

004.4(07)

М 545



№ 5636

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И
НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное
образовательное учреждение
высшего образования

«Южный федеральный университет»
Инженерно-технологическая академия
Институт компьютерных технологий и
информационной безопасности

Методические указания
к выполнению комплекса лабораторных работ
по общеинститутской дисциплине
«Операционные системы»

Для студентов всех направлений подготовки бакалавров и специалистов
Института компьютерных технологий и информационной безопасности

Таганрог

Издательство Южного федерального университета

2016

УДК 004.451(07.07)
ББК 32.973-018.2
М 545

Рецензент:

зав.кафедрой систем автоматического управления Южного федерального университета, доктор технических наук, профессор *В.И. Финаев*.

Нужнов, Е. В.

М545 Методические указания к выполнению комплекса лабораторных работ по общеинститутской дисциплине «Операционные системы» / Е. В. Нужнов, А. Н. Самойлов, А. Н. Беликов; Южный федеральный университет. – Таганрог: Издательство Южного федерального университета, 2016. – 143 с.

Комплекс лабораторных работ охватывает темы: «Установка и настройка ОС MS Windows», «Возможности и средства ОС MS Windows», «Средства сохранения и восстановления программной среды ОС MS Windows», «Установка и настройка ОС Linux или иной ОС по выбору (Solaris, UNIX, MacOS, и др.)», «Возможности и средства ОС Linux или иной ОС по выбору», «Освоение перспективных типов ОС (Live CD/DVD ОС, Web-ОС)».

Для студентов всех направлений подготовки бакалавров и специалистов Института компьютерных технологий и информационной безопасности. Методические указания могут быть использованы студентами других направлений ЮФУ, связанных с использованием перспективных информационных технологий.

Ил. 101. Табл. 8. Библиогр.: 53 назв.

УДК 004.451 (07.07)
ББК 32.973-018.2

ВВЕДЕНИЕ

В настоящее время перед IT специалистами при построении различных информационных систем часто встает задача выбора и сопровождения операционной системы. Так как глубокое знание конкретных операционных систем – прерогатива системных и сетевых администраторов, то системному аналитику, для грамотного выбора необходимо знать области применения ОС, классы решаемых ею задач, представлять себе ее структуру и организацию, а также сложность ее настройки и сопровождения. Для объективной оценки реальных ситуаций необходимо иметь практический опыт использования и настройки нескольких различных систем.

Данные методические указания предназначены для изучения возможностей различных операционных систем (Windows, Linux, Solaris и др.), а также практического освоения процедур их настройки и сопровождения. В свою очередь, решаемые практические задачи дадут представление о сложности работы и методах настройки систем.

Лабораторные работы представляют собой двухчасовые аудиторские занятия, которые проводятся в учебном классе, оснащенном персональными компьютерами. Каждая лабораторная работа включает в себя изучение теоретических сведений по рассматриваемой теме, освоение изучаемых операционных систем, получение практических навыков решения конкретных задач, выполнение индивидуального задания, оформление отчета о проделанной работе и его защиту у преподавателя.

В начале занятия преподаватель объясняет ход лабораторной работы, поясняет наиболее трудные теоретические положения, выдает задания и разъясняет требования к результатам работы. В случае возникновения у студентов проблем с выполнением лабораторной работы преподаватель консультирует их и поясняет моменты, вызвавшие затруднения.

После освоения теоретических сведений и выполнения практических и индивидуальных заданий каждый студент оформляет отчет, который должен отражать основные результаты работы.

Преподаватель оценивает работу студента на основании оформленного отчета, устного ответа студента по теме лабораторной работы, а также предъявления и защиты результатов выполнения индивидуального задания.

По каждой лабораторной работе для успевающих студентов могут предусматриваться развивающие творческие задания различной направленности и сложности.

В Приложении 1 приводится общий вариант оформления отчета по лабораторной работе.

ЛАБОРАТОРНАЯ РАБОТА № 1 «Установка и настройка ОС MS Windows»

1.1. Цель работы

Изучение возможностей настройки и оптимизации среды ОС Windows 7.

Получение практических навыков и освоение приемов конфигурирования и оптимизации среды ОС.

1.2. Теоретическая часть

Для многих пользователей Windows 7 уже стала любимой ОС. А рассмотренные нами настройки сделают работу в ней еще более приятной и комфортной. Оптимизация Windows 7 – дело очень полезное. Большинство изысканных свойств системы часто остаются неизвестными рядовому пользователю. Настройка и оптимизация Windows 7 проводятся с целью ускорения работы ОС и повышения ее общей эффективности. Достигается это при помощи отключения ненужных, навязчивых сервисов, изменения визуальных настроек, корректной настройки ОС. При этом следует иметь в виду, что если для настройки необходимо вносить изменения в системный реестр, то после их внесения компьютер нужно **перезагрузить** – только тогда изменения вступят в силу!

После того как ОС Windows 7 установлена, установлены драйверы всех устройств и все необходимые пакеты обновлений, можно приступать к оптимизации ее работы. Настройку ОС лучше всего начать с **Рабочего стола** и меню **Пуск**.

Системный реестр ОС Windows 7 представляет собой централизованную базу данных (БД) параметров настройки системы и работающих в ее среде приложений. В этом смысле реестр аналогичен INI-файлам, файлам AUTOEXEC.BAT и CONFIG.SYS, применявшимся в разных ОС ранее, призван снять неудобства и ограничения, связанные с использованием этих разрозненных файлов. Реестр содержит информацию обо всех аппаратных средствах, программном обеспечении, самой ОС и сетевых параметрах компьютера. Эта сложная, иерархическая БД принимает участие во всех аспектах работы Windows 7. Понимание принципов работы реестра, выполняемых им задач, а также умение манипулировать реестром необходимы многим категориям опытных пользователей. Системный реестр заслуживает отдельного рассмотрения, но из-за ограничения времени, отводимого на лабораторные работы,

возможности реестра (содержание, вызов, редактирование) рассматриваются далее в контексте обсуждаемых вопросов настройки и оптимизации среды Windows 7.

1.2.1. Индивидуальная настройка рабочего стола

После установки Windows 7, **Рабочий стол** содержит только один значок «Корзина». Для отображения значков «Мой компьютер», «Мои документы», «Сетевое окружение», «Internet Explorer», необходимо на Рабочем столе нажатием правой кнопки мыши вызвать контекстное меню, далее выбрать его пункт «Свойства», в открывшемся окне «Свойства: Экран» (рис.1.1) выбрать закладку «Рабочий стол», в ней – кнопку «Настройка рабочего стола».

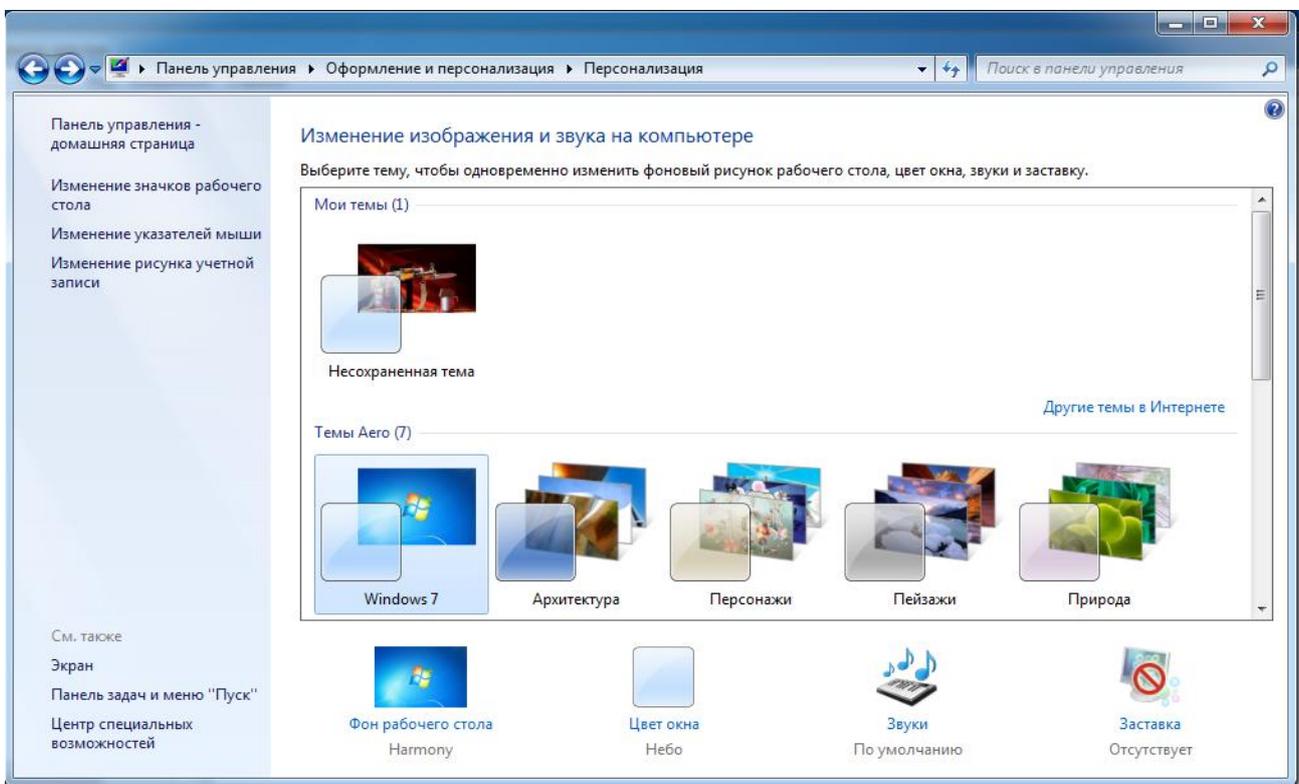


Рис.1.1. Окно «Персонализация»

Далее в появившемся окне «Элементы рабочего стола» (рис.1.2) можно выбрать значки элементов рабочего стола, которые необходимо отображать, выбрать их внешний вид, а также разрешить, либо запретить выполнение очистки рабочего стола. Один из оптимальных вариантов: отображение значков «Мой компьютер», «Мои документы» и запрет очистки рабочего стола.

Настройка меню «Пуск». В Windows 7 меню «Пуск» (рис.1.3) выглядит совершенно иначе, чем в предыдущих ОС семейства Windows. В

нем появилось много новых функций, которые при желании можно настроить в соответствии со своими потребностями. Например, по умолчанию в меню отображается шесть наиболее часто используемых программ.

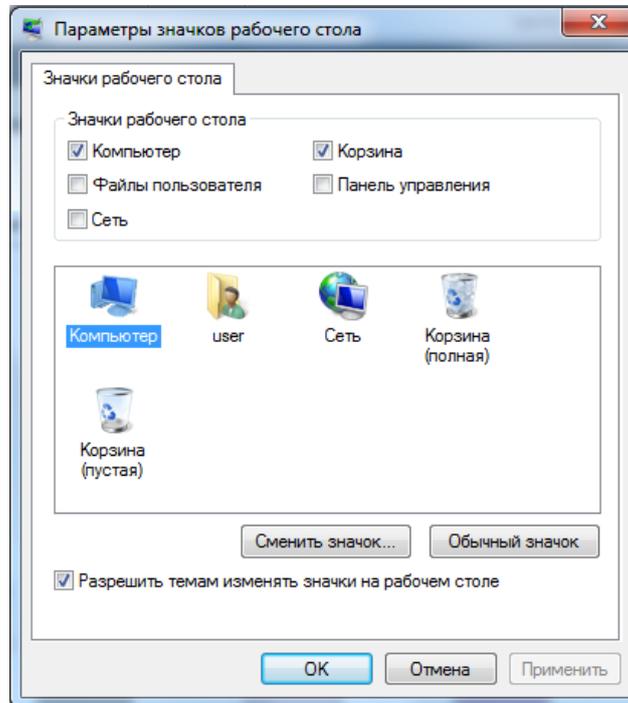


Рис.1.2. Параметры значков рабочего стола

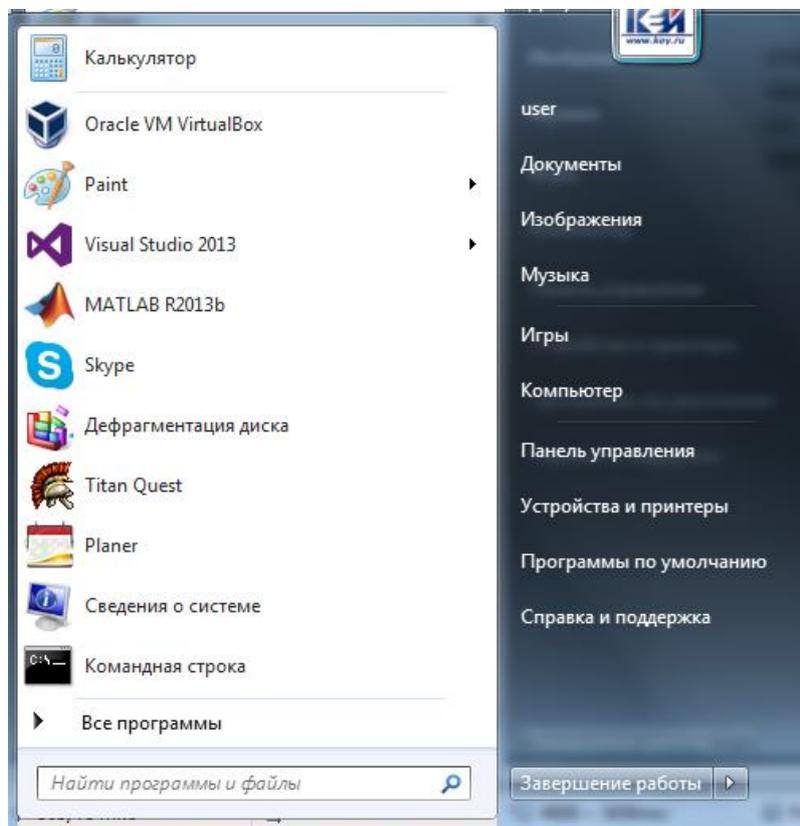


Рис.1.3. Меню «Пуск»

Чтобы изменить число этих программ или вообще отключить данную функцию, нажмите правой кнопкой мыши на кнопке «Пуск» и выберите в появившемся контекстном меню пункт «Свойства». Далее в появившемся окне «Свойства панели задач и меню «Пуск»» (рис.1.4) выберите вкладку и радиокнопку «Меню «Пуск»» и кнопку «Настроить».

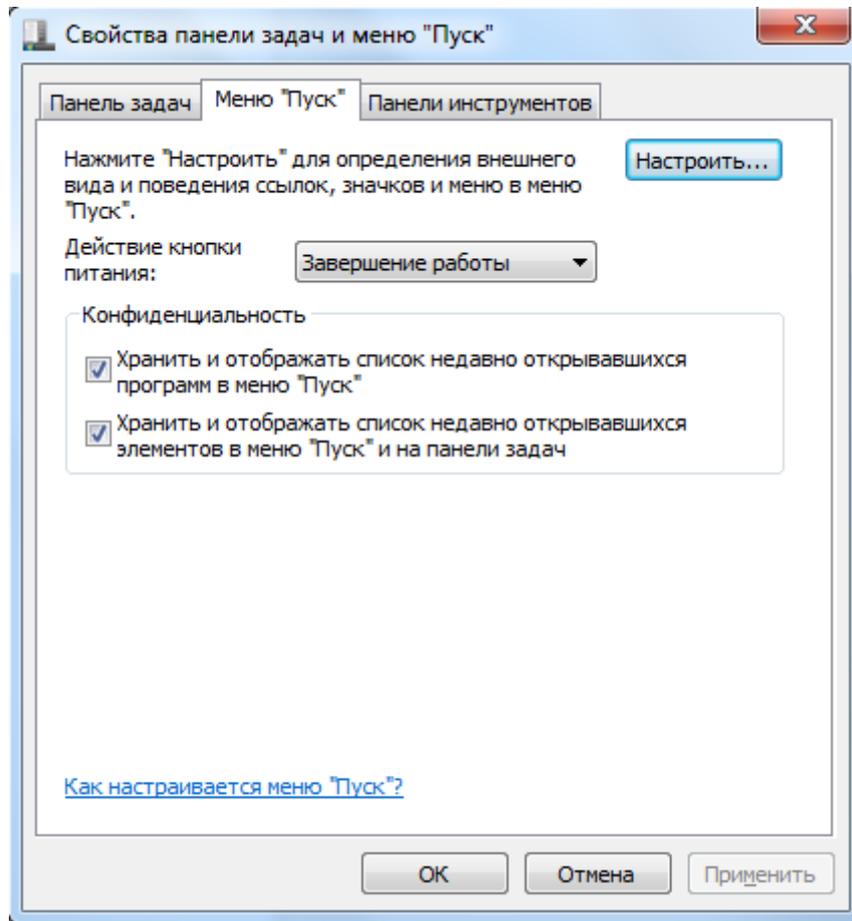


Рис.1.4. Свойство панели задач и меню «Пуск»

В появившемся окне (рис.1.5) на вкладке «Общие» можно изменить, например, число программ в меню «Пуск» или отказаться от появления в меню Internet-браузера и почтового клиента, которые система использует по умолчанию.

Для тех пользователей, которые привыкли к варианту меню «Пуск» из предыдущих версий Windows, нужно активизировать радиокнопку «Классическое меню «Пуск»» на вкладке «Меню «Пуск»» (рис.1.4), после чего настроить его.

Используя вкладку «Дополнительно» (рис.1.5), можно отключить раздражающее оповещение о вновь установленной программе и

полностью скорректировать список элементов меню «Пуск», а также способ их представления – в качестве меню или ссылки.

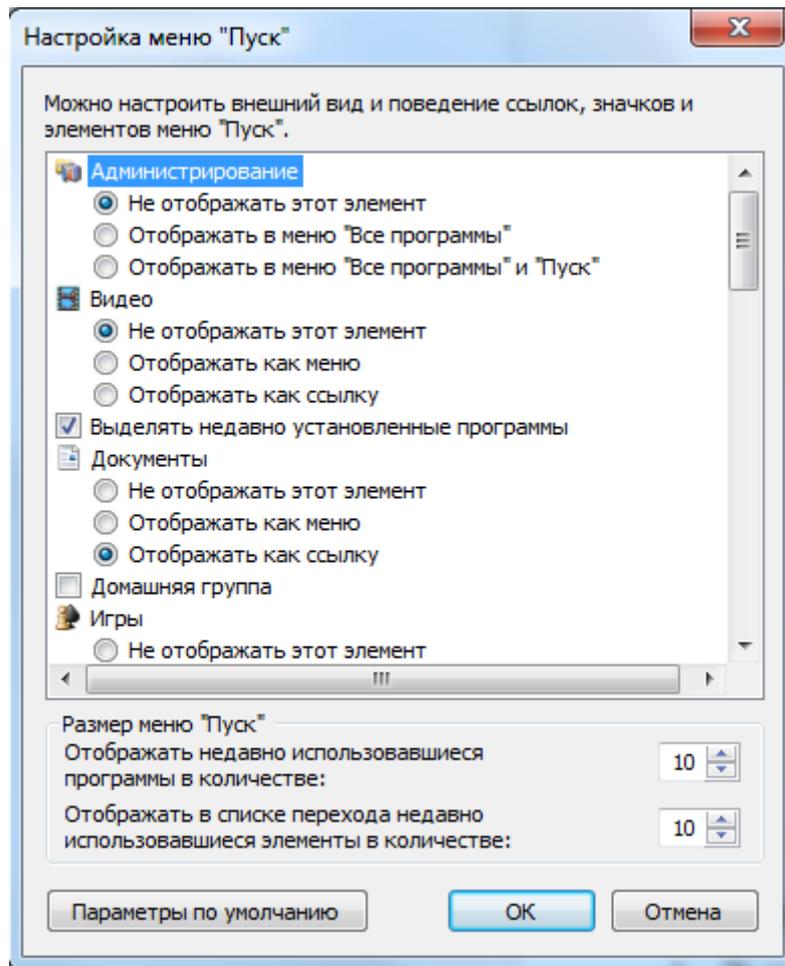


Рис.1.5. Настройка меню «Пуск»

Один из оптимальных вариантов: рекомендуется вернуться к классическому меню «Пуск», в настройках отключить «Использовать сокращенные меню», выбрать «Мелкие значки в главном меню» и «Раскрывать панель управления и сетевые подключения».

Ускорение работы меню «Пуск» обеспечивается следующим образом.

1. В меню «Пуск» выберите пункт «Выполнить», далее в появившемся окне «Запуск программы» в поле ввода «Открыть» введите regedit – имя утилиты редактирования системного реестра Windows 7.
2. В появившемся окне «Редактор реестра» (рис.1.6) на панели проводника выберите ветвь HKEY_CURRENT_USER\Control Panel\Desktop\.
3. В появившемся справа списке параметров выберите правой кнопкой

мыши параметр **MenuShowDelay** (рис.1.6), после чего в его контекстном меню выберите пункт «Изменить».

4. Измените значение параметра (чем оно меньше, тем быстрее открываются пункты меню; время в миллисекундах; значение по умолчанию – 400).
5. Нажмите **ОК**, закройте regedit и перезагрузите компьютер.

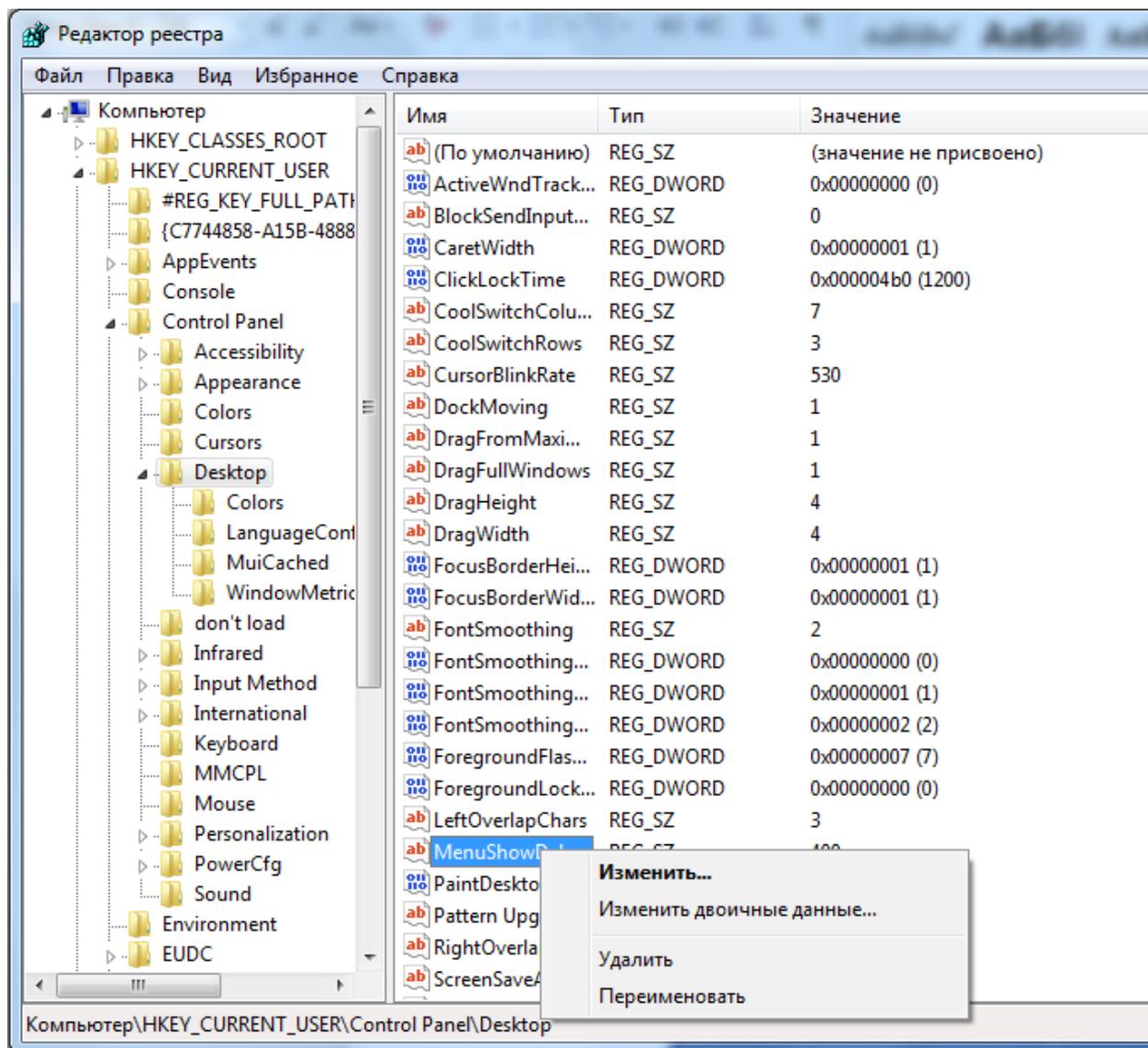


Рис.1.6. Редактор реестра

Настройка Панели задач. Чтобы открыть окно со свойствами **Панели задач**, нажмите правую кнопку мыши на пустом месте **Панели задач** и в контекстном меню выберите пункт «Свойства». В открывшемся окне «Свойства панели задач и меню «Пуск» (рис.1.7) можно настроить поля ввода «Оформление панели задач» и «Область уведомлений», устанавливая значения интересующих кнопок выбора.

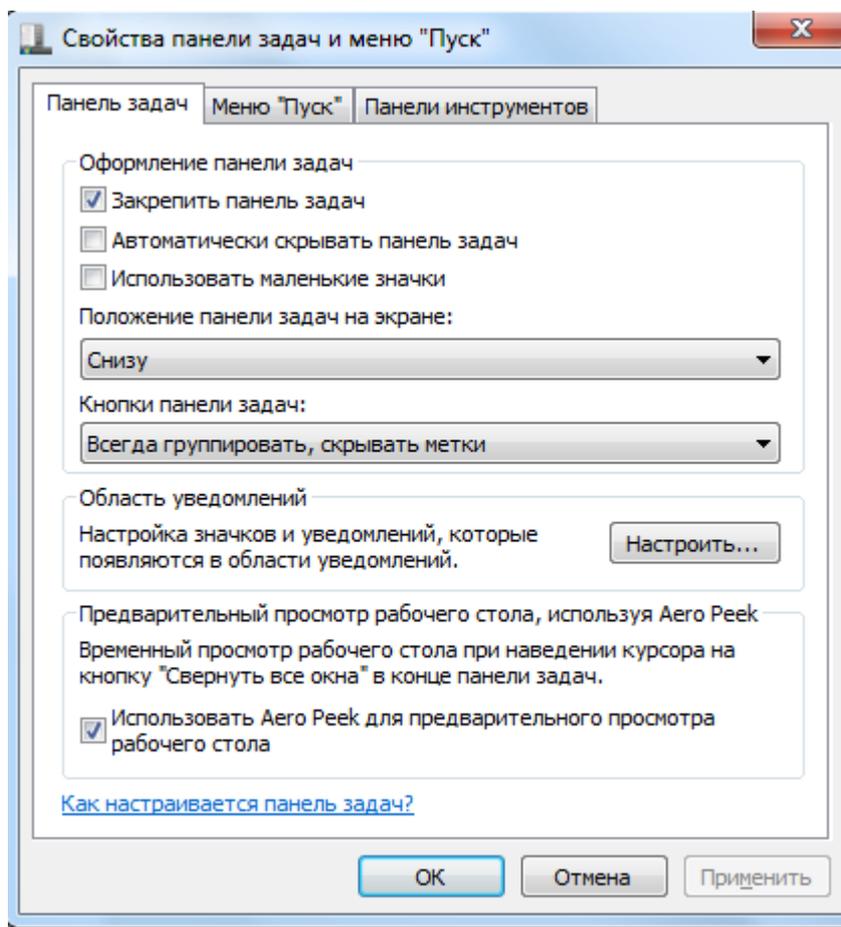


Рис.1.7. Свойства панели задач и меню «Пуск»

При одновременной работе с большим числом программ **Панель задач** переполняется кнопками. Поэтому в Windows 7 была добавлена функция автоматической группировки кнопок, помогающая управлять большим числом открытых документов и программ. Благодаря ей, при открытии в одной программе более трех окон Windows 7 объединяет все документы на **Панели задач** в виде одной кнопки с именем этой программы. Треугольник в правой части кнопки указывает на то, что в данной программе открыто несколько документов.

Считается, что так можно быстрее и удобнее перейти на нужный открытый документ. Но если для вас предпочтительнее старый вариант размещения открытых окон, то данную функцию можно просто отключить. Для этого уберите галочку в кнопке выбора «Группировать сходные кнопки панели задач».

Изменение области уведомлений Панели задач. В области уведомлений **Панели задач** обычно отображаются текущее время и иконки различных приложений. Данная область наполняется значками

уведомлений, отображающимися при выполнении определенных задач, например, при получении электронной почты. Можно настроить поведение значков уведомлений по своему усмотрению.

Чтобы настроить уведомления, щелкните правой кнопкой мыши на свободном месте в **Панели задач** и выберите в появившемся контекстном меню пункт «Свойства». На появившейся вкладке «Панель задач», в разделе **Область уведомлений** установите флажок «Скрывать неиспользуемые значки». После этого, нажав кнопку «Настроить», отредактируйте поведение каждого конкретного элемента, выбрав один из вариантов: **Всегда отображать**, **Всегда скрывать** или же **Скрывать, если не используется** (рис. 1.8). По окончании выбора нажмите кнопку «Применить».

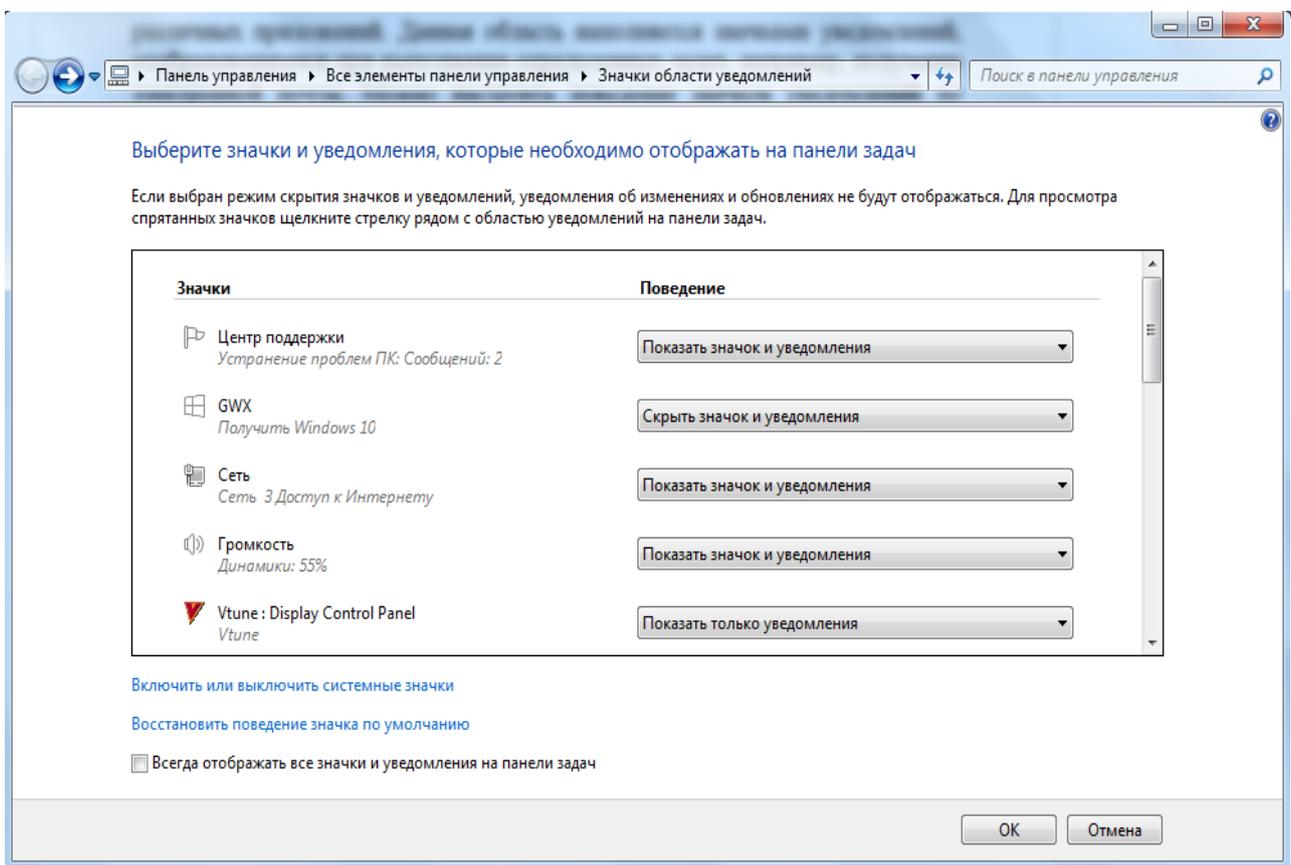


Рис.1.8. Значки области уведомлений.

Чтобы скрыть ненужные значки в **Панели задач** (в этом случае останутся лишь часы), в системном реестре найдите раздел `HKEY_CURRENT_USER \ Software \ Microsoft \ Windows \ CurrentVersion \ Policies \ Explorer` (рис.1.9) и присвойте параметру `NoTrayItemsDisplay` (тип `dword`) значение 1. Если параметр еще не существует, создайте его.

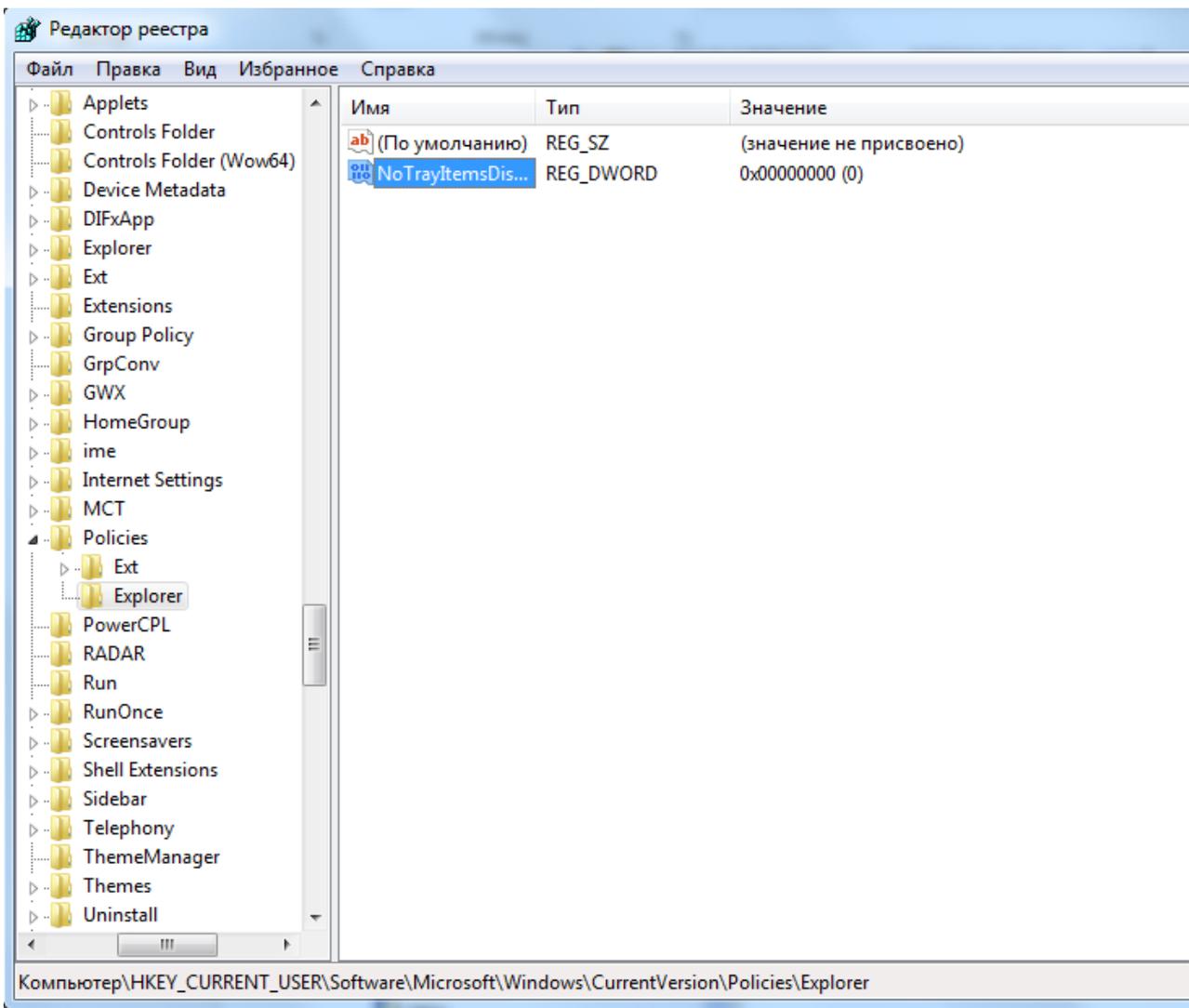


Рис.1.9. Редактор реестра

Для этого щелкните правой кнопкой мыши на пустом месте в области параметров, выберите пункт «Создать», параметр DWORD, введите имя параметра – **NoTrayItemsDisplay** (рис.1.10). Для запуска редактора реестра откройте меню «Пуск», далее в поле ввода введите имя **regedit**.

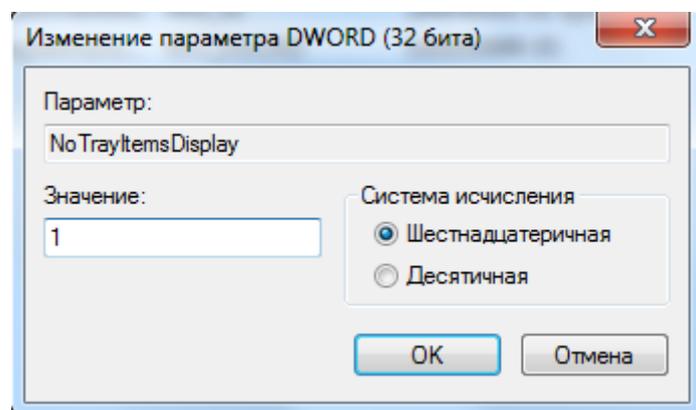


Рис. 1.10. Изменение параметра DWORD

Отказ от приветствия. Каждый раз при загрузке Windows 7 на экране появляется приветствие. Чтобы оно не мешало, откройте **Панель управления**, щелкните дважды кнопкой мыши по значку «Учетные записи пользователей», в появившемся одноименном окне (рис. 1.11) выберите команду «Изменение входа пользователей в систему» и уберите галочку в позиции «Использовать страницу приветствия». После выполнения этих действий щелкните кнопкой мыши на кнопке «Применение параметров».

Настройка режима работы программы с помощью мастера совместимости программ. Известной проблемой Windows 7 является тот факт, что не все программы, прекрасно функционировавшие в предыдущих версиях ОС Windows, могут работать в ней корректно. В ряде случаев, преимущественно при настройке игр и обучающих программ, на помощь может прийти мастер совместимости программ.

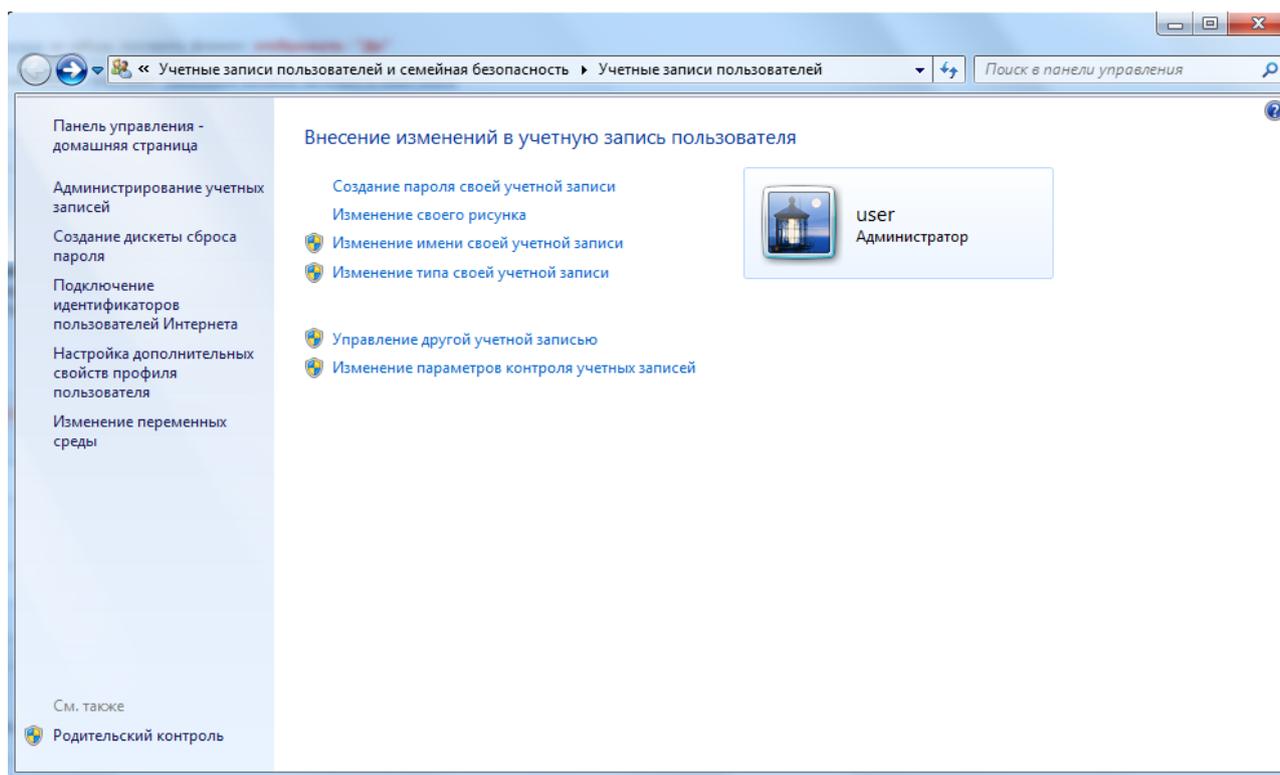


Рис.1.11. Учетные записи пользователей.

Для этого используйте меню «Пуск», пункт «Программы», а в нем – пункт «Стандартные», где активизируйте пункт «Мастер совместимости программ». Далее последовательно ответьте на вопросы мастера, выберите программу и ОС, в которой программа ранее корректно работала, а также установите нужный режим экрана.

Настройка свойств папки. Чтобы настроить параметры папок, откройте **Панель управления**, в ней – «Свойства папки», после чего на экране появится одноименное окно (рис.1.12).

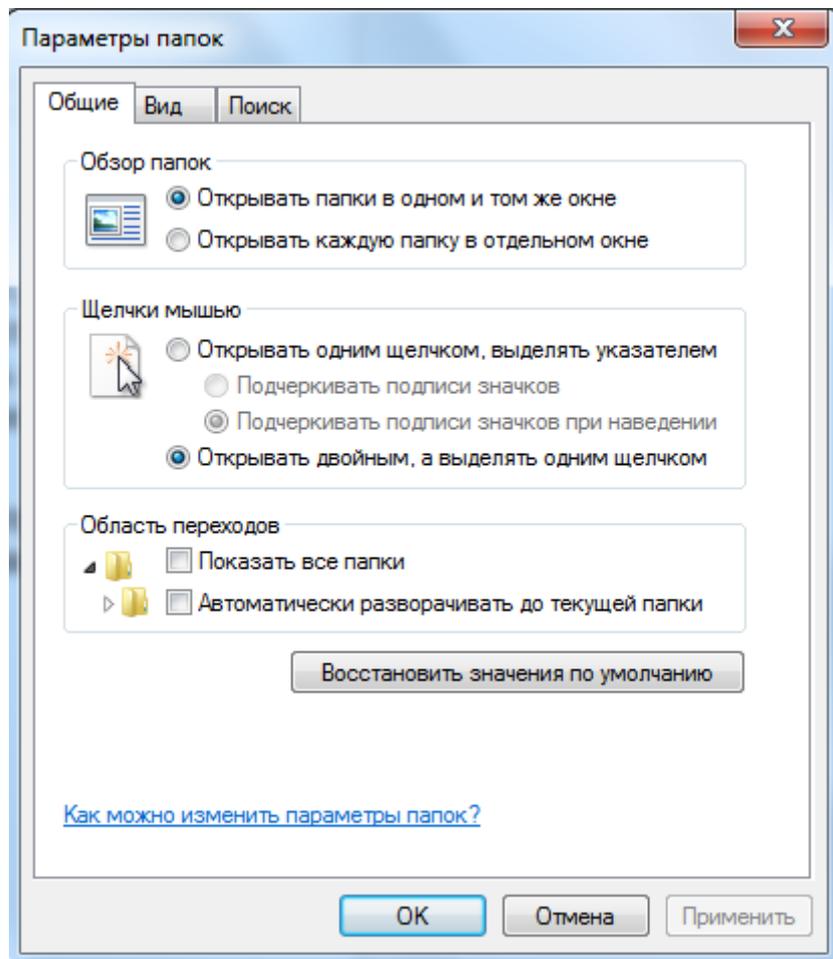


Рис.1.12. Параметры папок

На вкладке «Общие» можно выставить отображение папок, а также выбрать способ их выделения и открытия (одинарным или двойным щелчком мыши). На вкладке «Вид» можно детально настроить вид и параметры, как отдельно выбранной папки, так и применить изменения ко всем папкам, нажав после проведенных изменений кнопку «Применить ко всем папкам». При необходимости, на этой же вкладке можно вернуться к стандартным настройкам всех папок, нажав «Сброс для всех папок».

Изменение папки, открываемой по умолчанию. По умолчанию при запуске **Проводника** Windows 7 открывается папка «Мои документы», что далеко не всегда удобно. Чтобы изменить открываемую Проводником папку, нужно в системе меню «Пуск», «Программы», «Стандартные», щелкнуть правой кнопкой мыши на пункте «Проводник», вызвать в контекстном меню пункт «Свойства» и в появившемся окне на

вкладке «Ярлык» (рис.1.13) ввести, например, %SystemRoot%\explorer.exe /e,c:\. После этого при запуске Проводника сразу будет открываться диск С. Естественно, что таким образом можно выбрать любую папку диска.

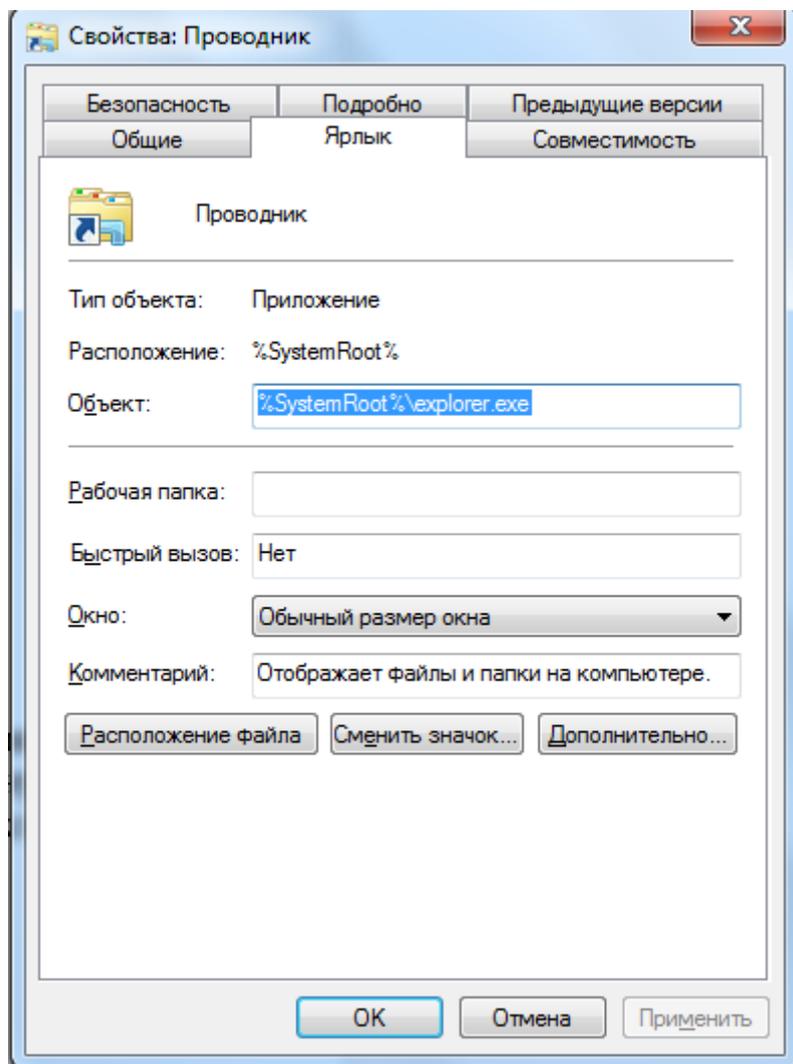


Рис. 1.13 – Свойства: Проводник

Настройка оформления Windows 7. Чтобы изменить стиль оформления ОС Windows 7, на **Рабочем столе** щелкните правой кнопкой мыши, выберите в контекстном меню пункт «Свойства», а в открывшемся окне выберите вкладку «Оформление» (рис.1.14).

В этой вкладке можно выбрать стиль оформления ОС: классический (разработанный ранее) или стиль Windows 7. Для каждого стиля можно выбрать цветовую схему и размер шрифта, который используется для всех текстовых элементов оформления ОС. Кнопки «Эффекты» и «Дополнительно» обеспечивают доступ к эффектам и дополнительным настройкам (рис.1.15, 1.16).

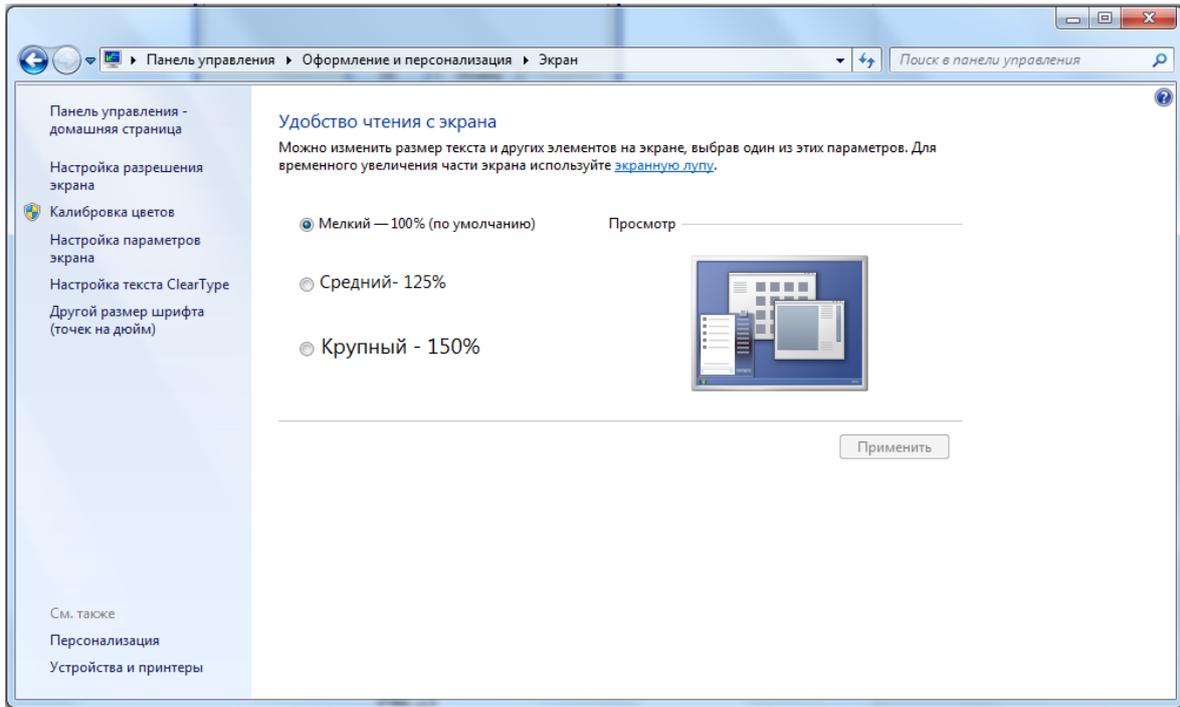


Рис.1.14. Оформление и персонализация

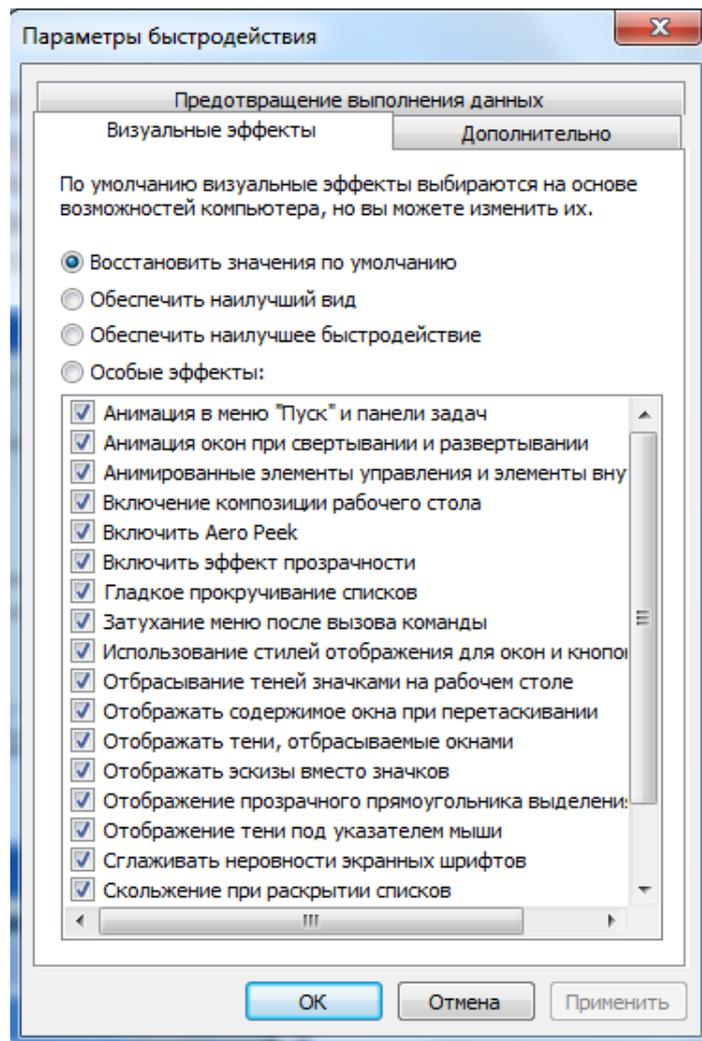


Рис.1.15. Параметры быстродействия

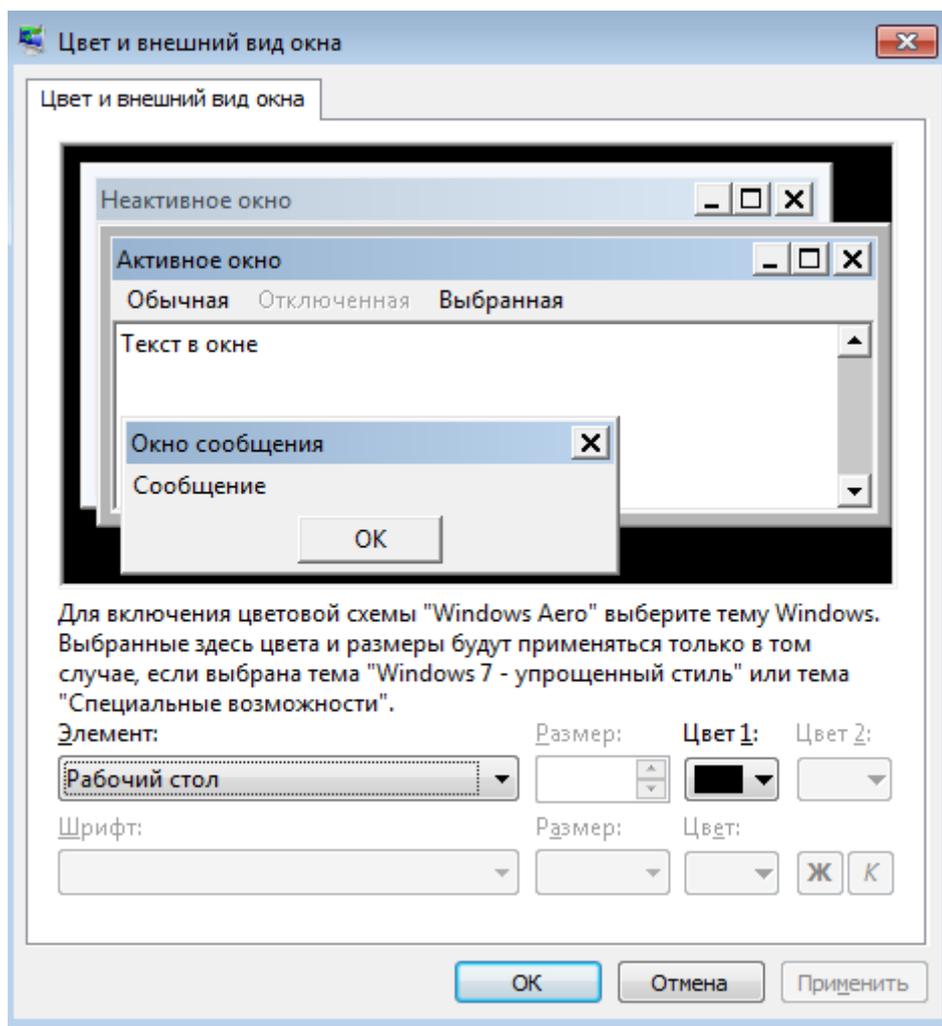


Рис.1.16. Цвет и внешний вид окна

1.2.2. Диспетчер задач (Task Manager)

Важнейшим средством мониторинга и оптимизации среды Windows 7 является **Диспетчер задач**, вызываемый нажатием правой кнопки мыши в свободном месте Панели задач и выбором из появившегося контекстного меню соответствующего пункта. В его главном окне (рис. 1.17) имеется пять вкладок:

- «**Приложения**» – показывает имена запущенных задач и их состояния. Содержит кнопки для снятия задачи, переключения на другую, запуска новой задачи;
- «**Процессы**» – показывает все запущенные процессы, запущенные в собственном адресном пространстве, включая все приложения и системные сервисы, а также их показатели (рис.1.17). Если требуется просмотреть 16-разрядные процессы, то в меню «Параметры» следует выбрать пункт «Отображать 16-разрядные задачи». Набор показателей можно изменять с помощью меню «Вид», пункт «Выбрать столбцы...»;

- «**Быстродействие**» – отслеживает в динамике различные показатели производительности системы (рис.1.18), интересные при выполнении лабораторных работ 4-5;
- «**Сеть**» – показывает состояние имеющегося подключения по локальной сети;
- «**Пользователи**» – показывает характеристики пользователей среды (имя, состояние, вид сеанса).

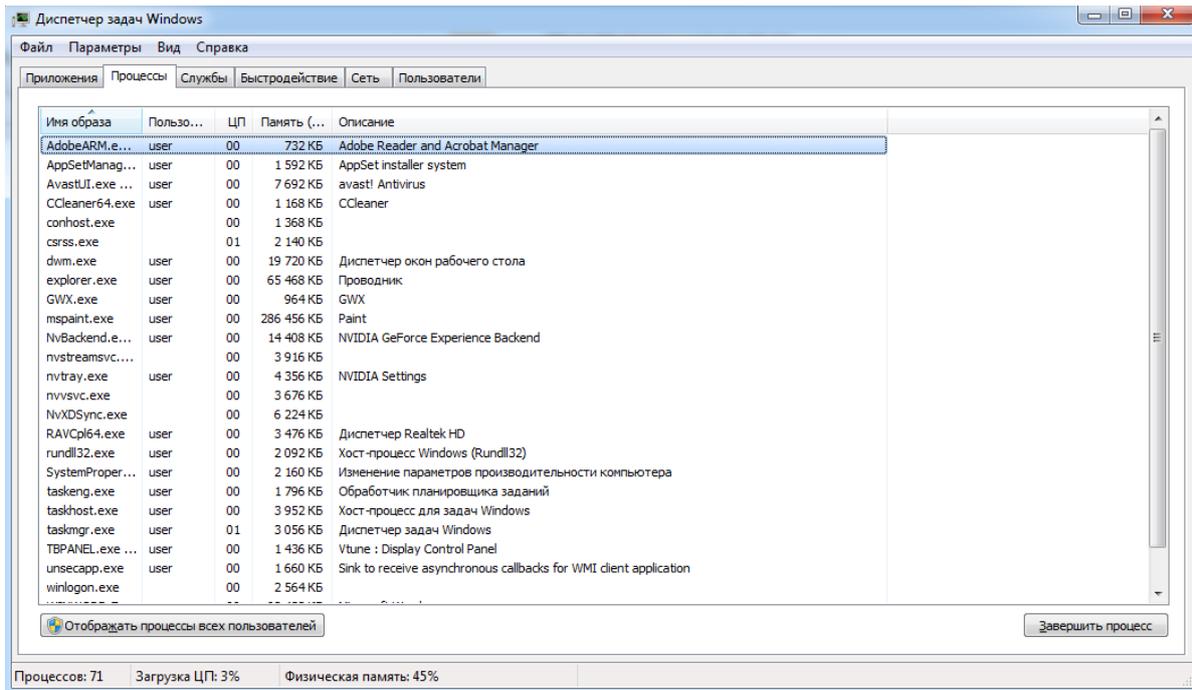


Рис.1.17. Диспетчер задач

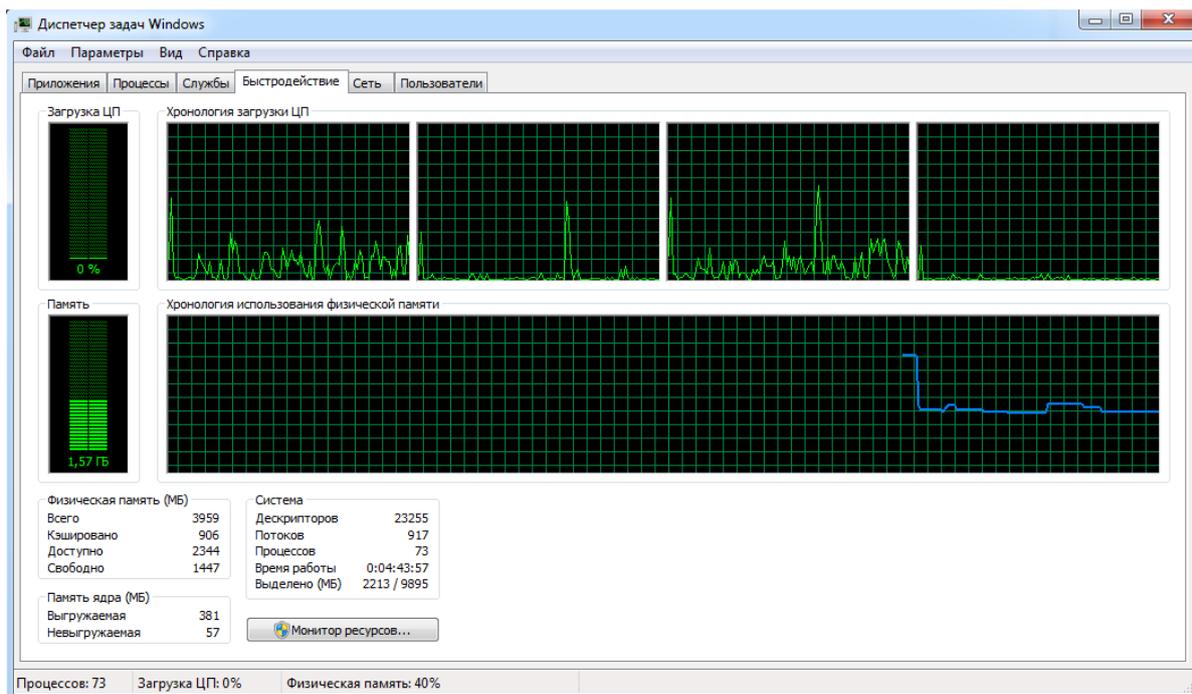


Рис.1.18. Диспетчер задач: Быстродействие

1.2.3. Планировщик заданий (Task Scheduler)

С помощью планировщика заданий можно составить расписание запуска командных файлов, документов, обычных приложений или различных утилит для обслуживания системы. Программы могут запускаться однократно, ежедневно, еженедельно, ежемесячно в заданные дни, при загрузке системы или регистрации в ней, а также при бездействии системы (idle state). Планировщик позволяет задавать достаточно сложное расписание для выполнения заданий, в котором задаются продолжительность задания, время его окончания, число повторов, зависимость от состояния источника питания и т.п. Задание сохраняется как файл с расширением **.job**, что позволяет перемещать его с одного компьютера на другой.

Планировщик заданий запускается из Панели управления, папка и окно «Назначенные задания» (рис.1.19).

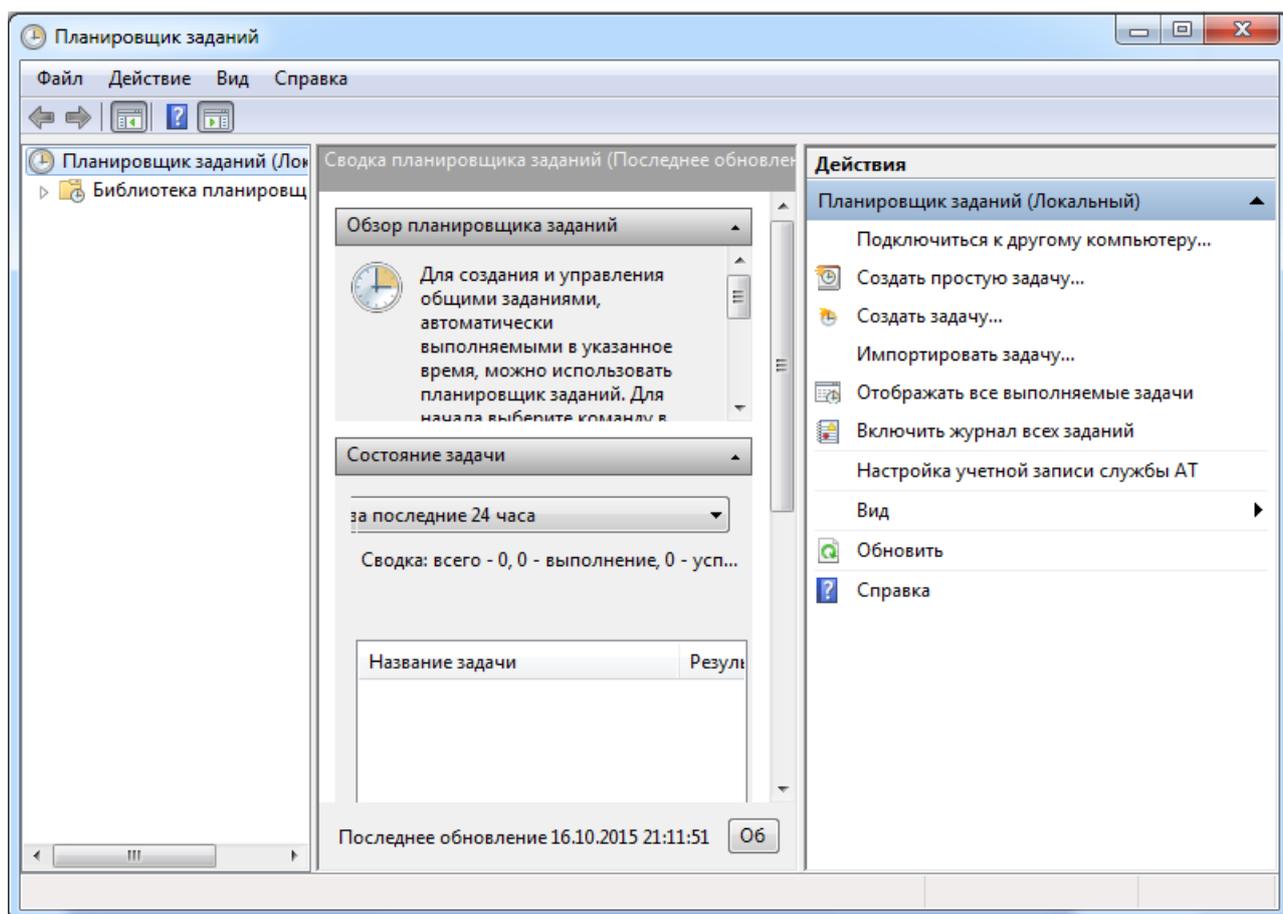


Рис. 1.19 – Планировщик заданий

Кроме того, упрощается отладка заданий, поскольку их легко проверить, запустив в любой момент непосредственно из папки заданий

командой «Выполнить» из контекстного меню. В главном окне Планировщика выводится основная информация о заданиях: расписание, время следующего и предыдущего запуска, состояние, результат выполнения задания, имя создателя задания.

Мастер планирования заданий запускается по команде «Добавить задание» и позволяет в интерактивном режиме указать все параметры для запуска запланированного задания. Задания могут иметь несколько принципиально различных расписаний.

1.2.4. Средства поддержки виртуальной памяти

При запуске приложения Windows 7 создает процесс со своей информационной структурой [3]. В рамках процесса запускается поток выполнения, который при необходимости может инициировать запуск множества других потоков, которые будут выполняться параллельно в рамках одного процесса. Множество запущенных процессов (представляющих в общем случае многопоточные приложения) также выполняется параллельно.

Если компьютер испытывает нехватку свободной оперативной памяти (**ОП**), а для завершения текущей задачи требуется дополнительная память, Windows 7 использует пространство жесткого диска для моделирования системной ОП. Такая память называется **виртуальной памятью**. Ее составляют физическая ОП и специальный скрытый **файл подкачки** на жестком диске. Последний ресурс является аналогом файла подкачки UNIX. По мере необходимости Windows 7 перемещает данные из файла подкачки в ОП (для их использования программой) и обратно (для освобождения места для новых данных). Файл подкачки называется также файлом виртуальной памяти.

Каждый процесс выполняется в своем собственном защищенном виртуальном адресном пространстве (**ВАП**). Потоки одного процесса выполняются в ВАП своего процесса.

Отображение различных ВАП исполняющихся процессов на физическую память в Windows 7 осуществляется по принципам страничной организации виртуальной памяти. Процессами выделения памяти, ее резервирования, освобождения и замещения страниц управляет

диспетчер виртуальной памяти (Virtual Memory Manager). Для замещения страниц выбрана дисциплина FIFO.

Когда процесс использует код или данные, находящиеся в физической памяти, Windows 7 резервирует место для всех соответствующих страниц в специальном файле подкачки pagefile.sys на диске. Этот файл может быть как непрерывным, так и сегментированным. Он может быть расположен на системном диске или на любом другом одном или даже нескольких дисках. Размер этого страничного файла ограничивает объем данных, которые могут храниться во внешней памяти при использовании механизмов виртуальной памяти. По умолчанию в Windows 7 начальный размер берется в 1,5 раза больше объема физической оперативной памяти. Проблема нехватки виртуальной памяти часто может быть решена за счет увеличения размера файла подкачки. Файл подкачки может быть не один – система поддерживает до 16 файлов подкачки, поэтому лучше создать их несколько и разместить на разных быстрых физических жестких дисках. При выборе таких дисков желательно придерживаться следующих правил:

- не размещайте файл подкачки на том же диске, где находятся системные файлы;
- не размещайте файл подкачки на отказоустойчивых дисках, например, на зеркальном томе или томе RAID-5. Для файлов подкачки не требуется обеспечивать отказоустойчивость; к тому же некоторые отказоустойчивые системы слишком медленно работают из-за того, что выполняют запись в несколько мест с целью резервирования;
- не размещайте несколько файлов подкачки в разных разделах одного и того же физического диска.

Режим использования памяти компьютера можно оптимизировать. Если компьютер в основном используется не как сервер, а как рабочая станция, можно отвести больше памяти для прикладных программ. В этом случае программы будут работать быстрее, а для системного кэша будет использоваться стандартный размер, установленный в Windows 7. Если же компьютер используется главным образом в качестве сервера, или если программам требуется кэш большого размера, можно выделить больше памяти для системного кэша.

Изменение размера файла подкачки. Для этого необходимо войти в систему с учетной записью «Администратор» или члена группы «Администраторы». Если компьютер подключен к сети, то параметры сетевой политики могут запретить выполнение данной процедуры. Для выполнения рассматриваемой процедуры, выполните следующие действия:

1. Откройте узел (оснастку) «Управление компьютером»: нажмите кнопку **Пуск**, затем используйте команды «Настройка» и «Панель управления». Дважды щелкните значок «Администрирование», а затем дважды щелкните значок «Управление компьютером». В результате появится окно (рис.1.20).

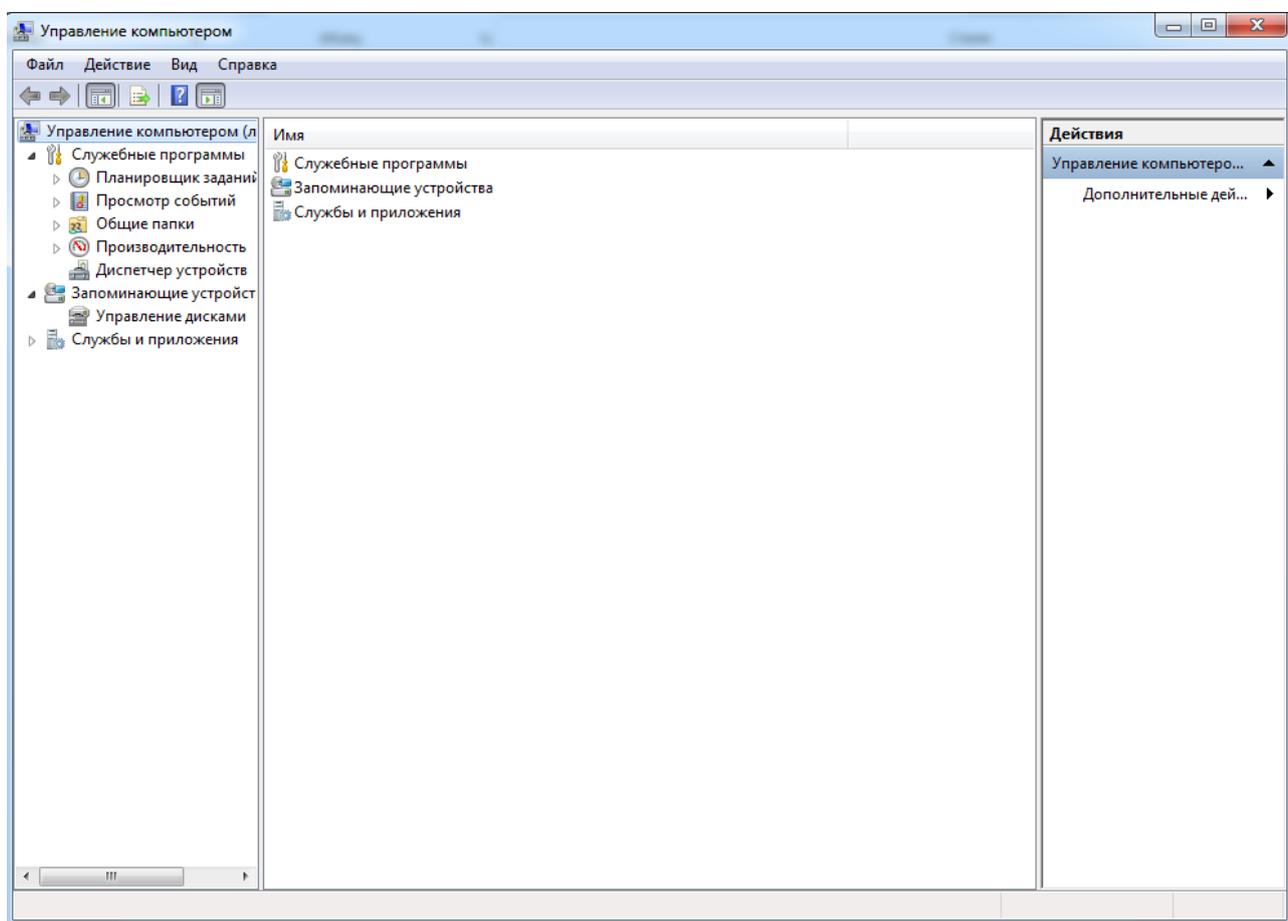


Рис.1.20. Управление компьютером

2. В дереве консоли щелкните правой кнопкой элемент «Управление компьютером (локальным)» и выберите команду «Свойства» (рис.1.21).
3. На вкладке «Дополнительно» нажмите кнопку «Параметры быстрогодействия» и в группе «Виртуальная память» нажмите кнопку «Изменить». Появится окно (рис.1.22).

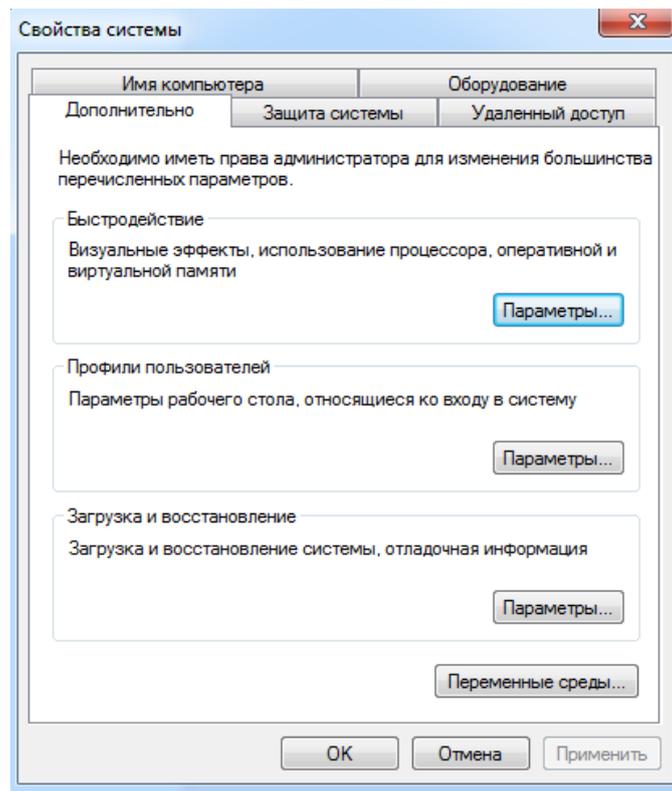


Рис.1.21. Свойства системы

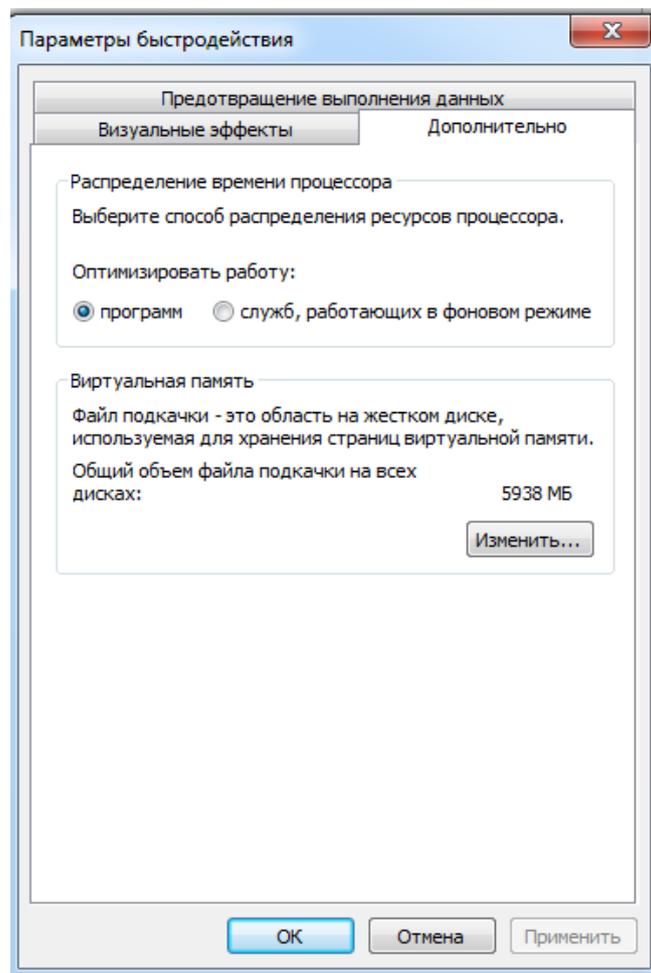


Рис.1.22. Параметры быстродействия

4. В списке «Диск» выберите диск, содержащий файл подкачки, размер которого необходимо изменить (рис. 1.23).

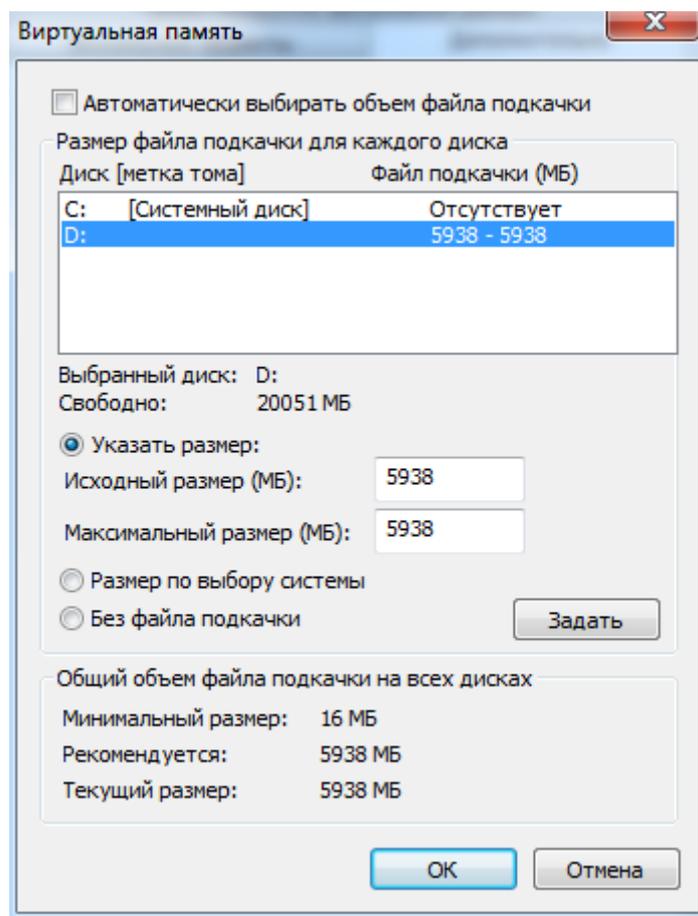


Рис.1.23. Виртуальная память

5. В группе «Размер файла подкачки для выбранного диска» введите новые значения (в Мбайтах) в поле «Исходный размер (МБ)» или в поле «Максимальный размер (Мб)», а затем нажмите кнопку «Установить». При уменьшении максимального или минимального размера файла подкачки необходимо перезагрузить компьютер, чтобы изменения вступили в силу. При увеличении размера перезагрузка, обычно, не требуется.

Выбор значения размера файла подкачки. Существует два варианта выбора значения размера файла подкачки: фиксированный размер и динамически изменяемый. В первом случае максимальный размер устанавливается равным минимальному размеру, что уменьшает фрагментацию диска, но может привести к зависанию программ, либо ОС, если им не хватит памяти (рекомендуется использовать, если вы уверены что размер вполне достаточен, к тому же сбой маловероятен, поскольку Windows 7 при необходимости увеличит размер файла подкачки

самостоятельно, не обращая внимания на «фиксированные» установки). Во втором случае максимальное значение устанавливается в 1,5-2 раза выше минимального, и файл подкачки будет изменяться в указанных пределах по мере необходимости, что будет приводить к фрагментации диска, фрагментации самого файла подкачки и, как следствие, замедлению работы системы. Когда оперативной памяти достаточно много, файл подкачки можно вообще отключить, выбрав пункт **«Без файла подкачки»**.

Дефрагментация файла подкачки (и реестра). Дефрагментируя файл подкачки и системный реестр, можно существенно повысить скорость работы Windows 7. Сделать это можно с помощью утилиты PageDefrag. Запускать ее рекомендуется раз в неделю, и этого будет достаточно.

Автоматическая очистка файла подкачки после завершения работы. Она необходима потому, что после выключения компьютера файл подкачки может содержать пароли, имена (login), конфиденциальную и прочую информацию, которая использовалась во время работы. Для извлечения этих данных достаточно загрузить ОС и хорошенько поискать нужную информацию в этом файле. Для очистки файла подкачки после завершения работы необходимо сделать следующее:

- Запустить regedit и в его проводнике найти ветвь [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management];
- установить параметр типа DWORD ClearPageFileAtShutdown равным 1.

Изменения вступят в силу после перезагрузки компьютера. Этот ключ не стирает файл подкачки с диска, а заполняет его весь нулями, то есть очищает.

1.2.5. Другие средства оптимизации

Перераспределение времени процессора между фоновыми и оперативными программами. Производительность компьютера в ряде случаев можно повысить и путем изменения режима использования ресурсов (времени процессора и памяти). Windows 7 можно настроить таким образом, чтобы оперативной программе, выполняющейся в данный

момент, выделялось больше времени процессора. В результате может уменьшиться время отклика программы и работа ускорится. В то же время, если при этом должны выполняться еще и некие фоновые программы (например, печать или архивация диска), то лучше настроить Windows 7 на равномерное распределение ресурсов процессора между программами, работающими в фоновом режиме, и оперативными программами.

Чтобы изменить быстродействие фоновых и оперативных программ, следует открыть в системе меню пункты: «Панель управления», «Система», «Дополнительно», после чего в группе «Быстродействие» нажать кнопку «Параметры». На вкладке «Дополнительно» необходимо установить переключатель «Распределение времени процессора» в нужное положение (рис.1.24, 1.25).

Изменение визуальных эффектов Windows 7. Для отключения различных анимационных подсказок, всплывающих меню, теней и прочей «красоты», затормаживающей систему, необходимо выполнить следующие шаги. Сначала надо щелкнуть правой кнопкой мыши на значке **Мой компьютер** и выбрать команду «Свойства». После этого на экране появится окно «Свойства системы» (вызвать его можно, используя комбинацию клавиш: <Win>+<Pause/Break>). Далее необходимо перейти на вкладку «Дополнительно». Здесь можно изменять параметры быстродействия, профили пользователей и параметры загрузки/восстановления. При нажатии кнопки «Параметры быстродействия», на экране появится следующее диалоговое окно (рис.1.24, 1.25).

Чем больше эффектов включено в открывшемся окне, тем красивее интерфейс. Но эти эффекты ощутимо замедляют работу компьютера. Так что на относительно медленной машине разумнее всего выбрать пункт «Обеспечить наилучшее быстродействие», или «Особые эффекты», тогда можно будет самому выбрать необходимые эффекты, а ненужные отключить.

На вкладке «Дополнительно», как правило, ничего изменять не надо. Однако, если компьютер используется, в качестве сервера сети, то стоит оптимизировать время процессора для «служб, работающих в фоновом режиме», а использование памяти – «для системного кэша».

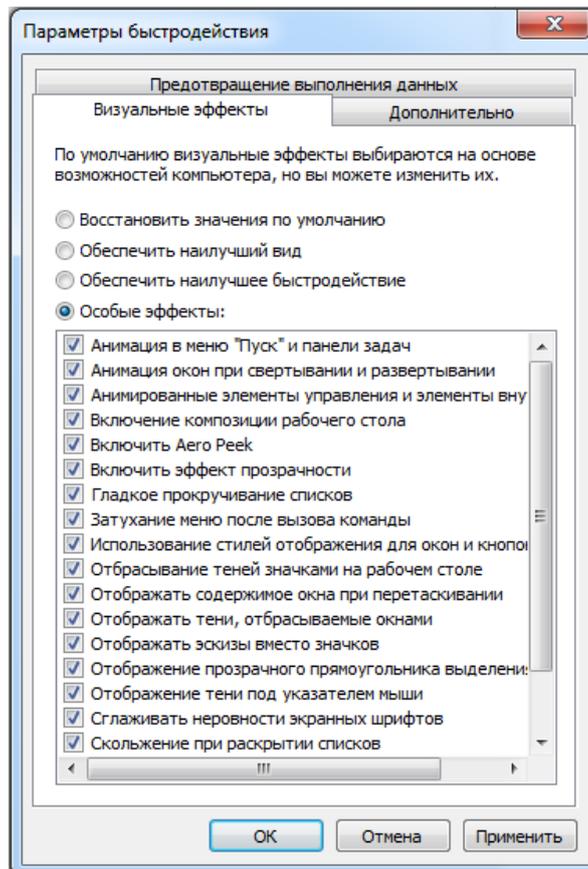


Рис.1.24. Параметры быстродействия. Визуальные эффекты

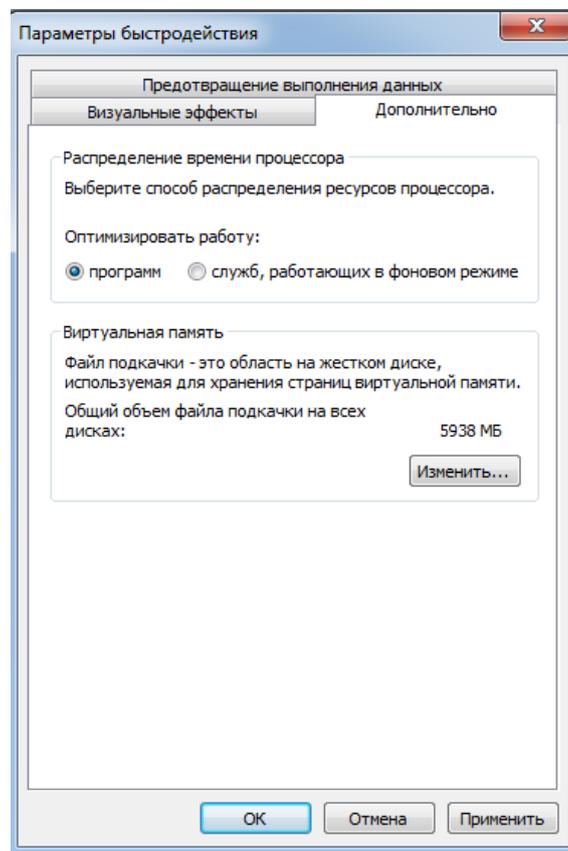


Рис.1.25. Параметры быстродействия. Дополнительно

В этом случае пользователи, работающие с БД, расположенными на сервере, ощутят существенный прирост производительности.

Увеличение скорости работы ОС. Для увеличения скорости работы системы применяют немало различных приемов. Например, можно оставлять исполняемый код ядра резидентным в памяти вместо вытеснения его, по мере необходимости, в виртуальную память на диск. Однако это безопасно, лишь при наличии не менее 256 Мбайт ОЗУ и отказе от ждущего и спящего режимов. Для реализации данной возможности в системном реестре в разделе HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Control \Session Manager \Memory Management следует изменить значения ключа DisablePagingExecutive (тип dword) на 1 (рис.1.26, 1.27).

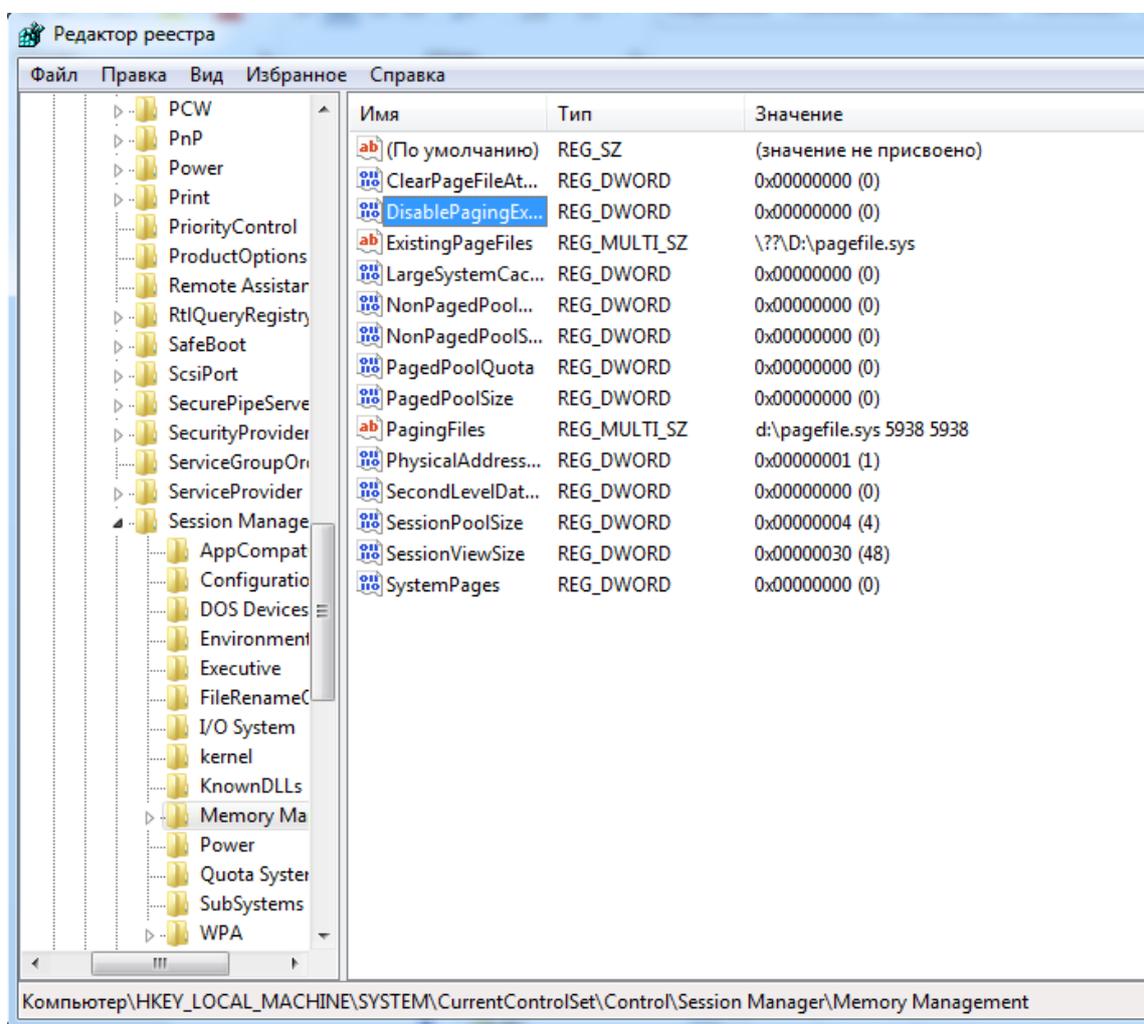


Рис.1.26. Редактор реестра

Несколько увеличить скорость работы поможет отключение, как правило, неиспользуемой подсистемы Posix. В этом случае нужно в разделе системного реестра HKEY_LOCAL_MACHINE \SYSTEM

\CurrentControlSet \Control \Session Manager \SubSystems удалить строки Optional и Posix (рис. 1.28).

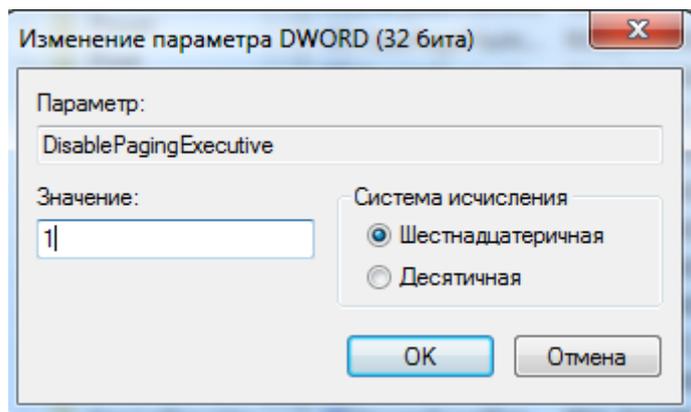


Рис.1.27. Изменение параметра DWORD

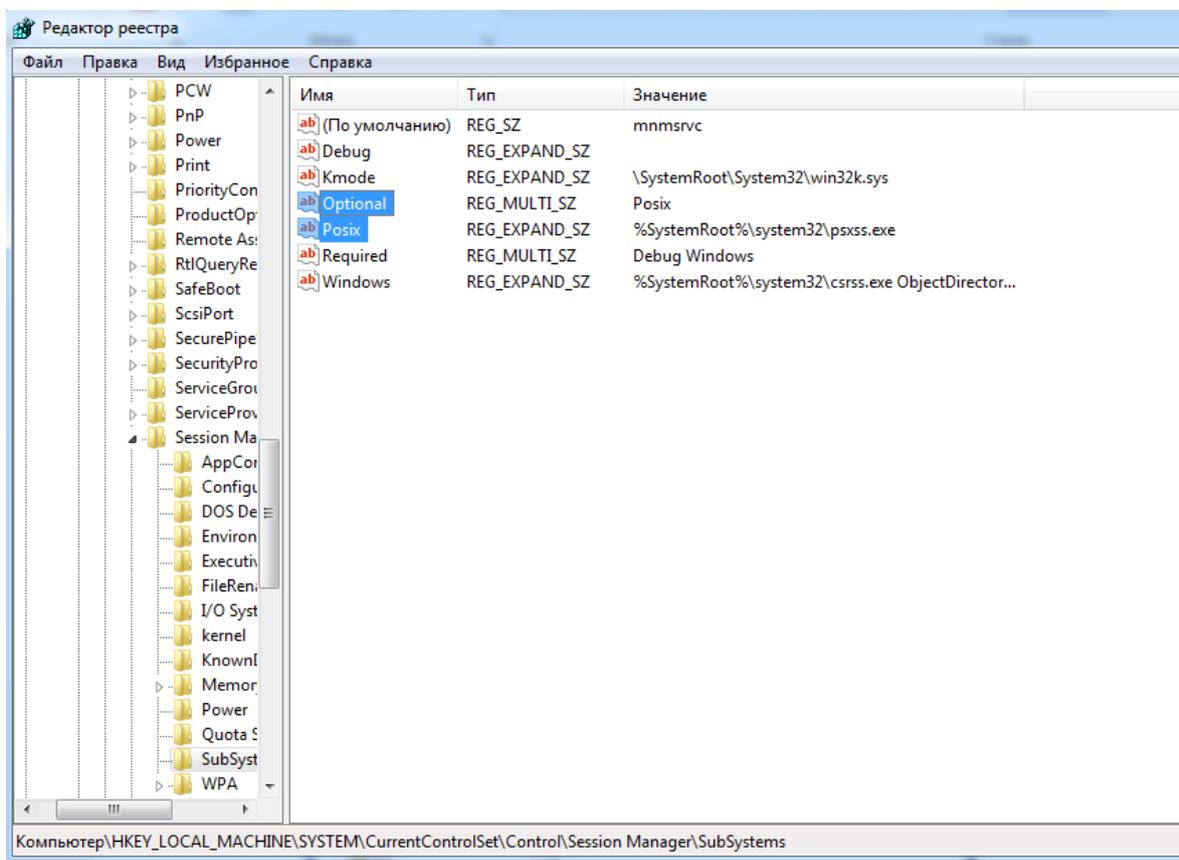


Рис.1.28. Редактор реестра

Включение опции «Большой системный кэш» (Large System Cache) также увеличивает общую производительность Windows 7, но оно рекомендовано только при условии, если вы имеете 256 Мбайт ОЗУ или больше. Необходимо открыть системный реестр и в разделе HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Control \Session Manager \Memory Management изменить значение ключа LargeSystemCache (тип dword) с 0 на 1 (рис.1.29, 1.30).

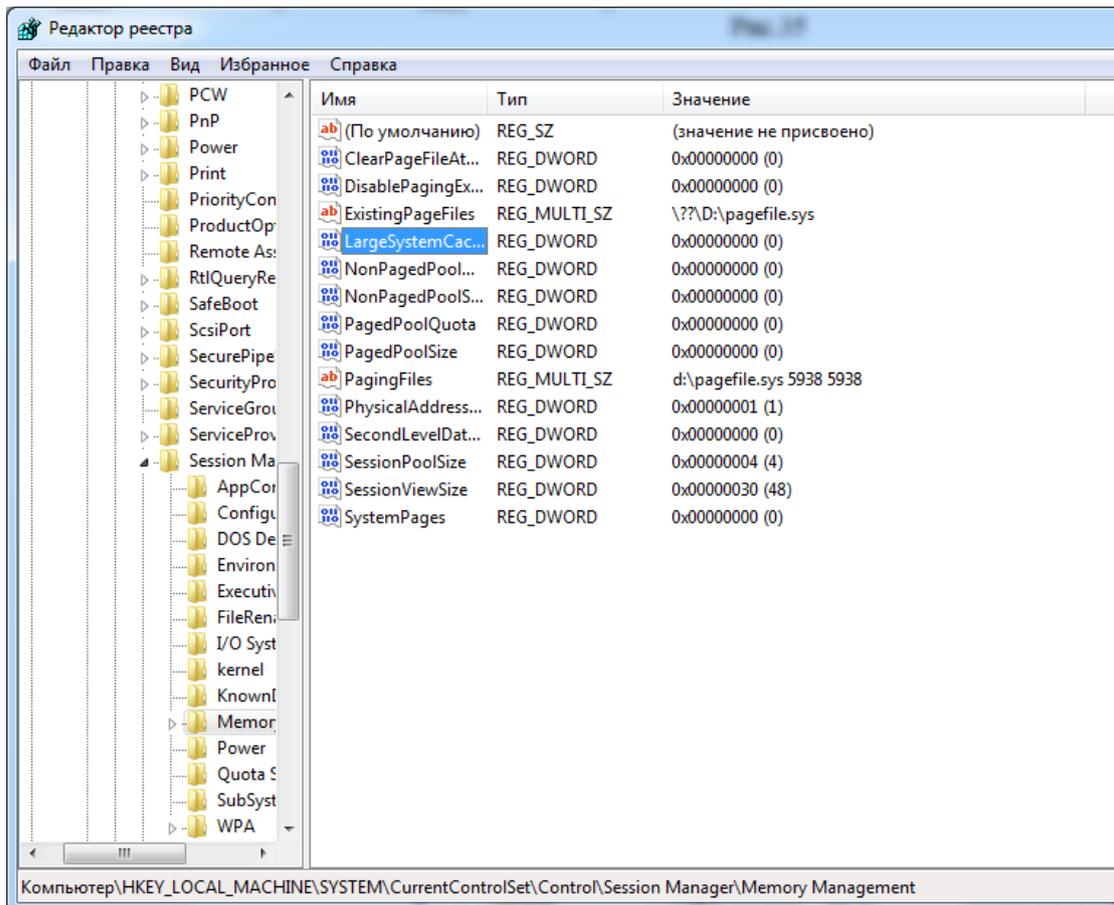


Рис.1.29. Редактор реестра

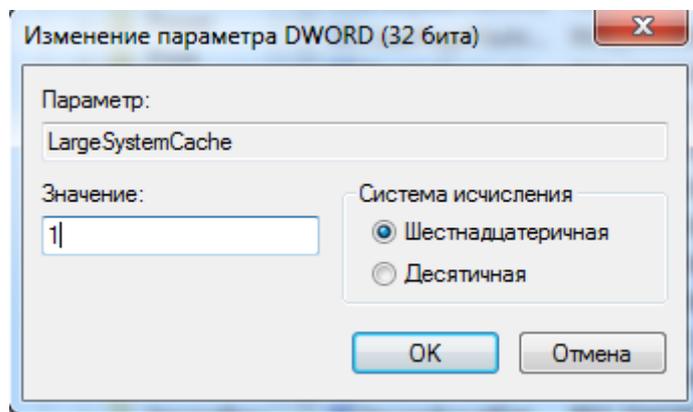


Рис.1.30. Изменение параметра DWORD

Слегка снижается производительность системы в результате установки размера L2-кэша для процессора по умолчанию. Поэтому лучше сделать это вручную. Правда, такая установка имеет смысл лишь для старых компьютеров с размером ОЗУ более 64 Мбайт с direct-mapped-кэшами второго уровня. Процессоры Pentium II и выше не имеют такого кэша, поэтому изменение размера L2-кэша для них не производится. По умолчанию для размера кэша используется значение 256 Кбайт. Чтобы установить его вручную, введите значение параметра

SecondLevelDataCache (тип dword) в разделе системного реестра HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Control \Session Manager \Memory Management в соответствии с его реальным размером. Значение вводится в байтах, а не в килобайтах (рис. 1.31, 1.32).

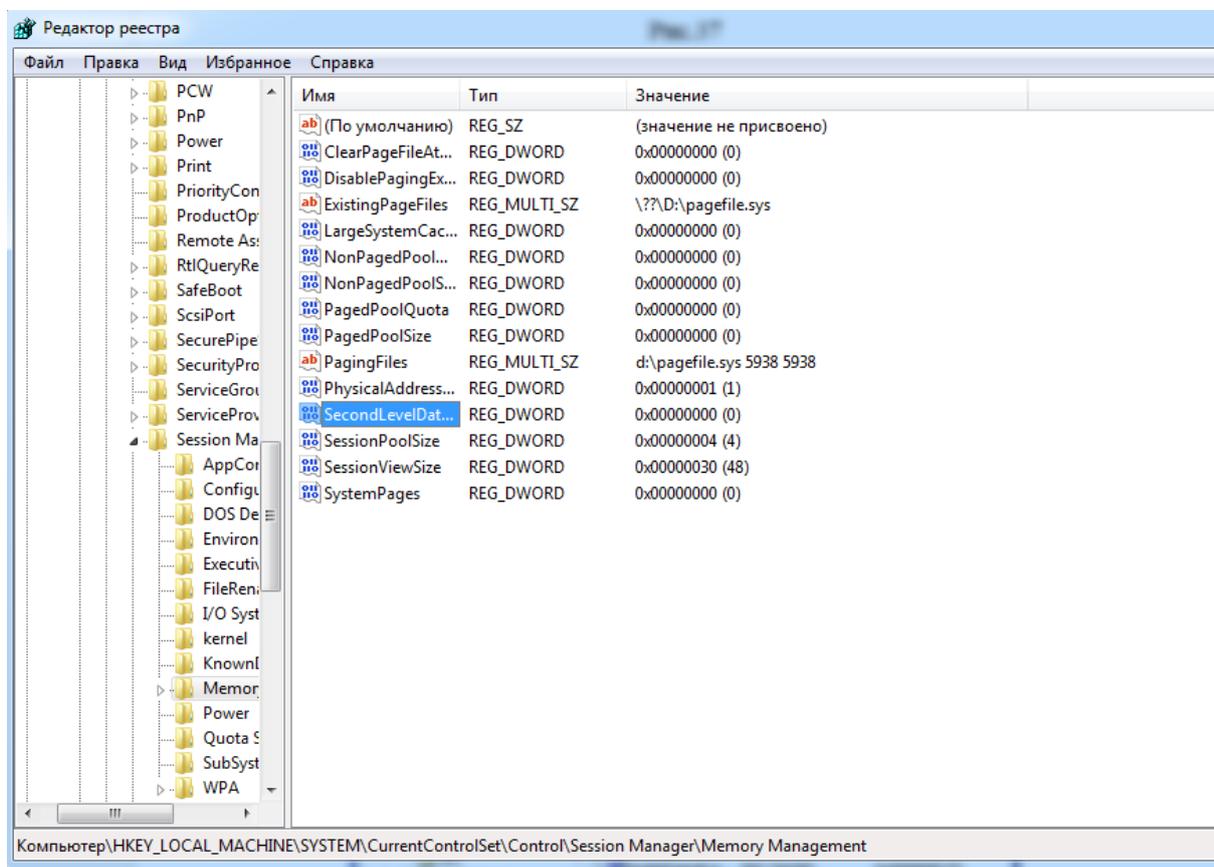


Рис.1.31. Редактор реестра

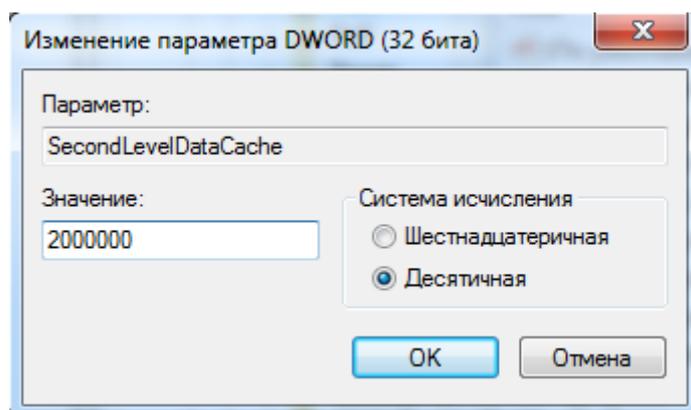


Рис.1.32. Изменение параметра DWORD

1.3. Порядок выполнения работы

Лабораторная работа выполняется в три этапа. Вначале студенты изучают возможности организации и поддержки. Далее студенты на своих компьютерах выполняют индивидуальную настройку рабочего стола. На

втором этапе студенты осваивают приемы работы с Диспетчером задач и Планировщиком заданий. На третьем этапе студенты изучают и осваивают некоторые приемы работы с системным реестром Windows 7, связанные с различными аспектами оптимизации ее среды.

Изучение материалов по рассматриваемой теме рекомендуется проводить в следующей последовательности.

1. Изучение теоретического материала по теме лабораторной работы.
2. Освоение приемов индивидуальной настройки рабочего стола.
3. Освоение приемов работы с Диспетчером задач и Планировщиком заданий.
4. Освоение приемов настройки виртуальной памяти.
5. Освоение других приемов оптимизации среды Windows 7.
6. Выполнение специальных заданий.

В ходе лабораторной работы необходимо изучить теоретический материал по данной теме, разобраться, для каких целей необходимо производить индивидуальную настройку и оптимизацию компонентов Windows 7, и получить опыт ее практического проведения.

1.4. Содержание отчета

1. Цель работы.
2. Описание результатов выполнения специальных заданий.
3. Выводы по проделанной работе.

1.5. Контрольные вопросы

1. Для чего необходима индивидуальная настройка и оптимизация Windows 7?
2. Что такое «Рабочий стол», каковы его основные параметры?
3. Каковы цели настройки Рабочего стола?
4. Перечислите основные направления настройки Рабочего стола.
5. На основе анализа состояния и вида Рабочего стола сформулируйте предложения по его оптимизации.
6. Как производится настройка отдельных элементов Рабочего стола?
7. Какие два стиля меню «Пуск» вы знаете и как можно перейти от одного к другому?
8. Расскажите об основных настройках меню «Пуск».

9. Каким образом можно изменить оформление Windows 7 (перейти к классическому стилю ОС)?
10. Перечислите основные настройки свойств папки.
11. К каким изменениям приводит отключение «Приветствия»?
12. Каким образом можно настроить **Панель задач**?
13. Каковы основные параметры **Панели задач**?
14. Какие параметры можно оптимизировать в элементе «Свойства системы»?
15. Произведите настройку рабочего стола компьютера по индивидуальному заданию преподавателя.
16. Опишите назначение и возможности Диспетчера задач.
17. Опишите основные приемы работы с Диспетчером задач.
18. Опишите назначение и возможности Планировщика заданий.
19. Опишите основные приемы работы с Планировщиком заданий.
20. Какими двумя подходами может быть осуществлена виртуализация памяти?
21. Что такое файл подкачки и для каких целей он предназначен?
22. Как можно изменить размер файла подкачки?
23. Возможна ли работа Windows 7 без файла подкачки?
24. Как осуществляется выбор оптимального значения файла подкачки?
25. Можно ли произвести дефрагментацию файла подкачки?
26. Как выполняется автоматическая очистка файла подкачки?
27. На основе анализа состояния и использования файловой системы сформулируйте предложения по их оптимизации (фрагментация, разделы, подкачка).
28. За счет каких средств можно добиваться ускорения работы Windows 7?
29. Опишите влияние визуальных эффектов Windows 7 на ее быстродействие.
30. Как можно изменять визуальные эффекты?
31. Каким образом распределение времени процессора может влиять на быстродействие Windows 7?
32. Дайте и реализуйте предложения по оптимизации рабочей среды многих пользователей данного компьютера (студентов).
33. Используя средства Диспетчера задач, дайте оценку оптимальности процесса автозагрузки данного компьютера.

ЛАБОРАТОРНАЯ РАБОТА № 2 «Возможности и средства ОС MS Windows»

2.1. Цель работы

Изучение назначения, возможностей и особенностей использования утилит Windows 7.

Получение навыков самостоятельного освоения утилит.

2.2. Теоретическая часть

Утилиты – вспомогательные (служебные) программы (для среды) ОС, решающие отдельные задачи управления и сопровождения компьютерной системы, например, программы сжатия дисков, архивирования данных и т.п. Обычно разделяют утилиты базового состава, входящие в состав ОС, и дополнительные утилиты для среды ОС, разрабатываемые другими производителями. Далее кратко рассмотрены основные утилиты базового состава Windows 7, подлежащие изучению и проверке возможностей использования в настоящей лабораторной работе.

Для вызова справки в среде Windows 7 можно использовать клавишу <F1>. Если необходимо получить справку по конкретной программе, то нужно в программе (утилите) вызвать в меню **Справка / Вызов справки**.

Рассматриваемые утилиты находятся в следующем месте системы меню: **Пуск/Программы/Стандартные/Служебные**. Программы дефрагментации и проверки диска также можно вызвать с рабочего стола:

Мой компьютер / Выбрать диск, далее правой кнопкой мыши выбрать **Свойства/Сервис**.

2.2.1. Программа получения сведений о системе

Она предназначена для отображения подробных сведений об оборудовании, системных компонентах и среде программного обеспечения (рис.2.1) и выводит информацию по следующим темам:

- общие сведения о системе (название ОС, ее версия, фирма-разработчик, папка Windows, системная папка, язык общения, имя пользователя, часовой пояс и т.д.);
- ресурсы аппаратуры;
- компоненты – CD-ROM, мультимедиа, звуковое устройство, дисплей

(видеокарта), ввод (клавиатура), модем и т.д.;

- программная среда – системные драйверы, подписанные драйверы, переменные среды, задания для принтера, сетевые подключения и т.д.;
- параметры обозревателя – Internet Explorer, язык, версия, путь приложения, шифр и т.д.;
- приложения;
- приложения установленного набора MS Office.

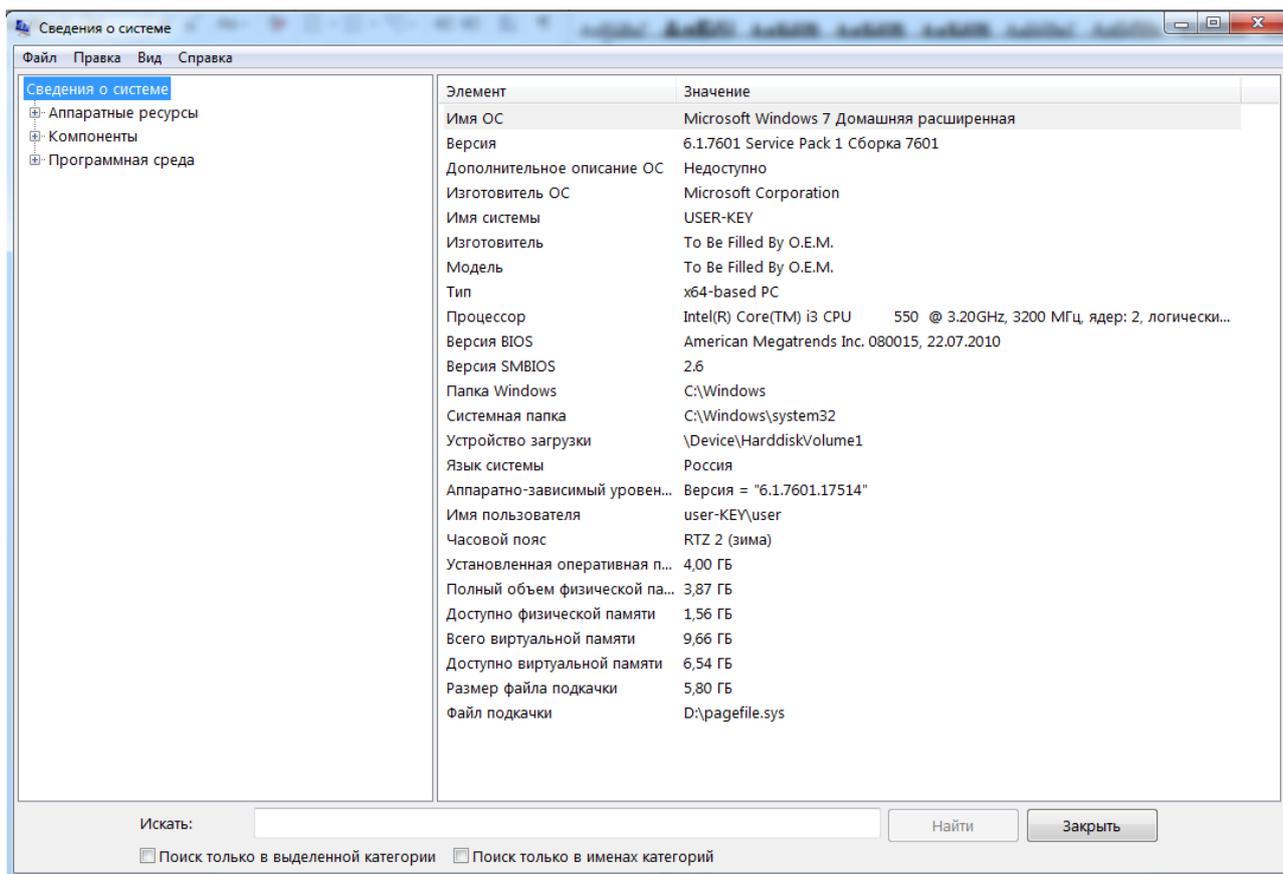


Рис.2.1. Сведения о системе

2.2.2. Программа архивации данных

Помогает создать копию данных на жестком диске. Если исходные данные будут случайно удалены, заменены или станут недоступны из-за неисправности жесткого диска, можно использовать резервную копию для их восстановления. Для правильной работы программы архивации должна быть запущена служба «Съемные ЗУ».

2.2.3. Программа проверки диска

Служит для обнаружения ошибок файловой системы и поврежденных секторов на жестком диске. Для ее вызова: откройте с

рабочего стола окно **Мой компьютер** и выберите локальный диск, который требуется проверить; в меню **Файл** выберите команду **Свойства**; на вкладке **Сервис** в группе **Проверка диска** нажмите кнопку **Выполнить проверку**; в группе **Параметры проверки диска** установите флажок **Проверить и восстанавливать поврежденные сектора**.

Перед запуском проверки диска закройте все файлы на нем. Если том используется, на экран будет выведено сообщение с предложением выполнить проверку диска после перезагрузки системы. При положительном ответе проверка диска будет запущена после перезагрузки компьютера. Во время проверки диск недоступен для выполнения других задач.

При использовании файловой системы NTFS ОС Windows 7 выполняет запись всех транзакций для файлов, автоматически заменяет поврежденные кластеры и сохраняет на томе NTFS копию наиболее важных данных о каждом из файлов.

В случае некорректного завершения работы ОС Windows 7, во время следующей загрузки ОС программа проверки диска автоматически проверит файловую систему и жесткий диск на наличие ошибок, и по возможности устранит их.

2.2.4. Программа дефрагментации дисков

Для выполнения некоторых задач может потребоваться войти в систему с учетной записью «Администратор» или члена группы «Администраторы».

Программа дефрагментации (рис.2.2) объединяет фрагментированные файлы и папки на жестком диске компьютера, после чего каждый файл или папка тома занимает единое непрерывное пространство. В результате доступ к файлам и папкам выполняется эффективнее, повышается быстродействие системы. Объединяя отдельные части файлов и папок, программа дефрагментации также объединяет в единое целое свободное место на диске, что делает менее вероятной фрагментацию новых файлов.

Программу дефрагментации также можно запустить из командной строки с помощью команды defrag.

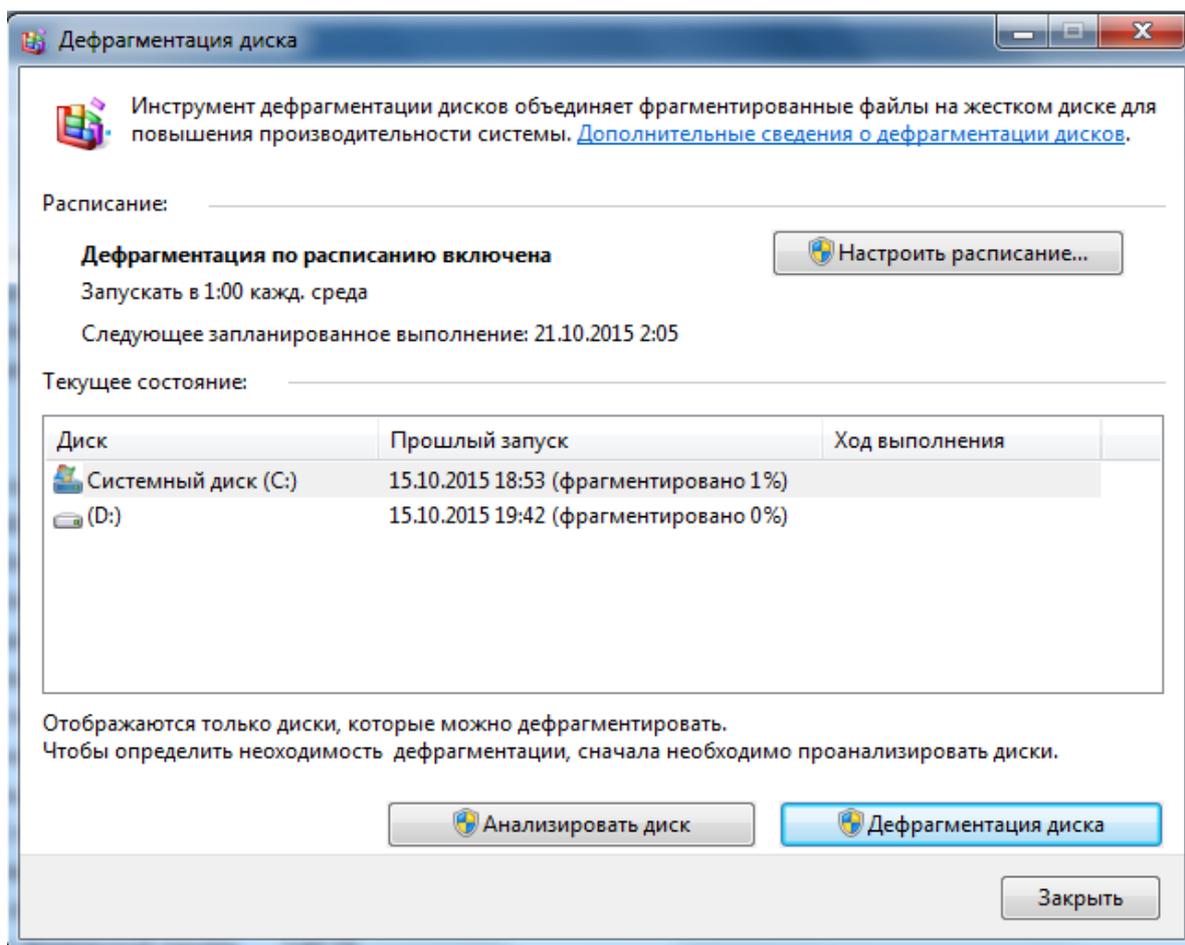


Рис.2.2. Дефрагментация диска

2.2.5. Программа очистки жесткого диска

Помогает очистить пространство на диске, проверяет диск и выводит перечень временных файлов, файлов кэша Internet, а также ненужных программных файлов, удаление которых не приведет к негативным последствиям. Можно выбрать удаление некоторых или всех этих файлов.

Объект рабочего стола **Корзина** выводит окно (рис.2.3) и обеспечивает дополнительную безопасность при удалении файлов или папок. При удалении файла или папки с жесткого диска, Windows 7 помещает его в корзину, заполняя и значок корзины. Элементы, удаляемые с сетевого или гибкого дисков, удаляются сразу, минуя корзину.

Файлы или папки остаются в корзине, пока пользователь не удалит их. Эти элементы все еще занимают место на диске и могут быть восстановлены в их исходное местоположение. Когда корзина заполняется, ОС Windows 7 автоматически очищает в корзине необходимое место для наиболее часто удаляемых файлов и папок.

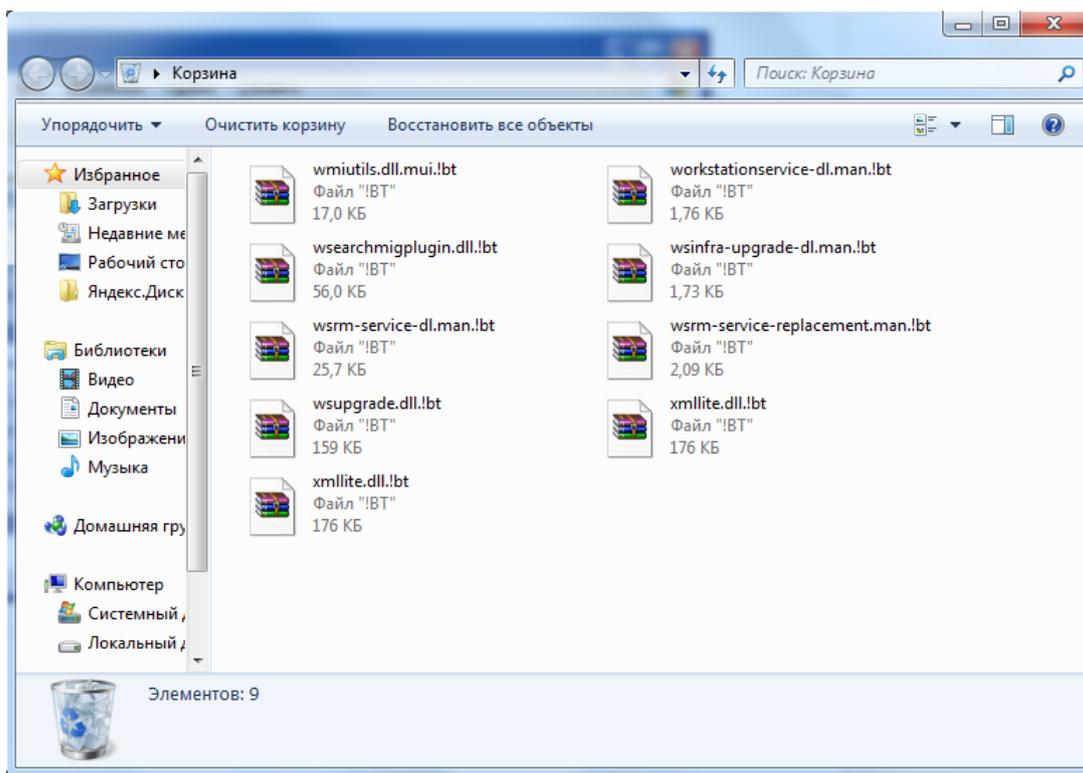


Рис.2.3. Корзина.

При нехватке места на жестком диске периодически освобождайте корзину. Для ограничения пространства, занимаемого корзиной можно изменить значение ее предельного объема. В ОС Windows 7 предусмотрена одна корзина для каждого раздела/жесткого диска со своим устанавливаемым предельным объемом корзины.

Чтобы удалить файлы без возможности их восстановления,

- щелкните правой кнопкой значок **Корзина** на рабочем столе и выберите пункт **Свойства**; установите флажок **Уничтожать файлы сразу после удаления, не помещая их в корзину**;
- при перетаскивании файлов в корзину удерживайте нажатой клавишу <SHIFT>;
- используйте для удаления файлов и папок комбинацию клавиш <SHIFT>+.

Если требуется задать различные настройки корзины для разных дисков, выберите параметр **Независимая конфигурация дисков**, а затем щелкните вкладку диска, для которого требуется изменить настройку.

Если требуется использовать общую настройку корзины для всех дисков, выберите параметр **Единые параметры для всех дисков**.

2.2.6. Программа «Восстановление системы»

В случае возникновения проблем в работе системы можно восстановить компьютер до последнего устойчивого состояния без потери файлов личных данных (документов, содержимого папки **Избранное** для работы в Internet, сообщений электронной почты). Программа «Восстановление системы» ведет наблюдение за изменениями в компьютере и периодически создает легко идентифицируемые **точки восстановления**. Эти точки восстановления позволяют вернуть систему к предыдущему состоянию.

«Восстановление системы» – компонент ОС Windows 7 Professional, с помощью которого при возникновении проблем можно восстановить предыдущее состояние компьютера без потери личных файлов (документов MS Word, перечня просмотренных страниц, рисунков, избранных файлов и сообщений электронной почты). Эта программа ведет наблюдение за изменениями системы и некоторыми файлами приложений и автоматически создает легко идентифицируемые точки восстановления. Точки восстановления позволяют вернуть систему к состоянию на данный момент времени. Они создаются ежедневно, а также во время существенных системных событий, таких как установка программ, утилит или драйверов для устройств. Пользователь также имеет возможность в любое время создавать именованные точки восстановления.

Программа «Восстановление системы» решает следующие задачи:

- восстановление предыдущего состояния компьютера. Например, если случайно были удалены или повреждены наблюдаемые программные файлы (такие, как файлы с расширением *.exe или *.dll), то можно восстановить состояние компьютера, предшествующее внесению изменений. По умолчанию отслеживаются и восстанавливаются все разделы и диски на компьютере. Она также отслеживает настройку всех приложений и дисков, которые осуществляют пользователи с помощью CD-ROM или гибкого диска, сервера управления системой (System Management Server, SMS) или IntelliMirror. В некоторых случаях при восстановлении системы восстанавливается папка, имя которой совпадает с именем существующей папки. Чтобы не переписывать существующие файлы, программа «Восстановление системы» переименовывает папку, добавляя к ее имени числовой суффикс;

- восстановление состояния компьютера без потери личных файлов или пароля. Не выполняется восстановление файлов в папке **Мои документы**, а также файлов данных с часто используемыми расширениями (*.doc, *.xls);
- сохранение прошлых точек восстановления с давностью 1-3 недели. Фактическое число сохраненных точек восстановления зависит от активности использования компьютера, размера жесткого диска (или размера раздела, содержащего каталог Windows 7 Professional) и количества места на жестком диске, выделенного для хранения сведений программой «Восстановления системы»;
- обнаружение дат, соответствующих точкам восстановления. При запуске программы появляется календарь, позволяющий пользователю найти даты, соответствующие точкам восстановления. Если компьютер используется не каждый день, то в течение нескольких дней может существовать только одна точка восстановления. Если компьютер используется ежедневно, то точка восстановления будет приходиться почти на каждый день, а некоторым датам может соответствовать несколько точек восстановления.

Обратимость. Если пользователю не нравится состояние компьютера после его восстановления, то можно отменить восстановление или выбрать другую точку восстановления. Все удачные операции восстановления обратимы. Все неудачные операции обновления автоматически отменяются программой восстановления системы.

Используются следующие **типы точек восстановления**:

- 1) **контрольные точки исходной системы** – создаются при первом запуске компьютера после обновления системы на Windows 7 Professional или при первом запуске нового компьютера. Выбор этой точки возвратит Windows 7 Professional и программы в начальное состояние. Все файлы данных с известными расширениями (такими как *.doc, *.htm, *.xls и так далее) и файлы в папке «Мои документы» не будут восстановлены;
- 2) **контрольные точки системы** – создаются автоматически, даже если в системе не было сделано никаких изменений (каждые 24 часа по календарному времени; каждые 24 часа при включенном компьютере).

Для создания точки восстановления по расписанию необходимо, чтобы компьютер находился в состоянии простоя несколько минут. Все файлы данных с известными расширениями (такими как *.doc, *.htm, *.xls и так далее) и файлы в папке «Мои документы» не будут восстановлены;

- 3) **точки восстановления установки имени программы.** При установке программ с помощью последних программ установки (InstallShield, Windows 7 Professional Installer), создается точка восстановления для отслеживания изменений в системе или для восстановления состояния компьютера, в котором он находился до установки программы. При выборе такой точки все установленные файлы будут удалены и исходные параметры реестра, все программы и системные файлы будут восстановлены в том виде, в котором они были до установки. Чтобы отменить изменения, внесенные программой, которая не использует один из указанных установщиков, выберите предшествующую точку восстановления, наиболее близкую к дате установки программы;
- 4) **точки восстановления для автоматического обновления Windows 7 Professional** – создаются до (при начале) установки обновленного программного обеспечения. Эти точки используются для отслеживания изменений в системе и, в редких случаях, отслеживают конфликт этих обновлений с прочими программами, установленными на компьютере;
- 5) **(собственные) точки восстановления, созданные вручную** – с помощью мастера восстановления системы. Такая точка отображается на экране «**Выберите контрольную точку восстановления**» с присвоенным ей именем и соответствующими значениями даты и времени ее создания. Точки восстановления можно создавать при настройке удачной конфигурации компьютера или перед внесением изменений, например перед установкой на компьютер программ, которые могут повлиять на его работу;
- 6) **точки восстановления операции восстановления** – создаются для отслеживания вносимых при восстановлении изменений и обеспечения возможности их отмены. Чтобы отменить восстановление, выберите соответствующую точку восстановления операции восстановления на экране «**Выберите контрольную точку восстановления**» мастера восстановления системы;

- 7) **точки восстановления неподписанных драйверов устройств** – создаются при обнаружении установки нового драйвера, не подписанного и не сертифицированного организацией Windows Hardware Quality Labs (WHQL). И если установка драйвера привела к нежелательным изменениям, можно отменить эти изменения и вернуть компьютер к состоянию до установки драйвера;
- 8) **точки восстановления программы Backup** – создаются до начала архивации. И если после операции восстановления компьютер перешел в неверное состояние, можно вернуть компьютер к состоянию до выполнения восстановления по архивной копии.

Замечание. Если выбранная точка восстановления была создана до установки программы, эта программа не будет работать после восстановления.

2.3. Порядок выполнения работы

Изучение материалов по рассматриваемой теме рекомендуется проводить в следующей последовательности.

1. Программа получения сведений о системе.
2. Программа проверки диска.
3. Программа дефрагментации диска.
4. Программа очистки диска.
5. Программа восстановление системы.
6. Программа архивации данных.

В ходе выполнения работы студент должен на практике проверить и проанализировать возможности основных встроенных утилит Windows 7.

2.4. Содержание отчета

1. Цель работы.
2. Перечень наиболее важных свойств и особенностей изученных утилит.
3. Результаты выполнения специального задания.
4. Выводы по проделанной работе.

2.5. Контрольные вопросы

1. Перечислите основные утилиты Windows 7 и назовите их назначение.
2. Для каких целей используется утилита «Получение сведений о системе», какова область ее применения?
3. Перечислите системные средства (из меню «Сервис»), входящие в утилиту «Получение сведений о системе».
4. Назовите область применения системных средств «Проверка подписи файла» и «Средство диагностики DirectX».
5. Для каких целей используется утилита «Проверка диска», какова область ее применения?
6. Выполните проверку дисков компьютера и оцените состояние их разделов.
7. Для чего и как используется утилита «Дефрагментация диска»?
8. Оцените состояние разделов жесткого диска и выполните их дефрагментацию.
9. Для чего и как используется утилита «Очистка диска»?
10. Оцените загруженность разделов жесткого диска и выполните их очистку.
11. Для чего и как используется утилита «Восстановление системы»?
12. Перечислите этапы восстановления более раннего состояния компьютера.
13. Перечислите этапы создания контрольной точки восстановления.
14. Перечислите и кратко охарактеризуйте типы точек восстановления.
15. Для чего и как используется утилита «Архивация данных»?

ЛАБОРАТОРНАЯ РАБОТА № 3 «Средства сохранения и восстановления программной среды ОС MS Windows»

3.1. Цель работы

Изучение основных сведений и способов работы с интегрированным пакетом программ Acronis True Image, обеспечивающим сохранение и защиту всей информации на компьютере.

Практическое освоение пакета Acronis True Image.

3.2. Теоретическая часть

Программа Acronis – это система резервного копирования диска. У нее довольно много полезных функций.

Традиционный пакет «Acronis» состоит из двух основных компонентов: **Acronis Disk Director** и **Acronis True Image**. Сегодня мы научимся работать с **Acronis True Image**.

Программа Acronis True Image, предназначена для создания резервной копии всей вашей системы и последующего восстановления таковой по мере необходимости.

Образ системы лучше всего хранить в безопасной области жесткого диска, созданной самой программой.

Acronis True Image 2016 – это интегрированный пакет программ, обеспечивающий защиту всей информации на компьютере. С его помощью можно создавать резервные копии документов, фотографий, электронных писем, выбранных разделов или целого диска, включая операционную систему, приложения, настройки и все данные. Резервные копии позволяют восстановить систему компьютера при потере данных, случайном удалении важных файлов и папок или полном отказе жесткого диска. Функция Online Backup позволяет хранить файлы и диски в Acronis Cloud. Данные будут защищены даже в случае потери, кражи или поломки компьютера, и при необходимости их можно будет полностью восстановить на новое устройство. Основные функции:

- Резервное копирование диска в локальное хранилище и Acronis Cloud.
- Резервное копирование файлов в локальное хранилище и Acronis Cloud.
- Загрузочный носитель.
- Клонирование жесткого диска.

- Архивирование данных.
- Защита домашних данных.
- Синхронизация файлов.
- Средства безопасности и конфиденциальности.

В **Acronis True Image 2016** появился ряд новых функций:

1. **Защита домашних данных.** В современной семье часто имеется более одного компьютера (ПК, Mac), а также несколько планшетов и смартфонов. Обычно кто-то из членов семьи обладает большим опытом работы с технологиями, чем другие. С помощью нового веб-приложения домашний ИТ-администратор может:
 - контролировать текущее состояние всех резервных копий и синхронизаций на всех домашних устройствах;
 - вручную запускать резервное копирование любых данных на любом компьютере;
 - восстанавливать данные из любой резервной копии в Acronis Cloud, включая резервные копии с ПК, компьютеров Mac и устройств под управлением iOS и Android;
 - устранять некоторые неполадки, связанные с продуктом.
2. **Архивирование данных.** **Acronis True Image 2016** просматривает выбранные папки и анализирует, как часто эти данные используются. Такой анализ позволяет найти редко используемые файлы. Их можно переместить в Acronis Cloud. В результате этого освободится место на жестких дисках, а оставшиеся на дисках данные будут храниться в оптимальном порядке. Архивированные файлы сохраняются в Acronis Cloud и будут доступны со всех ваших компьютеров и мобильных устройств, работающих под управлением операционной системы iOS или Android.
3. **Поддержка нескольких языков.** Язык интерфейса продукта можно просто изменить в настройках — переустанавливать Acronis True Image не требуется.
4. **Улучшенный интерфейс пользователя.** Для более удобного использования был переработан интерфейс Acronis True Image 2016, включая диалоговые окна резервного копирования и восстановления.
5. **Поддержка Windows 10.** Acronis True Image 2016 можно установить на ОС Windows 10 Insider Preview. Также продукт оптимизирован для запуска в этой операционной системе.

3.3. Порядок выполнения работы

Для создания образа сделаем следующие действия:

1. Выберите пункт меню «Создание зоны безопасности» в главном окне программы, или на вкладке «Сервис». Нажмите далее в появившемся окне (рис.3.1).

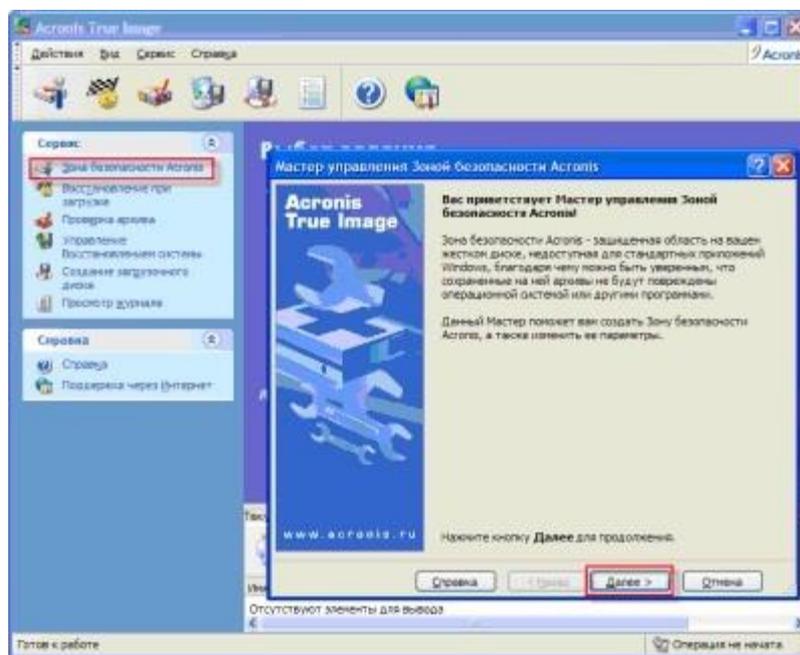


Рис.3.1. Мастер управления зоной безопасности Acronis True Image

2. Укажите размер зоны безопасности (не следует делать его меньше 3,5 Гб) и нажмите кнопку «Далее» (рис.3.2).

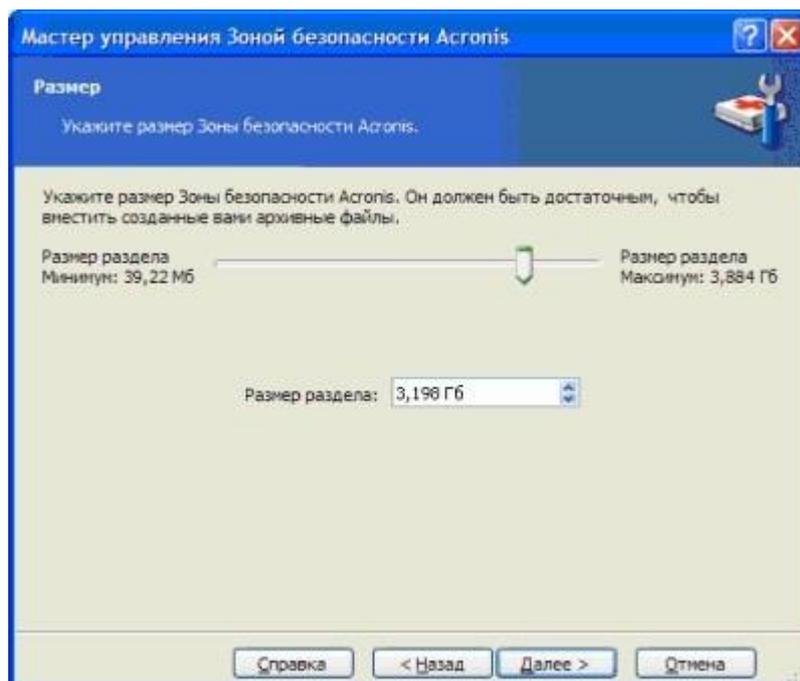


Рис.3.2. Указание размера зоны безопасности

3. После установки необходимого размера, в настройках программы отметьте пункт «Восстановление при загрузке». Благодаря этой функции вы сможете восстановить систему из образа при невозможности загрузки рабочей системы (рис.3.3).

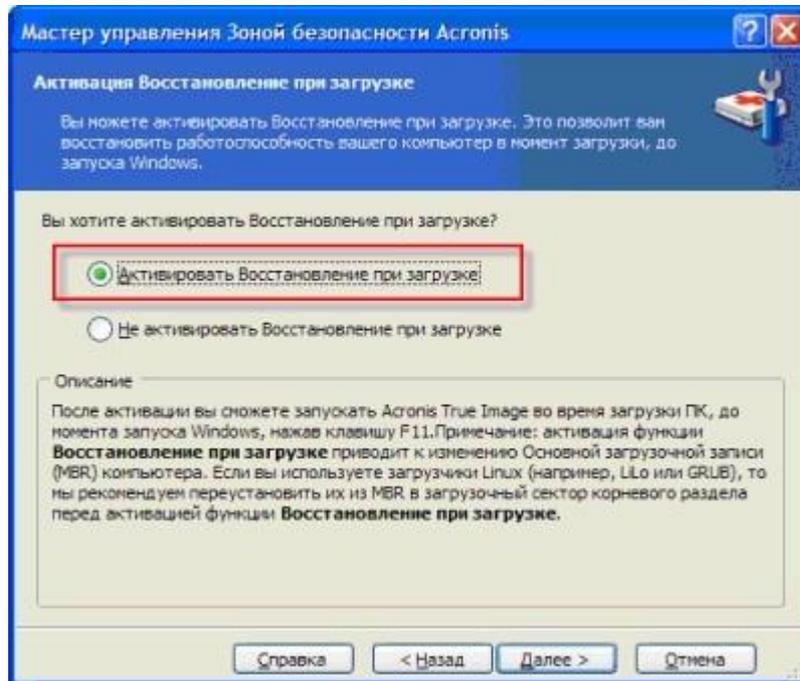


Рис.3.3. Выделение пункта «Восстановление при загрузке»

4. После этого нажмите на кнопку «Приступить» и дождитесь окончания операции (рис.3.4). По требованию программы обязательно перезагрузите компьютер.

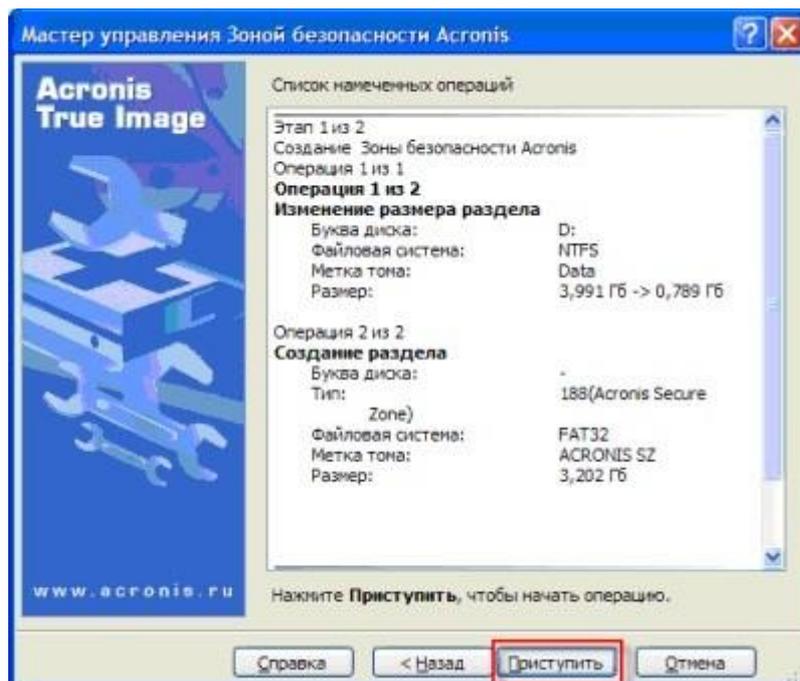


Рис.3.4. Создание полного архива

5. Далее создайте сам образ системы. Для этого в главном окне программы выберите опцию «Создать архив» (рис.3.5).

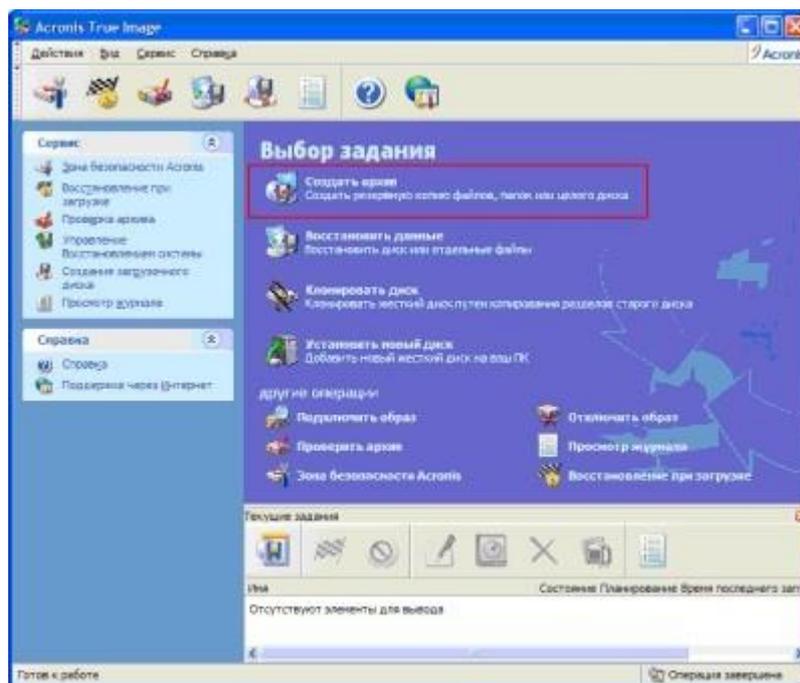


Рис.3.5. Создание полного архива

6. Для создания полного образа системы, отметьте пункт «Жесткий диск целиком» и нажмите «Далее» (рис.3.6)

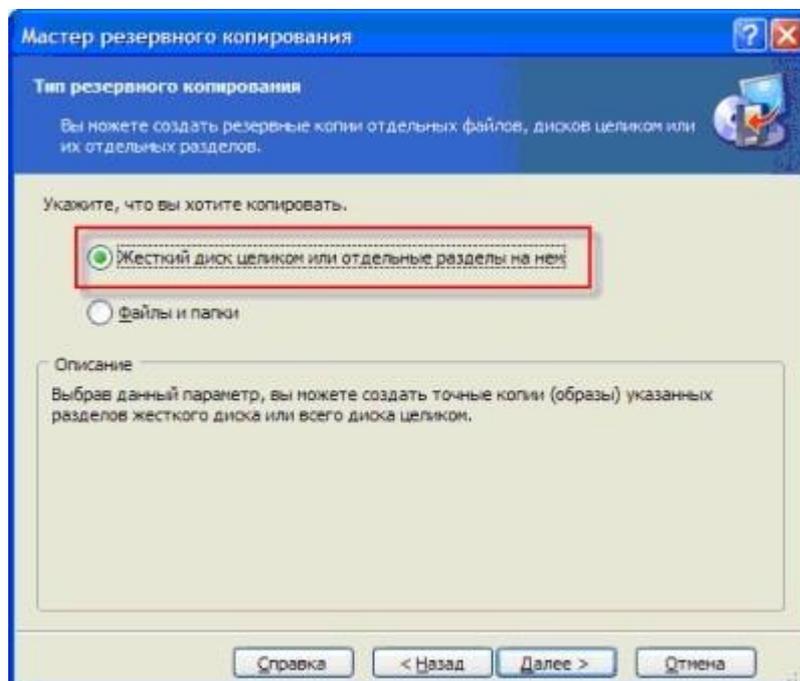


Рис.3.6. Создание полного архива

7. Затем, в списке дисков выберите системный раздел и нажмите «Далее» (рис.3.7).

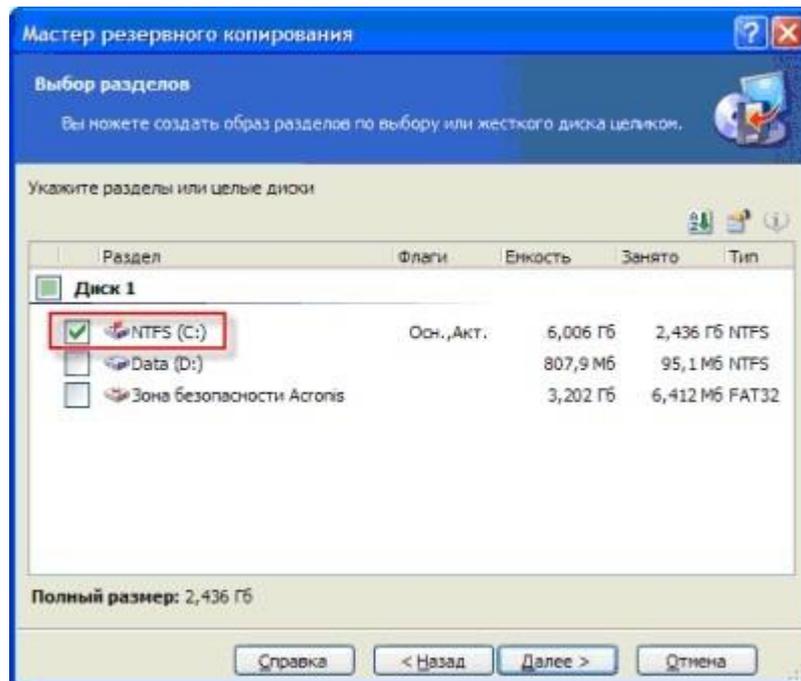


Рис.3.7. Образ системы

8. Выберите место хранения образа системы (в нашем случае зону безопасности Acronis) и нажмите «Далее» (рис.3.8).

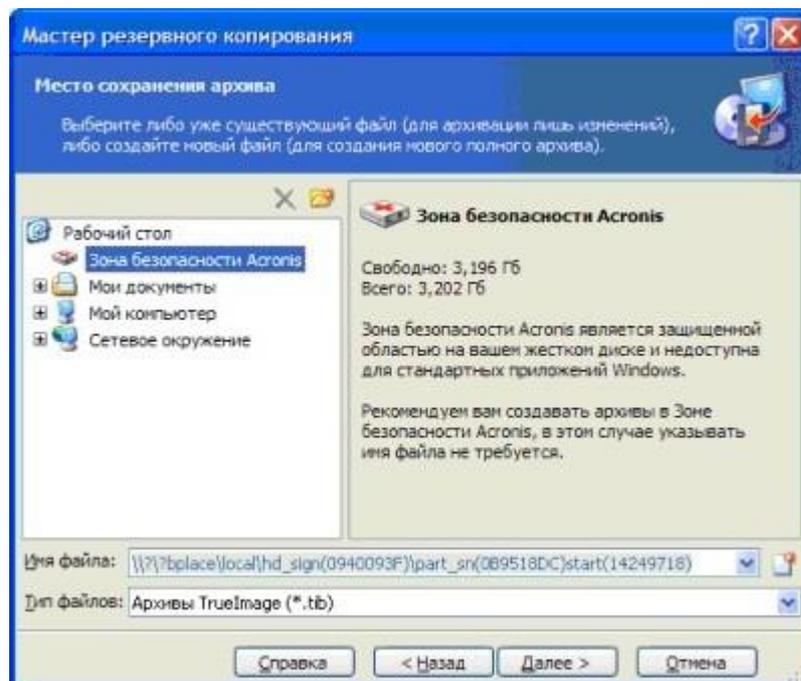


Рис.3.8. Выбор места хранения образа системы

При первом запуске программы следует выбрать опцию «**Создание полного архива**», который можно будет дополнять в будущем инкрементными архивами, содержащими внесенные изменения. А вообще, специалисты рекомендуют создать «чистый» образ системы со всеми нужными вам программами и инструментами на случай краха

системы, а затем создавать новые инкрементные архивы. В таком случае вы будете иметь **Исходный образ системы** и **Образ системы на текущий момент**.

После выбора типа архива, нажмите «Далее» и «Использовать параметры по умолчанию» (Рис.3.9).

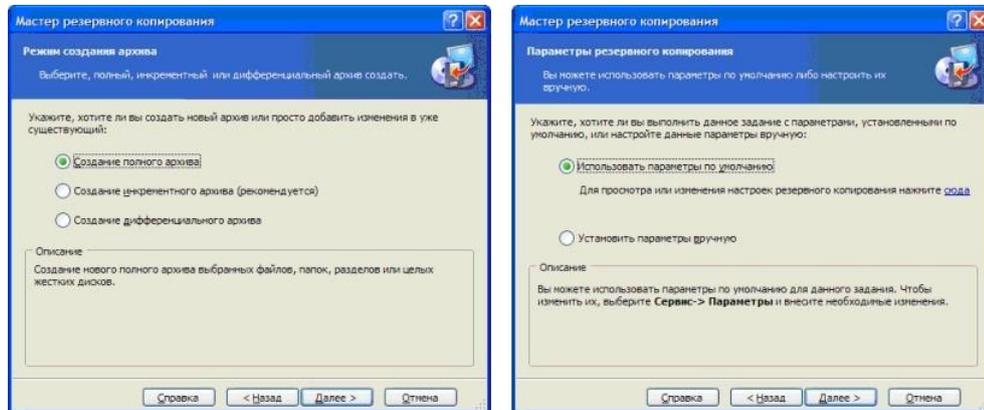


Рис.3.9. Образ системы

Для запуска процесса создания архива нажмите «Приступить» (Рис.3.10).

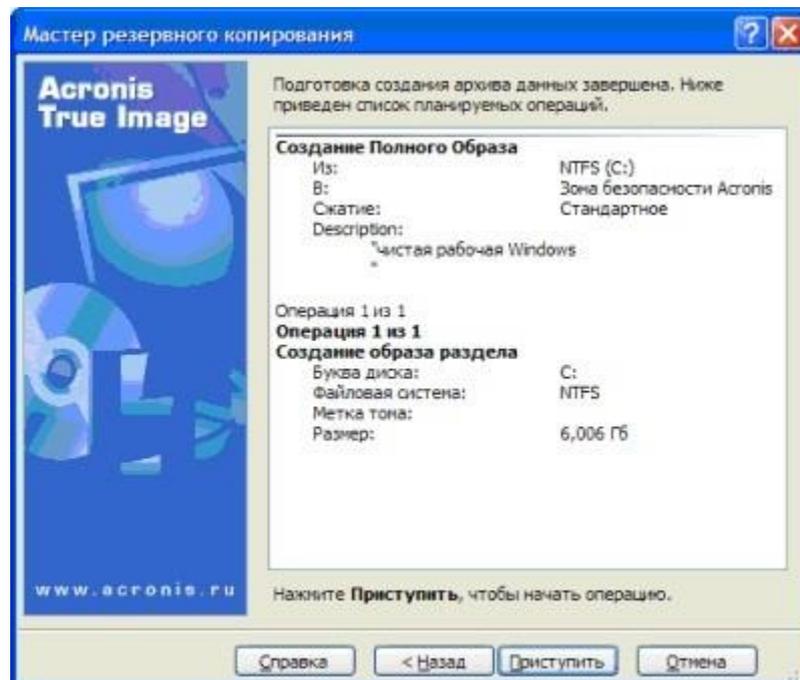


Рис.3.10. Восстановление при загрузке

После этого вы всегда сможете восстановить вашу систему в первоначальном виде. Для этого, во время загрузки нажмите F11 и дождитесь восстановления системы. Затем нажмите в окне программы «**Восстановить данные**» (Рис.3.11).

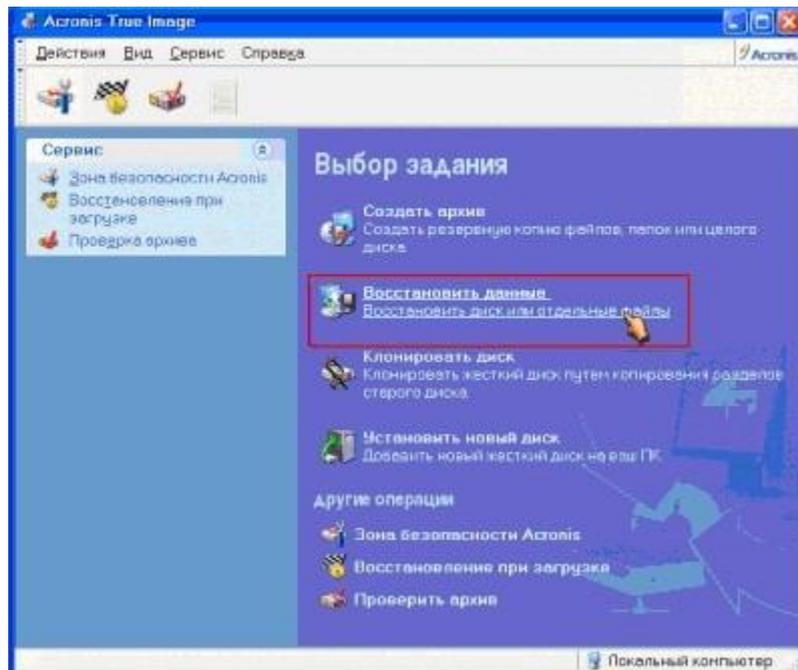


Рис.3.11. Восстановление при загрузке

Затем выберите архив для восстановления в зоне безопасности, поставьте отметку «**Восстановить диски и разделы**» (Рис.3.12).

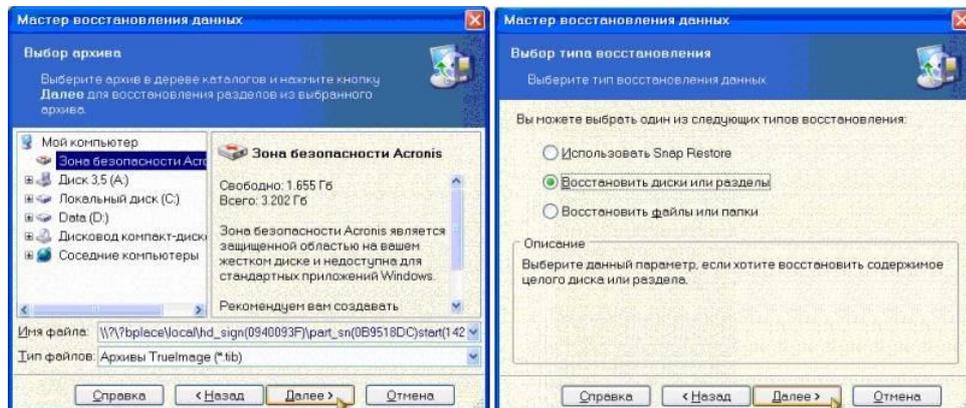


Рис.3.12. Восстановление при загрузке

Выберите раздел, который вы хотите восстановить и нажмите «Далее» (Рис.3.13).

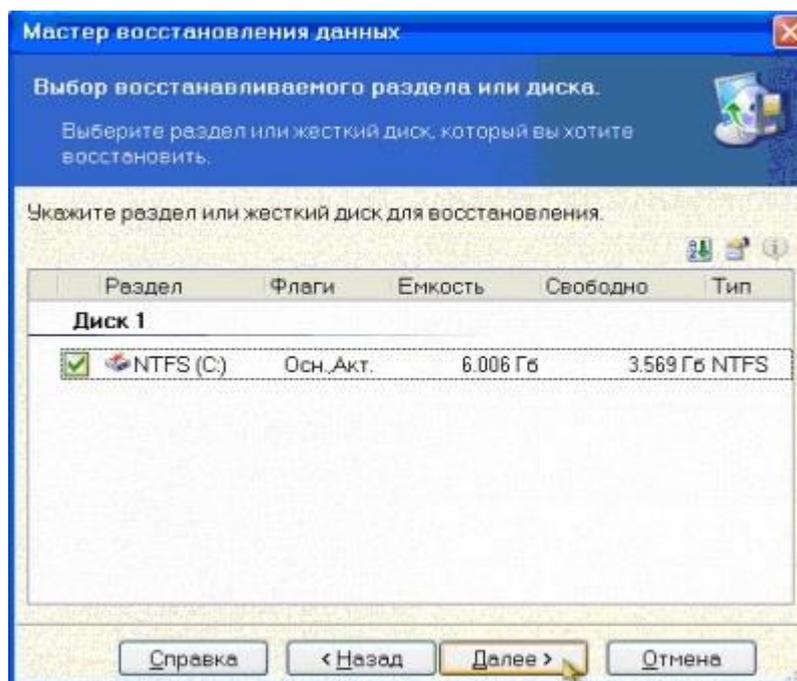


Рис.3.13. Восстановить данные

После этого, вам нужно будет заменить раздел с неработающей системой на раздел с архивной копией рабочей системы. Выберите активный раздел в списке и нажмите «Далее» (Рис.3.14).

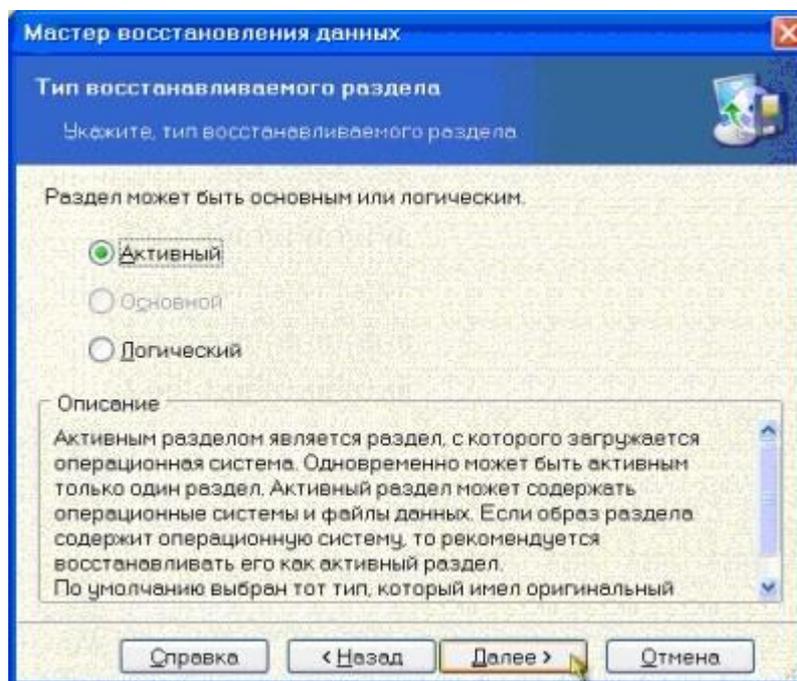


Рис.3.14. Восстановить данные

Затем назначьте букву для раздела. Для системного раздела обычно выбирают диск «С» (или выберите букву того раздела, на котором у вас ранее была установлена система). Нажмите «Далее» (Рис.3.15).

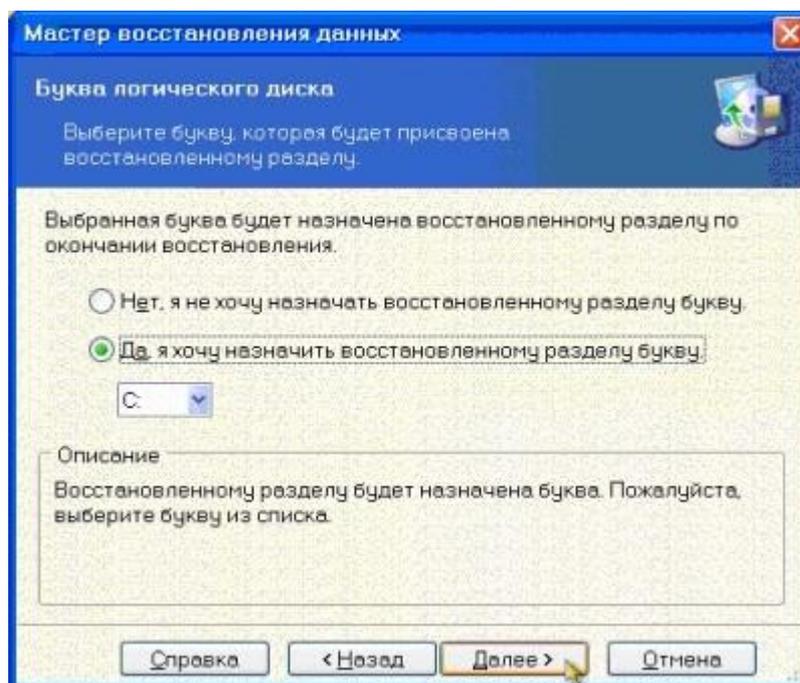


Рис.3.15. Восстановить диски и разделы

Проверку архивов и восстановленных файлов можно пропустить. При восстановлении единственного раздела, отметьте «нет, я не хочу» в окне запроса о восстановлении других разделов. Затем нажмите «Приступить» и ожидайте окончания работы программы (Рис.3.16).

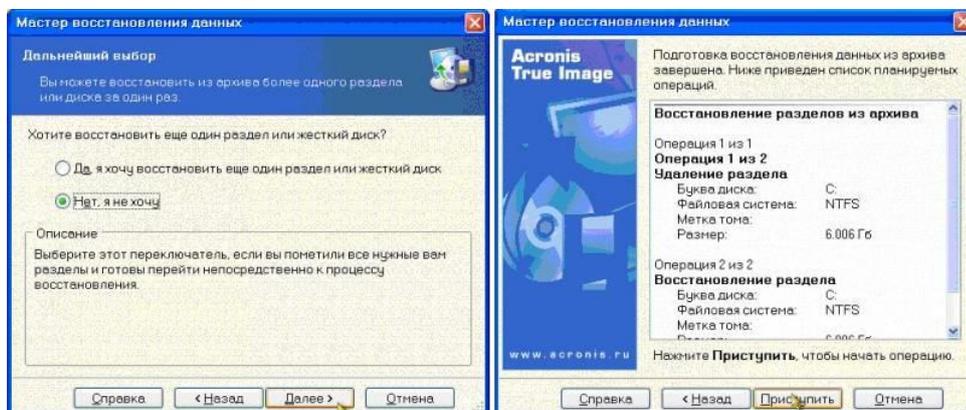


Рис.3.16. Восстановить диски и разделы

3.4. Индивидуальное задание

1. **Восстановление файлов и папок.** (Для выполнения задания, необходимо удалить на своем компьютере несколько файлов и папок, а после чего восстановить их (показать это в виде скриншотов)).

- Запустите Acronis True Image.
- На боковой панели нажмите кнопку Резервное копирование.

- Выберите из списка резервную копию, содержащую файлы и папки, которые необходимо восстановить.
- На правой панели нажмите кнопку Восстановить файлы.
- Выберите версию резервной копии (состояние данных в определенные дату и время)
- Выберите нужные файлы и папки и нажмите кнопку Далее.
- На компьютере укажите расположение, в которое нужно восстановить выбранные файлы или папки. Восстановите данные в прежнее хранилище или выберите другую папку. Для выбора нового хранилища нажмите кнопку Обзор.
- Если выбрано новое хранилище, выбранные объекты будут восстановлены по умолчанию без воссоздания их начального абсолютного пути. Также можно восстановить эти элементы на новое место со всей иерархией папок. В данном случае установите флажок Сохранить исходную структуру папок.
- При необходимости задайте параметры восстановления (приоритет процесса восстановления, параметры безопасности файлов и т. д.). Чтобы сделать это, нажмите Параметры. Параметры, заданные здесь, относятся только к текущей операции восстановления.
- Чтобы начать восстановление, нажмите кнопку Восстановить сейчас. Чтобы остановить восстановление, нажмите кнопку Отмена. Помните, что даже прерванное восстановление может вызвать изменения в целевой папке.

2. Резервное копирование файлов.

- Запустите Acronis True Image 2016.
- На боковой панели нажмите кнопку Резервное копирование.
- Щелкните значок Источник резервного копирования и выберите Файлы и папки.
- В открывшемся окне установите флажки напротив нужных файлов и папок и нажмите кнопку ОК.
- Щелкните значок Место назначения резервной копии и выберите место для сохранения резервной копии:

- Acronis Cloud — войдите в свою учетную запись Acronis и нажмите кнопку ОК.
- Внешний диск — если к компьютеру подключен внешний диск, его можно выбрать из списка.
- Обзор — выберите место назначения в дереве папок.
- Нажмите кнопку Создать резервную копию. Дополнительные сведения см. в разделе Резервное копирование файлов и папок.

3.5. Содержание отчета

1. Цель работы.
2. Перечень наиболее важных свойств и особенностей Acronis True Image.
3. Результаты выполнения специального задания.
4. Выводы по проделанной работе.

3.6. Контрольные вопросы

1. Для каких целей используется резервное копирование файлов?
2. Основные части AcronisTrueImage и их назначение.
3. Дайте определение функции «Архивирование данных».
4. Функция «Защита домашних данных» в AcronisTrueImage.
5. Дайте определение функции «Синхронизация файлов».
6. Средства безопасности и конфиденциальности в AcronisTrueImage.
7. Что такое «Зоны безопасности» в AcronisTrueImage.
8. Поддерживает ли AcronisTrueImage функцию UEFI / GPT?
9. Как создать резервную копию всего ПК?
10. Как активировать «Восстановление» при загрузке?

ЛАБОРАТОРНАЯ РАБОТА № 4 «Установка и настройка ОС Linux или иной ОС по выбору (Solaris, UNIX, MacOS, и др.)»

Часть 1 - «Настройка сетевых интерфейсов в ОС Solaris»

4.1. Цель работы

В ходе выполнения данной лабораторной работы студент получит первичные навыки конфигурирования сетевых интерфейсов, получит общее представление о сетевых возможностях операционной системы и оценит трудоемкость работ такого рода.

4.2. Теоретическая часть

4.2.1. Описание ОС Solaris 10

ОС Solaris 10 относится к семейству UNIX-систем и позиционируется производителем как операционная система, оптимизированная для работы на основных серверных платформах. Она построена с учетом требований, предъявляемых к критически важным информационным средам высокой готовности. Масштабируемость Solaris охватывает широкий спектр системных конфигураций — от ПК до суперкомпьютеров, обеспечивая гибкость в выборе аппаратной платформы с учетом конкретной задачи. Благодаря поддержке симметричного мультипроцессирования и многопоточности достигается приемлемое время отклика приложений и высокая пропускная способность в многопользовательских операционных окружениях. Кроме того, Solaris работает на наиболее популярных RISC- и CISC-платформах, включая SPARC, x86 и AMD64/EM64T, то есть обеспечивает независимость от аппаратной платформы и предоставляет возможность выбора. Поскольку все реализации Solaris опираются на единые исходные тексты, то и пользователи, и администраторы, и разработчики вправе рассчитывать на единообразное поведение ОС на всех платформах.

Создавая Solaris 10, специалисты корпорации Sun Micro Systems постарались наделить ее максимально возможным количеством функций. С помощью нового продукта Sun надеется обойти основных конкурентов на серверном рынке (в первую очередь, IBM, Hewlett-Packard и Dell).

4.2.2 Основные функциональные возможности

Так как Solaris 10 является системой UNIX семейства, она обладает большинством особенностей систем данного семейства. Но, для того чтобы занять лидирующие позиции на рынке серверных операционных систем компания Sun Micro Systems хорошо постаралась и снабдила систему множеством поддерживаемых технологий и дополнительных функций. В некоторых источниках указывается, что результатом работы специалистов Sun над новой версией Solaris стало появление в ней около 600 новых функций и нескольких уникальных программных механизмов. Проведем обзор самых главных на наш взгляд функций.

FireEngine - Подход "FireEngine" в Solaris 10 сводится к слиянию всех уровней протоколов в один полностью многопоточный модуль STREAMS. Внутри объединенного модуля вместо блокирования отдельных структур данных используется механизм синхронизации для отдельных процессоров, называемый "вертикальным периметром". "Вертикальный периметр" реализован с использованием абстракции очереди сериализации, называемой "queue". Каждая очередь "queue" связывается с ЦП, а каждое подключение в свою очередь связывается с очередью "queue", которая обеспечивает всю необходимую синхронизацию и взаимное исключение, требуемые структурами данных отдельных подключений.

Solaris Containers – встроенная контейнерная технология распределения системных ресурсов между отдельными процессами, группами процессов и пользователями. Она позволяет изолировать друг от друга различные участки ОС и приложения, благодаря чему существенно повышаются надежность и безопасность работы компьютера. На базе системы ИТ-администратор может создавать «виртуальные серверы», не пересекающиеся друг с другом ни по пользователям, ни по поддерживаемым процессам. Одновременное обслуживание нескольких десятков и даже сотен таких серверов не увеличит нагрузку на администратора: все «контейнеры» работают в рамках одной копии ОС Solaris и наследуют все установленные приложения, патчи и глобальные настройки системы.

Оперирование зонами и контейнерами в ОС Solaris 10 допускает создание на одном сервере до 8192 «виртуальных» операционных систем,

каждая из которых полностью автономна, со своими сетевыми адресами, пространством памяти, файловой системой, запущенными процессами и, разумеется, со своей базой пользователей. Этот набор независимых виртуальных серверов Solaris 10, работающих на одном физическом сервере, — не что иное, как технология виртуализации в чистом виде, но реализованная полностью программно.

Dynamic Tracing - технология динамической трассировки задач, которая позволяет оперативно производить диагностику и настраивать оптимальным образом работающие процессы и систему в целом в целях достижения максимальной производительности. Dynamic Tracing предоставляет достаточно богатые возможности. Во-первых, можно производить анализ взаимодействия пользовательских процессов в системе и находить возможные нестабильные либо неоптимальные места в их работе. Кроме того, отслеживаются проблемы производительности на большинстве программных уровней, а система настраивается на максимальную производительность с учетом текущей квоты ресурсов. Кроме того, можно определять причины сбоя в функционировании процессов и использовать собственные сценарии для проведения индивидуального анализа работы ОС и процессов. В некоторых случаях применение Dynamic Tracing дает возможность повысить скорость выполнения приложений до 30 раз.

Solaris ZFS (Zettabyte File System) - выглядит даже не как файловая система, а скорее как файловая среда, делающая совершенно прозрачной грань между локальной файловой системой и огромным корпоративным data-центром.

В ZFS отсутствует фрагментация данных, что позволяет динамически выделять или освобождать дисковое пространство логической файловой системе, причем это пространство может быть как в рамках физического накопителя, так и рассредоточено по нескольким сетевым дискам. Кроме того, имеет место переменный размер блока, что лучшим образом влияет на производительность, параллельность выполнения операций чтения-записи, а также 64-разрядный механизм использования контрольных сумм, сводящий к минимуму вероятность разрушения данных. Самое интересное — ZFS является 128-разрядной

файловой системой. Sun полагает, что такого задела хватит на обозримые двадцать лет.

Predictive Self-Healing - встроенный комплекс самодиагностики. Он служит для автоматического перезапуска приложений и сервисов, в которых были обнаружены нарушения работы, причем весь процесс занимает считанные миллисекунды.

Solaris Process Rights Management – технология управления правами на процессы. За счет ее применения появилась возможность определять расширенные права не только системным пользователям, но и процессам. Если ранее для выделения процессу безусловных полномочий требовалось присваивать ему статус суперпользователя (root), сейчас же в Solaris 10 можно ограничить права доступа для конкретного процесса или группы процессов только теми привилегиями, которые действительно необходимы. Даже в случае несанкционированного проникновения в систему и захвата процесса злоумышленник не сможет получить прав пользователя root.

Role Based Access Control (RBAC) - интегрированная система ролевого доступа, при которой пользователей можно наделять определенными правами. Она заключается в том, что, если какой-либо человек имеет полный доступ ко всем ресурсам системы, это чревато определенными рисками. RBAC контролирует разделение полномочий между пользователями. К примеру, одному из них предоставляется только право чтения системных журналов (скажем, для проведения аудита), другому — право добавлять новых пользователей, третьему — право подключать и конфигурировать внешние устройства.

Совместное использование Process Rights Management и Role Based Access Control позволяет существенно повысить надежность и безопасность ОС.

Solaris Security Toolkit (также известный как JASS — JumpStart Architecture and Security Scripts) - это набор скриптов и утилит, которые позволяют администратору автоматизировать процессы распределения прав доступа к сервисам ОС, контролировать и устанавливать безопасные атрибуты файлов.

4.2.3 Интерфейс пользователя

Интерфейс пользователя – это главное средство взаимодействия операционной системы и человека. На сегодняшний день графический интерфейс пользователя приобретает все большую популярность в современных операционных системах, так как в любом своем виде он базируется на одних и тех же принципах и соответственно понятен современному пользователю. Однако, в системах семейства UNIX традиционно принят текстовый интерфейс. Это сделано, прежде всего, из-за их ориентированности. Серверную систему достаточно часто приходится настраивать и обслуживать удаленно, а удаленный доступ с графическим интерфейсом требует высококачественных линий связи за счет большого объема передаваемой информации. Именно поэтому все настройки систем семейства UNIX приоритетно производятся из командной строки. ОС Solaris 10 имеет как традиционный для UNIX систем интерфейс командной строки (текстовый) так и графический представленный двумя оболочками.

4.2.4 Командная строка

Командная строка представляет собой область экрана, в которой пользователь вводит с клавиатуры различные команды, а система на них отвечает в текстовом режиме. По умолчанию в ОС Solaris запускается интерпретатор команд «korn», но его можно сменить на более дружелюбный и привычный пользователям Linux «bash» введя в командной строке его имя.

Базовый синтаксис команды UNIX включает: *команда параметр(ы) аргумент(ы)*

- Команда – выполняемая команда (указывает, что делать системе)
- Параметр – Изменяет команду (указывают системе, как выполнять команду). Параметры начинаются с символа дефиса (-).
- Аргумент – Имя файла, каталога или просто текст

Подробное описание любой команды можно получить с помощью команды: **man команда**.

Простейшее взаимодействие пользователя и ОС можно проиллюстрировать командой «netstat -rn», на которую система ответит отображением на экран таблицы маршрутизации (рис.4.1).

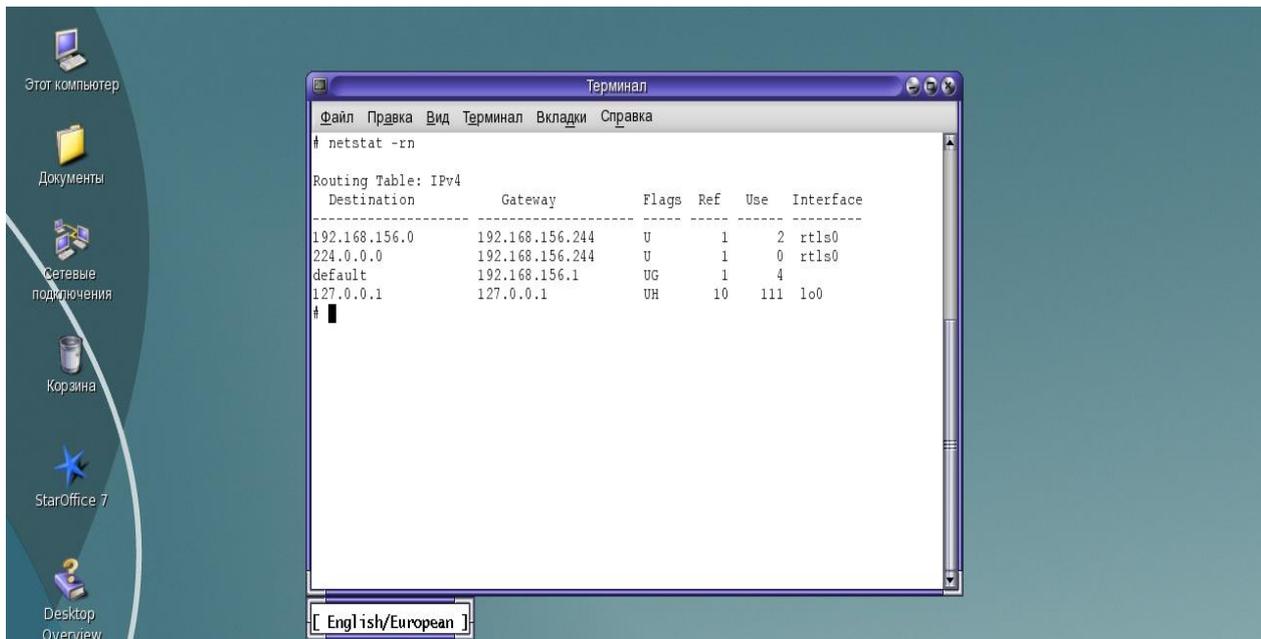


Рис.4.1. Демонстрация взаимодействия пользователя и ОС

4.2.5. Графический интерфейс

Изюминкой в Solaris 10 является новая графическая оболочка Java Desktop System 3, которая фактически представляет из себя Gnome 2.6. Она достаточно приятна в обращении и консолидирует в себе все современные представления о графическом интерфейсе. Вид основных меню и окон показан на рис.4.2.

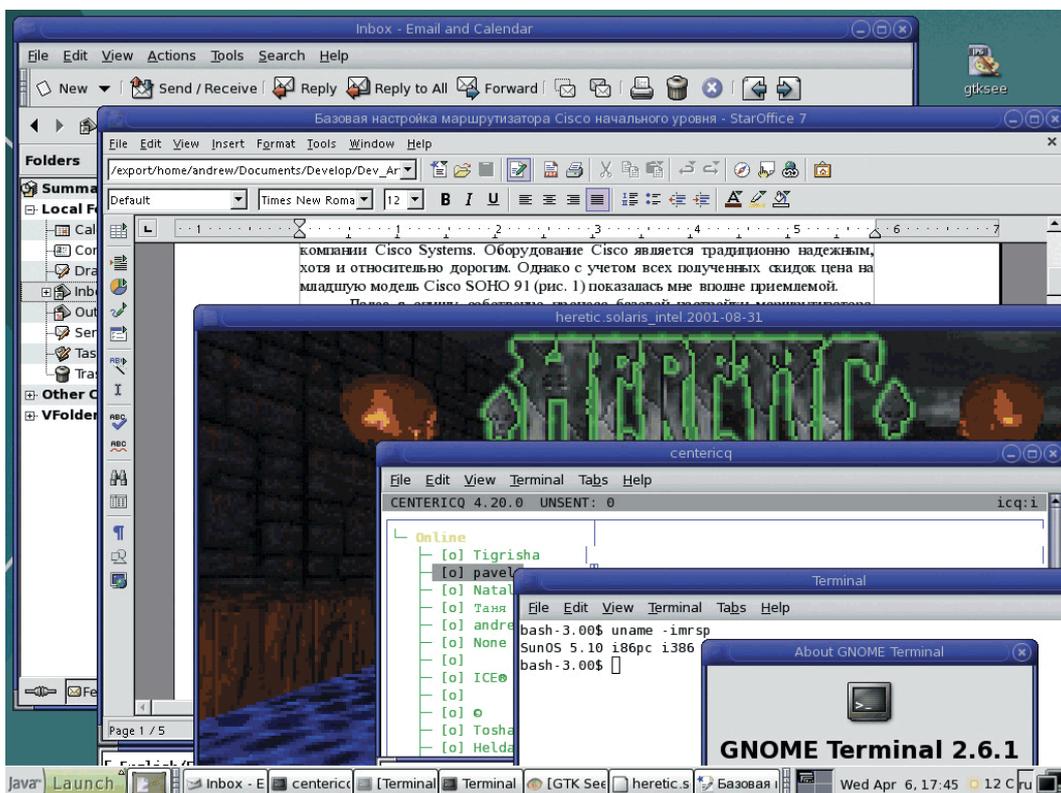


Рис.4.2. Вид основных окон и меню

С помощью графического интерфейса можно производить некоторые настройки системы, однако они по большей части касаются настройки самого графического интерфейса.

Более детальную и глубокую настройку системы, возможно осуществлять используя текстовые команды. При наличии физического доступа к серверу для более комфортабельной работы администратор имеет возможность воспользоваться графическим интерфейсом и уже внутри него для осуществления функций администрирования запустить терминал командной строки как показано на рис.4.3.

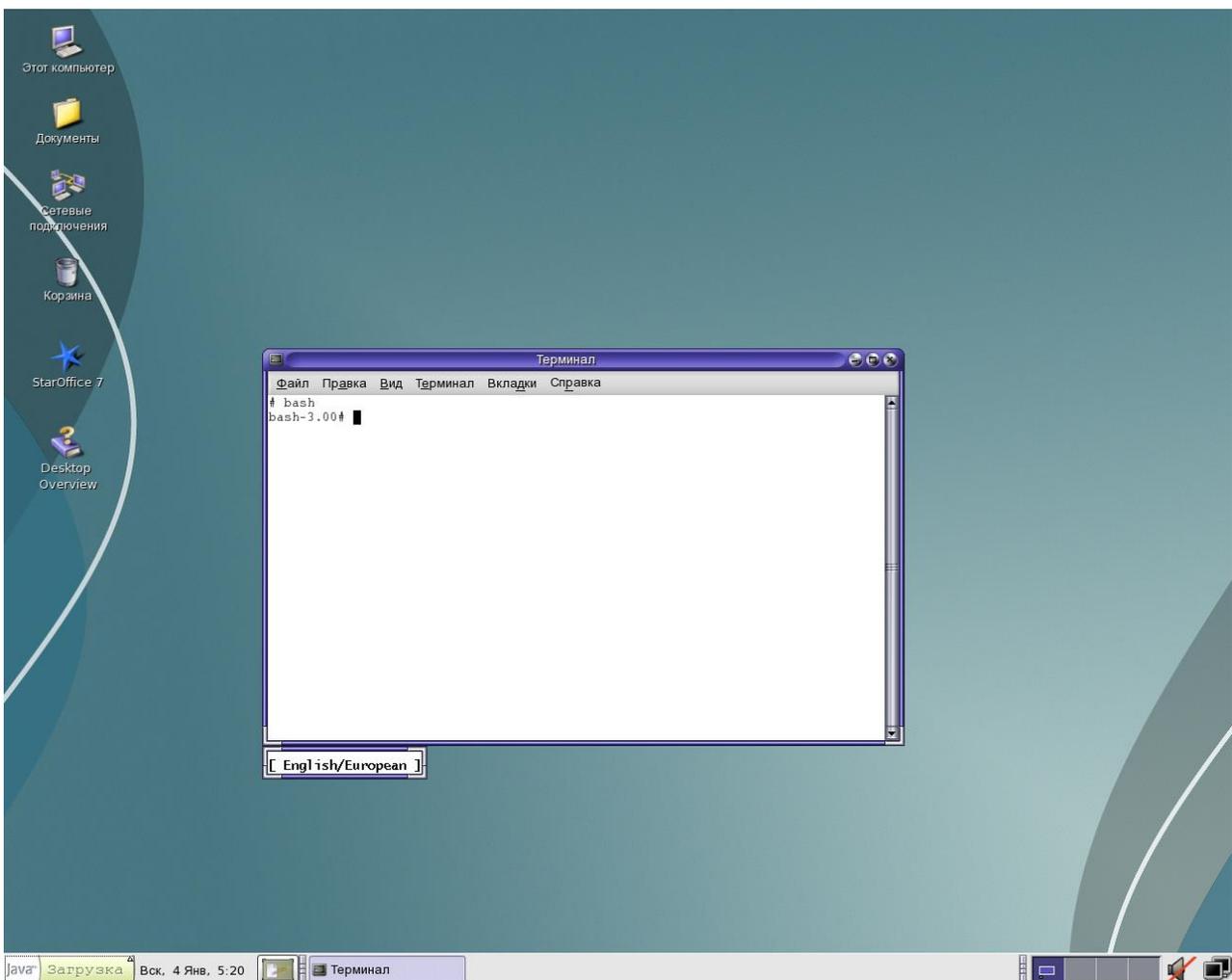


Рис.4.3. Терминал командной строки

Второй оболочкой является CDE (Common Desktop Environment) версии 1.6, она знакома пользователям предыдущих версий Solaris, не смотря на то, что она не так удобна и привычна как JDS 3, разработчики сохраняют ее поддержку. Оценить отличия можно с первого взгляда (рис.4.4).

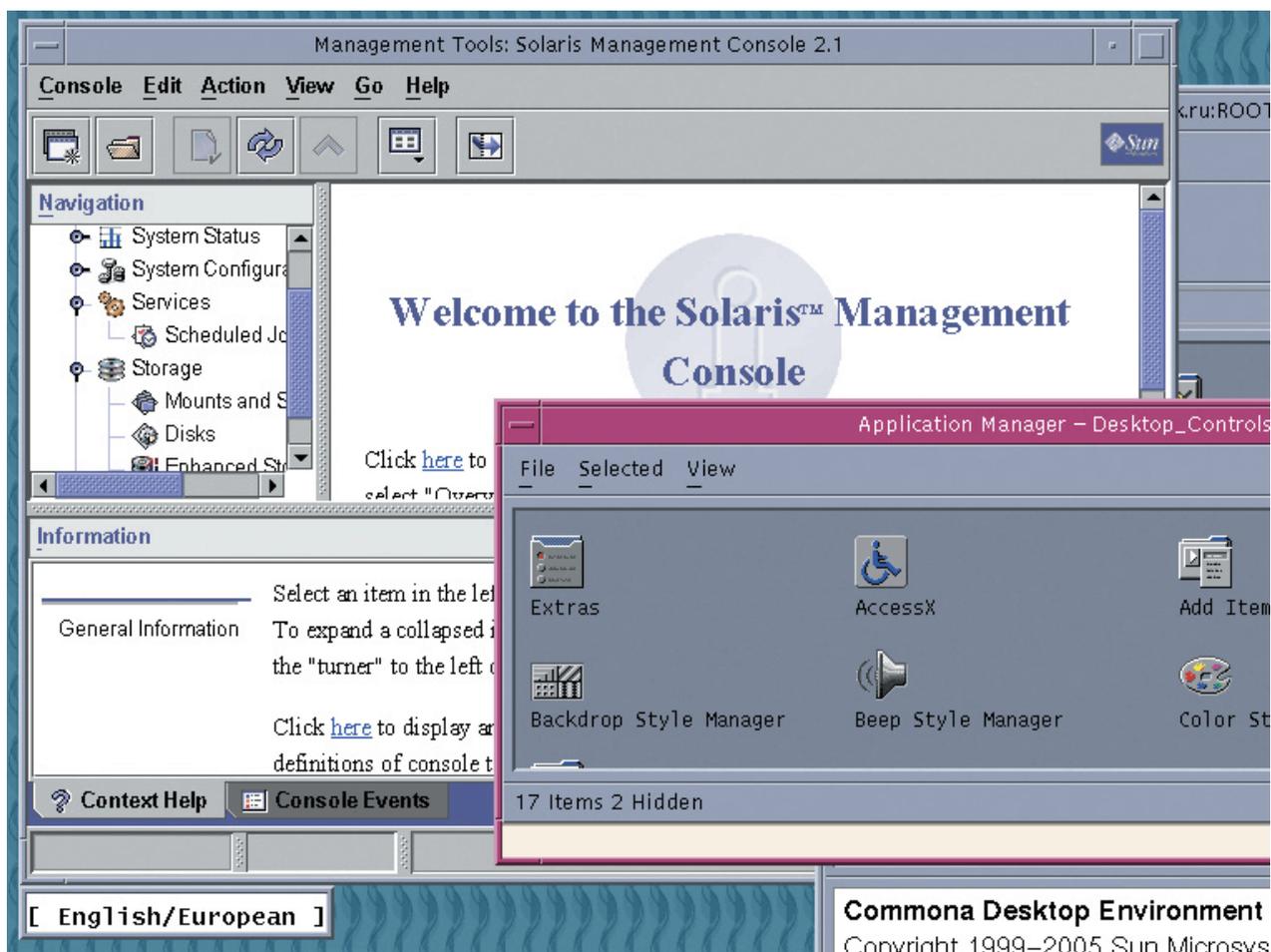


Рис.4.4. Внешний вид CDE

Да и внутри данной оболочки применяются нестандартные правила навигации по интерфейсу.

Основной сетевой подсистемы Solaris является сетевой интерфейс. Сетевой интерфейс – это абстракция, используемая для представления связи канального уровня сети с протоколом TCP/IP в Solaris.

Каждый сетевой интерфейс в системе имеет уникальное имя, состоящее из типа устройства и номера (0 или больше для однотипных устройств). Под типами устройств в Solaris понимается название драйвера устройства (Realtek – rtls). Интерфейс имеет набор параметров, большинство которых относятся к сетевому уровню (IP-адрес, маска сети и т.п.). Важным параметром сетевого интерфейса является аппаратный адрес (В Ethernet аппаратный адрес называется MAC-адрес и состоит из шести байтов, которые принято записывать в шестнадцатеричной системе исчисления и разделять двоеточиями). Создание сетевого интерфейса производится в момент загрузки драйвера сетевой карты или создания логического соединения (например, в PPP).

В Solaris всегда присутствует сетевой интерфейс lo, – так называемая заглушка (loopback), которая используется для организации сетевых взаимодействий компьютера с самим собой: любой посланный в заглушку пакет немедленно обрабатывается как принятый оттуда. Заглушка обычно имеет адрес 127.0.0.1.

Настройка сетевых параметров, связанных с интерфейсом производится командой `ifconfig`. Команда `ifconfig` имеет следующий синтаксис:

```
ifconfig [-L] [-m] interface [create] [address_family] [address
[dest_address]] [parameters] ifconfig interface destroy ifconfig -a [-L] [-d] [-m]
[-u] [address_family] ifconfig -l [-d] [-u] [address_family] ifconfig [-L] [-d] [-m]
[-u] [-C]
```

Основные опции команды:

- **интерфейс** – имя интерфейса (например, `rl0` в BSD или `eth0` в Linux).
- **Up** – вызывает активизацию интерфейса. Задается неявно при присвоении адреса интерфейсу.
- **down** – вызывает остановку работы драйвера для интерфейса.
- **[-]arp** – включает или отключает использование протокола ARP для интерфейса.
- **[-]promis** – включает или отключает неразборчивый режим (promiscuous mode) работы интерфейса. В этом режиме все проходящие по сети пакеты будут приниматься интерфейсом.
- **metric N** – устанавливает метрику интерфейса.
- **mtu N** – устанавливает максимальный размер пакета (Maximum Transfer Unit - MTU) для интерфейса.
- **inet адрес** – IP-адрес, присваиваемый интерфейсу.
- **netmask адрес** – устанавливает маску сети IP для этого интерфейса. По умолчанию используется обычная маска сети класса А, В или С (что определяется по IP-адресу интерфейса), но можно усановить любое значение.

Более подробное описание команды можно получить из встроенной в систему документации с помощью команды «`man ifconfig`».

4.2.6. Пример настройки сетевого интерфейса

Рассмотрим пошагово действия, которые необходимо осуществить для конфигурирования сетевого интерфейса и создания его виртуальных(логических) интерфейсов. Для выполнения этих действий необходимо использовать команду `ifconfig`.

Шаг 1. Включите компьютер с установленной на него ОС Solaris 10. После загрузки компьютера на экране появится окно приглашения пользователя в систему рисунок 4.5, в котором он должен выбрать графическую среду и ввести данные для авторизации.

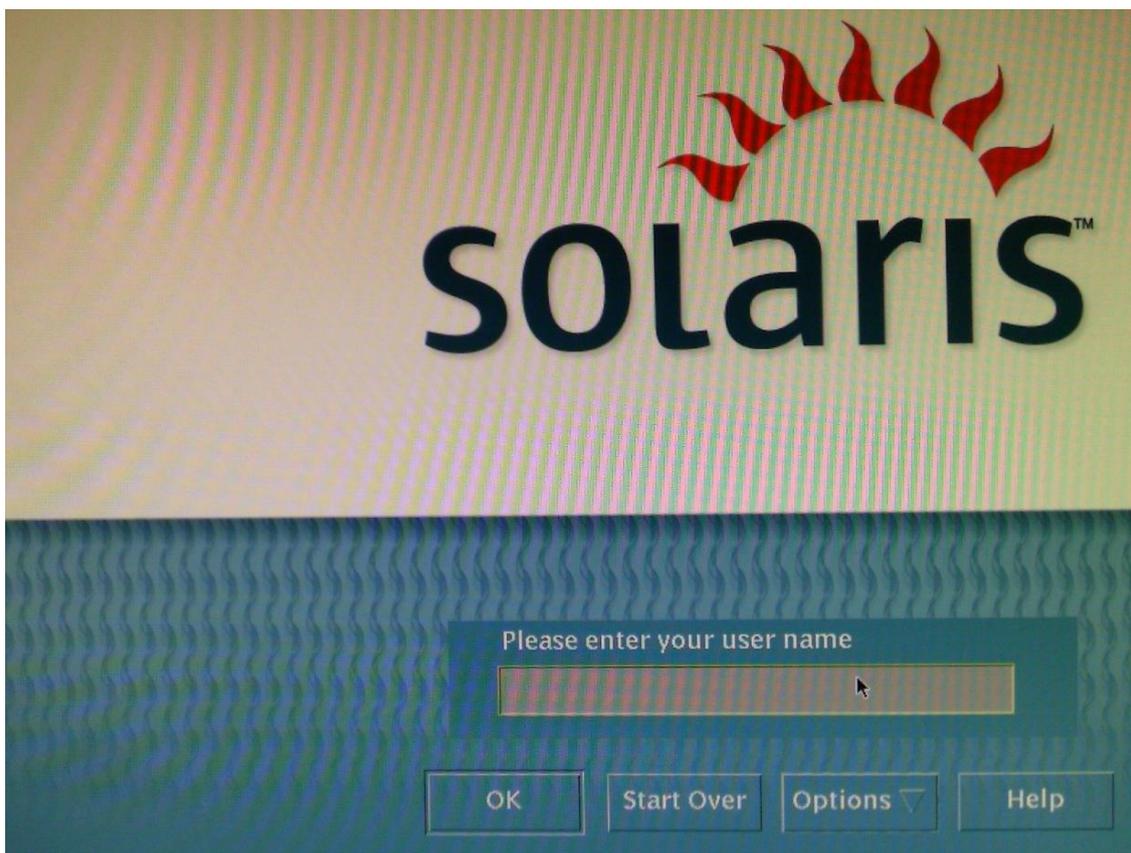


Рис. 4.5 Окно приглашения пользователя в систему

Шаг 2. С помощью кнопки «Options» в окне приглашения пользователя в систему, выберите подменю «Session», а в нем пункт «Java Desktop System Release 3». Для выполнения настройки системы требуются права суперпользователя, поэтому введите логин «root», нажмите Enter, а затем введите пароль и опять нажмите на клавишу «Enter». После выполнения этих действий будет произведена загрузка графического интерфейса и на экране появится рабочий стол, как показано на рис.4.6.

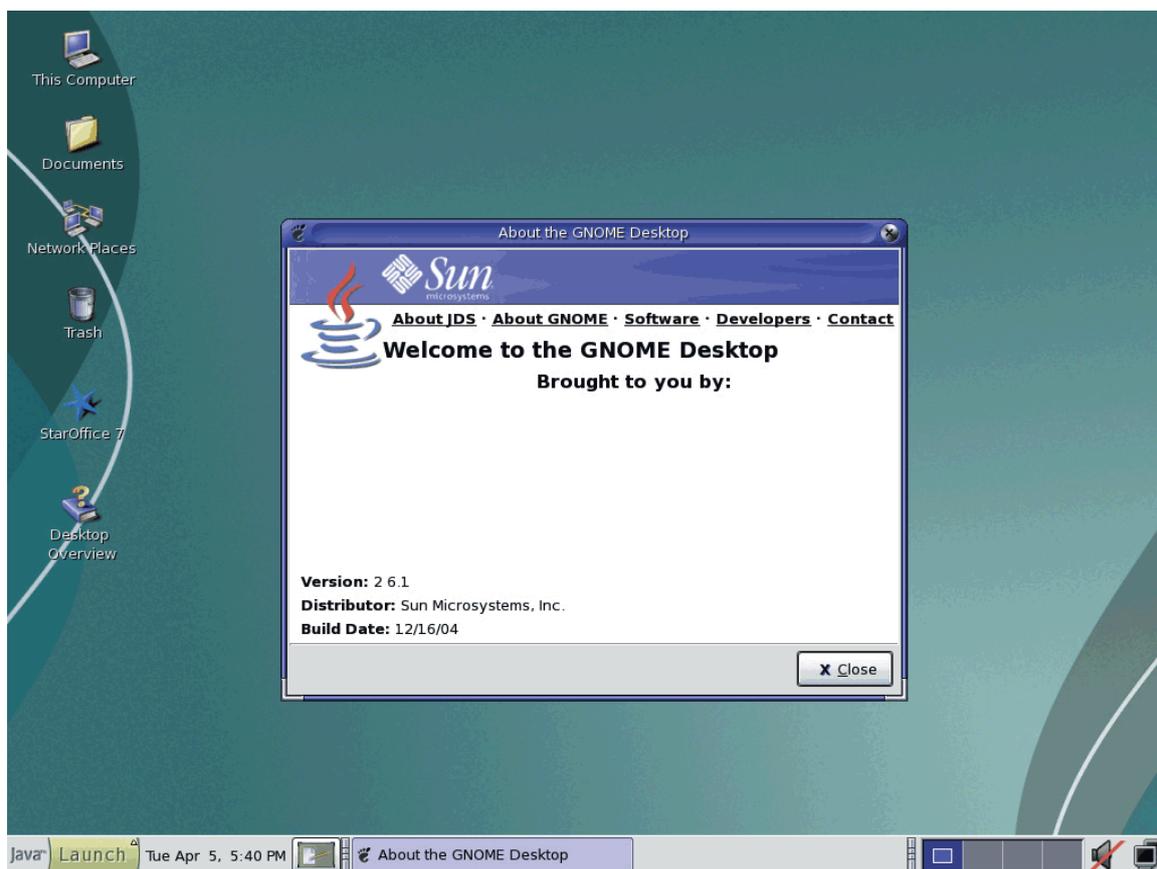


Рис.4.6. Рабочий стол интерфейса Java Desktop System Release 3

Шаг 3. Для конфигурирования системы необходимо запустить терминал командной строки, для этого нажмите мышью на значок «Java Загрузка» в левом нижнем углу экрана, в открывшемся меню выберите «Приложения->Utilities-> Терминал» (рис.4.7).

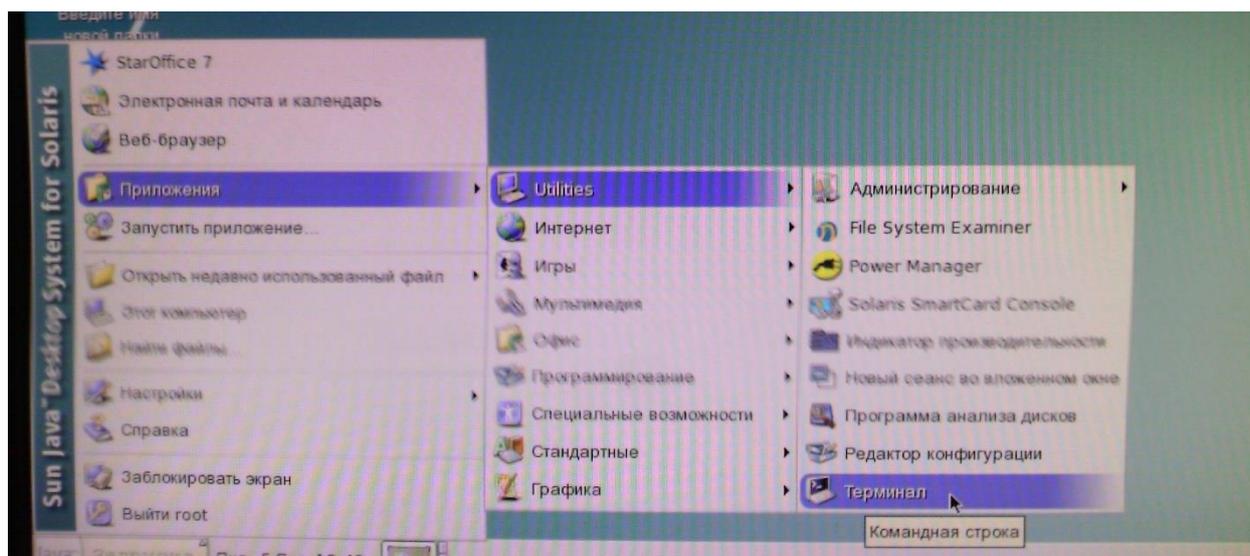


Рис.4.7. Запуск терминала командной строки

Шаг 4. По умолчанию в Solaris 10 запускается командный интерпретатор «korn». Настройку можно производить и в нем, но

рекомендуется запустить более дружелюбный - «bash». Для этого в окне командной строки необходимо набрать команду «bash» и нажать клавишу Enter как показано на рис.4.8.

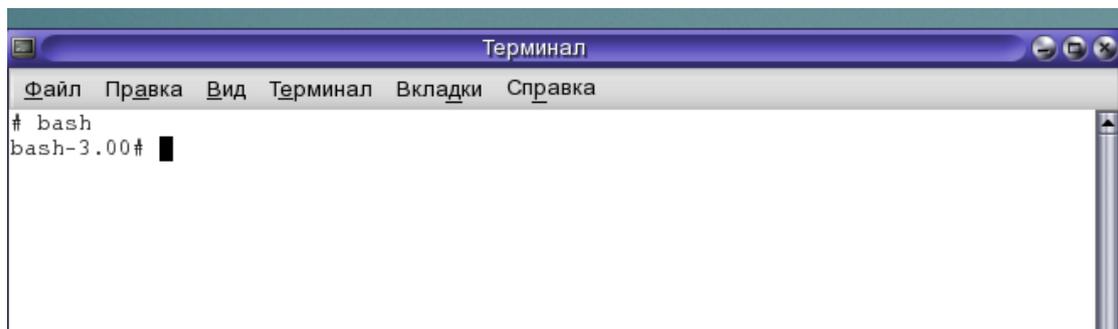


Рис.4.8. Переход к командному интерпретатору bash

Шаг 5. Теперь можно приступить к непосредственной настройке сетевого интерфейса, для этого необходимо узнать какие интерфейсы и с какими именами существуют в системе. Воспользуемся командой «ifconfig» с ключом «-а», в ответ на нее система отобразит список всех интерфейсов системы в том числе и неактивных (рис.4.9).

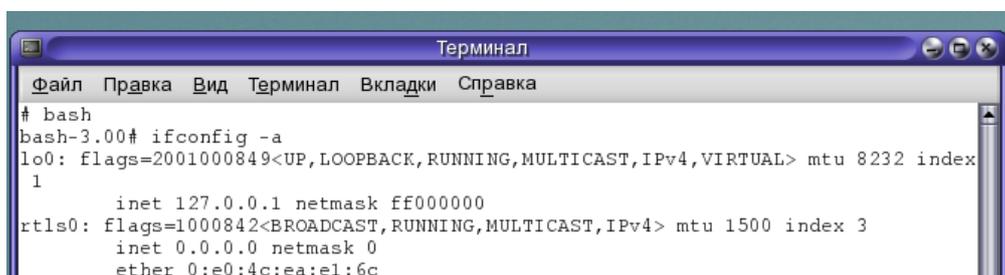
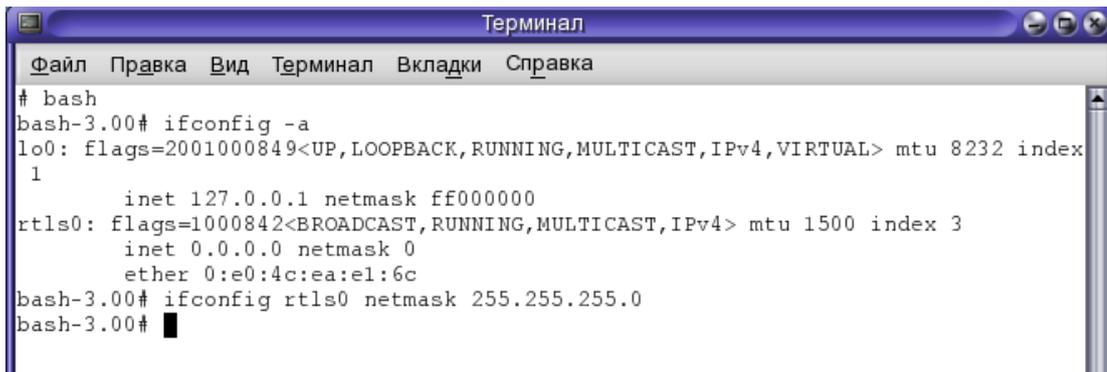


Рис.4.9. Просмотр информации о сетевых интерфейсах

Шаг 6. Теперь вы знаете, что в системе существует два сетевых интерфейса, один логический (lo0, является стандартным для Unix систем) и один физический (rtls0, в данном случае он представлен физическим Ethernet адаптером). Настраивать будем физический интерфейс rtls0. Для этого зададим маску подсети 255.255.255.0, сделать это можно с помощью команды «ifconfig rtls0 netmask 255.255.255.0», введите команду в командную строку и нажмите клавишу Enter как показано на рис, 4.10.



```

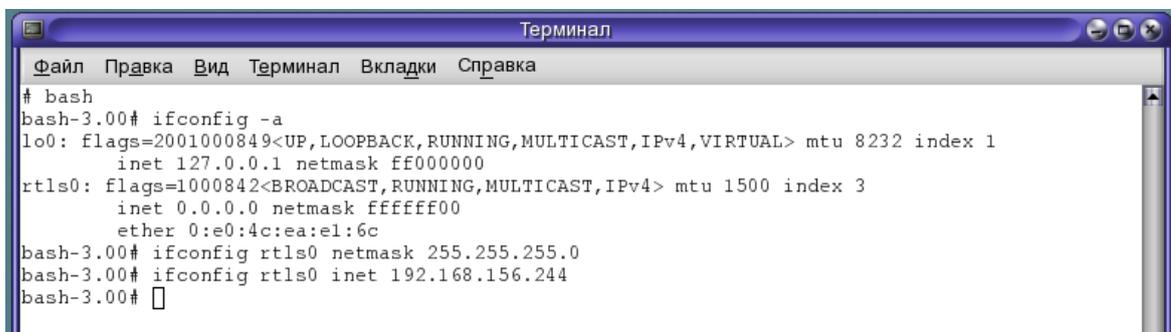
Терминал
Файл Правка Вид Терминал Вкладки Справка
# bash
bash-3.00# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index
  1
    inet 127.0.0.1 netmask ff000000
rtls0: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 0.0.0.0 netmask 0
    ether 0:e0:4c:ea:e1:6c
bash-3.00# ifconfig rtls0 netmask 255.255.255.0
bash-3.00# █

```

Рис.4.10. Задание маски подсети

Шаг 7. После успешного задания маски подсети, зададим IP-адрес для данного сетевого интерфейса (например, 192.168.156.244), для этого воспользуйтесь командой «ifconfig rtls0 inet 192.168.156.244», введите ее в командную строку и нажмите Enter как показано на рис.4.11.

Шаг 8. Для того чтобы убедиться, что параметры заданы правильно используйте команду «ifconfig rtls0», в ответ на эту команду интерпретатор выведет информацию о параметрах конфигурации сетевого интерфейса как показано на рис.4.12.

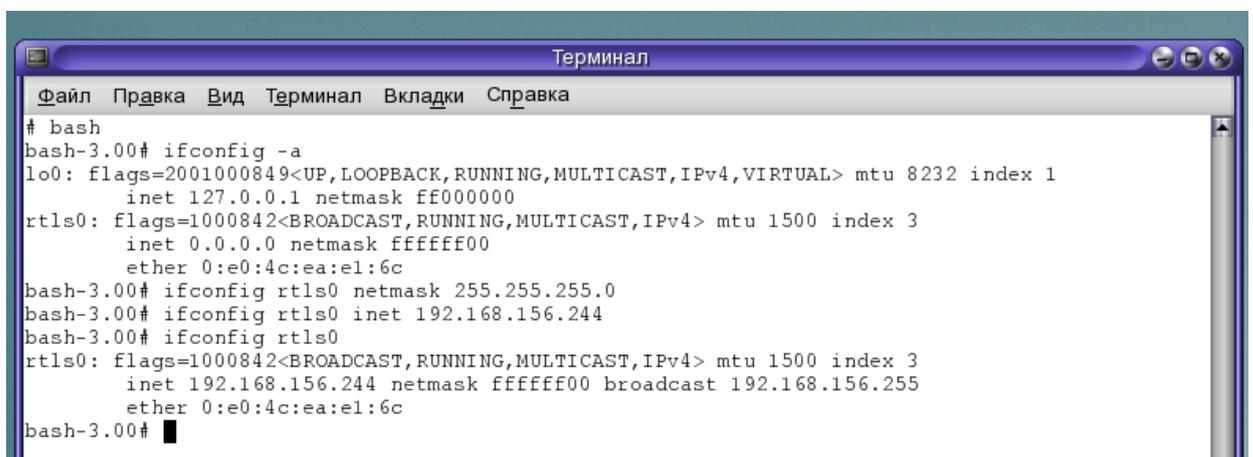


```

Терминал
Файл Правка Вид Терминал Вкладки Справка
# bash
bash-3.00# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
rtls0: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 0.0.0.0 netmask ffffffff
    ether 0:e0:4c:ea:e1:6c
bash-3.00# ifconfig rtls0 netmask 255.255.255.0
bash-3.00# ifconfig rtls0 inet 192.168.156.244
bash-3.00# █

```

Рис.4.11. Задание IP адреса



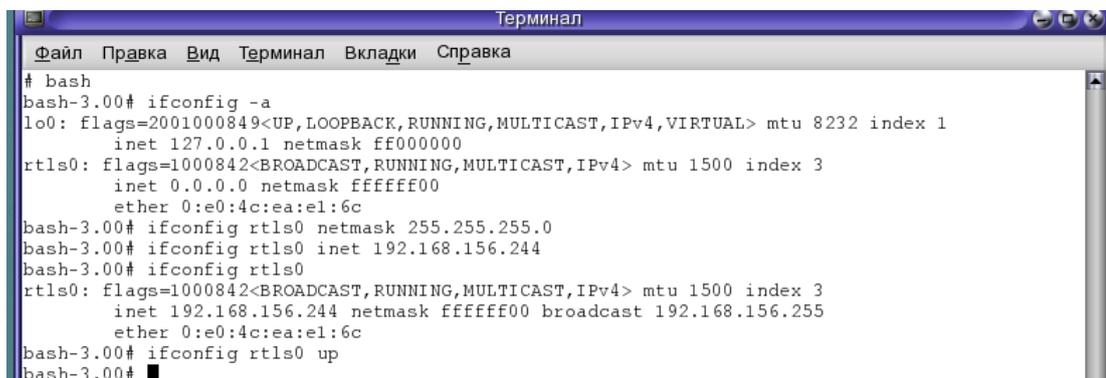
```

Терминал
Файл Правка Вид Терминал Вкладки Справка
# bash
bash-3.00# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
rtls0: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 0.0.0.0 netmask ffffffff
    ether 0:e0:4c:ea:e1:6c
bash-3.00# ifconfig rtls0 netmask 255.255.255.0
bash-3.00# ifconfig rtls0 inet 192.168.156.244
bash-3.00# ifconfig rtls0
rtls0: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 192.168.156.244 netmask ffffffff broadcast 192.168.156.255
    ether 0:e0:4c:ea:e1:6c
bash-3.00# █

```

Рис.4.12. Параметры конфигурации сетевого интерфейса

Шаг 9. После выполнения всех выше указанных действий необходимо произвести инициализацию интерфейса с помощью команды «ifconfig rtls0 up» как показано на рис.4.13.



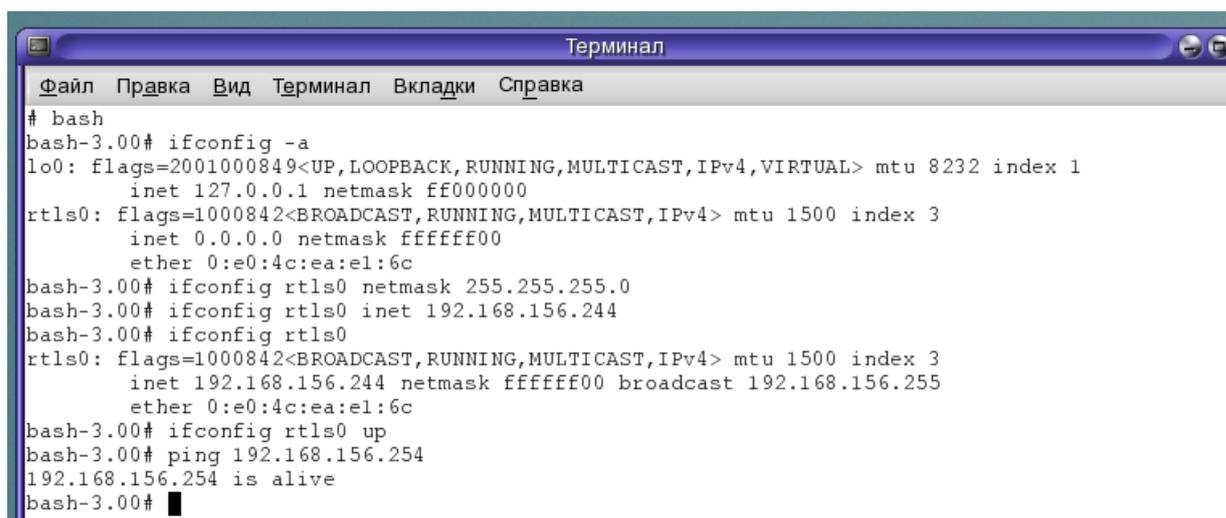
```

Терминал
Файл Правка Вид Терминал Вкладки Справка
# bash
bash-3.00# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
rtls0: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 0.0.0.0 netmask ffffffff0
    ether 0:e0:4c:ea:e1:6c
bash-3.00# ifconfig rtls0 netmask 255.255.255.0
bash-3.00# ifconfig rtls0 inet 192.168.156.244
bash-3.00# ifconfig rtls0
rtls0: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 192.168.156.244 netmask ffffffff0 broadcast 192.168.156.255
    ether 0:e0:4c:ea:e1:6c
bash-3.00# ifconfig rtls0 up
bash-3.00#

```

Рис.4.13. Инициализация сетевого интерфейса

Шаг 10. Проведем проверку работоспособности сети, попыткой обменяются пакетами с основным шлюзом сети с помощью команды «ping 192.168.156.254», в случае успешного обмена пакетами система ответит сообщением как показано на рис.4.14.



```

Терминал
Файл Правка Вид Терминал Вкладки Справка
# bash
bash-3.00# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
rtls0: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 0.0.0.0 netmask ffffffff0
    ether 0:e0:4c:ea:e1:6c
bash-3.00# ifconfig rtls0 netmask 255.255.255.0
bash-3.00# ifconfig rtls0 inet 192.168.156.244
bash-3.00# ifconfig rtls0
rtls0: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 192.168.156.244 netmask ffffffff0 broadcast 192.168.156.255
    ether 0:e0:4c:ea:e1:6c
bash-3.00# ifconfig rtls0 up
bash-3.00# ping 192.168.156.254
192.168.156.254 is alive
bash-3.00#

```

Рис.4.14. Обмен пакетами с основным шлюзом

Если ОС ответила сообщением ip_адрес_хоста is alive, значит интерфейс сконфигурирован правильно.

4.3. Задание для индивидуальной работы

1. Создать дополнительный логический интерфейс и присвоить ему IP адрес.
2. Изменить адрес основного шлюза путем редактирования таблицы маршрутизации.

3. Задать адрес DNS сервера, имя хоста и домен.

4.4. Содержание отчета

1. Цель работы.
2. Результаты выполнения специального задания.
3. Выводы по проделанной работе.

4.5. Контрольные вопросы

1. Опишите основы ip-адресации.
2. В чем заключаются отличия логического и физического сетевого интерфейса?
3. С помощью, какой команды можно сделать сетевой интерфейс временно неактивным?
4. Как удалить сетевой интерфейс и все связанные с ним файлы конфигурации?
5. Как определить работает ли сеть?
6. Опишите предоставляемые пользовательские интерфейсы.

Часть 2 - «Использование политики ролевого доступа Role Based Access Control»

4.6. Цель работы

получение первичных навыков создания новых пользователей;

получение общего представления о концепции Role Base Access Control (RBAC);

получение навыков гибкого контроля безопасности в системе на основе умения ассоциировать пользователя с ролями, а роли с профилями.

4.7. Теоретическая часть

На протяжении долгого времени безопасность в дистрибутивах Unix основывалась на двух параметрах: идентификаторах (IDs) и правах доступа (permissions). Каждый файл имеет UID (идентификатор пользователя) и GID (идентификатор группы), определяющие владельца

файла, а также набор атрибутов, определяющих права доступа пользователя-владельца, группы и всего остального мира. Права определенного пользователя на чтение, запись или исполнение файла зависят от его идентификаторов и прав доступа, хранящихся в описателе файла - inode.

На определенном этапе развития в Unix появились Списки Контроля Доступа (Access Control Lists - ACL), но они также основывались на идентификаторах и правах доступа. Пользователь, захотевший выполнить специфическую задачу, требующую прав суперпользователя, вначале должен получить идентификатор суперпользователя. Однако это позволит пользователю выполнить не только его специфическую задачу, а любую задачу. В соответствии со старой моделью пользователь не сможет выполнять задач администрирования, не имея UID 0 (root). Получив этот идентификатор, он сможет выполнять все администраторские задачи.

Role Based Access Control (RBAC) - интегрированная система ролевого доступа, при которой пользователей можно наделять определенными правами. Она контролирует разделение полномочий между пользователями. К примеру, одному из них предоставляется только право чтения системных журналов (скажем, для проведения аудита), другому — право добавлять новых пользователей, третьему — право подключать и конфигурировать внешние устройства.

У пользователя есть два идентификатора: реальный (Real UID) и эффективный (Effective UID). При входе в систему оба идентификатора идентичны и берутся из файла passwd. При запуске пользователем процесса, эффективный UID может отличаться от реального UID. Для изменения эффективного идентификатора пользователь обычно использует команду su. Когда пользователь с UID равным, к примеру, 2001 запускает su, у него будет запрошен пароль суперпользователя и после этого создан новый процесс. Этот процесс имеет реальный UID 2001 и эффективный UID 0, что означает, что на протяжении времени жизни нового процесса все команды пользователя будут выполняться с UID равным 0. Пользователь должен знать пароль другого пользователя, под идентификатор которого он хочет переключиться. root — единственный пользователь, который может переключиться под любой UID без знания пароля этого пользователя. Необходимость знания пароля

другого пользователя – главный недостаток в использовании su.

RBAC комбинирует пользователей, роли, профили, права и параметры запуска (Exec Attributes). Хорошей практикой является присоединение нескольких профилей к одной роли, позволяя пользователю выполнять различные задачи администрирования, всего лишь изменив свою роль. Так же как и при использовании su для смены идентификатора, пользователь должен знать пароль доступа к желаемой роли, однако в случае с ролями его права будут гораздо более ограничены в соответствии со следующим базисом:

Пользователь -> Роль -> Профиль -> Параметр запуска

Для выполнения практических заданий студент должен знать синтаксис команд:

Useradd

СИНТАКСИС

```
useradd [-A {method|DEFAULT},... ] [-c comment] [-d home_dir] [-e
expire_date] [-f inactive_time] [-g ini_tial_group] [-G group[,...]] [-m [-k
skeleton_dir]] [-s shell] [-u uid [ -o]] login
```

```
useradd -D [-g default_group] [-b default_home] [-f default_inactive] [-e
default_expiration] [-s default_shell]
```

ОПИСАНИЕ

Создание новых пользователей. Когда вызывается без опции -D , команда useradd создает новый бюджет пользователя используя значения определенные в командной строке и по умолчанию значения из системы. Новый бюджет пользователя будет записан в системные файлы, домашний каталог пользователя будет создан ,файлы инициализации скопированы. Все операции будут выполнены в зависимости от опций командной строки.

Основные опции используемые с командой useradd:

-d home_dir

Новый пользователь будет создан с помощью использования значения в качестве домашнего каталога home_dir По умолчанию это значение получается методом добавления login имени к default_home

-e expire_date

Дата после которой бюджет пользователя устареет. Дата определяется следующим форматом ММ/ДД/YY.

-f inactive_days

Число дней между днем когда пароль устарел и днем когда бюджет пользователя будет запрещен. Значение равное 0 запрещает бюджет пользователя ,в момент устаревания пароля, и значение -1 запрещает эту особенность. По умолчанию значение -1.

-g initial_group

Имя группы или номер группы которая будет присвоена пользователю после входа в систему. Группа с данным именем должна существовать. Номер группы должен ссылаться на существующую группу. По умолчанию номер группы 1.

-m

Домашний каталог пользователя будет создан если он не существует. Файлы находящиеся в skeleton_dir будут скопированы в домашний каталог пользователя если используется опция -k , в противном случае будут использоваться файлы находящиеся в каталоге /etc/skel. Любые каталоги находящиеся в каталогах skeleton_dir или /etc/skel будут созданы в домашнем каталоге пользователя. Опция -k доступно только месте с опцией -m. по умолчанию не создается каталога и не копируются файлы.

-s shell

Имя шела , который будет использоваться новым пользователем при входе в систему. Установка этого поля в пустое значение будет выбирать системный шел по умолчанию.

Passwd является текстовым файлом, который содержит список учётных записей в системе, где каждая учётная запись содержит некоторую полезную информацию, такую как идентификатор пользователя (UID), идентификатор группы (GID), домашний каталог, командный интерпретатор и т.д. Часто, в этом же файле для каждой учётной записи содержатся также и зашифрованные пароли. Файл должен быть доступен для чтения всем (многие утилиты, такие как ls(1)

используют его, чтобы отражать идентифкаторы пользователей на их имена), но доступ на запись должен предоставляться только суперпользователю. Если вы создаёте новую учётную запись, с начало в поле пароля помещается звёздочка, а затем с помощью команды `passwd` вы можете задать пароль. Каждая строка файла содержит только одну учётную запись в формате:

```
account:password:UID:GID:GECOS:directory:shell
```

Вот описание этих полей:

- **account** - имя пользователя в системе. Оно не должно содержать денежных символов.
- **password** - зашифрованный пароль пользователя или звёздочка. (В случае использования теневых паролей, если пароль задан, то в поле пароля ставится символ `x', а не звёздочка)
- **UID** - числовой идентификатор пользователя (UID).
- **GID** - числовой идентификатор первичной группы (GID) для этого пользователя.
- **GECOS** - данное поле является необязательным и используется только для информационных целей. Обычно, оно содержит полное имя пользователя.
- **directory** - домашний каталог пользователя \$HOME.
- **shell** - это программа, которая запускается после входа в систему (если это поле пустое, то используется /bin/sh).

Grep - выполняет поиск образца в текстовых файлах и выдает все строки, содержащие этот образец. Она использует компактный недетерминированный алгоритм сопоставления. Если имя_файла не указано, грег предполагает поиск в стандартном входном потоке.

На самостоятельное изучение отдаются следующие команды:

Roleadd – данная команда добавляет новую роль в систему. Более подробное описание команды смотрите в справочных руководствах `man`.

Rolemod – команда предназначена для модификации созданных ролей

Usermod – предназначена для модификации существующих пользователей

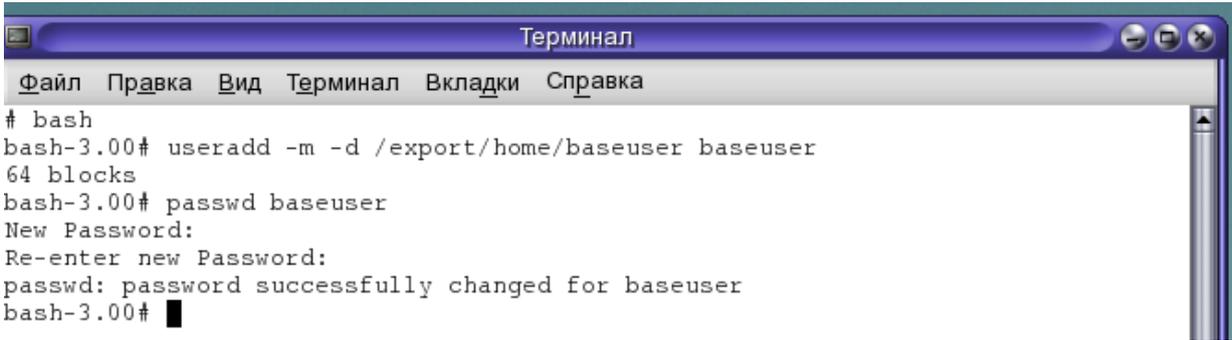
Более подробное описание каждой команды можно получить из встроенных справочных руководств, путем вызова команды «man имя_команды».

4.8. Порядок выполнения работы

В качестве примера использования RBAC рассмотрим создание нового пользователя, обладающего правами только на перезагрузку системы, и не на что более. Рассмотрим по шагам действия, которые необходимо осуществить для создания такого пользователя.

Шаг 1. Выполните шаги 1-4 из предыдущей лабораторной работы.

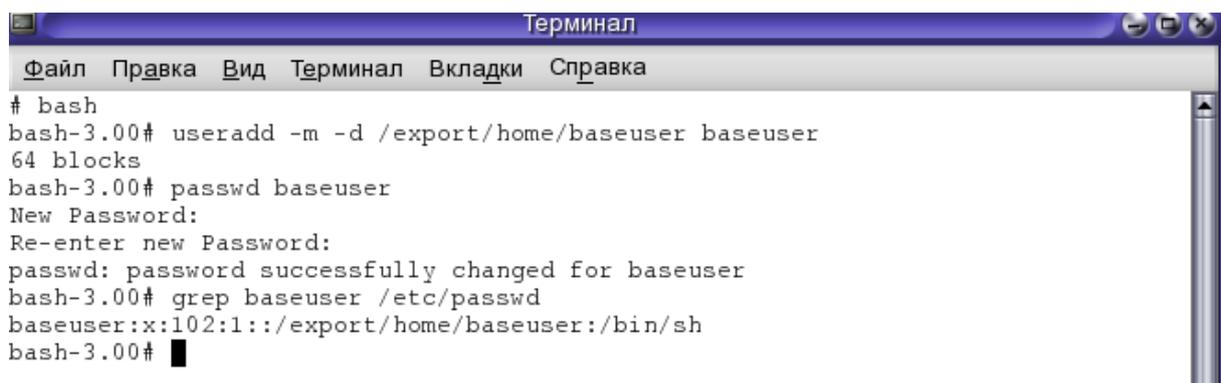
Шаг 2. Для начала добавим нового пользователя с именем «baseuser», с помощью команды «useradd -m -d /export/home/baseuser baseuser» и зададим ему новый пароль (к примеру: 123), с помощью команды «passwd baseuser» как показано на рис.4.15.



```
Терминал
Файл Правка Вид Терминал Вкладки Справка
# bash
bash-3.00# useradd -m -d /export/home/baseuser baseuser
64 blocks
bash-3.00# passwd baseuser
New Password:
Re-enter new Password:
passwd: password successfully changed for baseuser
bash-3.00#
```

Рис.4.15. Добавление нового пользователя

Шаг 3. Теперь необходимо удостовериться в создании пользователя и задании ему пароля путем поиска строки с именем пользователя «baseuser» в файле «/etc/passwd». Сделать это можно с помощью команды «grep baseuser /etc/passwd», как показано на рис.4.16. Если пользователь создан то строка будет найдена и командный интерпретатор выдаст строку следующего формата «baseuser:x:102:1::/export/home/baseuser:/bin/sh».



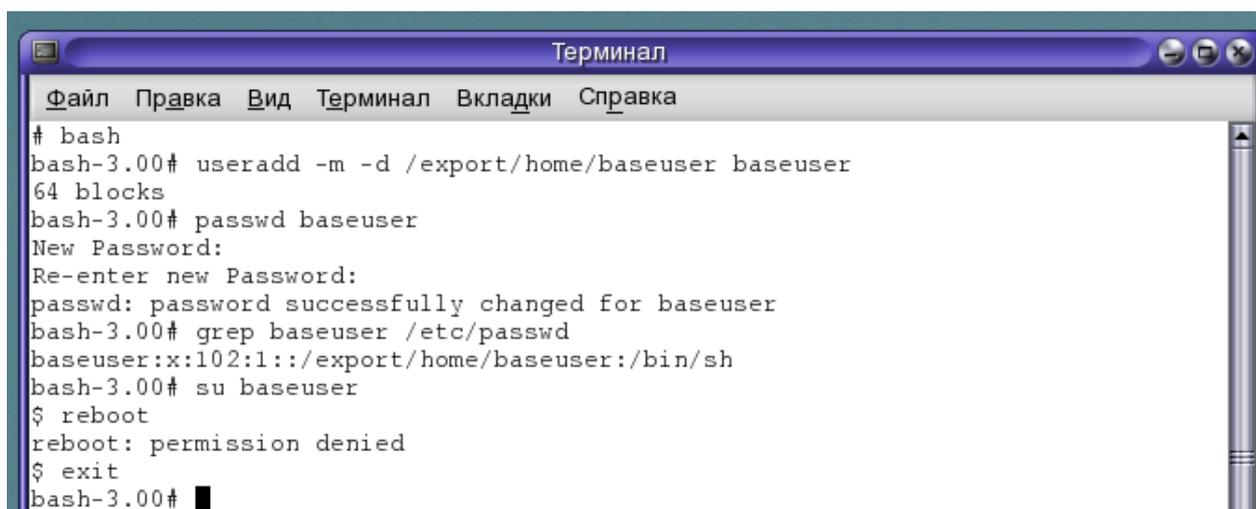
```

Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
# bash
bash-3.00# useradd -m -d /export/home/baseuser baseuser
64 blocks
bash-3.00# passwd baseuser
New Password:
Re-enter new Password:
passwd: password successfully changed for baseuser
bash-3.00# grep baseuser /etc/passwd
baseuser:x:102:1::/export/home/baseuser:/bin/sh
bash-3.00# █

```

Рис.4.16 Проверка создания пользователя

Шаг 4. Проверим, действительно ли у пользователя «baseuser» нет прав на перезагрузку системы. Для этого войдем в систему как пользователь «baseuser», используя команду «su baseuser» и попробуем выполнить команду «reboot», система должна ответить отказом «reboot: permission denied». Для возврата к правам пользователя «root» выполните команду «exit», как показано на рис.4.17.



```

Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
# bash
bash-3.00# useradd -m -d /export/home/baseuser baseuser
64 blocks
bash-3.00# passwd baseuser
New Password:
Re-enter new Password:
passwd: password successfully changed for baseuser
bash-3.00# grep baseuser /etc/passwd
baseuser:x:102:1::/export/home/baseuser:/bin/sh
bash-3.00# su baseuser
$ reboot
reboot: permission denied
$ exit
bash-3.00# █

```

Рис. 4.17 Проверка прав baseuser

Шаг 5. Далее создадим в системе роль с именем «reboot», с помощью команды «roleadd -m -d /export/home/reboot reboot» и зададим пароль, например, «321» для роли используя команду «passwd reboot», как показано на рис.4.18.

```

Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
# bash
bash-3.00# useradd -m -d /export/home/baseuser baseuser
64 blocks
bash-3.00# passwd baseuser
New Password:
Re-enter new Password:
passwd: password successfully changed for baseuser
bash-3.00# grep baseuser /etc/passwd
baseuser:x:104:1::/export/home/baseuser:/bin/sh
bash-3.00# su baseuser
$ reboot
reboot: permission denied
$ exit
bash-3.00# roleadd -m -d /export/home/reboot reboot
64 blocks
bash-3.00# passwd reboot
New Password:
Re-enter new Password:
passwd: password successfully changed for reboot
bash-3.00# █

```

Рис.4.18. Создание роли в системе

Шаг 6. Теперь необходимо удостовериться в создании роли и задании ей пароля путем поиска строки с именем роли «reboot» в файле «/etc/passwd». Сделать это можно с помощью команды «grep reboot /etc/passwd», как показано на рис.4.19. Если роль создана, то строка будет найдена и командный интерпретатор выдаст, строку следующего формата «reboot:x:5008:1::/export/home/reboot:/bin/pfsh». Обратите внимание на командный интерпретатор (/bin/pfsh). Этот шелл позволяет пользователю выполнять команду в профиле. Это не тот командный интерпретатор, с помощью которого вы можете войти в систему.

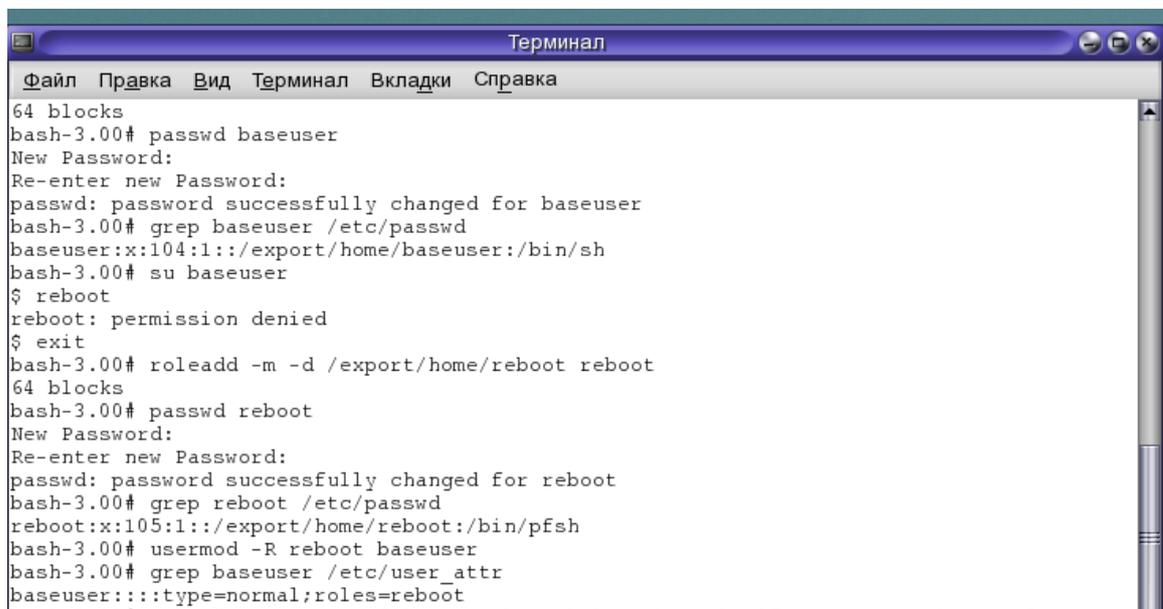
```

Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
# bash
bash-3.00# useradd -m -d /export/home/baseuser baseuser
64 blocks
bash-3.00# passwd baseuser
New Password:
Re-enter new Password:
passwd: password successfully changed for baseuser
bash-3.00# grep baseuser /etc/passwd
baseuser:x:104:1::/export/home/baseuser:/bin/sh
bash-3.00# su baseuser
$ reboot
reboot: permission denied
$ exit
bash-3.00# roleadd -m -d /export/home/reboot reboot
64 blocks
bash-3.00# passwd reboot
New Password:
Re-enter new Password:
passwd: password successfully changed for reboot
bash-3.00# grep reboot /etc/passwd
reboot:x:105:1::/export/home/reboot:/bin/pfsh
bash-3.00#

```

Рис.4.19. Проверка создания роли

Шаг 7. Присоединим пользователя “baseuser” к роле “reboot”, с помощью команды «`usermod -R reboot baseuser`», а затем выполните поиск строки, содержащей «baseuser» в файле «`/etc/user_attr`» используя команду «`grep baseuser /etc/user_attr`», как показано на рис.4.20. Обратите внимание на ключ «-R», он находится в верхнем регистре – командный интерпретатор чувствителен к регистру.



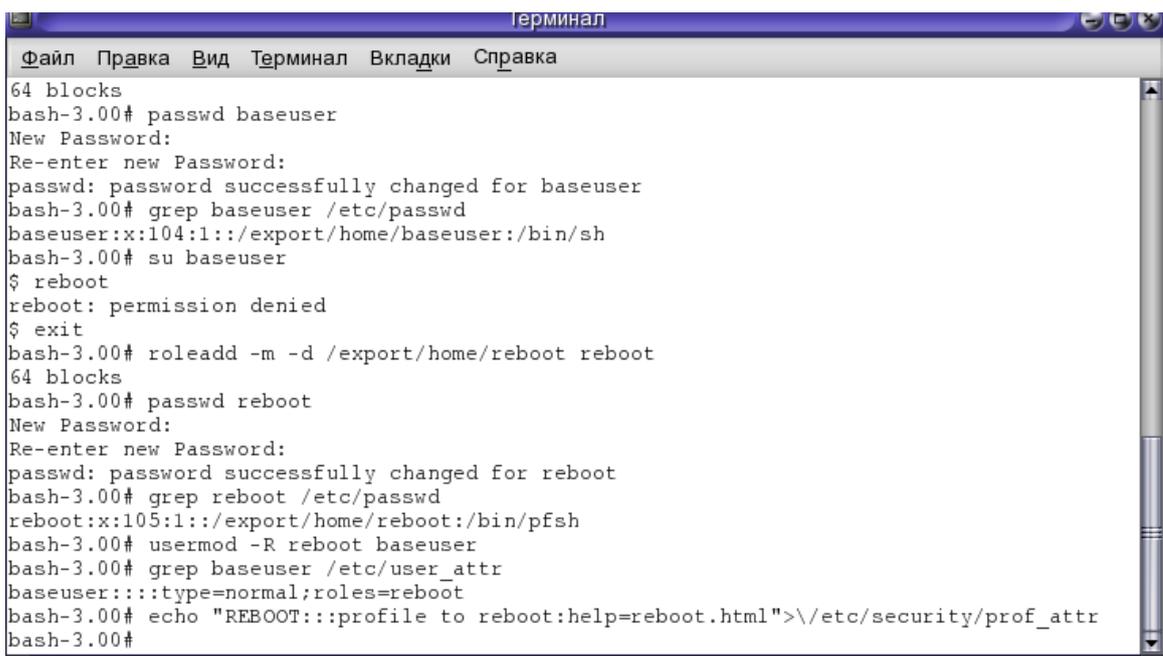
```

Терминал
Файл Правка Вид Терминал Вкладки Справка
64 blocks
bash-3.00# passwd baseuser
New Password:
Re-enter new Password:
passwd: password successfully changed for baseuser
bash-3.00# grep baseuser /etc/passwd
baseuser:x:104:1::/export/home/baseuser:/bin/sh
bash-3.00# su baseuser
$ reboot
reboot: permission denied
$ exit
bash-3.00# roleadd -m -d /export/home/reboot reboot
64 blocks
bash-3.00# passwd reboot
New Password:
Re-enter new Password:
passwd: password successfully changed for reboot
bash-3.00# grep reboot /etc/passwd
reboot:x:105:1::/export/home/reboot:/bin/pfsh
bash-3.00# usermod -R reboot baseuser
bash-3.00# grep baseuser /etc/user_attr
baseuser:::type=normal;roles=reboot

```

Рис.4.20. Присоединение пользователя к роли

Шаг 8. Создадим новый профиль REBOOT. Воспользуемся командой «`echo "REBOOT:::profile to reboot:help=reboot.html">\etc/security/prof_attr`», как показано на рис.4.21.



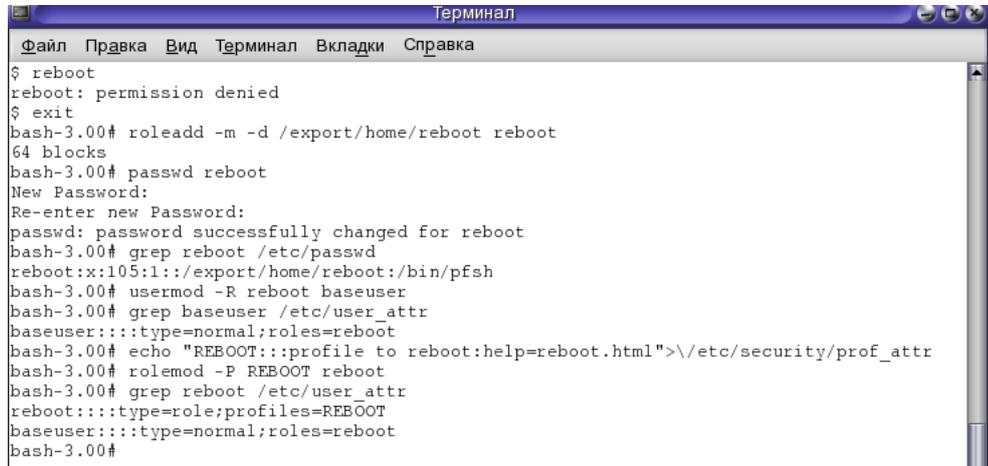
```

Терминал
Файл Правка Вид Терминал Вкладки Справка
64 blocks
bash-3.00# passwd baseuser
New Password:
Re-enter new Password:
passwd: password successfully changed for baseuser
bash-3.00# grep baseuser /etc/passwd
baseuser:x:104:1::/export/home/baseuser:/bin/sh
bash-3.00# su baseuser
$ reboot
reboot: permission denied
$ exit
bash-3.00# roleadd -m -d /export/home/reboot reboot
64 blocks
bash-3.00# passwd reboot
New Password:
Re-enter new Password:
passwd: password successfully changed for reboot
bash-3.00# grep reboot /etc/passwd
reboot:x:105:1::/export/home/reboot:/bin/pfsh
bash-3.00# usermod -R reboot baseuser
bash-3.00# grep baseuser /etc/user_attr
baseuser:::type=normal;roles=reboot
bash-3.00# echo "REBOOT:::profile to reboot:help=reboot.html">\etc/security/prof_attr
bash-3.00#

```

Рис.4.21 Создание профиля

Шаг 9. Соединим профиль “REBOOT” с ролью “reboot” и удостоверимся в установленных ассоциациях используя команды «rolemod -P REBOOT reboot» и «grep reboot /etc/user_attr», как показано на рис.4.22.



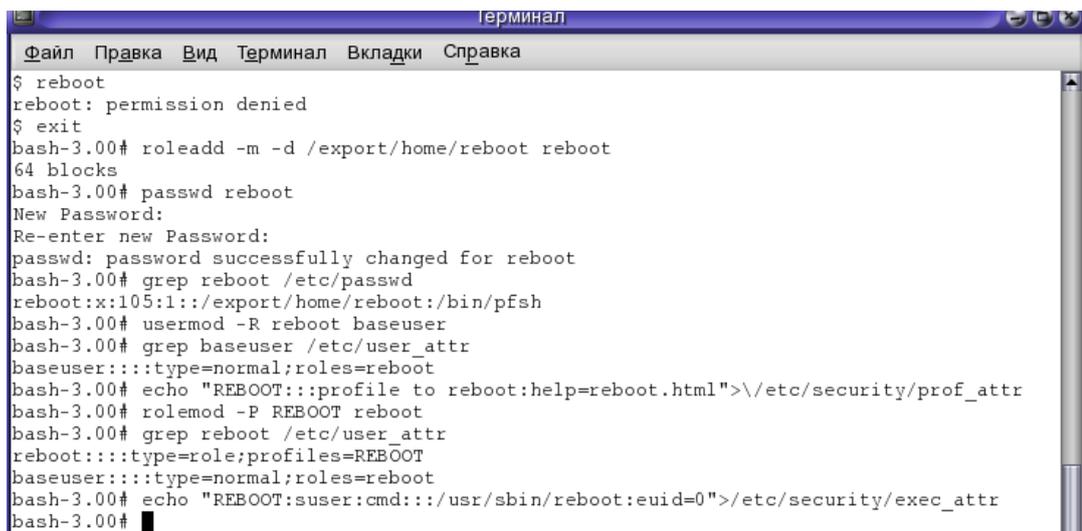
```

Терминал
Файл Правка Вид Терминал Вкладки Справка
$ reboot
reboot: permission denied
$ exit
bash-3.00# roleadd -m -d /export/home/reboot reboot
64 blocks
bash-3.00# passwd reboot
New Password:
Re-enter new Password:
passwd: password successfully changed for reboot
bash-3.00# grep reboot /etc/passwd
reboot:x:105:1::/export/home/reboot:/bin/pfsh
bash-3.00# usermod -R reboot baseuser
bash-3.00# grep baseuser /etc/user_attr
baseuser::::type=normal;roles=reboot
bash-3.00# echo "REBOOT::profile to reboot:help=reboot.html">\/etc/security/prof_attr
bash-3.00# rolemod -P REBOOT reboot
bash-3.00# grep reboot /etc/user_attr
reboot::::type=role;profiles=REBOOT
baseuser::::type=normal;roles=reboot
bash-3.00#

```

Рис.4.22. Соединение профиля с ролью

Шаг 10. Итак, у нас есть пользователь “baseuser” ассоциированный с ролью “reboot”. Роль “reboot” имеет профиль “REBOOT”. Нам осталось сделать так, чтобы профиль (REBOOT) разрешал роле (reboot) выполнить /usr/sbin/reboot с UID 0. Воспользуемся следующей командой «echo "REBOOT:suser:cmd:::/usr/sbin/reboot:euid=0">\/etc/security/exec_attr», как показано на рис.4.23.



```

Терминал
Файл Правка Вид Терминал Вкладки Справка
$ reboot
reboot: permission denied
$ exit
bash-3.00# roleadd -m -d /export/home/reboot reboot
64 blocks
bash-3.00# passwd reboot
New Password:
Re-enter new Password:
passwd: password successfully changed for reboot
bash-3.00# grep reboot /etc/passwd
reboot:x:105:1::/export/home/reboot:/bin/pfsh
bash-3.00# usermod -R reboot baseuser
bash-3.00# grep baseuser /etc/user_attr
baseuser::::type=normal;roles=reboot
bash-3.00# echo "REBOOT::profile to reboot:help=reboot.html">\/etc/security/prof_attr
bash-3.00# rolemod -P REBOOT reboot
bash-3.00# grep reboot /etc/user_attr
reboot::::type=role;profiles=REBOOT
baseuser::::type=normal;roles=reboot
bash-3.00# echo "REBOOT:suser:cmd:::/usr/sbin/reboot:euid=0">\/etc/security/exec_attr
bash-3.00# █

```

Рис.4.23. Задание разрешений профилю

Шаг 11. Теперь “baseuser” может переключиться под роль “reboot” и запустить /usr/sbin/reboot для перезагрузки машины, однако из административных задач пользователю «baseuser» разрешена только

перезагрузка, выполнение например команды «ifconfig» будет запрещено. Переключимся под пользователя «baseuser» используя команду «su baseuser», а затем войдем в роль «reboot» командой «su reboot». Попробуем выполнить команду «ifconfig», на что система ответит отказом как показано на рис.4.24.



```
bash-3.00# su baseuser
$ su reboot
Password:
$ ifconfig
ifconfig: not found
$
```

Рис. 4.24 Проверка прав пользователя

Шаг 12. Введите команду «reboot» и система перезагрузится.

4.9. Содержание отчета

1. Цель работы.
2. Результаты выполнения специального задания.
3. Выводы по проделанной работе.

4.10. Контрольные вопросы

1. Расскажите, на чем основывалась безопасность в ОС семейства UNIX?
2. В чем преимущество концепции RBAC?
3. В чем заключается базис, на котором основана концепция RBAC?
4. В каком файле в ОС Solaris 10 хранятся пароли пользователей и их числовые идентификаторы?
5. Команда grep?
6. Команды семейства user?
7. Команда echo?

ЛАБОРАТОРНАЯ РАБОТА № 5 «Возможности и средства ОС Linux»

Часть 1 - «Администрирование операционной системы Linux»

5.1. Цель работы

Получение практических навыков по администрированию операционной системы Linux, а именно управление пользователями и исследование локальной сети с помощью утилиты «netstat».

5.2. Теоретическая часть

5.2.1. Особенности и возможности ОС Linux

Linux – это многопользовательская, многозадачная, многотерминальная ОС семейства UNIX, под управлением которой могут одновременно выполняться несколько задач. Она предназначена для работы на серверах и рабочих станциях, обеспечивает подключение дополнительных терминалов и допускает в этом режиме использование графических оболочек.

Linux – это современная многопользовательская сетевая ОС с графическим интерфейсом. Все компоненты системы, включая исходные тексты, распространяются с лицензией на свободное распространение (Open source) и установку для неограниченного числа пользователей. Данная операционная система поддерживает стандарты открытых систем и протоколы сети Internet и совместима с системами Unix, MS Windows. Разработана ОС Linux Линусом Торвалдсом (Linus Torvalds) из университета Хельсинки. Первая версия ОС Linux появилась в 1991 г. и с тех пор постоянно развивается, благодаря огромному числу пользователей сети Internet, сотрудников исследовательских центров, фондов, университетов и т.д.

Преимущества ОС Linux:

- не подвержена вирусам, что позволяет хранить информации более надежно;
- распространяется по лицензии Open Source, что дает возможность бесплатно иметь современную ОС для использования как на работе, так и дома;
- обладает высоким быстродействием, т.е. позволяет обрабатывать некоторые операции быстрее, чем другие ОС.

- позволяет эффективно управлять многозадачностью и приоритетами, что дает Linux преимущества перед другими ОС;
- позволяет выполнять представленные в формате загрузки прикладные программы других ОС - различных версий Unix, MS Windows;
- обеспечивает использование большого числа разнообразных программных пакетов, свободно распространяемых вместе с исходным кодом. При этом пользователь может дорабатывать программы и операционную систему;
- предоставляет набор инструментальных средств для разработки прикладных программ любой степени сложности, включая системы класса клиент-сервер, объектно-ориентированные, с многооконным текстовым и/или графическим интерфейсом, пригодных для работы как в Linux, так и в других ОС;
- предоставляет учебную базу пользователем и особенно разработчику. Она предоставляется в виде документации и исходных текстов всех компонент, включая ядро самой ОС;
- **Характерные особенности ОС Linux:**
- Многозадачность. Позволяет нескольким программам выполняться одновременно;
- Экономная загрузка. Linux считывает с диска только те части программы, которые действительно используются для выполнения;
- Защита памяти процесса. Не вызывает зависание системы при сбое программы.
- Многопользовательский режим. Позволяет нескольким пользователям одновременно работать на одной и той же машине;
- Разделение страниц по записи между экземплярами выполняемой программы. Это значит, что процессы-экземпляры программы могут использовать при выполнении одну и ту же память.
- Виртуальная память со страничной организацией (т.е. на диск из памяти вытесняется не весь неактивный процесс, а только требуемая страница); при необходимости позволяет изменять размер виртуальной памяти во время выполнения программ;
- Общая память программ и дискового кэша: вся свободная память используется для буферизации обмена с диском;

- Поддержка национальных алфавитов и соглашений. В том числе для русского языка; возможность добавлять новые;
- Динамические загружаемые разделяемые библиотеки;
- Дамп программы для пост-мортем анализа. Позволяет анализировать отладчиком не только выполняющуюся, но и завершившуюся аварийно программу;
- Сертификация по стандарту POSIX.1, совместимость со стандартами System V и BSD на уровне исходных текстов;
- Множественные виртуальные консоли. Позволяет выводить на одном дисплее несколько одновременных независимых сеансов работы, переключаемых с клавиатуры;
- Через iBCS2-согласованный эмулятор совместимость с SCO, SVR3, SVR4 по загружаемым программам,
- Наличие исходного текста всех программ. Включая тексты ядра, драйверов, средств разработки и приложений. Эти тексты свободно распространяются.
- Управление заданиями в стандарте POSIX;
- Эмуляция сопроцессора в ядре. Позволяет приложению не заботиться об эмуляции сопроцессора;
- Поддержка ряда распространенных файловых систем (MINIX, Xenix, файловые системы System V); наличие собственной передовой файловой системы объемом до 4 Терабайт и с именами файлов до 255 знаков;
- Поддержка сети TCP/IP, включая ftp, telnet, NFS и т.д.

Программное обеспечение, которое поддерживает Linux

Инструментальные средства программирования. Основой средств программирования является компилятор GCC или его экспериментальные версии EGCS и PGCC для языков C и C++; модули поддержки других языков программирования (Objective C, Фортран, Паскаль, Modula-3, Ада, Java и др.); интегрированные среды и средства визуального проектирования: Kdevelop, Xwpe; средства адаптации привязки программ AUTOCONFIG, AUTOMAKE.

5.2.2 Работа с пользователями в ОС Linux

В LINUX существуют три типа пользователей: пользователь root; системные (фиктивные) пользователи; обычные пользователи.

Каждому пользователю кроме имени присваивается числовой идентификатор пользователя (UID, User IDentificator). Пользователь root имеет UID равный 0. Системные пользователи имеют идентификаторы от 1 до 499. Обычные пользователи имеют UID от 500. Для удобства управления доступом пользователей к файлам системы пользователи объединяются в группы. Каждый пользователь входит хотя бы в одну группу. Группе присваивается числовой идентификатор группы (GID - Group IDentificator).

Таблица 5.1 - Информация о пользователях хранится в файлах

Файл	Описание
/etc/passwd	пользователи
/etc/group	группы пользователей
/etc/shadow	зашифрованные пароли пользователей
/etc/gshadow	зашифрованные пароли групп

Таблица 5.2 - Дополнительная информация располагается в файлах

Файл	Описание
/etc/default/useradd	свойства по умолчанию для новых пользователей
/etc/login.defs	настройки новых пользователей
/etc/skel/	каталог, файлы из которого копируются в домашний каталог нового пользователя

Файл /etc/passwd имеет следующий формат записи:

имя_пользователя:пароль:UID:GID:комментарий:домашний_каталог
:командная_оболочка.

Все элементы записи разделяются символом : (двоеточие) и пишутся без пробелов. В случае, если пароль хранится в зашифрованном виде в файле /etc/shadow, то вместо пароля указывается символ x (икс).

Файл /etc/group имеет следующий формат записи:
имя_группы:пароль:GID:пользователь1,пользователь2,пользователь3

Все элементы записи разделяются символом : (двоеточие) и пишутся без пробелов. Список пользователей группы разделяется символом , (запятая).

К файлам /etc/passwd и /etc/group должны быть заданы права доступа: чтение и запись для пользователя root и только чтение - для всех остальных.

Таблица 5.3 - Для управления пользователями служат команды

useradd	добавить нового пользователя
passwd	установить пароль пользователя
usermod	изменить параметры учетной записи пользователя
userdel	удалить учетную запись пользователя

Таблица 5.4 - Для управления группами служат команды.

groupadd	создать новую группу
gpasswd	установить пароль группы
groupmod	изменить параметры группы
groupdel	удалить группу

5.2.3 Управление ограничением доступа к файлам и папкам

В LINUX файлами считаются обычные файлы, каталоги, а также специальные файлы, соответствующие периферийным устройствам (каждое устройство представляется в виде файла). Доступ ко всем файлам однотипный, в том числе, и к файлам периферийных устройств. Такой подход обеспечивает независимость программы пользователя от

особенностей ввода/вывода на конкретное внешнее устройство.

В LINUX различаются 3 уровня доступа к файлам и каталогам:

1. доступ владельца файла;
2. доступ группы пользователей, к которой принадлежит владелец файла;
3. остальные пользователи.

Для каждого уровня существуют свои байты атрибутов, значение которых расшифровывается следующим образом:

1. r – атрибут разрешающий чтение;
2. w – атрибут разрешающий запись;
3. x – атрибут разрешающий выполнение;
4. - – атрибут показывающий отсутствие разрешения.

Первый символ байта атрибутов определяет тип файла и может интерпретироваться со следующими значениями:

1. - – обычный файл;
2. d – каталог;
3. l – символическая связь;
4. в – блок-ориентированный специальный файл, который соответствует таким периферийным устройствам, как накопители на магнитных дисках;
5. с – байт-ориентированный специальный файл, который может соответствовать таким периферийным устройствам как принтер, терминал.

В домашнем каталоге пользователь имеет полный доступ к файлам (READ, WRITE, EXECUTE; r, w, x).

Атрибуты файла можно просмотреть командой **ls -l** и они представляются в следующем формате:

```
d  rwx  rwx  rwx
|  |  |  |
|  |  |  | Доступ для остальных пользователей
```

```
| | | Доступ к файлу для членов группы
| | | Доступ к файлу владельца
| | | Тип файла (директория)
```

Пример. Командой `ls -l` получим листинг содержимого текущей директории `student`:

```
- rwx --- --- 2 student 100 September 5 09:40 file_1
- rwx --- r-- 1 adm    200 September 15 10:00 file_2
- rwx --- r-- 1 student 100 September 25 10:30 file_3
```

После байтов атрибутов на экран выводится следующая информация о файле:

1. Число связей файла;
2. Имя владельца файла;
3. Размер файла в байтах;
4. Дата создания файла (или модификации);
5. Время;
6. Имя файла.

5.2.4 Сетевая утилита `netstat`

Утилита `netstat` выводит информацию о локальной сети и средствах TCP/IP. Данная утилита позволяет пользователю просмотреть сетевые соединения, таблицы маршрутизации, статистику интерфейсов, NAT-соединения, и членство в *multicast*-группах. Именно с данной утилитой приходится иметь дело администраторам, чтобы быстро найти причину неисправности в сети TCP/IP. Содержание и форма выходной информации зависят от операционной системы, но обычно выводятся следующие данные: список соединений, статистика сетевых интерфейсов, информация по буферам данных, содержание таблицы маршрутизации, статистика работы протокола. Характер выводимой информации можно выбирать с помощью опции командной строки.

Таблица 5.5 – Параметры команды netstat

-a	Параметр предоставляет просматривать состояние всех сокетов; обычно сокет, используемый серверными процессами, не показывается.
-A	Показывает адреса любых управляющих блоков протокола, связанных с сокетами; используется для отладки.
-i	Показывает состояние автоматически сконфигурированных (auto-configured) интерфейсов.
-n	Показывает сетевые адреса как числа. Netstat обычно показывает адреса как символы. Эту опцию можно использовать с любым форматом показа.
-r	Показать таблицы маршрутизации. При использовании с опцией -s, показывает статистику маршрутизации.
-s	Показать статистическую информацию по протоколам. При использовании с опцией -r, показывает статистику маршрутизации.
-f	Ограничить показ статистики или адресов управляющих блоков только указанным семейством адресов.
-I	Выделять информацию об указанном интерфейсе в отдельный столбец; по умолчанию (для третьей формы команды) используется интерфейс с наибольшим объемом переданной информации с момента последней перезагрузки системы. В качестве интерфейса можно указывать любой из интерфейсов, перечисленных в файле конфигурации системы, например, emd1 или lo0.
-p	Отобразить идентификатор/название процесса создавшего сокет (-p, --programs display PID/Program name for sockets)

5.3. Порядок выполнения работы

Локальную учетную запись в ОС Linux можно создать с помощью команды: `sudo useradd "имя_пользователя"`, информацию об учетной записи сохраняется в файле `/etc/passwd`. В этом файле хранится такая

информация, как имя пользователя, домашняя директория, командная оболочка и некоторые комментарии, связанные с учетной записью. Обычно эти учетные записи называются *локальными учетными записями UNIX*. Создаем локальную учетную запись с именем пользователя *students* и описанием *Students SAiT* в разделе комментариев (-с). Для того чтобы команда выполнялась необходимо иметь права 'root', для этого используем команду `sudo`.

`sudo useradd -с "имя_пользователя"`,

1. Добавление учетной записи:

`sudo passwd "имя_пользователя"`.

2. Добавление группы пользователей

`Sudo groupadd "имя_группы"`

3. Добавление пользователя в группу

`sudo usermod -a -G "имя_группы" "имя_пользователя"`

Также можно одной командой создать пользователя и сразу добавить его в уже имеющуюся группу.

`Sudo useradd -G "имя_группы" "имя_пользователя"`

4. Вывод на экран список групп, к которым принадлежит пользователь

`Sudo groups "имя_пользователя"`

5. Удаление пользователя из группы

`Sudo gpasswd -d "имя_пользователя" "имя_группы"`

6. Удаление пользователей и групп

`Sudo groupdel "имя_группы"`

`Sudo7 userdel 'имя_пользователя'`

7. Создание нового каталога для директории в которой вы в данный момент находитесь:

`sudo mkdir "название каталога"`

8. Информацию о правах доступа к файлам и папкам:

`Ls -l`

```

artyom@artyom-1015P: /
artyom@artyom-1015P:/$ sudo mkdir newfolder
artyom@artyom-1015P:/$ ls -l
итого 92
drwxr-xr-x  2 root root  4096 2013-02-02 15:52 bin
drwxr-xr-x  3 root root  4096 2013-02-02 16:14 boot
drwxr-xr-x  2 root root  4096 2013-02-02 18:55 cdrom
drwxr-xr-x 16 root root  4200 2013-10-02 22:00 dev
drwxr-xr-x 138 root root 12288 2013-10-02 22:40 etc
drwxr-xr-x  5 root root  4096 2013-10-02 22:19 home
lrwxrwxrwx  1 root root    33 2013-02-02 16:07 initrd.img -> /boot/initrd.img-3
.0.0-30-generic
lrwxrwxrwx  1 root root    32 2013-02-02 18:57 initrd.img.old -> boot/initrd.im
g-3.0.0-12-generic
drwxr-xr-x 20 root root  4096 2013-02-02 21:29 lib
drwx----- 2 root root 16384 2013-02-02 18:51 lost+found
drwxr-xr-x  6 root root  4096 2013-10-02 22:05 media
drwxr-xr-x  2 root root  4096 2011-10-09 11:29 mnt
drwxr-xr-x  2 root root  4096 2013-10-02 22:50 newfolder
drwxr-xr-x  4 root root  4096 2013-06-04 21:58 opt
dr-xr-xr-x 157 root root    0 2013-10-02 02:44 proc
drwx-----  6 root root  4096 2013-02-03 21:28 root
drwxr-xr-x 20 root root   720 2013-10-01 22:45 run
drwxr-xr-x  2 root root  4096 2013-02-02 16:10 sbin
drwxr-xr-x  2 root root  4096 2011-06-21 22:43 selinux

```

Рис.5.1. Информация о правах доступа к файлам

9. Изменим правообладателя каталога newfolder

Команда `chown` “*имя_пользователя*” “*имя_папки или файла*” сменит пользователя-владельца файла или каталога

10. Для смены группы отдельно можно использовать команду вида `chgroup` “*имя_группы*” “*имя_файла*”

```

artyom@artyom-1015P: /
artyom@artyom-1015P:/$ sudo chown students newfolder
artyom@artyom-1015P:/$ ls -l
итого 92
drwxr-xr-x  2 root  root  4096 2013-02-02 15:52 bin
drwxr-xr-x  3 root  root  4096 2013-02-02 16:14 boot
drwxr-xr-x  2 root  root  4096 2013-02-02 18:55 cdrom
drwxr-xr-x 16 root  root  4200 2013-10-02 22:00 dev
drwxr-xr-x 138 root  root 12288 2013-10-02 22:40 etc
drwxr-xr-x  5 root  root  4096 2013-10-02 22:19 home
lrwxrwxrwx  1 root  root    33 2013-02-02 16:07 initrd.img -> /boot/initrd.i
mg-3.0.0-30-generic
lrwxrwxrwx  1 root  root    32 2013-02-02 18:57 initrd.img.old -> boot/initr
d.img-3.0.0-12-generic
drwxr-xr-x 20 root  root  4096 2013-02-02 21:29 lib
drwx----- 2 root  root 16384 2013-02-02 18:51 lost+found
drwxr-xr-x  6 root  root  4096 2013-10-02 22:05 media
drwxr-xr-x  2 root  root  4096 2011-10-09 11:29 mnt
drwxr-xr-x  2 students root  4096 2013-10-02 22:50 newfolder
drwxr-xr-x  4 root  root  4096 2013-06-04 21:58 opt
dr-xr-xr-x 158 root  root    0 2013-10-02 02:44 proc
drwx-----  6 root  root  4096 2013-02-03 21:28 root
drwxr-xr-x 20 root  root   720 2013-10-01 22:45 run
drwxr-xr-x  2 root  root  4096 2013-02-02 16:10 sbin
drwxr-xr-x  2 root  root  4096 2011-06-21 22:43 selinux

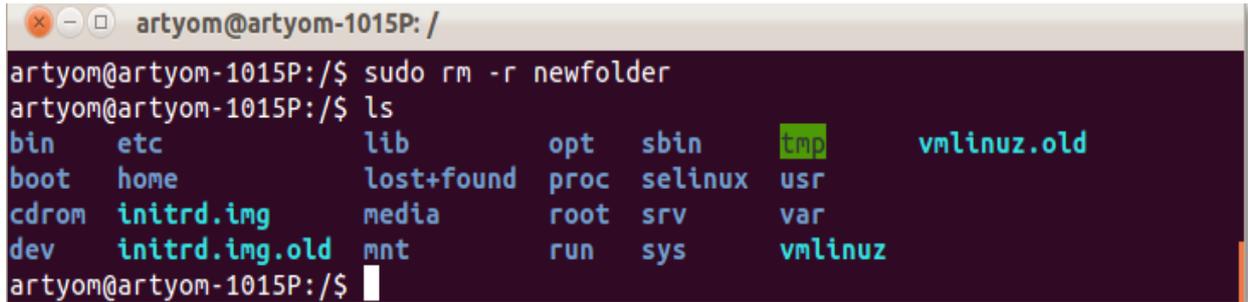
```

Рис.5.2. Список команд

11. Удаление ранее созданного каталога (Рис.5.3)

`Rm -r` “*имя_каталога*”

Rm “имя_файла”



```

artyom@artyom-1015P: /
artyom@artyom-1015P:/$ sudo rm -r newfolder
artyom@artyom-1015P:/$ ls
bin      etc      lib      opt      sbin     tmp      vmlinuz.old
boot    home    lost+found  proc    selinux  usr
cdrom   initrd.img  media    root    srv      var
dev     initrd.img.old  mnt      run     sys      vmlinuz
artyom@artyom-1015P:/$

```

Рис.5.3. Удаление ранее созданного каталога

12. Команда `netstat -a` обладает набором ключей для отображения портов, находящихся в активном и/или пассивном состоянии. Таким образом, можно получить список всех серверных приложений, работающих на данном компьютере. Информация выводится столбцами. В первом из них указан протокол, затем размеры очередей приема и передачи для установленного соединения на данной машине (на другом конце соединения размеры очередей могут быть другими), локальный и удаленный адреса и текущее состояние соединения.

Вывод состояния всех сокетов (как прослушиваемых, так и имеющих другое состояние):

- `Netstat -a | more`
- Просмотр только TCP соединений:
- `Netstat -at`
- Просмотр только UDP соединений:
- `Netstat -au`
- Виды состояний (state):
- ESTABLISHED - соединение установлено;
- SYN_SENT - сокет пытается установить соединение;
- SYN_RECV - начальная синхронизация соединения;
- FIN_WAIT1 - сокет закрыт, ожидание отключения соединения;
- FIN_WAIT2 - ожидание отключения удаленной стороны;
- TIME_WAIT - ожидание после закрытия повторной передачи отключения удаленной стороны;
- CLOSED - соединение закрыто, сокет не используется;

- CLOSE_WAIT - удаленный узел отключился, сокет скоро будет закрыт;
- LAST_ACK - сначала отключилась удаленная сторона, а затем сокет будет закрыт;
- LISTEN - сокет ожидает входящих соединений;
- CLOSING - закрытие сокета, точнее сокет закрыт, а затем отключился удаленный узел;
- UNKNOWN - состояние сокета неизвестно.

Сокеты (англ. socket — разъём) — название программного интерфейса для обеспечения обмена данными между процессами. Процессы при таком обмене могут исполняться как на одной ЭВМ, так и на различных ЭВМ, связанных между собой сетью. Сокет — абстрактный объект, представляющий конечную точку соединения.

Следует различать клиентские и серверные сокеты. Клиентские сокеты грубо можно сравнить с оконечными аппаратами телефонной сети, а серверные — с коммутаторами. Клиентское приложение (например, браузер) использует только клиентские сокеты, а серверное (например, веб-сервер, которому браузер посылает запросы) — клиентские и серверные сокеты. Интерфейс сокетов впервые появился в BSD Unix. Программный интерфейс сокетов описан в стандарте POSIX.1 и в той или иной мере поддерживается всеми современными ОС.

```

artyom@artyom-desktop: ~
Файл Правка Вид Терминал Справка
artyom@artyom-desktop:~$ netstat -a | more
Активные соединения с интернетом (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 localhost:ipp *.* LISTEN
tcp 0 0 192.168.117.129:56612 srv118-131.vkontakt:www ESTABLISHED
tcp 0 0 192.168.117.129:36287 cache.google.com:https ESTABLISHED
tcp 0 0 192.168.117.129:39000 a23-3-91-132.deploy:www ESTABLISHED
tcp 0 0 192.168.117.129:51721 srv117-131.vkontakt:www ESTABLISHED
tcp 0 0 192.168.117.129:51289 jujube.canonical.co:www ESTABLISHED
tcp 0 0 192.168.117.129:51722 srv117-131.vkontakt:www ESTABLISHED
tcp 0 0 192.168.117.129:56602 srv118-131.vkontakt:www ESTABLISHED
tcp 0 0 192.168.117.129:49778 host42.rax.ru:www TIME_WAIT
tcp 0 0 192.168.117.129:35916 la-in-f139.1e100.ne:www ESTABLISHED
tcp 0 0 192.168.117.129:60384 arn06s01-in-f3.1e10:www ESTABLISHED
tcp 0 0 192.168.117.129:56603 srv118-131.vkontakt:www ESTABLISHED
tcp 0 0 192.168.117.129:55424 srv99-131.vkontak:https ESTABLISHED
tcp 0 0 192.168.117.129:38520 a23-3-91-122.deploy:www ESTABLISHED
tcp 0 0 192.168.117.129:58035 srv120-131.vkontakt:www ESTABLISHED
tcp 0 0 192.168.117.129:58036 srv120-131.vkontakt:www ESTABLISHED
tcp 0 0 192.168.117.129:41730 a23-3-91-33.deploy.:www ESTABLISHED
tcp 0 0 192.168.117.129:51728 srv117-131.vkontakt:www ESTABLISHED
tcp 0 0 192.168.117.129:56601 srv118-131.vkontakt:www ESTABLISHED
tcp 0 0 192.168.117.129:47588 la-in-f102.1e100.ne:www ESTABLISHED

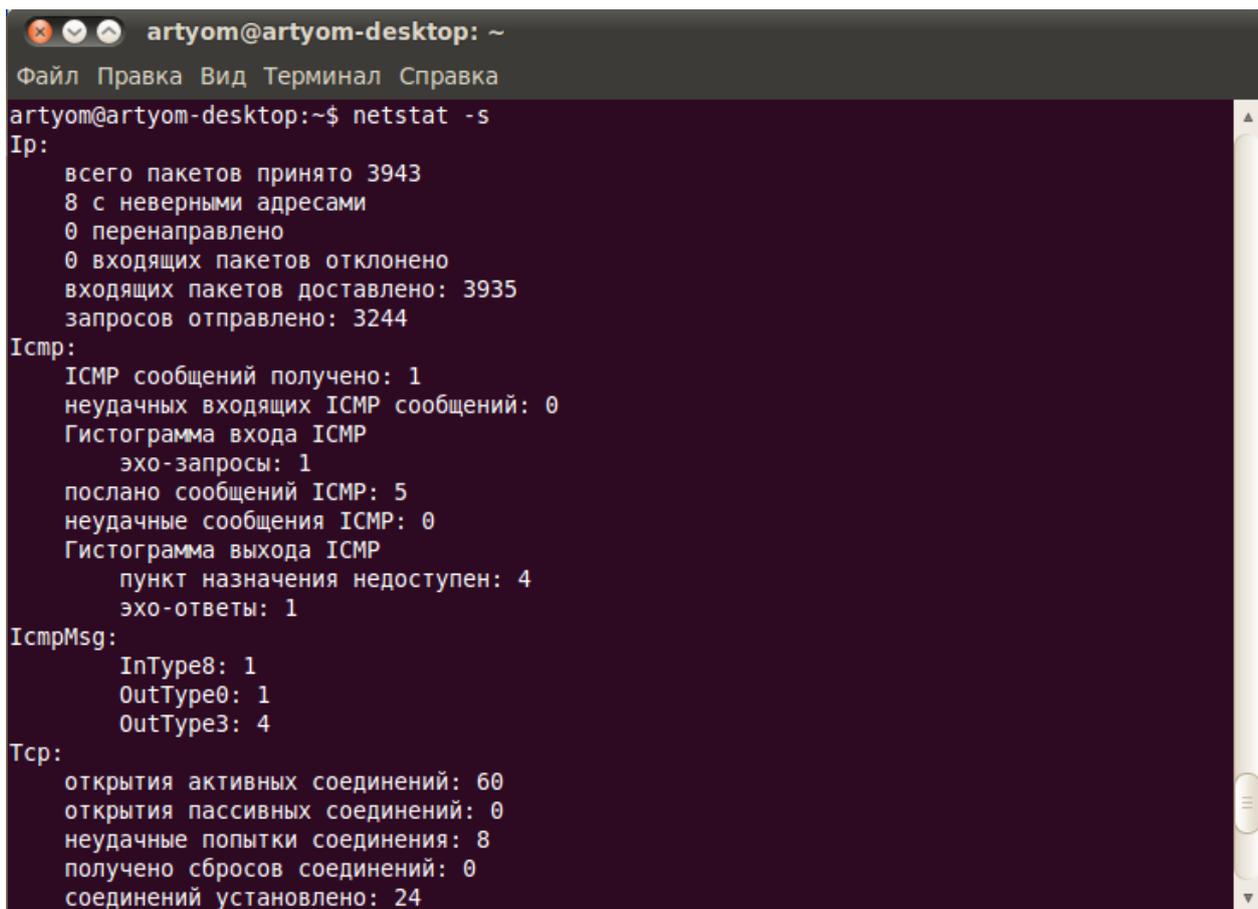
```

Рис.5.4. Вывод активных соединений с интернетом

13. Показ статистической информации по протоколам.

- Для протокола TCP *netstat -st*
- Для протокола UDP *netstat -su*
- Общая статистика *netstat -s*

Команда *netstat -s* выдает содержимое счетчиков сетевых программ. В выходной информации есть разделы, относящиеся к различным протоколам: IP, ICMP, TCP, UDP. С ее помощью можно определить место появления ошибки в принятом пакете.



```

artyom@artyom-desktop: ~
Файл Правка Вид Терминал Справка
artyom@artyom-desktop:~$ netstat -s
Ip:
  всего пакетов принято 3943
  8 с неверными адресами
  0 перенаправлено
  0 входящих пакетов отклонено
  входящих пакетов доставлено: 3935
  запросов отправлено: 3244
Icmp:
  ICMP сообщений получено: 1
  неудачных входящих ICMP сообщений: 0
  Гистограмма входа ICMP
    эхо-запросы: 1
  послано сообщений ICMP: 5
  неудачные сообщения ICMP: 0
  Гистограмма выхода ICMP
    пункт назначения недоступен: 4
    эхо-ответы: 1
IcmpMsg:
  InType8: 1
  OutType0: 1
  OutType3: 4
Tcp:
  открытия активных соединений: 60
  открытия пассивных соединений: 0
  неудачные попытки соединения: 8
  получено сбросов соединений: 0
  соединений установлено: 24
  
```

Рис.5.5. Показ статистической информации по протоколам

14. Каждое соединение машины с сетью называется сетевым интерфейсом. Машина, имеющая более одного интерфейса, может принимать данные по одному интерфейсу и передавать их по-другому, таким образом осуществляя пересылку данных между сетями. Эта функция называется маршрутизацией, а машина, выполняющая ее – шлюзом.

Данные маршрутизации хранятся в одной из таблиц ядра. Для направления пакета по конкретному адресу ядро подбирает наиболее подходящий маршрут. Если такой маршрут отсутствует и нет маршрута

по умолчанию, то отправителю возвращается сообщение об ошибке.

Команда *netstat -r* позволяет отображать таблицу маршрутизации. Пункты назначения и шлюзы могут показываться или именами машин, или их IP-адресами. Флаги дают оценку маршрута.

```

artyom@artyom-desktop: ~
Файл Правка Вид Терминал Справка
artyom@artyom-desktop:~$ netstat -r
Таблица маршрутизации ядра протокола IP
Destination Gateway Genmask Flags MSS Window irtt Iface
192.168.117.0 * 255.255.255.0 U 0 0 0 eth0
link-local * 255.255.0.0 U 0 0 0 eth0
default 192.168.117.2 0.0.0.0 UG 0 0 0 eth0
artyom@artyom-desktop:~$

```

Рис.5.6. Таблица маршрутизации

Таблица 5.6. Команды маршрутизации

Название столбца	Расшифровка
Gateway	Имена используемых шлюзов
Genmask	Маска, используемая для отображения общей части адреса, соответствующего данному маршруту
Flags	Флаги, описывающие маршрут: G Маршрут использует шлюз
	U Интерфейс активен, может использоваться для передачи данных
	H Данные можно передавать только одному узлу
	D Запись создана перенаправляющим сообщением протокола ICMP
	M Запись модифицирована перенаправляющим сообщением протокола ICMP
Iface	Интерфейс, используемый для передачи пакетов

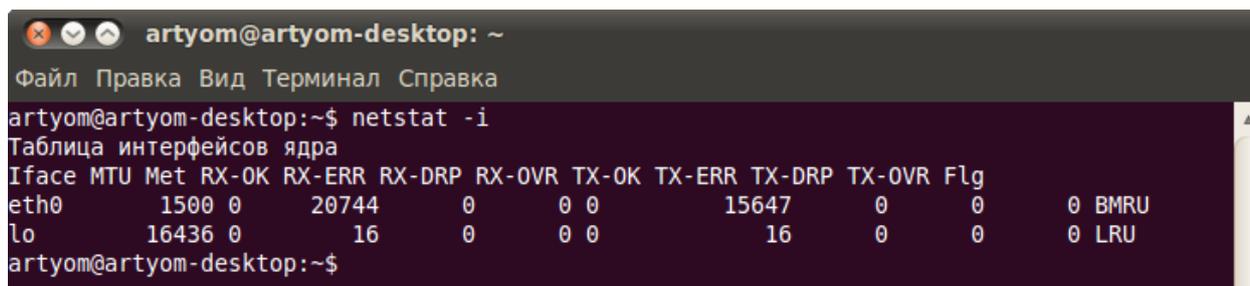
Отображение сетевых интерфейсов и информации по ним:

Netstat -i

Расширенная информация по интерфейсам

Netstat -ie

При использовании ключа `-i` команды `netstat` на экран будут выведены статистические данные всех используемых интерфейсов. Исходя из них, можно выяснить, исправно ли соединение с сетью.



```

artyom@artyom-desktop: ~
Файл Правка Вид Терминал Справка
artyom@artyom-desktop:~$ netstat -i
Таблица интерфейсов ядра
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0    1500 0    20744    0    0 0    15647    0    0    0 BMRU
lo      16436 0     16     0    0 0     16     0    0    0 LRU
artyom@artyom-desktop:~$

```

Рис. 5.7 – Отображение сетевых интерфейсов.

Ошибки являются следствием проблем в кабельной системе. В нормально работающей сети количество конфликтов (RX-OVR, TX-OVR) не должно превышать 3% от числа пакетов, а другие ошибки не должны составлять более 0,5% от общего числа пакетов.

5.4. Задания для лабораторной работы:

- 1) Создать два пользователя “students и prepodavateli”
- 2) Задать этим пользователям пароли
- 3) Создать две группы “SAIT и SAPR”
- 4) Добавить пользователей в обе группы.
- 5) Проверить наличие этих пользователей в группах, путем вывода всех пользователей состоящих в каждой группе.
- 6) Удалить пользователя “students” из группы “SAIT”
- 7) Создать каталог “newfolder”
- 8) Вывести информацию о каталоге, а именно о правах доступа к нему
- 9) Изменить правообладателя каталога “newfolder” на “students”, и убедиться в этом
- 10) Удалить каталог и ранее созданные группы и пользователей.
- 11) С помощью утилиты `netstat` получить список соединений.

12) Получить таблицу маршрутизации. Указать, через какие интерфейсы с какими сетями происходит связь, имя шлюза, маски локальных сетей.

13) Получить статистику сетевых интерфейсов. Объяснить значение установленных флагов. Сравнить количество ошибок с требованиями, предъявляемыми к работе сетей; сделать вывод о работе локальной сети.

14) Получить статистику сетевых интерфейсов. Проанализировать работу каждого из протоколов.

5.5. Содержание отчета

1. Цель работы.
2. Результаты выполнения специального задания.
3. Выводы по проделанной работе.

5.6. Контрольные вопросы

1. Перечислите основные особенности операционной системы Linux.
2. Перечислите программное обеспечение, которое поддерживает система LINUX.
3. Назовите типы пользователей, используемых в ОС Linux.
4. Что присваивается каждому пользователю?
5. Назовите команды по управлению пользователями.
6. Назовите команды по управлению группами пользователей.
7. Перечислите уровни доступа к файлам и каталогам.
8. Расшифруйте байты атрибутов, каждого уровня доступа к файлам и каталогам.
9. Какая информация выводится после байтов атрибутов?
10. Поясните назначение утилиты netstat.

Часть 2 - «Диспетчеризация операционной системы Linux»

5.7. Цель работы

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

5.8. Теоретические часть

5.8.1. Интерактивные процессы

Интерактивные процессы инициализируется и контролируется через терминальную сессию. Иными словами, кто-то должен быть подключен к системе, чтобы эти процессы запустились; они не запускаются автоматически, как часть функциональности системы. Эти процессы могут работать на переднем плане, занимая терминал, который запустил программу, и вы не сможете запускать другие приложения до тех пор, как этот процесс работает таким образом. Иначе они могут работать в фоновом режиме, таким образом, терминал, в котором вы запустили программу, может принимать новые команды, пока та программа тоже работает. До сих пор мы в основном уделяли внимание программам, работающим на переднем плане - продолжительность времени, необходимого для их работы, была слишком короткой, чтобы заметить - но просмотр файлов командой `less` - хороший пример, когда команда занимает терминал сессии. В этом случае, активная программа ожидает от вас какого-нибудь действия. Программа по-прежнему связана с терминалом, откуда она была запущена, а терминал пригоден только для ввода команд для этой программы, только их он может понять. Другие команды просто приведут к ошибкам или зависанию системы.

Несмотря на то, что процесс выполняется в фоновом режиме, пользователь все равно не имеет возможности делать другие вещи в терминале, в котором он запустил программу, пока программа работает.

Оболочка предлагает функцию управления заданиями, которая позволяет легко обрабатывать множество процессов. Этот механизм переключает процессы между передним и задним фоном. Используя эту систему, программы также могут быть запущены в фоновом режиме сразу же.

Запуск процесса в фоновом режиме, имеет смысл только для программ, которые не нуждаются в пользовательском контроле (через

shell). Перевод задания в фоновый режим делается, как правило, когда для его выполнения, как ожидается, потребуется длительного времени.

В таблице 5.7 представлены команды для работы процесса в фоновом режиме:

Таблица 5.7 – Команды ОС Linux

regular_command	Запуск данной команды на переднем плане.
command &	Запуск команды в фоновом режиме (в версии терминала).
jobs	Показывает команды запущенные в фоновом режиме.
Ctrl+Z	Приостановить (остановить, но не выйти) процесс, выполняемый на переднем плане (засыпание).
Ctrl+C	Прервать (прекратить и выйти) процесс, выполняемый на переднем плане.
%n	Каждый процесс, выполняемый в фоновом режиме, получает присвоенный ему номер. Используя выражение %, указание может быть отнесено к процессу путем использования его номера, например, fg %2 .
bg	Возобновить приостановленную программу в фоновом режиме.
fg	Поместить фоновое задание на передний план.
kill	Завершение процесса

5.8.2. Свойства процессов

Процесс имеет ряд особенностей, которые могут быть просмотрены с помощью команды **ps**:

Идентификатор процесса или PID: уникальный идентификационный номер, используемый для обращения к процессу.

PPID родительского процесса или PPID: номер процесса (PID), который запустил данный процесс.

Nice-число: степень дружелюбия этого процесса по отношению к другим процессам (не путать с приоритетом процесса, который рассчитывается на основе этого милого числа и последнего использования процессора этим процессом).

Терминал или TTY: терминал, к которому привязан данный процесс.

Имя реального и эффективного пользователя (RUID и EUID): владелец процесса. Реальным владельцем является пользователь, который ввел команду, эффективный пользователь — тот, кто определяет порядок доступа к системным ресурсам. RUID и EUID обычно это одно и то же, и процесс имеет те же права доступа, которые есть у пользователя его запустившего

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

```
ps -ax > test.txt
```

```
ps -ax > test.txt - добавит в конец файла
```

Группы команд

```
command-1;command-2;command-3
```

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

Отношения между процессами. Отношения между процессами могут быть отображены с использованием команды **pstree**:

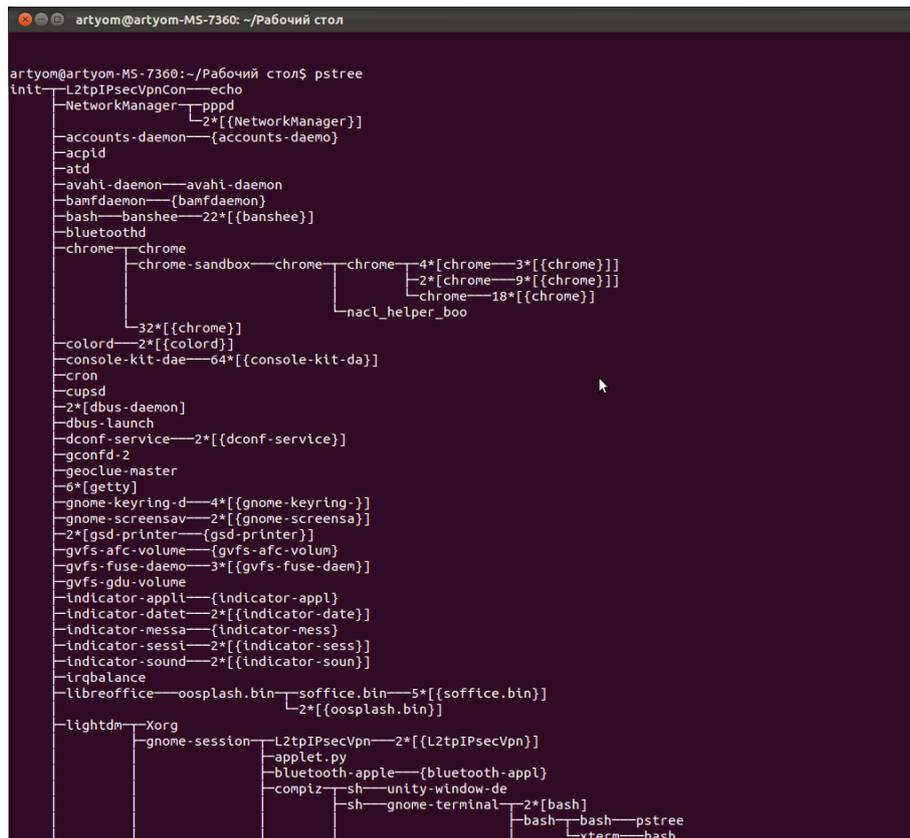


Рис.5.8. Отношение между процессами в Linux

Опции `-u` и `-a` дают дополнительную информацию. Для знакомства с другими опциями, а также тем, что они делают, направляем к info-страницам.

Сигналы. Процессы заканчиваются, если они получают сигнал. Есть несколько сигналов, которые вы можете отправить процессу. Используйте команду `kill`, чтобы послать сигнал процессу. Команда `kill -i` отображает список сигналов. Большинство сигналов предназначены для внутреннего использования системой или для программистов, когда они пишут код.

Часто используемые сигналы. Список сигналов можно получить по команде: `kill -l`.

Таблица 5.8 – Список сигналов

№	Имя	Описание	Можно перехватывать	Можно блокировать	Комбинация клавиш
1	HUP	Hangup. Отбой	Да	Да	
2	INT	Interrupt. В случае выполнения простых команд вызывает прекращение выполнения, в интерактивных программах — прекращение активного процесса	Да	Да	<Ctrl>+<C> или
3	QUIT	Как правило, сильнее сигнала Interrupt	Да	Да	<Ctrl>+<\>
4	ILL	Illegal Instruction. Центральный процессор столкнулся с незнакомой командой (в большинстве случаев это означает, что допущена программная ошибка). Сигнал отправляется программе, в которой возникла проблема	Да	Да	
8	FPE	Floating Point Exception. Вычислительная ошибка, например, деление на ноль	Да	Да	
9	KILL	Всегда прекращает выполнение процесса	Нет	Нет	
11	SEGV	Segmentation Violation. Доступ к недозволённой области памяти	Да	Да	
13	PIPE	Была предпринята попытка передачи данных с помощью конвейера или очереди FIFO, однако не существует процесса, способного принять эти данные	Да	Да	
15	TERM	Software Termination. Требование закончить процесс (программное завершение)	Да	Да	
17	CHLD	Изменение статуса порожденного процесса	Да	Да	
18	CONT	Продолжение выполнения приостановленного процесса	Да	Да	
19	STOP	Приостановка выполнения процесса	Нет	Нет	
20	TSTR	Сигнал останова, генерируемый клавиатурой. Переводит процесс в фоновый режим	Да	Да	<Ctrl>+<Z>

Завершение процесса. Когда процесс завершается нормально (его не завершили или иным способом неожиданно не прервали), программа возвращает его *exit status* (статус выхода) родителю. Этот *exit status* является числом, возвращаемым программой, предоставившей результаты своего выполнения. Система передачи информации об исполненной работе берет свое начало с языка программирования Си, на котором был написан UNIX.

Коды возврата могут быть интерпретированы родителем или скриптами. Значения кодов возврата специфичны для программ. Эту информацию обычно можно найти в man-страницах указанной программы, например команда **grep** возвращает -1, если не найдено совпадений, при этом в строке может быть выведено сообщение: "Нет найденных файлов". Другим примером является встроенная команда Bash **true**, которая не делает ничего, кроме возвращения статуса выхода 0, что означает успех.

Если вы хотите остановить процесс, потому что он висит или собирается полностью вывести из рабочего состояния ввод-вывод, создание файла или использование других ресурсов системы, используйте команду **kill**. Если у вас есть возможность, попробуйте сначала завершить процесс "по тихому", отправив ему сигнал SIGTERM. Это указание к прекращению любой активности в соответствии с процедурами, описанными в коде программы.

Приоритет. Приоритет или важность работы определяется ее числом *nice*. Программа с большим *nice*-числом дружелюбна к другим программам, другим пользователям и системе; т.е. эта программа - не важное дело. Низкое *nice*-число обозначает более важное задание, которое потребует больше ресурсов, не делясь ими.

Улучшение выполнения задания за счет увеличения его *nice*-числа полезно только для процессов, которые используют много процессорного времени (компиляторы, математические приложения и т.п.). Процессы, которые постоянно обращаются к вводу/выводу имеют более высокий приоритет (низкое *nice*-число), например, ввод с клавиатуры всегда получает наивысший приоритет в системе.

Определение приоритетов программ осуществляется с помощью команды **nice**.

Большинство систем также предоставляют команду BSD **renice**, которая позволяет изменять дружелюбность выполняющейся команды. Опять же, читайте man-страницы для получения системно-специфичной информации.

Процесс загрузки, инициализация и завершение работы. Одна из наиболее мощных сторон Linux связана с ее открытым методом запуска и остановки операционной системы; Linux загружает определенные программы, используя их особую конфигурацию, что позволяет вам изменять эти конфигурации для контроля процесса загрузки и останавливать работу системы элегантно и организовано.

Помимо задач контроля загрузки и процесса завершения работы, открытый характер Linux облегчает в большинстве случаев точное определение источников проблем, связанных с запуском и остановкой системы. Понимание основ этого является весьма полезным для всех, кто использует Linux.

Множество систем Linux используют **lilo** (Linux LOader) для загрузки операционных систем. Мы обсудим только GRUB, т. к. он проще в использовании и является более гибким. Если вам необходима информация о **lilo**, обратитесь к man-страницам и HOWTO. Оба загрузчика поддерживают двойные загрузки; мы ссылаемся на HOWTO по этому вопросу для практических примеров и справочной информации.

Загрузочный процесс. Когда загружается компьютер x86, процессор просматривает конец системной памяти на обнаружение BIOS (Basic Input/Output System) и запускает его. (Программа BIOS записывается в постоянной памяти только для чтения, и всегда доступна для использования.) BIOS обеспечивает низкоуровневый интерфейс для периферийных устройств и контролирует первый шаг процесса загрузки.

BIOS тестирует систему, ищет и проверяет периферию, и затем ищет устройство, чтобы использовать его для загрузки системы. Обычно BIOS проверяет устройство для дискет (или на многих современных системах дисковод CD-ROM) на наличие загрузочного диска, если их там нет, то смотрит на жестком диске.

Последовательность устройств, используемых для загрузки, обычно контролируется индивидуальными для системы настройками BIOS. После того, как Linux установлен на жесткий диск, BIOS ищет главную загрузочную запись (Master Boot Record - MBR), начиная с первого сектора первого жесткого диска, загружает его содержимое в память и передает ему управление.

MBR содержит инструкции о том, как загрузить загрузчик GRUB (или LILO), использующий предварительно заданные операционные системы. MBR загружает загрузчик, который принимает на себя процесс (если загрузчик установлен в MBR).

В стандартной конфигурации Red Hat Linux GRUB использует настройки в MBR для отображения параметров загрузки в меню. Как только GRUB получил корректные инструкции для запуска операционной системы либо из его командной строки или из файла конфигурации, он находит необходимые загрузочные файлы и удаляется от управления машиной в данной операционной системе.

5.8.3. Особенности GRUB

Этот метод загрузки называется *прямой загрузкой*, поскольку инструкции используются для непосредственной загрузки операционной

системы, без какого-либо промежуточного кода между загрузчиком и основными файлами операционной системы (например, ядром). Процесс загрузки, используемый в других операционных системах, может незначительно отличаться от вышеуказанного. Например, операционные системы Microsoft (DOS и Windows) полностью переписывают MBR, когда они устанавливаются без учета какой-либо текущей конфигурации MBR. Все они стирают любую другую информацию, хранящуюся в MBR от других операционных систем, таких как Linux. Операционные системы Microsoft, также как и другие разные проприетарные операционные системы, загружаются с помощью метода цепной загрузки. При использовании этого метода, MBR указывает на первый сектор раздела, на котором находится операционная система, где находит специальные файлы, необходимые для фактической загрузки данной операционной системы.

GRUB поддерживает оба загрузочных метода, что позволяет использовать его практически с любой операционной системой, наиболее популярными файловыми системами, и практически любым жестким диском, который сможет распознать ваш BIOS.

GRUB содержит ряд других особенностей. Из них наиболее важные:

GRUB обеспечивает истинно командное, предшествующее ОС, окружение на машинах x86 для обеспечения максимальной гибкости при загрузке операционной системы с определенными параметрами или сбора информации о системе.

GRUB поддерживает режим логической адресации блоков (LBA), необходимый для доступа ко многим IDE и всем SCSI жестким дискам. До LBA жесткие диски могли столкнуться с 1024-цилиндровый пределом, когда BIOS после этого места не может найти файл.

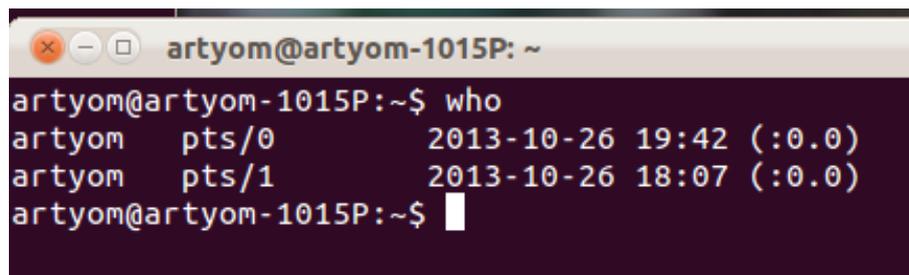
Конфигурационный файл GRUB читается с диска каждый раз при загрузке системы, освобождая вас от необходимости писать поверх MBR при каждом изменении параметров загрузки.

Init. Ядро, как только оно загружено, находит **init** в `sbin` и выполняет его. Когда **init** запускается, то становится родителем и прародителем всех процессов, которые запускаются автоматически в вашей системе Linux. Первое, что делает **init**, читает его файл инициализации `/etc/inittab`. Он инструктирует **init** прочитать первоначальный сценарий конфигурации

для окружения, в котором определены пути, запускается своппинг, проверяются файловые системы и так далее. В принципе, этот шаг позаботится обо всем, что ваша система должна сделать при своей инициализации: установка часов, инициализация последовательных портов и т.д.

Затем **init** продолжает читать файл `/etc/inittab`, который описывает, каким образом система должна подниматься на каждом уровне запуска и устанавливает *уровень выполнения* по умолчанию. *Уровень выполнения* определяет конфигурацию процессов. Все UNIX-подобные системы могут работать в различных конфигурациях, среди них есть однопользовательский режим, который называется уровнем выполнения 1 или S (или s). В этом режиме только системный администратор может подключиться к системе. Он используется для выполнения задач по обслуживанию без риска повреждения системы или пользовательских данных. Естественно, в такой конфигурации нам не следует предоставлять пользовательские службы, так что все они будут отключены. Другой уровень выполнения — это режим перезагрузки, или уровень 6, который завершает все запущенные службы, выполнив надлежащие процедуры, и перезагружает систему.

Используйте **who** для проверки, каким является ваш текущий уровень выполнения:



```

artyom@artyom-1015P:~$ who
artyom pts/0 2013-10-26 19:42 (:0.0)
artyom pts/1 2013-10-26 18:07 (:0.0)
artyom@artyom-1015P:~$

```

Рис.5.9. Вывод текущего уровня выполнения процессов

После того, как для вашей системы определяется уровень выполнения по умолчанию, **init** запускает все фоновые процессы, необходимые для запуска системы, просматривая соответствующую для данного уровня выполнения директорию `rc`. **init** выполняет каждый `kill`-сценарий (их файловые имена начинаются с K) с параметром `stop`. Затем он проходит все запускающие сценарии (их имена файлов начинаются с S) в соответствующем уровню выполнения каталоге, с тем, чтобы все

службы и приложения запустились правильно. На самом деле вы вручную можете выполнить те же сценарии после того, как система завершит загрузку, командами подобными `/etc/init.d/httpd stop` или `service httpd stop`, войдя в систему с правами *root*; в этом случае останавливается веб-сервер.

Затем **init** проходит через уровень выполнения, чтобы задать уровень по умолчанию, сценарий `/etc/inittab` разветвляется процессом **getty** для каждой виртуальной консоли (приглашение на вход в текстовом режиме). **getty** открывает строки `tty`, устанавливает их режимы, печатает строку входа, получает имя пользователя, и затем начинает процесс входа для данного пользователя. Это позволяет пользователям представиться системе и использовать ее. По умолчанию, большинство систем предлагают 6 виртуальных консолей, но, как вы можете увидеть из файла `inittab`, это можно изменить.

`/etc/inittab` может также указать **init**, как тот должен обрабатывать пользовательское нажатие **Ctrl + Alt + Delete** в консоли. Поскольку система должна надлежащим образом выключаться и перезапускаться, а не прекращать немедленно работу, **init** сообщается выполнить команду `/sbin/shutdown -t3 -r now`, например, когда пользователь нажимает эти клавиши. Кроме того, `/etc/inittab` устанавливает, что **init** должен делать в случае отключения электропитания, если у вашей системы есть `UPS unit`, прикрепленный к ней.

В большинстве систем, основанных на RPM, графический экран для входа запускается на уровне выполнения 5, где `/etc/inittab` запускает скрипт под названием `/etc/X11/prefdm`. Скрипт `prefdm` преимущественно работает в графическом менеджере X, основываясь на содержимом каталога `/etc/sysconfig/desktop`. Это, как правило, **gdm**, если вы работаете на GNOME или **kdm**, если вы используете KDE, но они могут быть смешаны, кроме того, есть **xdm**, который предоставляется при стандартной установке X.

Уровни выполнения Init.

Идея функционирования различных служб на разных уровнях выполнения по существу вращается вокруг того факта, что различные системы могут быть использованы по-разному. Некоторые службы не могут быть использованы, пока система находится в определенном

состоянии или режиме, таком как быть готовой к более чем одному пользователю или доступной сети.

Бывают периоды, в которые вы можете работать с системой в низкоуровневом режиме. Примеры: решение проблем испорченного диска на уровне выполнения 1, так чтобы у других пользователей не было возможности быть в системе, или оставить сервер на уровне выполнения 3 (без работающей X сессии). В этих случаях, функционирование сервисов, которые зависят от высокоуровневого режима системы, не имеет смысла, т. к. они не будут работать правильно в любом случае. Уже имеющимся службам назначается их запуск, когда будет достигнут определенный уровень выполнения; вы обеспечиваете последовательный запуск процессов, и вы можете быстро изменить режим машины, не беспокоясь о том, что вручную придется запускать и останавливать службы.

Доступные уровни выполнения, как правило, описаны в файле `/etc/inittab`,

Когда вы работаете в текстовом режиме, т. к. не был предоставлен экран графического входа на вашей машине, обычно вы можете переключаться на консоль

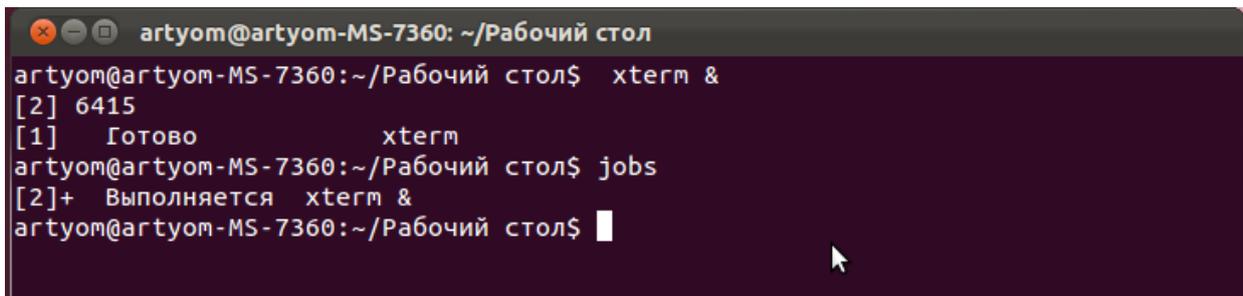
Если это не так, проверить текущий режим работы с помощью команды **who** -г. Если он установлен на что-нибудь другое, чем есть оригинальные установки по умолчанию в `/etc/inittab`, то вероятно это может быть причина того, что система не запускается в графическом режиме по умолчанию. В таком случае обратитесь к системному администратору или прочитайте **man** `init`. Заметим, что переключение режимов работы осуществляется преимущественно с использованием команды **telinit**; переключение из текстовой в графическую консоль или наоборот не подразумевает изменение режима выполнения.

Мы стараемся в данном руководстве обсуждать режимы выполнения, скрипты и конфигурации в общих чертах. Существует множество вариаций. Например, Gentoo Linux хранит скрипты в уровнях `/etc/gun`. Другие системы могут запускаться через первый нижний режим выполнения и выполнить все сценарии в нем до прибытия конечного режима выполнения и исполнения его сценариев. Обратитесь к системной документации для получения дополнительной информации. Вы также можете прочитать скрипты, которые относятся к `/etc/inittab`, чтобы лучше понять, что происходит в вашей системе.

5.9. Порядок выполнения работы

1. Запуск процесса в фоновом режиме

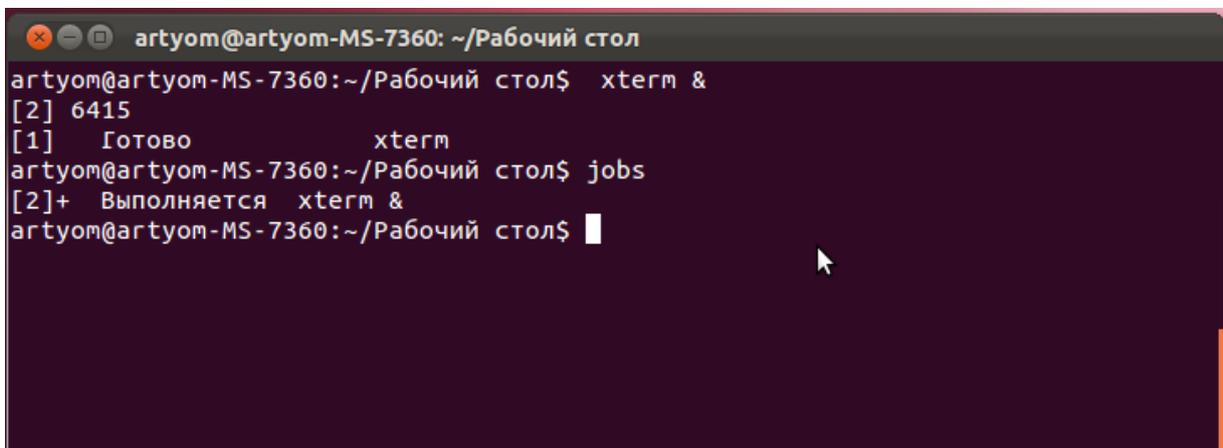
Если вы запускаете какой-то процесс путем запуска программы из командной строки, то обычно процесс запускается, как говорят, "на переднем плане". Это значит, что процесс "привязывается" к терминалу, с которого он запущен, воспринимая ввод с этого терминала и осуществляя на него вывод. Но можно запустить процесс в фоновом режиме, когда он не связан с терминалом. Для запуска процесса в фоновом режиме в конце командной строки запуска программы добавляют символ `&`.



```
artyom@artyom-MS-7360: ~/Рабочий стол
artyom@artyom-MS-7360:~/Рабочий стол$ xterm &
[2] 6415
[1] Готово          xterm
artyom@artyom-MS-7360:~/Рабочий стол$ jobs
[2]+  Выполняется xterm &
artyom@artyom-MS-7360:~/Рабочий стол$ █
```

Рис.5.10. Запуск процесса в фоновом режиме

Все задания, находящиеся в фоновом режиме можно просмотреть с помощью команды `jobs`.



```
artyom@artyom-MS-7360: ~/Рабочий стол
artyom@artyom-MS-7360:~/Рабочий стол$ xterm &
[2] 6415
[1] Готово          xterm
artyom@artyom-MS-7360:~/Рабочий стол$ jobs
[2]+  Выполняется xterm &
artyom@artyom-MS-7360:~/Рабочий стол$ █
```

Рис.5.11. Просмотр задач

2. Просмотр информации о процессе

Команда `ps` является одним из инструментов для визуализации процессов. Эта команда имеет несколько опций, которые могут быть объединены для отображения различных свойств процесса.

При отсутствии конкретных опций `ps` выдает только информацию о текущей оболочке и возможных процессах:

```

artyom@artyom-MS-7360: ~/Рабочий стол
artyom@artyom-MS-7360:~/Рабочий стол$ ps
  PID TTY          TIME CMD
 6196 pts/2        00:00:00 bash
 6415 pts/2        00:00:00 xterm
 6586 pts/2        00:00:00 ps
artyom@artyom-MS-7360:~/Рабочий стол$

```

Рис.5.12. Просмотр информации о процессе

Поскольку это не дает достаточно информации - как правило, по меньшей мере, сотня процессов запущены в вашей системе – выберем отдельные процессы из списка всех процессов, используя команду **grep** в конвейере:

```

artyom@artyom-MS-7360: ~
artyom@artyom-MS-7360:~$ ps -ef | grep mozilla
artyom 9810 5079 0 23:32 pts/1 00:00:00 grep --color=auto mozilla
artyom@artyom-MS-7360:~$

```

Рис.5.13. Выбор процессов из списка

Следующий пример показывает все процессы с процессорным именем **bash**, наиболее распространенной оболочкой в системах Linux:

```

artyom@artyom-MS-7360: ~/Рабочий стол
artyom@artyom-MS-7360:~/Рабочий стол$ ps auxw | grep bash
artyom 2954 0.0 0.1 9232 4016 pts/0 Ss+ Oct23 0:00 bash
artyom 5079 0.0 0.1 9232 4012 pts/1 Ss+ 22:41 0:00 bash
artyom 5377 0.0 0.0 6640 1388 ? S 22:46 0:00 bash /usr/bin/banshee
--redirect-log --play-enqueued /media/2057-3F77/Muse - The 2nd Law [2012)/12. The 2nd
Law Unsustainable.mp3
artyom 6196 0.0 0.1 9236 4160 pts/2 Ss 23:33 0:00 /bin/bash
artyom 6417 0.0 0.1 9232 4020 pts/3 Ss+ 23:41 0:00 bash
artyom 6621 0.3 0.1 9236 4148 pts/2 S 23:53 0:00 bash
artyom 6680 0.0 0.0 5716 772 pts/2 S+ 23:54 0:00 grep --color=auto bas
h

```

Рис.5.14 – Сортировка процессов по имени.

В этих случаях команда **grep** находит строки, содержащие строку *bash*, которые множество раз отображаются также в системах, которые имеют много простоев. Если вы не хотите, чтобы это произошло, используйте команду **pgrep**.

Bash-оболочки представляют собой особый случай: этот

процессорный список также показывает, какие из них являются оболочками входа (где вы должны ввести ваше имя пользователя и пароль, например, когда вы входите в текстовом режиме или же совершаете удаленный вход, в отличие от оболочек не требующих логина, например, нажав значок окна терминала). Такие оболочки, требующие логин, начинаются с дефиса (-).

Дополнительная информация может быть найдена как обычно: **ps --help** или **man ps**. GNU **ps** поддерживает различные форматы опций;

Команда **ps** выдает только сиюминутное состояние активных процессов, это одномоментный срез. Программа **top** дает более точный обзор путем обновления результатов, определенных **ps** (с множеством опций), один раз в 5 секунд, создавая новый список процессов, что периодически вызывает большие нагрузки, поэтому используется подключение дополнительной информации о файле подкачки и состоянии процессора; из файла `proc` системы:

```

artyom@artyom-MS-7360: ~/Рабочий стол
top - 23:56:57 up 1 day, 1:09, 4 users, load average: 1.63, 1.35, 1.33
Tasks: 179 total, 1 running, 177 sleeping, 0 stopped, 1 zombie
Cpu(s): 5.4%us, 0.4%sy, 0.0%ni, 93.3%id, 0.8%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 3098128k total, 2870864k used, 227264k free, 371440k buffers
Swap: 3143676k total, 176k used, 3143500k free, 807272k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 2289 artyom    20   0 579m 351m 25m  S   36  11.6 104:32.28 chrome
 2709 artyom    20   0 599m 315m 30m  S   4  10.4  25:11.37 chrome
 3060 artyom    20   0 196m  48m 21m  S   4   1.6   0:15.79 chrome
 1617 artyom    20   0 7040 3580 928  S   2   0.1   4:20.08 dbus-daemon
 1653 artyom    20   0 323m 161m 40m  S   2   5.3   8:41.22 compiz
 2072 artyom    20   0 467m 122m 39m  S   2   4.0   7:30.60 chrome
 2659 artyom    20   0 294m 139m 27m  S   2   4.6   8:09.86 chrome
   1 root      20   0 3340 1824 1244 S   0   0.1   0:00.56 init
   2 root      20   0   0   0   0   S   0   0.0   0:00.00 kthreadd
   3 root      20   0   0   0   0   S   0   0.0   0:06.04 ksoftirqd/0
   5 root      20   0   0   0   0   S   0   0.0   0:00.22 kworker/u:0
   6 root      20   0   0   0   0   S   0   0.0   0:00.00 migration/0
   7 root      20   0   0   0   0   S   0   0.0   0:00.00 migration/1
   9 root      20   0   0   0   0   S   0   0.0   0:04.21 ksoftirqd/1
  11 root     0 -20   0   0   0   S   0   0.0   0:00.00 cpuset
  12 root     0 -20   0   0   0   S   0   0.0   0:00.00 khelper
  13 root     0 -20   0   0   0   S   0   0.0   0:00.00 netns
  14 root     20   0   0   0   0   S   0   0.0   0:00.27 kworker/u:1
  15 root     20   0   0   0   0   S   0   0.0   0:00.10 sync_supers
  16 root     20   0   0   0   0   S   0   0.0   0:00.00 bdi-default
  17 root     0 -20   0   0   0   S   0   0.0   0:00.00 kintegrityd
  18 root     0 -20   0   0   0   S   0   0.0   0:00.00 kblockd
  19 root     0 -20   0   0   0   S   0   0.0   0:00.00 ata_sff
  20 root     20   0   0   0   0   S   0   0.0   0:00.06 khubb
  21 root     0 -20   0   0   0   S   0   0.0   0:00.00 md
  22 root     20   0   0   0   0   S   0   0.0   0:00.03 khungtaskd
  23 root     20   0   0   0   0   S   0   0.0   0:00.13 kswapd0
  24 root     25   5   0   0   0   S   0   0.0   0:00.00 ksmd
  25 root     39  19   0   0   0   S   0   0.0   0:00.00 khugepaged
  26 root     20   0   0   0   0   S   0   0.0   0:00.04 fsnotify_mark
  27 root     20   0   0   0   0   S   0   0.0   0:00.00 ecryptfs-kthrea
  28 root     0 -20   0   0   0   S   0   0.0   0:00.00 crypto
  36 root     0 -20   0   0   0   S   0   0.0   0:00.00 kthrotld
  38 root     20   0   0   0   0   S   0   0.0   0:00.00 scsi_ah_0
  39 root     20   0   0   0   0   S   0   0.0   0:00.00 scsi_ah_1

```

Рис.5.15. Список процессов

На рис.5.15 выводится таблица, характеризующая отдельные процессы. Число строк, отображаемых в этой таблице, определяется размером окна. Содержимое окна обновляется каждые 3 секунды. Нажатие клавиши **h** выводит справку по командам, которые позволяют

изменить формат отображаемых данных и управлять некоторыми параметрами процессов в системе:

s или **d** - изменить интервал обновления окна.

z - цветное/черно-белое отображение

n или **#** - количество отображаемых процессов

Можно отсортировать список процессов по использованию процессора (**P**), памяти (**M**), времени выполнения (**T**) и т. п.

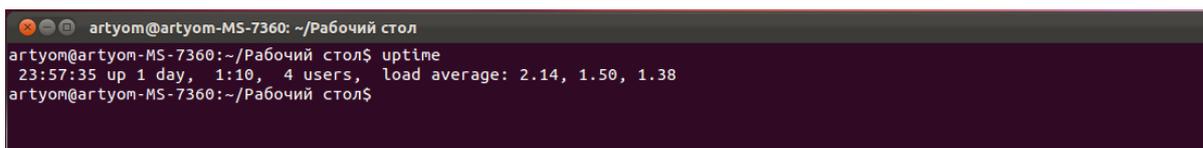
Однако, есть и более полезные команды, которые позволяют управлять процессами в интерактивном режиме:

k - снять (kill) задачу. По этой команде будет запрошен идентификатор процесса (PID), после ввода которого, процесс будет завершен.

r - изменить текущий приоритет задачи (renice). Как и при выполнении предыдущей команды, будет запрошен PID, и затем, новое значение приоритета (отображается в колонке NI). Диапазон значений приоритетов - от минус 20 (наивысший) до плюс 19. Отрицательные значения и 0 может устанавливать только пользователь root.

Для выхода из top нажмите **q**

Первая строка **top** содержит такую же информацию, которая отображается командой **uptime**:



```
artyom@artyom-MS-7360: ~/Рабочий стол
artyom@artyom-MS-7360:~/Рабочий стол$ uptime
23:57:35 up 1 day, 1:10, 4 users, load average: 2.14, 1.50, 1.38
artyom@artyom-MS-7360:~/Рабочий стол$
```

Рис.5.16. Пример ввода команды «uptime»

Данных для этих программ хранятся среди других в /var/run/utmp (информация о текущих подключенных пользователях) и в виртуальной файловой системе /proc, например, /proc/loadavg (обычная загрузочная информация). Есть разные виды графических приложений для просмотра этой информации, такие как Gnome System Monitor и *lavaps*.

3. Информация о процессах через system monitor

Информацию о процессах так же можно вывести с помощью утилиты system monitor:

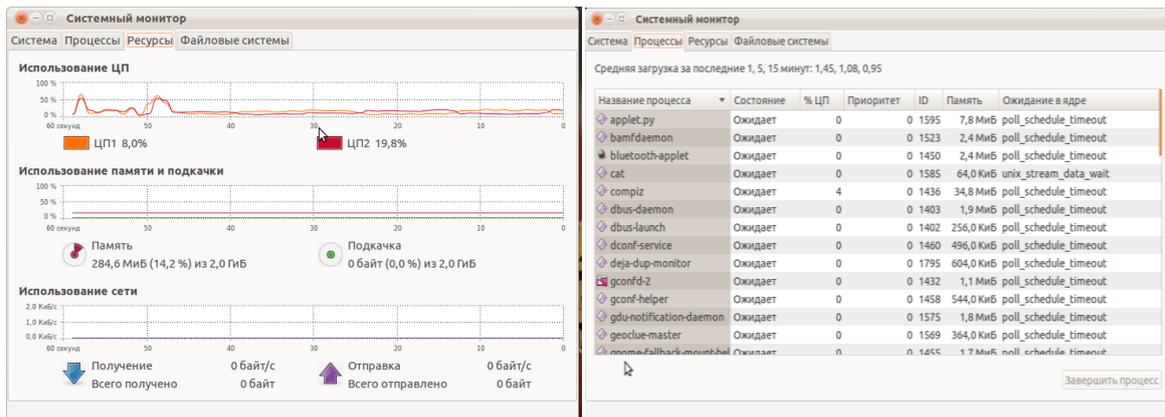


Рис.5.17. Работа утилиты «system monitor»

4. Завершение процесса

Для того чтобы послать определенный сигнал выбранному процессу, нужно ввести:

- `kill < -номерсигнала > номер PID` выбранного процесса:

```

artyom@artyom-MS-7360: ~
artyom@artyom-MS-7360:~$ ps -ef | grep mozilla
artyom  9866  5079  0 23:34 pts/1    00:00:00 grep --color=auto mozilla
artyom@artyom-MS-7360:~$ kill -15 5079
artyom@artyom-MS-7360:~$

```

Рис.5.18. Завершение процесса

5. Изменение приоритетов - команды nice и renice

При запуске процесса, его приоритет задается значением родительского процесса, например, терминала xterm или командной оболочки shell. Команда nice позволяет запустить процесс с приоритетом, равным сумме родительского (например 8) и числа, задаваемого в качестве опции команды nice:

- `nice -< число > команда`
- `nice -4 gvim` - запустит gvim с приоритетом = 8+4=12

Команда renice используется для изменения приоритета уже запущенного процесса:

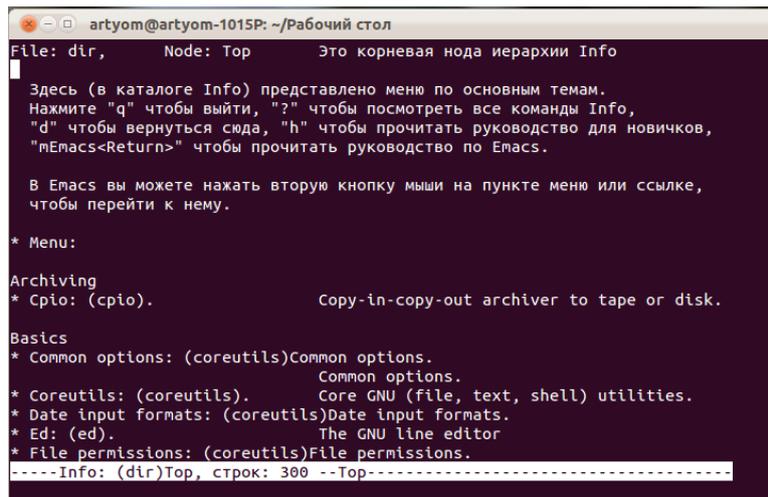
- `renice < число > -p PID`
- `renice 4 -p 5848` - установить значение приоритета для процесса Gvim (PID=5848) равным 4.

После выполнения, `renice` выдаст строку результата: `5848: old priority 12, new priority 4` - старый приоритет 12, новый приоритет 4

Задание нулевого и отрицательного числа приоритета возможно только с правами `root`.

6. Команда `sleep`

Info-страница о сне, вероятно, одна из самых коротких.



```

File: dir,      Node: Top      Это корневая нода иерархии Info
|
Здесь (в каталоге Info) представлено меню по основным темам.
Нажмите "q" чтобы выйти, "?" чтобы посмотреть все команды Info,
"d" чтобы вернуться сюда, "h" чтобы прочитать руководство для новичков,
"mEmacs<Return>" чтобы прочитать руководство по Emacs.

В Emacs вы можете нажать вторую кнопку мыши на пункте меню или ссылке,
чтобы перейти к нему.

* Menu:

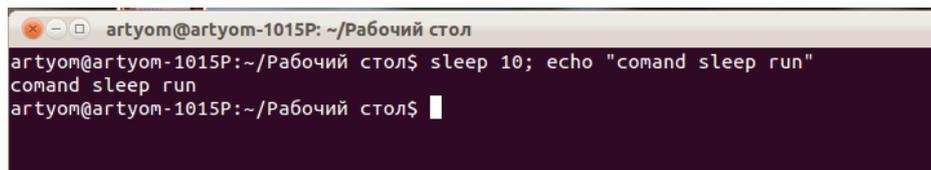
Archiving
* Cpio: (cpio).          Copy-in-copy-out archiver to tape or disk.

Basics
* Common options: (coreutils)Common options.
                        Common options.
* Coreutils: (coreutils).  Core GNU (file, text, shell) utilities.
* Date input formats: (coreutils)Date input formats.
* Ed: (ed).                The GNU line editor
* File permissions: (coreutils)File permissions.
-----Info: (dir)Top, строк: 300 --Top-----

```

Рис. 5.19 – Info-страница о сне.

`sleep` только ждет. По умолчанию время ожидания выражается в секундах.



```

artyom@artyom-1015P: ~/Рабочий стол
artyom@artyom-1015P:~/Рабочий стол$ sleep 10; echo "comand sleep run"
comand sleep run
artyom@artyom-1015P:~/Рабочий стол$

```

Рис. 5.20 – Запуск «Comand sleep run».

7. Команда `at`

Для запуска процесса в определенное время используется команда «`at`». После её ввода на экране видим приглашение для ввода команд задания - `at>`. Далее вводим команду, которую необходимо выполнить и нажимаем `Enter`. После ввода команды, которую мы хотим выполнить в назначенное время, нажимаем комбинацию клавиш `Ctrl+D`. На экране, после назначенной нами команды, появится еще одна - и будут указаны номер и дата задания.

```
artyom@artyom-1015P:~$ at 23:00
warning: commands will be executed using /bin/sh
at> gvim<EOT>
job 1 at Tue Oct  1 23:00:00 2013
artyom@artyom-1015P:~$ █
```

Рис.5.21. Запуск команды «at»

Для указания времени можно использовать значения согласно спецификации POSIX.2. Они представляют из себя вид в стандарте НН:ММ (где НН - часы, а ММ - минуты). Можно также использовать расширенную спецификацию:

- at 7am 10202008 - 20-го октября 2013 года в 7 утра
- at 12:00 20.10.2008 - 20 октября 2013 года в 12 дня
- at 10pm today - сегодня в 10 вечера
- at 3am tomorrow - завтра (считается от сегодняшней даты) в 3 часа утра

8. Проверка и изменение уровня выполнения

Для ввода ниже перечисленных команд необходимо иметь права root

Проверка уровня выполнения осуществляется с помощью команды **runlevel**:

```
artyom@artyom-1015P: ~
artyom@artyom-1015P:~$ runlevel
N 2
artyom@artyom-1015P:~$ █
```

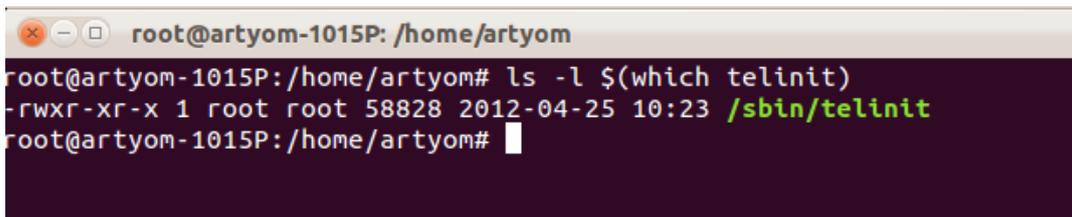
Рис.5.22. Запуск команды «runlevel»

Чтобы перейти на другой уровень выполнения используйте команду **telinit**:

```
root@artyom-1015P: /home/artyom
artyom@artyom-1015P:~$ su
Пароль:
root@artyom-1015P:/home/artyom# telinit 5
root@artyom-1015P:/home/artyom# █
```

Рис.5.23. Запуск команды «telinit»

Если с помощью команды `ls` вы отобразите подробные сведения о команде `telinit`, то увидите, что на самом деле она является символической ссылкой на команду `init`:



```

root@artyom-1015P: /home/artyom
root@artyom-1015P:/home/artyom# ls -l $(which telinit)
-rwxr-xr-x 1 root root 58828 2012-04-25 10:23 /sbin/telinit
root@artyom-1015P:/home/artyom#

```

Рис.5.24. Свойства команды «telinit»

9. Shutdown – выключение

UNIX была создана как система, которая не останавливается, но если вам это действительно надо, используйте команду **shutdown**. Опция `-h` будет остановить систему, а `-r` будет перезагружать ее.

Команды **reboot** и **halt** теперь способны вызывать **shutdown**, когда система находится в режимах выполнения 1-5, и таким образом обеспечить надлежащее завершение ее работы, но это плохая привычка, поскольку не во всех версиях UNIX/Linux есть такая возможность.

Пока ваш компьютер сам не выключится, вы не должны выключать его, пока не увидите сообщение с указанием, что система остановилась или все процессы завершились, т. к. система должна отключить все разделы. Нетерпеливость может привести к потере данных.

5.10. Задания для самостоятельной работы

1. Запустите Linux.
2. Запустите процесс в фоновом режиме, просмотрите все процессы находящиеся в фоновом режиме, выводите из него.
3. Выведите информацию о работающих процессах, с различным количеством столбцов и процессов. Сортируйте эту информацию по столбцам.
4. Запишите выходной поток процесса в файл.
5. Просмотрите информацию о процессах с помощью системного монитора (`system monitor`).

6. Выведите информацию о работающих процессах. Пошлите сигнал на прекращение процесса по ID и по имени.
7. Подготовьте задание на запуск процессов в определённое время, в определённую дату и с определённым приоритетом (в очереди на запуск), покажите этот список, удалите задание из списка
8. Запустите процесс с конкретным приоритетом.
9. Измените приоритет конкретного процесса
10. Узнайте свой текущий уровень выполнения. Перейдите с вашего уровня выполнения по умолчанию на уровень 1 и обратно.
11. Задайте **at** задание, которое копирует все файлы в домашней директории в /var/tmp в течение получаса. Вы можете создать подкаталог в /var/tmp.
12. Используйте **sleep** для создания напоминания.

5.11. Содержание отчета

1. Цель работы.
2. Результаты выполнения специального задания.
3. Выводы по проделанной работе.

5.12. Контрольные вопросы

1. Перечислите свойства процессов.
2. Перечислите часто используемые сигналы.
3. Опишите действия системы по завершению процесса.
4. Дайте определение понятию приоритет.
5. Что такое GRUB?
6. Назовите основные особенности GRUB.
7. Особенности работы процесса «Init».

ЛАБОРАТОРНАЯ РАБОТА №6 «Освоение перспективных типов ОС (Live-CD/DVD ОС, Web-ОС)»

Данная лабораторная работа является ознакомительной, связана с изучением особенностей и возможностей перспективных типов ОС и практическим освоением процессов их запуска, настройки и использования. Она описывает новые возможности запуска ОС не только с локального жесткого диска компьютера, но и с других накопителей, исключая процесс их инсталляции, а также создания и поддержки защищенной рабочей среды пользователя.

6.1. Цель работы

Изучить возможности и особенности перспективных типов ОС (Live-CD/DVD-ОС, Web-ОС), а также оригинальные виды их запуска, создания и поддержки защищенной индивидуальной и групповой рабочей среды.

Получить навыки практической работы в среде Live-CD/DVD ОС, Web-ОС.

6.2. Теоретическая часть

6.2.1. Live-CD ОС

Live-CD – ОС, загружающаяся со сменного носителя (CD, DVD, USB-накопителя) и не требующая для своего функционирования установки на жёсткий диск. Этим же понятием обозначают и носители с такими ОС (иногда различают Live CD, Live DVD и Live USB).

Live-CD позволяют быстро начать работу с компьютером, избегая процесса установки ОС в постоянную память (винчестер). Запуск Live-CD занимает обычно несколько минут, в то время как инсталляция и первый запуск традиционных ОС зачастую требуют от одного до нескольких часов. Большинство используемых Live-CD созданы на основе ОС GNU/Linux, существуют также Live-CD на основе других ОС семейства Unix: FreeBSD, OpenBSD, NetBSD, OpenSolaris и др.

До широкого распространения CD в качестве загрузочных носителей использовались загрузочные дискеты, обычно содержавшие компактную версию ОС и ограниченный набор необходимых приложений. Программисты приспособили CD, изначально созданные для хранения аудиоданных, для хранения и распространения больших объёмов

информации. Позже, с увеличением объёмов оперативной памяти, стало удобным загружать компьютер напрямую с CD для того, чтобы установить ОС на жёсткий диск.

Первые приводы оптических дисков для персональных компьютеров были слишком медленными для работы сложных ОС. Часто в компьютерах попросту отсутствовала возможность загрузки с CD. Когда ОС стали распространяться на CD, использовалась загрузочная дискета или загрузка с самого компакт-диска, но только для того, чтобы установить ОС на жёсткий диск. Первым Live-CD была FM Towns OS, впервые выпущенная в 1989 г.

Некоторые Live-CD созданы на основе ОС семейства Unix: GNU/Linux, OpenBSD, OpenSolaris. Большинство же используемых Live-CD созданы на основе Linux. Использование Windows Live-CD является нарушением авторских прав, да и возможности «живых» дисков на базе Windows минимальны.

Применение Live-CD.

Чаще всего Live-CD используются для ознакомления с возможностями новых ОС. Многие Live-CD также предоставляют возможность установки на жёсткий диск. Таким образом, можно быстро запустить и попробовать в использовании ту или иную ОС, и, если система понравится, установить её уже у качестве основной для повседневной работы.

Другое применение Live-CD – восстановление данных и/или работоспособности основной ОС после сбоя. Live-CD позволяют работать с компьютером в тех случаях, когда основная ОС не работоспособна.

Преимуществом Live-CD является удобство при перемещении. Например, вы можете взять Live-CD с собой, если вам нужно некоторое время работать на чужом компьютере. Таким образом, вы можете на любом компьютере, имеющем CD-ROM-дисковод, быстро получить доступ к привычной ОС и не бояться навредить чужой системе и чужим данным во время своей работы.

Некоторые Live-CD позволяют использовать полезную возможность – полную загрузку содержимого CD в оперативную память компьютера. При использовании такой возможности достигается высокая скорость

работы ОС и всех входящих в её состав программ, однако это возможно только при небольшом размере системы и достаточно большом объёме оперативной памяти.

Существуют программы, позволяющие создать собственный Live-CD на основе уже существующих систем.

Недостатки Live-CD:

- **низкая скорость.** Хотя Live-CD позволяют сэкономить много времени при первом запуске ОС, при следующих запусках традиционные ОС, как правило, значительно быстрее. Это обусловлено сравнительно небольшой скоростью работы CD-ROM приводов, повышенным расходом ОП и необходимостью настраивать Live-CD в соответствии с конфигурацией используемого оборудования;
- **проблема хранения настроек и данных.** Поскольку произвольная запись на CD невозможна, пользователи Live-CD сталкиваются с проблемой хранения своих данных и настроек. Для этого приходится использовать различные дополнительные носители данных, например Flash-накопители или сетевые диски. Многие Live-CD предоставляют специальные программы для организации хранения настроек и данных на подобных носителях. Также некоторые Live-CD позволяют дописывать свои данные на тот же CD с помощью так называемого мультисессионного режима записи;
- **дефекты CD.** CD подвержены различным повреждениям – царапинам, а также трещинам (вплоть до полного разрушения) и отслоению записываемого слоя. Надёжность CD сильно зависит как от качества изготовления, так и от бережного обращения при эксплуатации.

Чтобы загрузиться с Live CD диска необходимо в настройках BIOS в разделе Boot Device указать первым устройством загрузки CD/DVD привод или USB порт, а на втором месте оставить жесткий диск. При включении компьютера будет предложено загрузиться с CD/DVD или с жесткого диска, необходимо выбрать внешний носитель и дождаться загрузки с Live CD.

Live-USB – подключаемый по шине USB носитель данных, содержащий ОС с возможностью ее загрузки. Обычно в качестве Live-USB используются флеш-накопители, в силу наибольшей компактности,

однако возможно применение других носителей. По своей сути Live-USB близок к Live-CD, зачастую отличается от него только носителем информации. В таком случае, наследуемой особенностью Live-USB является использование специальных версий ОС, которые обычно рассчитаны на возможность запуска на максимально широком спектре совместимых компьютеров. В этих версиях зачастую отсутствуют или существенно ограничены возможности установки требуемых пользователю приложений, настройки и сохранения параметров системы. Однако нередко возможности ОС, блокируемые на Live-CD, при использовании в варианте Live-USB становятся доступными. Кроме того, скорость загрузки системы и приложений с любого современного USB-накопителя выше, чем с Live-CD, а сам носитель может быть использован для оперативной записи созданных в процессе работы данных без подключения дополнительного накопителя. Поэтому Live-USB в ряде случаев предпочтительнее Live CD.

Создание Live-CD флеш-накопителя.

Создания Live-CD носителя с ОС позволяет работать с ОС со всеми ее компонентами, при этом не нужно отдельно ее устанавливать на диски и прочее. Например, можно использовать образ Kali Linux и сделать ее загрузаемой с флеш-накопителя (Рис.6.1).

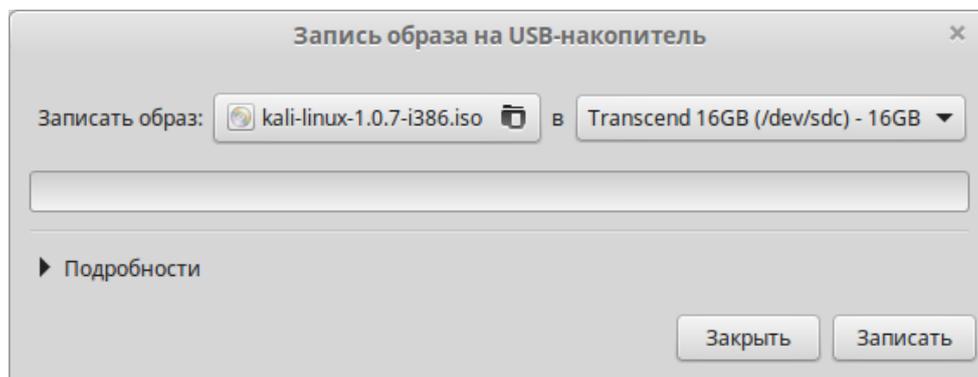


Рис.6.1. Создание загрузочного флеш-накопителя

Live-CD ОС, заслуживающие внимания:

- **Knoppix** – Live-CD дистрибутив ОС GNU/Linux. Достаточно большой по размеру Linux-дистрибутив (примерно 4Гб), очень много самого разнообразного программного обеспечения на все случаи жизни. Это один из первых Live-CD дистрибутивов, построен на основе Debian. Knoppix можно загрузать с DVD и Flash дисков;

- **Frenzy** (<http://www.livecd.su/livecd/livecd5.php>) – ОС, основанная на ядре BSD. Это «портативный инструмент системного администратора», Live-CD на базе ОС FreeBSD, загрузившись с которого, администратор получает полностью работоспособную систему с набором программного обеспечения для настройки, проверки и анализа сети, тестирования аппаратуры компьютера и ряда других задач;
- **Ubuntu – Linux** (<http://ubuntu.ru/get>);
- **Parrot Linux** (ParrotSec) (<http://pingvinus.ru/distribution/parrot-linux>) – специализированный дистрибутив Linux, основанный на Debian и предназначенный для тестирования проникновения и защиты данных. Обеспечивает конфиденциальность при работе в сети. Для скачивания доступны 32- и 64-битные версии дистрибутива (ISO образы). Поддерживается LiveCD режим. Если в LiveCD-режиме заблокируется экран, для разблокировки потребуется ввести пароль:toor;
- **ReactOS** (<http://pingvinus.ru/distribution/reactos>) – свободная ОС с открытым исходным кодом, совместимая с драйверами и программами Windows;
- **Antergos** (<http://pingvinus.ru/distribution/antergos>) – дистрибутив Linux, основанный на Arch Linux. Прежнее название Cinnarch. Дистрибутив полностью готов к работе после установки и не требует особых навыков, в отличие от Arch Linux;
- **Solus** (<http://pingvinus.ru/distribution/solus>) – красивый оригинальный дистрибутив. Использует собственную среду рабочего стола Budgie. Имеет современный и удобный рабочий стол;
- **Xubuntu** (<http://pingvinus.ru/distribution/xubuntu>) – дистрибутив, основанный на Ubuntu, использующий среду рабочего стола Xfce. Дистрибутив имеет хороший набор предустановленного программного обеспечения, качественный интерфейс и может работать на относительно слабых компьютерах;
- **Android x86** (<http://pingvinus.ru/distribution/android-x86>) – дистрибутив ОС Android, который предназначен для запуска на персональных компьютерах с процессорами Intel x86 и AMD;
- **Kubuntu** (<http://pingvinus.ru/distribution/kubuntu>) – официальная версия дистрибутива Ubuntu, использующая рабочее окружение KDE.

Дистрибутив использует репозитории Ubuntu, обновляется синхронно с Ubuntu;

- **Netrunner** (<http://pingvinus.ru/distribution/netrunner>) – дистрибутив Linux для персональных компьютеров, использующий среду рабочего стола Plasma Desktop (KDE). Дистрибутив имеет две версии, одна основана на Kubuntu, другая – на **ArchLinux (Manjaro)**;
- **Kali Linux** (ранее BackTrack Linux) (<http://pingvinus.ru/distribution/kali-linux>) – специальный дистрибутив, основанный на **Debian**, содержащий в себе множество программ и утилит, связанных с проверкой безопасности и взломом. Дистрибутив используется для проверки безопасности и поиска уязвимости систем;
- **Deepin Linux** (<http://pingvinus.ru/distribution/deepin-linux>) – красивый и стильный дистрибутив, основанный на **Ubuntu**. Включает несколько качественных собственных приложений. Имеет элегантный и проработанный интерфейс, основанный на технологии HTML5;
- **Simply Linux** (<http://simplylinux.ru/index.php/download>) – свободная ОС, доступна для скачивания и использования без ограничений. Образ гибридный и может использоваться для работы без установки на жесткий диск (Live-режим); установки на жесткий диск из Live-режима DVD; работы с Flash-носителя без установки на жесткий диск; установки на жесткий диск из Live-режима Flash-носителя;
- **Tails** (<http://pingvinus.ru/distribution/tails>) – Live-дистрибутив Linux, основанный на Debian, и созданный для конфиденциального и анонимного использования сети (шифрование, анонимизация и безопасность перехвата/прослушки трафика). Содержит набор программ и утилит, которые позволяют полностью анонимно выходить в интернет и пользоваться другими сервисами и службами (почта, im-сообщения и прочее). Для достижения приватности дистрибутив использует Tor-сеть;
- **Manjaro** (<http://pingvinus.ru/distribution/manjaro>) – дистрибутив Linux, основанный на Arch Linux, имеет простой графический инсталлятор и «предустановленное» графическое окружение (XFCE, Openbox и другие). Manjaro намного проще в установке по сравнению с Arch Linux. Дистрибутив использует свои собственные репозитории, но при

этом совместим с AUR (Arch User Repositories).

6.2.2. Web-ОС

Термин «Web-ОС» имеет множество альтернативных наименований: WebOS, «операционная система в вебе», Internet OS, браузерная ОС, облачная ОС, виртуальная ОС, онлайн ОС).

С каждым годом обычные настольные приложения все больше и больше вытесняются Web-сервисами. Сегодня уже можно без проблем найти онлайн-графический редактор, конвертер, IM-клиент для быстрого обмена сообщениями, средства для распознавания текста, средства для создания-распаковки архивов и многие другие онлайн-замены обычным утилитам. Но работать с большим числом сервисов не всегда удобно, ведь ссылку на нужный ресурс легко можно потерять и не вспомнить в самый ответственный момент.

Отличная альтернатива отдельным сервисам – виртуальные операционные системы или Web-ОС. Такие ресурсы представляют собой уже готовую рабочую среду с набором наиболее востребованных приложений. Зарегистрировавшись на сервисе, пользователь получает доступ к полностью рабочей системе, в которой можно хранить файлы и обмениваться ими с другими людьми, работать над документами, смотреть фильмы, слушать музыку и т.д. Таким образом, все нужные инструменты оказываются в одном месте, и все, что требуется от пользователя – помнить данные для входа в свою учетную запись.

Работа с виртуальными системами может быть удобна в самых разных случаях. Такие сервисы могут использоваться для онлайн-хранения файлов, к которым необходимо получать доступ из разных мест. Поскольку вся работа с Web-ОС выполняется в браузере, все данные, которые в них хранятся, могут быть доступны вне зависимости от того, на какой платформе запущен ваш браузер. Web-ОС также хороши тем, что могут служить личным хранилищем для любой важной или конфиденциальной информации, которую вы не хотите держать на рабочем или домашнем компьютере, боясь, что она попадет на глаза другим людям. Такой способ ограничения доступа к своим файлам гораздо надежнее, чем флеш-накопитель, который может потеряться или выйти из строя.

Web-ОС – Web-приложение, организующее платформу (операционную среду с набором готовых функций API) для выполнения других Web-приложений. Многие Web-ОС содержат в своем составе Web-браузер, обычно построенный на основе HTML. Главной составляющей Web-ОС является её графический интерфейс. Его особенности: внешний вид и функциональные возможности, присущие традиционным ОС, то есть наличие рабочего стола, ярлыков, окон и т. п. Благодаря этому, интерфейс Web-ОС многофункционален, удобен и понятен подавляющему большинству пользователей. Web-ОС различаются лишь набором сервисов, некоторые объединяют в себе возможности оболочек управления контентом и систем управления содержимым, другие ограничиваются возможностью удалённого рабочего стола и органайзером.

Web-ОС – по сути это тоже самое, что и обычные ОС, только они не управляют компьютером, а лишь работают с данными. Они могут применяться не только в Интернете, но и в локальной сети (интранете), Web-приложения могут заменить большинство или даже все офисные программы. Для интернета Web-ОС – это, в первую очередь, инструмент для работы с потоками информации, причём это может быть как личный инструмент на своем сайте, так и общий для рабочих групп, сообществ, социальных сетей. Web-ОС чётко отделяют оформление данных от самой информации, оформление пользователь может сделать своё, а загружать данные с других сайтов. И самое главное: всё, что нужно для Web-ОС (ОС, доступ в Интернет и браузер), сегодня это есть даже у мобильных устройств.

Итак, если есть возможность запустить браузер и выйти в Интернет, тогда в окне браузера можно открыть рабочий стол онлайн-ОС. Для этого просто надо зайти на сайт этой Web-ОС и осуществить стандартный вход (логин + пароль). И тогда получаем в окне браузера рабочий стол с набором приложений, своим браузером, местом для хранения файлов. В чем плюсы использования Web-ОС: 1) получаем дополнительное место для размещения своих файлов в Интернете. Можно получить бесплатно до 30ГБ; 2) можно получить доступ к своему рабочему столу с любого компьютера (домашнего, на работе, у друзей, в другой стране); 3) используя браузер онлайн-ОС, получаем возможность безопасного серфинга.

Web-ОС позволяют использовать свой рабочий стол на любом компьютере, где есть доступ в Интернет. Системы не зависят от аппаратной части компьютера, на котором вы работаете, что делает их аппаратно-независимыми и не «стесненными» системными требованиями. Большинство из них имеют очень красивый внешний вид и минимально необходимый функционал для работы (почта, RSS, файловый менеджер, калькулятор, офисные пакеты, контакты, календарь, чат, MP3-плеер и многое другое). Кроме этого, каждая система предоставляет дополнительные возможности, такие как, например, увеличение дискового пространства, а работа в Web-ОС ничем не отличается от работы в традиционной ОС вашего компьютера. То есть Web-ОС – это, как минимум, упорядоченное собрание сервисов.

Web-ОС представляет собой уже готовую рабочую среду с набором наиболее востребованных приложений. Зарегистрировавшись на сервисе, пользователь получает доступ к полностью рабочей системе, в которой можно хранить файлы и обмениваться ими с другими людьми, работать над документами, смотреть фильмы, слушать музыку и т.д. Таким образом, все нужные инструменты оказываются в одном месте, и все, что требуется от пользователя – помнить данные для входа в свою учетную запись.

Итак, Web-ОС – это клиент-серверное гибридное программное обеспечение, в общем случае, базирующееся на парадигме Cloud Computing, обычно функционирующее в среде современного Web-браузера.

С развитием подобных разработок было проведено более точное позиционирование облачных, интернет ОС и Web-ОС:

- **облачная ОС** – вычислительная среда, базирующаяся на мощном сервере. Обработку большей части информации выполняет именно серверное оборудование и программное обеспечение, а клиентская часть работает лишь в виде небольшой утилиты, обеспечивающей графический интерфейс и функции управления;
- **интернет ОС** – то же, что и облачная, но в ней реализованы более продвинутые и перспективные механизмы управления и работы. Реализован ряд функций, ранее присущих только полноценным ОС (возможность запуска приложений непосредственно на компьютере

пользователя). Ее достоинство в том, что при отсутствии соединения с Интернетом, важная информация может сохраняться на компьютере клиента с возможностью последующей синхронизации с облачным хранилищем;

- **Web-ОС** – в отличие от облачной и интернет ОС, Web-приложение, организующее платформу для выполнения других Web-приложений на основе браузера. Имеет дружелюбный и понятный любому пользователю графический интерфейс. Системы различаются набором служб и различной функциональностью.

Основные черты Web-ОС:

- файловая система, идентифицирующая информацию по однозначному ее местоположению;
- не зависящая от расположения ресурсов система их имен;
- безопасное удаленное выполнение популярных приложений;
- охраняемый доступ к данным пользователя;
- безопасная передача информации.

Необходимые типовые функциональные компоненты Web-ОС:

- 1) менеджер окон, запущенный на клиентской стороне, в котором находятся приложения в виде окон и фреймов;
- 2) менеджер окон, где приложения выводятся в виде двигающихся окон-слоев, являющихся частью Web-страницы, с запущенными в ней приложениями;
- 3) система добавления/удаления приложений на лету на серверной и клиентской сторонах;
- 4) средства управления несколькими пользователями и их индивидуальными настройками;
- 5) средства динамического распределения ресурсов, управления приложениями так, что они занимают как можно меньше места, когда запущены;
- 6) средства предотвращения столкновений и конфликтов приложений и их данных, точная настройка общей программной среды для успешной совместной работы.

Все это осуществляет именно Web-ОС, а не сами запущенные пользователем приложения. Это означает, что все они находятся в окружении, которое контролирует их существование и все то, что они могут и не могут делать.

Достоинства Web-ОС:

- всегда находится в сети Интернет и готова к работе;
- высокая скорость загрузки, входа (log-in) в Интернет, получения и отправления электронной почты и др.;
- обеспечивает надёжность работы;
- обеспечивает тесную интеграцию с интернет-сервисами;
- обеспечивает доступ к требуемым приложениям и актуальным данным с любого сетевого компьютера;
- предъявляет в целом невысокие системные (аппаратные ресурсы, вычислительные характеристики) требования к клиенту (ряд Web-ОС специально на это ориентирован);
- использует только лицензионное программное обеспечение (на стороне удаленного сервера Интернет);
- не требует дополнительного программного обеспечения для организации своей работы;
- гарантирует внешний контроль работоспособности и оперативных обновлений предоставляемых пользователю приложений;
- обеспечивает бесплатный характер использования среды и минимально необходимых пользователю ресурсов (соблюдается, к сожалению, не для всех Web-ОС);
- обеспечивает возможность хранения в системе любых типов файлов;
- обеспечивает возможность использования системы в качестве хранилища личной информации пользователя или общей информации группы пользователей;
- для входа в систему необходимо иметь и помнить свой логин и пароль;
- обеспечивает высокий уровень безопасности и конфиденциальности информации пользователя;
- обеспечивает высокий уровень антивирусной защиты системы и отсутствие вирусов.

Выявлены следующие обобщенные недостатки Web-ОС:

- необходимость подключения компьютера пользователя к Интернету;
- необходимость наличия у пользователя современного браузера в актуальной версии (целый ряд браузеров не поддерживается в отдельных системах);
- необходимость установки пользователем плагинов (например, Adobe

(Macromedia) Flash) в отдельных системах;

- отсутствие у пользователя контроля над собственными данными;
- ограничение объема памяти, выделенной системой для файлов пользователя при его регистрации (бесплатно или с оплатой по тарифу);
- наметилась тенденция наличия у пользователя компьютера с производительностью, достаточной для полноценной работы в Интернет;
- полная зависимость от поставщика сетевых/облачных сервисов;
- разная функциональная направленность, а также наполненность сервисами и приложениями отдельных разработок и систем. Все Web-ОС различаются набором Web-служб и своим функционалом;
- явная недостаточность полноты наборов сервисов и приложений;
- отсутствие у пользователя возможности устанавливать собственные приложения (ситуация постепенно улучшается, имеются отдельные примеры устранения данного недостатка);
- в отдельных системах отсутствие поддержки русского языка;
- наметилась тенденция ввода привилегий отдельных групп пользователей (например, iCloud доступна только для обладателей аппаратуры компании Apple), а также закрытия регистрации отдельных пользователей (Ghost).

Web-ОС, заслуживающие внимания:

- AstraNOS (<http://astranos.org>);
- DesktopTwo (<http://www.desktoptwo.com/>);
- Exo-Enterprise;
- EyeOs (<http://www.eyeos.org/>) (my.Eye-Os.org) (<https://eyeos.com/en/>);
- Ghost (Global Hosting Operating System) (<http://g.ho.st/>) (Ghost.cc);
- Glide OS (<http://www.glidedigital.com/>) (www.GlideOS.com);
- Chrome OS (<https://www.google.com/chrome/browser/desktop/index.html>);
- Goowy (<http://www.goovy.com/>);
- iCloud (<http://icloud.com/ru>);
- Joli OS;
- Joose;

- LiVirtual (<http://livirtual.com/>);
- MyGoya (<http://mygoya.de/uk>);
- myGoya Online Desktop (<http://mygoya.de>);
- Online Operating System (<https://www.oos.cc>);
- Orca (<http://www.orcaa.com/>);
- Peepel;
- SilveOS (<http://www.silveos.com/>);
- Widgetop(<http://www.widgetop.com/>)
(<http://www.widgetop.com/downloads.html>);
- Xindesk (<http://www.xindesk.com/>);
- YouOs (<https://www.youos.com/>);
- ZeroPC (<http://www.zeropc.com/>);
- Zimdesk (<http://www.zimdesk.com/>).

Таким образом, существует немало Web-ОС, задачей которых является перенос обычных настольных пользовательских приложений в среду Интернет (браузер, облако, онлайн). То есть, имея доступ в Интернет и браузер, остальные приложения пользователь получает в Интернете, используя внешнюю удаленную ОС, возможно, принципиально отличающуюся от той, в которой он привык работать. Некоторые из таких систем представляют собой привычный рабочий стол, в то время как другие (стартовые страницы) собирают информацию со всей сети. В этой связи среди Web-ОС выделяют: удаленные ОС ([AstraNOS](#); [Desktoptwo](#); [eyeOS](#); [Glide](#); [G.ho.st](#); [goowy](#); [myGoya](#); [OOS](#); [YouOS](#); [ZimDesk](#)); самостоятельные Web-ОС ([eyeOS](#); [Virtual-OS](#); [ZKDesktop](#)); удаленные рабочие столы ([DesktopOnDemand](#)); стартовые страницы ([Favoor](#); [iGoogle](#) (бывший Google IG); [Netvibes](#); [Pageflakes](#); [Schmedley](#); [Webwag](#); [yourminis](#)).

В течение последних нескольких лет сектор Web-ОС активно развивается: одни системы пополняются все новыми и новыми функциями, другие, увы, вообще перестают существовать. Несмотря на то, что виртуальные ОС все еще находятся на самой ранней стадии своего развития, вряд ли кто-то будет оспаривать тот факт, что за подобными

проектами будущее. Именно поэтому присмотреть Web-ОС для себя стоит уже сейчас. Среди существующих проектов явного лидера нет: у каждой из Web-ОС есть свои преимущества, и каждая, к сожалению, имеет свои недостатки. Социальные функции хорошо реализованы в iCloud. Эта система также хороша тем, что поддерживает работу со многими популярными форматами файлов, однако плохая поддержка альтернативных браузеров является ее серьёзным недостатком. myGoYa Online Desktop, наоборот, работает одинаково хорошо во всех браузерах, но в этой системе нет поддержки русского языка. GlideOS интересна поддержкой нескольких учетных записей, в eyeOS есть удобные средства для совместной работы над проектами, а Online Operating System отличается удобными возможностями загрузки файлов, в том числе и по FTP.

Описание Web-ОС на примере iCloud

Регистрация на iCloud занимает не более полуминуты – придумав подходящий логин и выбрав пароль, можно сразу же приступить к работе с онлайн-системой (Рис.6.2).

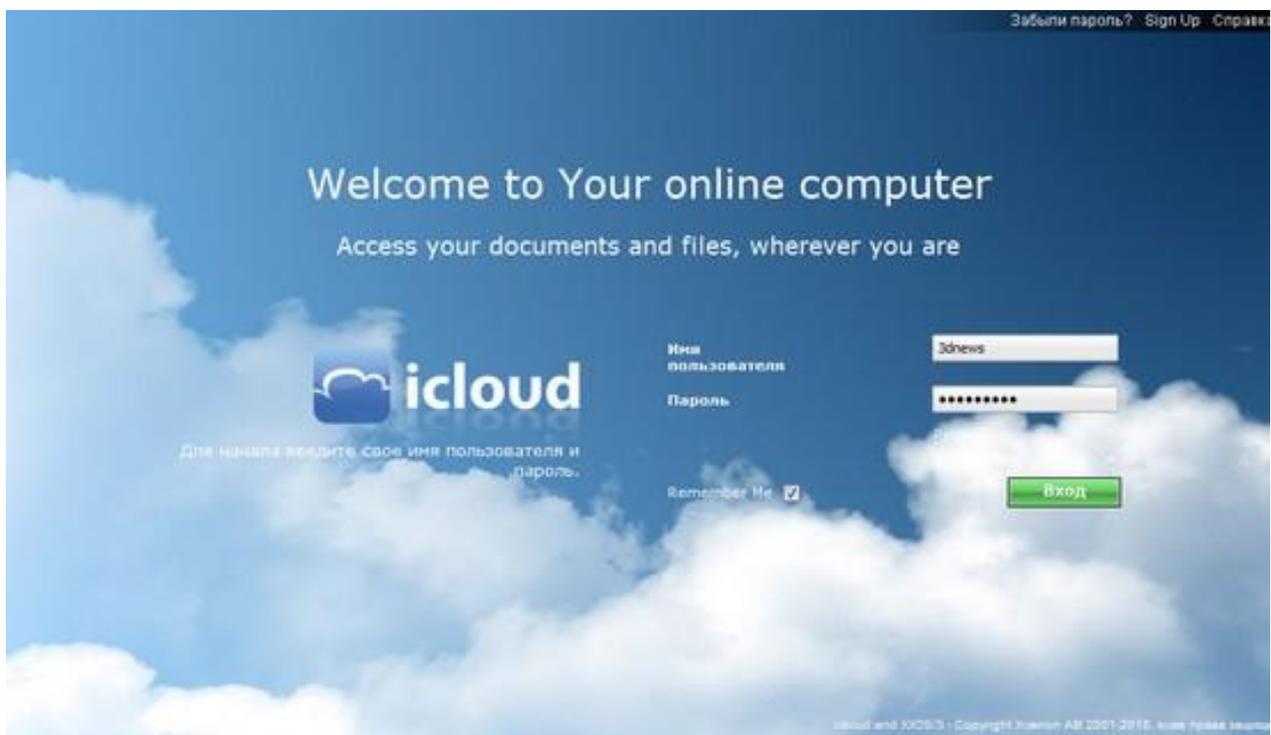


Рис.6.2. Регистрация пользователя

На загрузку рабочего стола потребуется пара минут, после чего перед вами предстанет десктоп, готовый к работе (Рис.6.3).



Рис.6.3. Представление графического интерфейса пользователя Web-OS

Виртуальная система уже укомплектована всеми наиболее востребованными приложениями: тут есть почтовый клиент, средство для просмотра изображений, плеер для просмотра фильмов (это могут быть как ролики, загруженные на YouTube, так и фильмы, которые хранятся в операционной системе), графический редактор, блокнот, калькулятор, инструмент для ведения домашней бухгалтерии. Что касается офисных приложений, то собственные разработки iCloud, включенные в состав операционной системы, пока что не радуют большим числом функций. Однако для тех, кому нужен более серьезный текстовый редактор, а также средства для работы с электронными таблицами и презентациями, предлагаются онлайн-инструменты Zoho, интегрированные в интерфейс системы. iCloud бесплатно предоставляет до 3 Гбайт для хранения пользовательских файлов. Заплатив 40 долларов за год, можно увеличить этот лимит до 100 Гбайт. Стоит иметь в виду, что на самом деле объем предоставляемого хранилища в два раза больше, так как iCloud дает столько же места для резервного копирования данных.

Загружать файлы в виртуальную ОС можно несколькими способами. Самый простой - щелкнуть по иконке, расположенной в верхней части рабочего стола iCloud, и выбрать нужные файлы на своем жестком диске.

Благодаря наличию встроенных офисных приложений Zoho, система iCloud свободно открывает файлы Word, Excel и PowerPoint, в том числе и

те, которые были созданы в последних версиях MS Office в новом формате для хранения документов. Для просмотра файлов PDF необходимо, чтобы на компьютере был установлен Adobe Reader. В этом случае документ открывается в окне просмотрщика iCloud. Что касается работы с архивами, в системе имеется приложение Unzip Archive, при помощи которого можно извлечь содержимое из загруженных файлов. К сожалению, пока что утилита поддерживает лишь файлы ZIP.

Еще одна особенность iCloud, которая касается социальных сервисов, - возможность общения с другими пользователями системы. Виртуальную ОС можно использовать и как площадку для обмена данными с другими пользователями, а также для общения с ними. В системе реализована возможность открытия доступа к определенным файлам другим людям, есть возможность объединения пользователей в группы. Если пользователь даст согласие на то, чтобы его профиль был общедоступным, другие пользователи iCloud смогут просматривать его, оставлять записи в гостевой книге, скачивать файлы и т.д.

К сожалению, iCloud стабильно работает только в Internet Explorer. В других же популярных браузерах виртуальная система или вообще отказывается загружаться, или работает с ошибками.

6.3. Порядок выполнения работы

Лабораторная работа выполняется в два этапа. Вначале студенты получают программное обеспечение (образ) предложенной преподавателем Live-ОС, самостоятельно находят всю дополнительную информацию о ней, осуществляют ее запуск, выполняют необходимые настройки, осваивают ее функционал, изучают ее средства и возможности, представление графического интерфейса пользователя, а также производят их сравнение с аналогичными средствами и возможностями известных и популярных традиционных ОС.

После этого подобная работа производится на уровне возможностей и особенностей предложенной преподавателем Web-ОС. Студенты получают ссылки на ее местонахождение, самостоятельно находят всю дополнительную информацию о рассматриваемой Web-ОС, осуществляют доступ и регистрацию на ее сайте, запуск системы, выполняют необходимые настройки, осваивают ее функционал, изучают ее средства и возможности, представление графического интерфейса пользователя, а

также производят их сравнение с аналогичными средствами и возможностями известных и популярных традиционных ОС.

Изучение материалов по рассматриваемой теме рекомендуется проводить в следующей последовательности.

1. Самостоятельное предварительное изучение информации о предложенной Live-ОС.
2. Получение загружаемого комплекта Live-ОС.
3. Запуск Live-ОС, выполнение необходимых настроек.
4. Изучение и освоение среды рассматриваемой Live-ОС: ее функционала, ее средств и возможностей, представления графического интерфейса пользователя системы.
5. Сравнение изученных средств и возможностей с характеристиками известных и популярных традиционных ОС.
6. Самостоятельное предварительное изучение информации о предложенной Web-ОС.
7. Получение ссылок на месторасположение предложенной Web-ОС, доступ и регистрация на ее сайте.
8. Запуск системы, выполнение необходимых настроек.
9. Изучение и освоение среды рассматриваемой Web-ОС: ее функционала, ее средств и возможностей, представления графического интерфейса пользователя системы.
10. Сравнение изученных средств и возможностей с характеристиками известных и популярных традиционных ОС.
11. Формирование отчета по лабораторной работе.

В ходе выполнения работы студент должен оценить:

- возможности графического интерфейса пользователя рассматриваемой Live-ОС / Web-ОС (удобство представления, функциональная полнота, удобство работы и настройки, интуитивность, эффективность реализации компонентов интерфейса);
- возможности эффективной организации работы с ресурсами Internet;
- функциональные возможности и направления использования системы.

6.4. Содержание отчета

1. Цель работы.
2. Результаты выполнения индивидуального задания, связанного с освоением предложенных преподавателем Live-ОС и Web-ОС.
3. Выводы по проделанной работе.

6.5. Контрольные вопросы

1. Дайте определение Live-CD/DVD- ОС.
2. Опишите назначение, особенности и недостатки Live-CD/DVD ОС.
3. Опишите назначение, особенности и недостатки Live-USB ОС на флеш-накопителе.
4. Сравните особенности и возможности систем Live-CD, Live-DVD и Live-USB.
5. Опишите Live-CD ОС, заслуживающие внимания.
6. Перечислите основные шаги создания Live-CD ОС.
7. Опишите общие тенденции развития Live-ОС.
8. Опишите понятие «Web-ОС» и различные, связанные с ним, альтернативные наименования.
9. Опишите более точное позиционирование облачных, интернет ОС и Web-ОС главные отличия каждой из них.
10. Дайте определение Web-ОС.
11. Проанализируйте основные черты Web-ОС.
12. Опишите основные достоинства Web-ОС.
13. Поясните выявленные обобщенные недостатки Web-ОС и возможные пути их устранения.
14. Опишите популярные Web-ОС и их главные особенности и возможности.
15. Опишите общие тенденции развития Web-ОС.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ватаманюк А.И. Установка, настройка и восстановление Windows 7 на 100%. – СПб.: Питер, 2010. – 340 с.
2. Гладкий А.А. Настройка Windows 7 своими руками. Как сделать, чтобы работать было легко и удобно. - М.: ЛитРес, 2012. - 194 с. - <http://www.libros.am/book/read/id/357518/slug/nastrojka-windows-7-svoimi-rukami-kak-sdelat-chtoby-rabotat-bylo-legko-i-udobno>.
3. Зозуля Ю. Windows 7 на 100%. – СПб.: Питер, 2010. – 472 с.
4. Карп Д. Хитрости Windows 7. Для профессионалов. – СПб.: Питер, 2011. – 512 с.
5. Климов А. Реестр Windows 7. - СПб: Питер, 2010. – 208 с.
6. Меженный О.А. Microsoft Windows 7. Краткое руководство. М.: Вильямс, 2013. - 281 с.
7. Райтман М. Установка и настройка Windows 7 для максимальной производительности (+ DVD ROM). – СПб.: БХВ-Петербург, 2010. - 368 с.
8. Acronis True Image. Документация. Руководство пользователя РС. - http://www.acronis.com/ru-ru/personal/computer-backup/#section_block-nodeblock-2779.
9. Козлова С. Как работать с программой Акронис (Acronis)? Часть 1, 2011. – <http://www.kurs-pc-dvd.ru/blog/programmy/kak-rabotat-s-programmoj-akronis-acronis-chast-1.html>.
10. Козлова С. Как работать с программой Акронис (Acronis)? Часть 2, 2011. – <http://www.kurs-pc-dvd.ru/blog/programmy/kak-rabotat-s-programmoj-akronis-acronis-chast-2.html>.
11. Введение в операционные системы: практический подход в рамках проекта OpenSolaris» Практическое руководство. – Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A.
12. Операционная система Solaris: учеб.пособие. - М.: ИНТУИТ; БИНОМ.Лаборатория знаний, 2006. - 472с. (Основы информационных технологий).
13. Операционная система Solaris / Ф.И. Торчинский, Е.С. Ильин – 2-е

изд., испр. – М.: ИНТУИТ; БИНОМ. Лаборатория знаний, 2009. – 600 с. (Серия «Основы информационных технологий»).

14. Операционная система Linux – передовая технология для всех. – https://linux4u.jinr.ru/usoft/WWW/194_44_93.1/linux/laosa.html.
15. Колисниченко Д.Н. Linux своими руками. – <http://lib.rus.ec/b/173845>.
16. Колисниченко Д.Н. Linux. От новичка к профессионалу. 4-е изд. – СПб.:БХВ-Петербург, 2012. – 704 с.
17. Баррет Д.Дж. Linux: основные команды. Карманный справочник. – М.: КУДИЦ-ОБРАЗ, 2007. – 288 с.
18. Бондаренко С., Бондаренко М. Операционная система в браузере, пять лучших WebOS, 2010. – <https://www.3dnews.ru/601548>.
19. Чеботарев А. Операционные системы на съёмных носителях. – http://www.zahodi-ka.ru/stat/program/live_cd.shtml
20. Портативная операционная система в кармане, 2011. – <http://www.softiq.ru/news/?id=13>.
21. Живой CD, 2016. – <http://dic.academic.ru/dic.nsf/ruwiki/919069>.
22. Список Live-CD дистрибутивов. – <http://dic.academic.ru/dic.nsf/ruwiki/1633395>.
23. Live CD, 2015. – https://ru.wikipedia.org/wiki/Live_CD.
24. Live USB, 2016. – http://biceps-spb.ru/sport-pitanie/Live_USB.
25. Live USB, 2015. – https://ru.wikipedia.org/wiki/Live_USB.
26. Parted Magic OS 2013.06.15 ML/Rus - CD/USB операционная система Live CD, 2015. – http://portable.moy.su/load/livecd_winpe/parted_magic_os_2013_06_15_ml_rus_cd_usb_operacionnaja_sistema_live_cd/18-1-0-2227.
27. Скачать Linux. Live-CD и USB, образы установочных дисков, 2016. - http://desktoplinux.ru/unix_guide/skachat_linux_live_cd_obrazy_ustanovочnykh_diskov.
28. Linux. The LiveCD List. - <http://livecdlist.com/os/linux/>.
29. Создание Linux LiveCD/ - <https://www.ibm.com/developerworks/ru/library/l-LiveCD/index.html>.
30. Simply Linux 6.0.1 LiveCD [i586] (2012) PC - скачать торрент. - <http://rutor.in/viewtopic.php?f=29&t=139933>.
31. Руководство по Ubuntu для новичков. Загрузка Ubuntu с LiveCD/ - <http://help.ubuntu.ru/manual/%D0%B7%D0%B0%D0%B3%D1%80%D1%>

- [83%D0%B7%D0%BA%D0%B0_%D1%81_livecd.](#)
32. WebOS, 2017. – <http://encyclopaedia.bid/%D0%B2%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F/WebOS.>
33. 5 лучших операционных веб-систем (Web OS), 2009. – [http://www.coolwebmasters.com/operating-system/77-top-of-5-web-operating-systems.html.](http://www.coolwebmasters.com/operating-system/77-top-of-5-web-operating-systems.html)
34. Арсентьев Ю. Браузерные операционные системы (Web Operating System), 2008. – [http://www.flcd.ru/os/reviews/web_operating_system.](http://www.flcd.ru/os/reviews/web_operating_system)
35. Лучшие WebOS, 2007. – [https://xakep.ru/2007/11/26/41262/.](https://xakep.ru/2007/11/26/41262/)
36. Web OS, 2010. – [http://www.alshar.ru/web-os.html.](http://www.alshar.ru/web-os.html)
37. Емельянов И. 15 лучших онлайн-операционных систем, 2010. – [https://lifehacker.ru/2010/03/04/webos/.](https://lifehacker.ru/2010/03/04/webos/)
38. Виртуальная операционная система в браузере, или что такое WebOS, 2011. – [http://tibezh.blogspot.ru/2011/04/webos.html.](http://tibezh.blogspot.ru/2011/04/webos.html)
39. Обзор существующих WebOS, 2007. – [http://www.livebusiness.ru/news/229/.](http://www.livebusiness.ru/news/229/)
40. Что такое облачные операционные системы, 2012. – [http://www.ildarmukhutdinov.ru/2012/02/23/web-os/.](http://www.ildarmukhutdinov.ru/2012/02/23/web-os/)
41. WebOS, 2016. – [http://www.sbup.com/wiki/WebOS.](http://www.sbup.com/wiki/WebOS)
42. Онлайн операционные системы (WebOS). – [http://shkamer.ru/2012/02/09/onlajn-operacionnye-sistemy-webos/.](http://shkamer.ru/2012/02/09/onlajn-operacionnye-sistemy-webos/)
43. Бондаренко С., Бондаренко М. Операционная система в браузере: пять лучших WebOS, 2010. – [http://apocalypse.moy.su/publ/34-1-0-4645.](http://apocalypse.moy.su/publ/34-1-0-4645)
44. Обзор Web OS, 2010. – [http://korzh.net/2010-05-obzor-web-os.html.](http://korzh.net/2010-05-obzor-web-os.html)
45. WEB OS: 40+ сетевых операционных систем, 2007. – [https://networkforpeople.wordpress.com/2007/11/16/web-os-40-%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D1%8B%D1%85-%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D1%85-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC/.](https://networkforpeople.wordpress.com/2007/11/16/web-os-40-%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D1%8B%D1%85-%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D1%85-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC/)
46. Веб-операционная система онлайн, 2010. – [http://womtec.ru/2010/10/online-web-operating-systems/.](http://womtec.ru/2010/10/online-web-operating-systems/)
47. Виртуальная Web ОС уже сегодня, 2010. – [http://nyukers.blogspot.ru/2010/11/web.html#axzz4cnx9YepW.](http://nyukers.blogspot.ru/2010/11/web.html#axzz4cnx9YepW)

48. 9 облачных операционных систем, которые вы можете опробовать бесплатно. – <http://www.interface.ru/home.asp?artId=31816>.
49. Онлайн операционные системы, 2009. – http://www.livebusiness.ru/tags/onlajn_operacionnye_sistemy.
50. Савчук И. Шпаргалка по облачным операционным системам, 2011. – <http://internetno.net/category/obzoryi/crib/cloud-os-2011/>.
51. Подборка облачных операционных систем, которые вы можете опробовать бесплатно, 2013. – <https://webtun.com/webservices/4076-9-cloud-operating-system-you-can-try-out-for-free.html>.
52. 9 облачных операционных систем, которые вы можете опробовать бесплатно, 2013. – <http://www.itshop.ru/9-oblachnyh-operatsionnyh-sistem-kotorye-vy-mozhete-oprobovat-besplatno/19i31816>.
53. Lawton G. Moving OSs to Web. Computer, March 2008, pp. 16-19; Перемещение операционных систем в Web, 2008. – http://khpi-iip.mipk.kharkiv.edu/library/transl/05/t05_4.html.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ЛАБОРАТОРНАЯ РАБОТА № 1 «Установка и настройка ОС MS Windows».....	5
1.1. Цель работы	5
1.2. Теоретическая часть	5
1.2.1. Индивидуальная настройка рабочего стола.....	6
1.2.2. Диспетчер задач (Task Manager)	18
1.2.3. Планировщик заданий (Task Scheduler)	20
1.2.4. Средства поддержки виртуальной памяти	21
1.2.5. Другие средства оптимизации	26
1.3. Порядок выполнения работы	32
1.4. Содержание отчета.....	33
1.5. Контрольные вопросы	33
ЛАБОРАТОРНАЯ РАБОТА № 2 «Возможности и средства ОС MS Windows».....	35
2.1. Цель работы	35
2.2. Теоретическая часть.....	35
2.2.1. Программа получения сведений о системе.....	35
2.2.2. Программа архивации данных	36
2.2.3. Программа проверки диска.....	36
2.2.4. Программа дефрагментации дисков	37
2.2.5. Программа очистки жесткого диска	38
2.2.6. Программа «Восстановление системы»	40
2.4. Содержание отчета.....	43
2.5. Контрольные вопросы	44
ЛАБОРАТОРНАЯ РАБОТА № 3 «Средства сохранения и восстановления программной среды ОС MS Windows».....	45
3.1. Цель работы	45
3.2. Теоретическая часть.....	45
3.3. Порядок выполнения работы	47
3.4. Индивидуальное задание	54
3.5. Содержание отчета.....	56
3.6. Контрольные вопросы	56
ЛАБОРАТОРНАЯ РАБОТА № 4 «Установка и настройка ОС Linux или иной ОС по выбору (Solaris, UNIX, MacOS, и др.)».....	57

Часть 1 - «Настройка сетевых интерфейсов в ОС Solaris»	57
4.1. Цель работы	57
4.2. Теоретическая часть	57
4.2.1. Описание ОС Solaris 10	57
4.2.2 Основные функциональные возможности	58
4.2.3 Интерфейс пользователя	61
4.2.4 Командная строка	61
4.2.5. Графический интерфейс	62
4.2.6. Пример настройки сетевого интерфейса	66
4.3. Задание для индивидуальной работы.....	70
4.4. Содержание отчета.....	71
4.5. Контрольные вопросы	71
Часть 2 - «Использование политики ролевого доступа Role Based Access Control»	71
4.6. Цель работы	71
4.7. Теоретическая часть	71
4.8. Порядок выполнения работы	76
4.9. Содержание отчета.....	81
4.10. Контрольные вопросы	81
ЛАБОРАТОРНАЯ РАБОТА № 5 «Возможности и средства ОС Linux»	82
Часть 1 - «Администрирование операционной системы Linux».....	82
5.1. Цель работы.....	82
5.2. Теоретическая часть	82
5.2.1. Особенности и возможности ОС Linux	82
5.2.2 Работа с пользователями в ОС Linux.....	85
5.2.3 Управление ограничением доступа к файлам и папкам.....	86
5.2.4 Сетевая утилита netstat	88
5.3. Порядок выполнения работы	89
5.4. Задания для лабораторной работы:	96
5.5. Содержание отчета.....	97
5.6. Контрольные вопросы	97
Часть 2 - «Диспетчеризация операционной системы Linux»	98
5.7. Цель работы	98
5.8. Теоретическая часть	98
5.8.1. Интерактивные процессы.....	98
5.8.2. Свойства процессов	99

5.8.3. Особенности GRUB	104
5.9. Порядок выполнения работы	109
5.10. Задания для самостоятельной работы.....	116
5.11. Содержание отчета.....	117
5.12. Контрольные вопросы	117
ЛАБОРАТОРНАЯ РАБОТА №6 «Освоение перспективных типов ОС (Live-CD/DVD ОС, Web-ОС)»	118
6.1. Цель работы	118
6.2. Теоретическая часть.....	118
6.3. Порядок выполнения работы	133
6.4. Содержание отчета.....	135
6.5. Контрольные вопросы	135
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	136

Учебное издание

Нужнов Евгений Владимирович

Самойлов Алексей Николаевич

Беликов Александр Николаевич

Методические указания
к выполнению комплекса лабораторных работ
по общеинститутской дисциплине
«Операционные системы»

Редактор Белова Л.Ф.

Корректор Селезнева Н.И.

Подписано к печати 15.02.2017

Формат 60x84 1/16

Бумага офсетная.

Печать офсетная.

Усл. п. л. – 9,0

Уч.-изд. л. – 8,7

Заказ № 76

Тираж 20 экз.

«С»

Издательство Южного федерального университета

344091, г. Ростов-на-Дону, пр. Стачки, 200/1. Тел. (863) 2478051.

Отпечатано в г. Таганроге в отделе полиграфической, корпоративной и сувенирной продукции ИПК КИБИ МЕДИА ЦЕНТРА ЮФУ.

ГСП 17А, Таганрог, 28, Энгельса, 1.

Тел. (8634) 371717, 371655.