символ в наборе символов. Если это символ 'd', то тип файла - директория, если там стоит символ то это регулярный файл. Следующие три символа определяют права

доступа для хозяина файла, следующие три - для пользователей, входящих в группу хозяев файла, и последние три -для всех остальных пользователей. Наличие символа (r, w или x), соответствующего праву, для некоторой категории пользователей означает, что данная категория пользователей обладает этим правом.

Для получения полной информации о команде ls воспользуйтесь утилитой man.

1.7 Команда cat и создание файла. Перенаправление ввода и вывода

Вы уже умеете перемещаться по логической структуре файловой системы и рассматривать ее содержимое. Следует уметь также и просматривать содержимое файлов, и создавать их.

Для просмотра содержимого небольшого текстового файла на экране можно воспользоваться командой cat.

Если набрать ее в виде

$ cat имя_файла, то на экран выведется все его содержимое.

Если ваш текстовый файл большой, то вы увидите только его последнюю страницу. Большой текстовый файл удобнее рассматривать с помощью утилиты more (описание ее использования вы найдете в UNIX Manual).

Если в качестве параметров для команды cat задать не одно имя, а имена нескольких файлов:

$ cat файлі файл2 ... файлів, то система выдаст на экран их содержимое в указанном порядке. Вывод команды cat можно перенаправить с экрана терминала в какой-нибудь файл, воспользовавшись символом перенаправления выходного потока данных - знаком «>» («больше»).

Команда ***$ cat файлі файл2 ... файлУ > файлрезультата*** сольет содержимое всех файлов, чьи имена стоят перед знаком ">", воедино в файл_результата - конкатенирует их (от англ. ‘concatenate’ -объединять - и произошло название команды).

Прием перенаправления выходных данных со стандартного потока вывода (экрана) в файл является стандартным для всех команд, выполняемых командным интерпретатором. Вы можете получить файл, содержащий список всех файлов

текущей директории, если выполните команду ls-a с перенаправлением выходных данных:

$ ls -a > новыйфайл

Если имена входных файлов для команды cat не заданы, то она будет использовать в качестве входных данных информацию, которая вводится с клавиатуры, до тех пор, пока вы не наберете признак окончания ввода - комбинацию клавиш ***и .***

Таким образом, команда

$ cat > новый файл позволяет создать новый текстовый файл с именем «новый файл» и содержимым, которое пользователь введет с клавиатуры.

У команды cat существует множество различных опций. Посмотреть ее полное описание можно в UNIX Manual.

Заметим, что наряду с перенаправлением выходных данных существует другой способ перенаправить входные данные. Если во время выполнения некоторой команды требуется ввести данные с клавиатуры, можно положить их заранее в файл, а затем перенаправить стандартный ввод этой команды с помощью знака «меньше» - «<» - и следующего за ним имени файла с входными данными.

1.8 Шаблоны имен файлов

Шаблоны имен файлов могут применяться в качестве параметра для задания набора имен файлов во многих командах операционной системы. При использовании шаблона просматривается вся совокупность имен файлов, находящихся в файловой системе, и те имена, которые удовлетворяют шаблону, включаются в набор. В общем случае шаблоны могут задаваться с использованием следующих метасимволов:

- • * - соответствует всем цепочкам литер, включая пустую;
- • ? - соответствует всем одиночным литерам;
- • [•••]- соответствует любой литере, заключенной в скобки.

Пара литер, разделенных знаком минус, задает диапазон литер.

Так, например, шаблону *.c удовлетворяют все файлы текущей директории, чьи имена заканчиваются на .c. Шаблону [a-c1]*удовлетворяют все файлы текущей директории, чьи имена начинаются с букв a, b, c, d. Существует одно ограничение на

использование метасимвола * в начале имени файла, например, в случае шаблона *с. Для таких шаблонов имена файлов, начинающиеся с символа точка, считаются не соответствующими шаблону.

1.9 Простейшие команды работы с файлами - cp, rm, mkdir, mv

Для нормальной работы с файлами необходимо не только уметь создавать файлы, просматривать их содержимое и перемещаться по логическому дереву файловой системы. Нужно уметь создавать собственные поддиректории, копировать и удалять файлы, переименовывать их. Это минимальный набор операций, не владея которым, нельзя чувствовать себя уверенно при работе с компьютером.

Для создания новой поддиректории используется команда mkdir (сокращение от make directory). В простейшем **виде** команда выглядит следующим образом:

$ mkdir имядиректории, где «имядиректории» - полное или относительное имя создаваемой директории. У команды mkdir имеется набор опций, описание которых можно просмотреть с помощью утилиты man.

Для копирования файлов и директорий применяется команда cp. Данная команда может применяться в следующих формах.

$ cp файлисточник файлназначения - служит для копирования одного файла с именем «файл источник» в файл с именем «файл_назначения».

Команда cp в форме

$ cp файлі файл2 ... файлN дир_назначения - служит для копирования файла или файлов с именами *«файлі», «файл2», ... «файлів»* в уже существующую директорию с именем «дирназначения» под своими именами. Вместо имен копируемых файлов могут использоваться их шаблоны.

$ cp -г диристочник дирназначения - служит для рекурсивного копирования одной директории с именем «дир источник» в новую директорию с именем «дир назначения». Если директория «дир назначения» уже существует, то мы получаем команду cp в следующей форме

$ cp -г дир! дир2 ... дирN дир назначения - служит для рекурсивного копирования директории или директорий с именами «дирі», «дир2», ... «дирМ» в уже

существующую директорию с именем «дир_назначения» под своими собственными именами. Вместо имен копируемых директорий могут использоваться их шаблоны.

Для удаления файлов или директорий применяется команда rm (сокращение от remove). Если вы хотите удалить один или несколько регулярных файлов, то простейший вид команды rm будет выглядеть следующим образом:

$ rm файл! файл2 ... файлів, где «файлі», «файл2», ... «файлЫ» - полные или относительные имена регулярных файлов, которые требуется удалить. Вместо имен файлов могут использоваться их шаблоны. Если вы хотите удалить одну или несколько директорий вместе с их содержимым (рекурсивное удаление), то к команде добавляется опция -r:

$ rm -r dirі dir2 ... dirN, где «дирі», «дир2», ... «дирМ» -полные или относительные имена директорий, которые нужно удалить. Вместо непосредственно имен директорий также могут использоваться их шаблоны.

У команды rm есть еще набор полезных опций, которые описаны в UNIXManual и могут быть просмотрены с помощью команды man. Командой удаления файлов и директорий следует пользоваться с осторожностью. Удаленную информацию восстановить невозможно. Если вы системный администратор и ваша текущая директория - это корневая директория, пожалуйста, не выполняйте команду rm -r *!

Для перемещения файлов и директорий используется команда mv (сокращение от move). Данная команда может применяться в следующих формах:

$ mv имяисточника имяназначения - для переименования или перемещения одного файла (неважно, регулярного или директории) с именем «имяисточника» в файл с именем «имя_назначения». При этом перед выполнением команды файла с именем «имя_назначения» существовать не должно.

$ mv имя! имя2 ... имяК дирназначения - служит для перемещения файла или файлов (неважно, регулярных файлов или директорий) с именами «имяі», «имя2», ... «имяЫ» в уже существующую директорию с именем «дирназначения» под собственными именами. Вместо имен перемещаемых файлов могут использоваться их шаблоны.

1.10 Пользователь и группа. Команды chown и chgrp. Права доступа к файлу

Как уже говорилось, для входа в операционную систему UNIX каждый пользователь должен быть зарегистрирован в ней под определенным именем. Вычислительные системы не умеют оперировать именами, поэтому каждому имени пользователя в системе соответствует некоторое числовое значение - его идентификатор UID (user identificator).

Все пользователи в системе делятся на группы. Например, студенты одной учебной группы могут составлять отдельную группу пользователей. Группы пользователей также получают свои имена и соответствующие идентификационные номера - GID (group identificator). В одних версиях UNIX каждый пользователь может входить только в одну группу, в других - в несколько групп.

Команда chown предназначена для изменения собственника (хозяина) файлов. Нового собственника файла могут назначить только предыдущий собственник файла или системный администратор.

$ chown owner файлі файл2... файлУ - параметр owner задает нового собственника файла в символьном виде, как его username, или в числовом виде, как его UID. Параметры «файлі», «файл2», ... «файлК» - это имена файлов, для которых производится изменение собственника. Вместо имен могут использоваться их шаблоны.

Для каждого файла, созданного в файловой системе, запоминаются имена его хозяина и группы хозяев. Заметим, что группа хозяев не обязательно должна быть группой, в которую входит хозяин. Упрощенно можно считать, что в операционной системе Linux при создании файла его хозяином становится пользователь, создавший файл, а его группой хозяев - группа, к которой этот пользователь принадлежит. Впоследствии хозяин файла или системный администратор могут передать его в собственность другому пользователю или изменить его группу хозяев с помощью команд chown и chgrp.

Команда chgrp предназначена для изменения группы собственников (хозяев) файлов.

S ***chgrp group файлі файл2 ... фашМ -*** новую группу собственников файла могут назначить только собственник файла или системный администратор. Параметр group задает новую группу собственников файла в символьном виде, как имя группы, или в числовом виде, как ее GID. Параметры «файлі», «файл2», ... «файлИ» - это

имена файлов, для которых производится изменение группы собственников. Вместо имен могут использоваться их шаблоны.

Для каждого файла выделяется три категории пользователей:

- • пользователь, являющийся хозяином файла;
- • пользователи, относящиеся к группе хозяев файла;
- • все остальные пользователи.

Для каждой из этих категорий хозяин файла может определить различные права доступа к файлу. *Различают три вида прав доступа: право на чтение файла - г (от слова read), право на модификацию файла - w (от слова write) и право на исполнение файла — x (от слова execute).*

Команда chmod предназначена для изменения прав доступа к одному или нескольким файлам.

$ chmod [who] { + I - | = } [perm] файлі файл2 ... файлN -права доступа к файлу могут менять только собственник (хозяин) файла или системный администратор.

Параметр *who* определяет, для каких категорий пользователей устанавливаются права доступа. Он может представлять собой один или несколько символов:

a - установка прав доступа для всех категорий пользователей. Если параметр who не задан, то по умолчанию применяется a. При определении прав доступа с этим значением заданные права устанавливаются с учетом значения маски создания файлов;

u - установка прав доступа для собственника файла;

g - установка прав доступа для пользователей, входящих в группу собственников файла;

O - установка прав доступа для всех остальных пользователей. Операция, выполняемая над правами доступа для заданной категории пользователей, определяется одним из следующих символов:

+ - добавление прав доступа;

- - отмена прав доступа;

= - замена прав доступа, т.е. отмена всех существовавших и добавление перечисленных.

Если параметр *perm* не определен, то все существовавшие права доступа отменяются.

Параметр *perm* определяет права доступа, которые будут добавлены, отменены или установлены взамен соответствующей командой. Он представляет собой комбинацию следующих символов или один из них:

r - право на чтение;

w - право на модификацию;

x - право на исполнение.

Параметры ***файл1, файл2,... файлN -*** это имена файлов, для которых производится изменение прав доступа. Вместо имен могут использоваться их шаблоны.

Хозяин файла может изменять права доступа к нему, пользуясь командой chmod.

1.11 Вход удаленным пользователем

Для входа удаленным пользователем в систему UNIX используется утилита ssh (security shell). Для доступа к другим UNIX системам с UNIX машины

$ ssh -l <Имя пользователя> <1P адрес удаленной машины> Пользователь может набрать команду

$ ssh -l <Имя пользователя> localhost для доступа по ssh к «своей» (локальной) машине.

1.12 Команды write и wall

Очень часто устанавливают многопользовательскую поддержку, многие люди работают на том же сервере через различные удаленные доступные рабочие варианты. Однако все эти пользователи системы Linux могут работать на соответствующем проекте или в команде, и даже если они не связаны, они принадлежат к одному рабочему месту. Таким образом, вполне вероятно, им придется общаться в любой момент времени. **C** утилитой Linux **write,** Linux пользователи имеют удобный способ общения друг с другом. Можно послать сообщение любому

другому пользователю и разрешить войти в Linux машину. Более того, можно посылать сообщения любому пользователю в той же сети, даже на другой машине хозяина.

Синтаксис выглядит так: **$ write person** Здесь **person -** имя

пользователя, если он находится на той же машине, машины или username @hostname в случае, если пользователь принадлежит к другой машине хозяина и необходима идентификация, когда один пользователь заходит более чем один раз. Рассмотрим, как передается сообщения среди пользователей на той же машине.

$ w -s — данная команда выводит всех активных пользователей системы;

$write person — вместо **person** мы можем вписать имя любого активного пользователя. После нажатия клавиши ввода появляется возможность писать наше сообщение.

Команда **wall** используется для передачи сообщения всем пользователям системы. Однако для получения этого сообщения пользователи должны установить разрешение их mesg на "да". В использовании довольно проста, что легко понять на следующем примере;

$ wall - как только ввод сообщения завершен, нужно нажать комбинацию клавиш **+ .**

После этого все пользователи получают сообщение.

Таблица 1.1 - Основные информационные команды

Команды	**Описание**
hostname	Вывести или изменить сетевое имя машины.
whoami	Вывести имя, под которым я зарегистрирован.
date	Вывести или изменить дату и время.

time	Получить информацию о времени, нужном для выполнения процесса
who	Определить, кто из пользователей работает на машине.
rwho -a	Определение всех пользователей, подключившихся к вашей сети. Для выполнения этой команды требуется, чтобы был запущен процесс rwho. Если такого нет -запустите "setup" под суперпользователем.
finger [имя_пользователя]	Системная информация о зарегистрированном пользователе. Попробуйте: finger root
ps -a	Список текущих процессов
df-h	(=место на диске) Вывести информацию о свободном и используемом месте на дисках (в читабельном виде).

Окончание табл. 1.1

Arch или uname -m uname -r	отобразить архитектуру компьютера отобразить используемую версию ядра
find / -name file1 find / -user user1	найти файлы и директории с именем file1. Поиск начать с корня (/) найти файл и директорию, принадлежащие пользователю user1. Поиск начать с корня (/)
top	отобразить запущенные процессы,

	используемые ими ресурсы и другую полезную информацию (с автоматическим обновлением данных)
Kill-9 98989 или kill -KILL 98989	«убить» процесс с PID 98989 «на смерть» (без соблюдения целостности данных)

1.13 Командный интерпретатор Shell

Командный интерпретатор в среде UNIX выполняет две основные функции:

- - представляет интерактивный интерфейс с пользователем, т. е. выдает приглашение и обрабатывает вводимые пользователем команды;
- - обрабатывает и исполняет текстовые файлы, содержащие команды интерпретатора (командные файлы).

В последнем случае операционная система позволяет рассматривать командные файлы как разновидность исполняемых файлов. Соответственно, различают два режима работы интерпретатора: интерактивный и командный.

Существует несколько типов оболочек в мире UNIX. Две главные - это «Bourne shell» и «C shell». Bourne shell (или просто shell) использует командный синтаксис, похожий на первоначальный для UNIX. В большинстве UNIX-систем Bourne shell имеет имя /bin/sh (где sh сокращение от «shell»). C shell используется иной синтаксис, чем-то напоминающий синтаксис языка программирования Си. В большинстве UNIX-систем он имеет имя /bin/csh.

В Linux есть несколько вариаций этих оболочек. Две наиболее часто используемые - это Новый Bourne shell (Bourne Again Shell) или

«Bash» (/bin/bash) и Tcsh (/bin/tcsh). Bash - это развитие прежнего shell с добавлением многих полезных возможностей, частично содержащихся в C shell.

Поскольку Bash можно рассматривать как надмножество синтаксиса прежнего shell, любая программа, написанная на sh shell должна работать и в Bash. Tcsh является расширенной версией C shell.

При входе в систему пользователю загружается командный интерпретатор по умолчанию. Информация о том, какой интерпретатор использовать для конкретного пользователя, находится в файле /etc/passwd.

Настройка Shell. Файлы инициализации, используемые в bash: /etc/profile (устанавливается системным администратором, выполняется всеми экземплярами начальных пользовательских bash, вызванными при входе пользователей в систему), $HOME/.bash_profile (выполняется при входе пользователя) и $HOME/.bashrc (выполняемый всеми прочими не начальными экземплярами bash). Если .bash_profile отсутствует, вместо него используется .profile. Переменная HOME указывает на домашний каталог пользователя, tcsh использует следующие сценарии инициализации: /etc/csh.login (выполняется всеми пользовательскими tcsh в момент входа в систему), $HOME/.tcshrc (выполняется во время входа в систему и всеми новыми экземплярами tcsh) и $HOME/.login (выполняется во время входа после .tcshrc). Если .tcshrc отсутствует, вместо него используется .cshrc.

Командные файлы. Командный файл в UNIX представляет собой обычный текстовый файл, содержащий набор команд UNIX и команд Shell.

Для того чтобы командный интерпретатор воспринимал этот текстовый файл как командный, необходимо установить атрибут на исполнение.

Установку атрибута на исполнение можно осуществить командой chmod или через mc по клавише F9 выйти в меню и выбрать вкладку File, далее выбрать изменение атрибутов файла.

Например.

$ echo “ps -af” > commandfile

$ chmod +x *commandfile*

$ ./commandfile

В представленном примере команда echo “ ps -af ” > commandfile создаст файл с одной строкой “ ps -af “, команда chmod 4-x commandfile установит атрибут на исполнение для этого файла, команда ./commandfile осуществит запуск этого файла.

Переменные shell. Имя shell-переменной - это начинающаяся с буквы последовательность букв, цифр и подчеркиваний. Значение shell-переменной - строка символов.

Например: Var = “ String ” или Var = String

Команда *echo $Var* выведет на экран содержимое переменной Var, т. е. строку ‘String’ , на то, что мы выводим содержимое переменной, указывает символ ‘$’.

Так, команда echo Var выведет на экран просто строку ‘Var’.

Еще один вариант присвоения значения переменной *Var = ' набор команд UNIX'*

Обратные кавычки говорят о том, что сначала должна быть выполнена заключенная в них команда, а результат ее выполнения, вместо выдачи на стандартный выход, приписывается в качестве значения переменной.

CurrentDate = date' - Переменной CurrentDate будет присвоен результат выполнения команды date.

Можно присвоить значение переменной и с помощью команды «read», которая обеспечивает прием значения переменной с (клавиатуры) дисплея в диалоговом режиме.

Например:

echo “Введите число”

read XI

echo “вы ввели $X1

Несмотря на то что shell-переменные в общем случае воспринимаются как строки, т. е. «35» - это не число, а строка из двух символов «3» и «5», в ряде случаев они могут интерпретироваться иначе, например, как целые числа.

Разнообразные возможности имеет команда "expr".

Например, командный файл:

x=7

j=2

rez=expr $x + $y'

echo резулыпат=$гег

выдаст на экран результат=9

Порядок выполнения работы:

- 1. Объясните основные моменты работы с системой UNIX в терминальном режиме: вход в систему обычным символьным терминалом,

переключение между терминалами; регистрация удаленных терминалов с помощью протокола ssh; запуск утилиты tc; получение информации о пользователях, зарегистрированных в системе (команды who, w, finger).

- 2. Объясните организацию структуры каталогов в UNIX, рассмотрите основные каталоги /etc, /bin, /usr, /proc, их назначение.
- 3. Рассмотрите основные информационные команды и команды управления процессами.
- 4. Рассмотрите настройку shell (bash) и переменных среды окружения.
- 5. Рассмотрите основы написания сценариев на языке shell, изучите основные команды языка shell (bash).
- 6. Напишите свой собственный сценарий на языке shell с использованием изученных команд.
- 7. Получите подробную информацию о файлах домашней директории.
- 8. Создайте новый файл и посмотрите на права доступа к нему, установленные системой при его создании.
- 9. Убедитесь, что вы находитесь в своей домашней директории, и создайте новый текстовый файл. Введите туда информацию: ФИО студента, № группы. Скопируйте этот файл в другую директорию.
- 10. Реализуйте командный файл, который выводит: дату, системную информацию и текущего пользователя.