

Министерство образования и науки Российской Федерации
ГБОУ СПО «Торжокский педагогический колледж им. Ф.В. Бадюлина»

Комплект
контрольно-оценочных оценочных средств
учебной дисциплины
Операционные системы и среды
основной образовательной программы (ОПОП)
по направлению подготовки (специальности)
230701 «Прикладная информатика (по отраслям)»

Разработал:
Преподаватель информатики
Шефер Елена Михайловна

Торжок
2012 г.

1. Общие положения

Контрольно-оценочные средства (КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «Операционные системы и среды»

КОС включают контрольные материалы для проведения текущего контроля и промежуточной аттестации в форме зачета (1 семестр), дифференцированный зачет (2 семестр, экзамен (3 семестр).

КОС разработаны на основании положений: основной профессиональной образовательной программы по направлению подготовки ВПО (специальности СПО) 230701 «Прикладная информатика (по отраслям)»; программы учебной дисциплины «Операционные системы и среды».

2. Результаты освоения дисциплины, подлежащие проверке

Результаты обучения (освоенные умения, усвоенные знания)
Умение использовать средства операционных систем для решения практических задач
Умение использовать сервисные средства, поставляемые с операционной системой
Умение использовать различные операционные системы
Умение подключать к операционной системе новые сервисные средства
Умение решать задачи операционных систем
Знание основных функций операционных систем
Знание машинно-независимых свойств операционных систем
Знание принципов построения операционных систем
Знание сопровождения операционных систем

3. Распределение оценивания результатов обучения по видам контроля

Наименование элемента умений или знаний	Виды аттестации	
	<i>Текущий контроль</i>	<i>Промежуточная аттестация</i>
У1. Умение использовать средства операционных систем для решения практических задач	Лабораторные работы – 5	Практическая работа - 10
У2. Умение использовать сервисные средства, поставляемые с операционной системой	Лабораторные работы – 10	Практическая работа - 9
У3. Умение использовать различные операционные системы	Лабораторные работы – 4	Практическая работа - 3
У4. Умение подключать к операционной системе новые сервисные средства	Лабораторные работы – 2	Практическая работа - 4
У5. Умение решать задачи операционных систем	Лабораторные работы – 6 устный ответ-1	Практическая работа - 7
З1. Знание основных функций операционных систем	Лабораторные работы – 6 устный ответ-6	Устный ответ – 19
З2. Знание машинно-независимые свойства операционных систем	лабораторные. работы – 3 устный ответ-2	Устный ответ – 23
З3. Знание принципов построения операционных систем	Лабораторные работы – 4 устный ответ-4	Устный ответ – 8
З4. Знание сопровождения операционных систем	Лабораторные работы – 14 устный ответ-2	Устный ответ - 19

4. Распределение типов контрольных заданий по элементам знаний и умений.

Содержание учебного материала по программе УД	Тип контрольного задания								
	У1	У2	У3	У4	У5	31	32	33	34
Раздел 1. Основы операционных систем									
Тема 1.1. Понятие ОС. Основные функции								Устный ответ	
Тема 1.2. Основные понятия операционных систем. Классификация операционных систем	Лаб. раб. №1,2					Лаб. раб. №1			
Тема 1.3. Процессы	Лаб. раб. №4	Лаб. раб. №3,5				Лаб. раб. №3,5		Лаб. раб. №3,4,5	
Тема 1.4. Планирование процессов						Устный ответ			
Тема 1.5. Взаимодействие процессов		Лаб. раб. №6					Лаб. раб. №6		
Тема 1.16. Управление памятью. Простейшие схемы управления памятью							Устный ответ		
Тема 1.7. Виртуальная память							Устный ответ		
Тема 1.8. Аппаратно-независимый уровень управления виртуальной памятью						Устный ответ			
Тема 1.9. Файловые системы. Файлы с точки зрения пользователя.	Лаб. раб. №8,9	Лаб. раб. №7				Лаб. раб. №8,9			Лаб. раб. №7
Тема 1.10. Логическая структура файлового архива	Лаб. раб. №10								
Тема 1.11. Реализация файловой системы						Устный ответ			
Тема 1.12. Физические основы организации управления вводом-выводом				Лаб. раб. №11,12					Лаб. раб. №11,12
Тема 1.13. Логические принципы организации ввода-вывода						Устный ответ			
Тема 1.14. Сетевые операционные системы.								Устный ответ	
Тема 1.15. Проблема адресации в сети					Лаб. раб. №13				Лаб. раб. №13
Тема 1.16. Безопасность операционных систем						Устный ответ			
Тема 1.17. Авторизация и аудит								Устный ответ	
Тема 1.18. Контрольная работа №1.						Устный ответ			
Раздел 2. Работа в конкретных операционных системах									
Тема 2.1. Состав операционных систем Windows и Linux								Устный ответ	
Тема 2.2. Установка операционных систем					Лаб. раб. №14,15				Лаб. раб. №14,15
Тема 2.3. Изменение настроек и пользовательских интерфейсов					Лаб. раб. №16				Лаб. раб. №16
Тема 2.4. Установка дополнительного программного обеспечения		Лаб. раб. №17,18							Лаб. раб. №17,18
Тема 2.5. Файловые менеджеры		Лаб. раб. №19,20							Лаб. раб. №19,20
Тема 2.6. Теоретические основы сжатия информации		Лаб. раб. №21							
Тема 2.7. Понятие о системном администрировании								Устный ответ	
Тема 2.8. Способы входа в систему			Лаб. раб. №22			Лаб. раб. №22			
Тема 2.9. Основы работы с командными интерпретаторами							Устный ответ		
Тема 2.10. Управление учетными записями и правами доступа			Лаб. раб. №23				Лаб. раб. №23		
Тема 2.11. Командные файлы в файлах сценариев					Лаб. раб. №24				Лаб. раб. №24
Тема 2.12. Файловые системы		Лаб. раб. №25	Лаб. раб. №25				Лаб. раб. №25		
Тема 2.13. Установка дополнительного оборудования					Лаб. раб. №26			Лаб. раб. №26	
Тема 2.14. Сетевые ресурсы. Клиентская часть	Лаб. раб. №27								Лаб. раб. №27
Тема 2.15. Сетевые ресурсы. Серверная часть.								Устный ответ	
Тема 2.16. Дополнительные возможности.			Лаб. раб. №28						Лаб. раб. №28

5. Распределение типов и количества контрольных заданий по элементам знаний и умений, контролируемых на промежуточной аттестации. (примечание: *Билет №, № вопроса в билете*)

Содержание учебного материала по программе ПМ	Тип контрольного задания								
	У1	У2	У3	У4	У5	З1	З2	З3	З4
Раздел 1. Основы операционных систем									
Тема 1.1. Понятие ОС. Основные функции						Б1,1; устный ответ			Б33,2; устный ответ
Тема 1.2. Основные понятия операционных систем. Классификация операционных систем						Б2,1; устный ответ	Б8,1; устный ответ		
Тема 1.3. Процессы	Б1,3; Б8,3; практич. работа					Б3,1; устный ответ			Б10,1; Б11,1; устный ответ
Тема 1.4. Планирование процессов		Б9,3; практич. работа			Б13,3; практич. работа	Б34,1; устный ответ		Б17,2; устный ответ	Б9,1; Б12,1; устный ответ
Тема 1.5. Взаимодействие процессов						Б30,2; устный ответ		Б26,1; устный ответ	Б13,1; устный ответ
Тема 1.6. Управление памятью. Простейшие схемы управления памятью				Б14,3; практич. работа		Б26,2; Б35,2; устный ответ			
Тема 1.7. Виртуальная память							Б23,2; Б28,1; устный ответ		
Тема 1.8. Аппаратно-независимый уровень управления виртуальной памятью								Б29,2; устный ответ	
Тема 1.9. Файловые системы. Файлы с точки зрения пользователя.	Б4,3; Б5,3; практич. работа	Б7,3; практич. работа					Б1,2; Б3,2; Б4,2; Б6,2; Б8,2; Б11,2; Б12,2 устный ответ		
Тема 1.10. Логическая структура файлового архива	Б10,3; практич. работа						Б5,2; Б7,1; Б9,2; Б13,2; устный ответ		
Тема 1.11. Реализация файловой системы	Б16,3; практич. работа			Б12,3; практич. работа	Б6,3; практич. работа		Б7,2; Б10,2; устный ответ		
Тема 1.12. Физические основы организации управления вводом-выводом		Б29,3; практич. работа				Б34,2; Б35,1; устный ответ			
Тема 1.13. Логические принципы организации ввода-вывода					Б31,3; практич. работа			Б4,1; устный ответ	Б5,1; устный ответ
Тема 1.14. Сетевые операционные системы.	Б30,3; практич. работа					Б20,1; Б21,1; Б24,1; устный ответ			
Тема 1.15. Проблема адресации в сети									Б27,1; устный ответ
Тема 1.16. Безопасность операционных систем									Б6,1; Б15,2; устный ответ
Тема 1.17. Авторизация и аудит						Б32,1; устный ответ		Б31,1; устный ответ	
Тема 1.18. Контрольная работа №1.									

Содержание учебного материала по программе ПМ	Тип контрольного задания								
	У1	У2	У3	У4	У5	31	32	33	34
Раздел 2. Работа в конкретных операционных системах									
Тема 2.1. Состав операционных систем Windows и Linux						Б18,1; Б23,1; устный ответ			
Тема 2.2. Установка операционных систем								Б21,2; устный ответ	
Тема 2.3. Изменение настроек и пользовательских интерфейсов						Б18,2; Б30,1; устный ответ			Б33,1; устный ответ
Тема 2.4. Установка дополнительного программного обеспечения							Б14,1; устный ответ		Б29,1; Б31,2; устный ответ
Тема 2.5. Файловые менеджеры	Б17,3; практич. работа	Б2,3; практич. работа			Б3,3; практич. работа		Б16,1; Б20,2; Б22,2; устный ответ		
Тема 2.6. Теоретические основы сжатия информации		Б24,3; Б25,3; практич. работа		Б20,3; практич. работа			Б16,2; устный ответ		Б28,2; устный ответ
Тема 2.7. Понятие о системном администрировании		Б11,3; практич. работа				Б25,1; устный ответ			
Тема 2.8. Способы входа в систему			Б26,3; практич. работа	Б19,3; практич. работа			Б14,2; устный ответ		
Тема 2.9. Основы работы с командными интерпретаторами		Б10,3; Б33,3; практич. работа			Б23,3; практич. работа				Б24,2; устный ответ
Тема 2.10. Управление учетными записями и правами доступа					Б15,3; Б32,3; практич. работа	Б27,2; устный ответ			
Тема 2.11. Командные файлы в файлах сценариев	Б34,3; Б35,3; практич. работа		Б21,3; практич. работа					Б22,2; устный ответ	
Тема 2.12. Файловые системы			Б22,3; практич. работа				Б25,2; устный ответ	Б19,1; устный ответ	
Тема 2.13. Установка дополнительного оборудования									Б19,2; устный ответ
Тема 2.14. Сетевые ресурсы. Клиентская часть									Б15,1; устный ответ
Тема 2.15. Сетевые ресурсы. Серверная часть.									Б15,1; устный ответ
Тема 2.16. Дополнительные возможности.									Б19,2; устный ответ

6. Структура контрольного задания

6.1. Текст задания

6.1.1. Лабораторные работы

Лабораторная работа №1 (1 час) Знакомство с операционной системой Linux

Понятия login и password

Операционная система Linux является многопользовательской операционной системой. Для обеспечения безопасной работы пользователей и целостности системы доступ к ней должен быть санкционирован. Для каждого пользователя, которому разрешен вход в систему, заводится специальное регистрационное имя – username или login и сохраняется специальный пароль – password, соответствующий этому имени.

Узнайте у своего системного администратора регистрационные имена и пароли, установленные для обучающихся.

Вход в систему

Если в системе установлена графическая оболочка наряду с обычными алфавитно-цифровыми терминалами, лучше всего это сделать с алфавитно-цифрового терминала или его эмулятора. Чтобы перейти из графической оболочки в алфавитно-цифровой терминал нажмите **<ctrl>+<alt>+<Fx>**, где Fx – одна из функциональных клавиш F1, F2...F6. Вы попадете в алфавитно-цифровую консоль. Номер консоли указывает указатель tty. Например: tty2- означает, что вы находитесь во второй консоли. По умолчанию таких консолей 7. Причем седьмая консоль-графическая. Переходить между консолями можно нажимая клавиши **<alt>+<Fx>**.

Потренируйтесь переходить между консолями

В каждой консоли вам предлагают зарегистрироваться. Таким образом в разных консолях вы можете работать под разными именами и, что еще более важно, с разными правами. Итак, регистрируемся в консоли.

На экране появляется надпись, предлагающая ввести регистрационное имя, как правило, это **"login:"**. Набрав свое регистрационное имя, нажмите клавишу **<Enter>**. Система запросит у вас пароль, соответствующий введенному имени, выдав специальное приглашение – обычно **"password:"**. Внимательно наберите пароль, установленный для вас системным администратором, и нажмите клавишу **<Enter>**. **Вводимый пароль на экране не отображается, поэтому набирайте его аккуратно!** Если все было сделано правильно, у вас на экране появится приглашение к вводу команд операционной системы.

Команда man – универсальный справочник

По ходу изучения операционной системы Linux вам часто будет требоваться информация о том, что делает та или иная команда или системный вызов, какие у них параметры и опции, для чего предназначены некоторые системные файлы, каков их формат и т.д.

Для получения интерактивной справки служит утилита **man**

Пользоваться утилитой man достаточно просто – наберите команду

man имя

где имя – это имя интересующей вас команды, утилиты, системного вызова, библиотечной функции или файла.

Попробуйте с ее помощью посмотреть информацию о командах pwd, who, whoami, last.

Чтобы пролистать страницу полученного описания, если оно не поместилось на экране полностью, следует нажать клавишу **<пробел>**. Для прокрутки одной строки воспользуйтесь клавишей **<Enter>**. Вернуться на страницу назад позволит одновременное нажатие клавиш **<Ctrl>** и ****. Выйти из режима просмотра информации можно с помощью клавиши **<q>**.

Иногда имена команд интерпретатора и системных вызовов или какие-либо еще имена совпадают. Тогда чтобы найти интересующую вас информацию, необходимо задать утилите man категорию, к которой относится эта информация (номер раздела). В Linux, принято следующее деление:

1. Исполняемые файлы или команды интерпретатора.
2. Системные вызовы.
3. Библиотечные функции.
4. Специальные файлы (обычно файлы устройств)
5. Формат системных файлов и принятые соглашения.
6. Игры (обычно отсутствуют).
7. Макропакеты и утилиты – такие как сам man.
8. Команды системного администратора.
9. Подпрограммы ядра (нестандартный раздел).

Если вы знаете раздел, к которому относится информация, то утилиту man можно вызвать в Linux с дополнительным параметром

man номер_раздела имя

В других операционных системах этот вызов может выглядеть иначе. Для получения точной информации о разбиении на разделы, форме указания номера раздела и дополнительных возможностях утилиты man наберите команду

man man

Используя команды `who`, `whoami`, `last` получите сведения о пользователях, работающих в системе.

Завершение работы в Linux

В ОС Linux, нельзя выключать компьютер простым отключением питания, Дело в том, что в любой момент времени в системе запущено несколько процессов. Некоторые из этих процессов могут работать с файлами, причем система не записывает все изменения файлов на диск сразу после внесения этих изменений пользователем или процессом, а сохраняет их временно в оперативной памяти (кэширует). Если просто выключить питание, эти изменения не будут сохранены и пропадут, что иногда может привести даже к невозможности последующей загрузки системы. Так что надо уметь правильно завершить работу системы перед выключением компьютера. Это делается командой `shutdown`.

Команда `shutdown` может быть выполнена только пользователем `root`

Из опций программы `shutdown` наиболее часто используются две:

- `-h` — полная остановка системы (компьютер будет выключен);
- `-r` — перезагрузить систему.

Когда вы захотите просто выключить компьютер. Эквивалентом команды

`shutdown -h 0`

является команда `halt`. При нажатии известной комбинации клавиш `<Ctrl>+<Alt>+` в Red Hat Linux выполняются действия, аналогичные команде

`shutdown -r 0`

так что таким образом тоже можно выключить компьютер, только надо в момент перезагрузки отключить питание.

Выполните перезагрузку системы

Выполните завершение работы

Лабораторная работа № 2 (1 час) Знакомство с текстовым редактором Vim.

Vim экранный текстовый редактор. Большая часть экрана используется для отображения редактируемого файла. Последняя строка экрана используется для ввода команд и вывода различной информации. Редактор может находиться либо в режиме редактирования, либо в режиме ввода команд. Для того, чтобы совершать какие либо действия Вы должны находиться в нужном режиме.

После запуска редактор находится в командном режиме. Для перехода из режима редактирования в командный режим используется клавиша `Esc`.

Для того, чтобы начать редактирование файла используется команда `vi имя_файла`.

Основные возможности в командном режиме

• Перемещение по файлу:

h, left-arrow - переместить курсор влево на один символ

j, down-arrow - переместить курсор вниз на одну строку

k, up-arrow - переместить курсор вверх на одну строку

l, right-arrow - переместить курсор вправо на один символ

/text<cr> - найти строку **text** в файле и поместить курсор на ее первый символ. После этого можно использовать клавиши **n** и **Shift-n** для перемещения к следующему или предыдущему включению строки.

• Переход в режим редактирования:

i - начать ввод текста перед курсором

a - начать ввод текста после курсора

o - вставить строку после текущей и начать ввод текста в ней

O - вставить строку перед текущей и начать ввод текста в ней

• Копирование, вставка и удаление:

yy **u\$** **yw** - скопировать строку, строку от позиции курсора до конца, слово.

dd **d\$** **dw** - удалить строку, строку от позиции курсора до конца, слово.

x - удалить символ

p - вставить содержимое буфера после курсора

P - вставить содержимое буфера перед курсором

u - Отменить последнюю операцию

• Сохранение и чтение файлов, выход из редактора:

:w<cr> - сохранить файл

:w filename<cr> - сохранить файл под указанным именем

:r filename<cr> - вставить содержимое указанного файла

:q<cr> - выйти из редактора

:wq<cr> - сохранить файл и выйти из редактора

:q!<cr> - выйти без сохранения файла

Самостоятельная работа

При помощи редактора `vim` создайте следующие файлы:

1

Я живу в маленьком доме на дюнах. Все Рижское взморье в снегу. Он все время слетает с высоких сосен длинными прядями и рассыпается в пыль.

Слетает он от ветра и оттого, что по соснам прыгают белки. Когда очень тихо, то слышно, как они шелушат сосновые шишки.

Дом стоит у самого моря. Чтобы увидеть море, нужно выйти за калитку и немного пройти по протоптанной в снегу тропинке мимо заколоченной дачи.

На окнах этой дачи еще с лета остались занавески. Они шевелятся от слабого ветра

2

Море не замерзло. Снег лежит до самой кромки воды. На нем видны следы зайцев.

Когда на море подымается волна, то слышен не шум прибоя, а хрустенье льда и шорох оседающего снега, Балтика зимой пустынна и угрюма.

Латыши называют ее «Янтарным морем» («Дзинтара юра»). Может быть, не только потому, что Балтика выбрасывает много янтаря, но еще и потому, что ее вода чуть заметно отливает янтарной желтизной.

Сохраните созданные файлы, дав им свое имя и фамилию

Лабораторная работа № 3 (1 час) **Процессы в операционной системе Linux**

1 Процессы

Процесс (process) — блок адресного пространства в котором выполняются одна или более нитей, экземпляр выполняемой программы. Любой процесс может запускать другие процессы.

Таким образом, процессы в среде UNIX образуют иерархическую структуру. На вершине этой структуры находится процесс init, являющийся предком всех остальных процессов.

1.1 Атрибуты процессов

С каждым процессом связан набор атрибутов, которые помогают системе контролировать выполнение процессов и распределять между ними ресурсы системы.

Идентификатор процесса (process ID) это целое число однозначно идентифицирующее процесс. Процесс с идентификатором 1 это процесс init.

Идентификатор родительского процесса (parent process ID) указывает на родительский процесс.

Идентификатор группы процессов (process group ID).

Процессы могут объединяться в группы. Каждая группа обозначается идентификатором группы. Процесс, идентификатор которого совпадает с идентификатором группы, называется лидером группы.

Идентификатор сеанса (session ID).

Каждая группа процессов принадлежит к сеансу. Сеанс связывает процессы с управляющим терминалом. Когда пользователь входит в систему, все создаваемые им процессы будут принадлежать сеансу, связанному с его текущим терминалом.

Программное окружение (programm environment) это просто набор строк, заканчивающихся нулевым символом. Строки называются переменными окружения и имеют следующий формат: имя переменной = значение переменной

Дескрипторы открытых файлов.

Дескриптор файла — некоторое число, которое используется для обращения к файлу.

При запуске процесс наследует дескрипторы от родительского процесса.

Текущий рабочий каталог это каталог от которого система производит разрешение относительных имен.

Текущий корневой каталог это каталог от которого производится разрешение абсолютных имен. Процесс не имеет доступа к файлам находящимся выше корневого каталога.

Идентификаторы пользователя и группы.

С каждым процессом связаны действительные идентификаторы пользователя (real userID) и группы (real group ID), совпадающие с соответствующими идентификаторами пользователя, запустившего процесс. Кроме того, с процессом связаны эффективные идентификаторы пользователя (effective user ID) и группы, определяющие права процесса в системе. Обычно, действительные и эффективные идентификаторы совпадают.

Приоритет (nice).

Значение nice ("дружелюбность") показывает готовность процесса уступить свое процессорное время другим процессам. Чем больше значение nice, тем ниже приоритет процесса.

2 Команды для работы с процессами

ps [-axewjlu] [-o формат] [-U пользователь] [-p pid]

Выводит список и статус процессов работающих в системе. Без аргументов выводит список процессов текущего пользователя, подключенных к терминалу. Значения параметров следующие:

-a вывести информацию о процессах всех пользователей.

-x вывести информацию о процессах не подключенных к терминалу.

-e вывести значения переменных окружения процесса.

-w использовать строки длиной 132 символа. Если указан несколько раз, то строки не обрезаются совсем.

-j, -l, -u - меняют формат вывода информации.

-o формат вывести информацию в указанном формате.

-U пользователь вывести информацию о процессах указанного пользователя.

-p pid вывести информацию о процессе с указанным идентификатором.

Значение формата для параметра **-o** является списком из следующих ключевых слов разделенных запятыми (без пробелов):

Command командная строка и аргументы.
nice уровень nice (приоритет).
pgid идентификатор группы процессов.
pid идентификатор процесса.
ppid идентификатор родительского процесса.
rgid, ruid реальные идентификаторы группы и пользователя.
uid реальный идентификатор пользователя.
tty управляющий терминал
top выводит сведения о запущенных процессах в динамике. Прекратить работу команды можно нажав клавишу q

Самостоятельная работа

1. Освойте работу с командой **ps**. Попробуйте запускать ее с различными аргументами. Если вывод команды не помещается на экране, используйте команду **less**.
2. Запишите в отчет следующую информацию о запущенных Вами процессах:
pid, ppid, tty, ruid, command. Вывод должен быть отсортирован по номеру процесса.
3. Ключи -c, -v, -j, -u изменяют формат вывода команды. Попробуйте выполнить команду **ps** с каждым из этих ключей. Результат запишите в отчет .
4. Выведите сведения о процессах в динамике. Запишите в отчет дополнительные сведения, которые были представлены командой **top**.

Лабораторная работа №4 (1 час) Управление процессами Linux

Linux имеет большое число механизмов межпроцессного взаимодействия. Наиболее популярными средствами являются сигналы, программные каналы (pipes) и именованные каналы (FIFO).

Сигналы

Сигналы обеспечивают простой метод прерывания работы процессов. Сигналы используются в основном для обработки исключительных ситуаций. Процесс может определять действия выполняемые при поступлении сигнала, блокировать сигналы, посылать сигналы другим процессам.

Существует более двадцати различных сигналов. Основные:

SIGCHLD - сигнал о завершении дочернего процесса.

SIGHUP - сигнал освобождения линии. Посылается всем процессам, подключенным к управляющему терминалу при отключении терминала. Многие демоны при получении данного сигнала заново просматривают файлы конфигурации и перезапускаются.

SIGINT - сигнал посылается всем процессам сеанса, связанного с терминалом, при нажатии пользователем клавиши прерывания (CTRL-C).

SIGTERM - сигнал приводит к немедленному прекращению работы получившего сигнал процесса.

SIGKILL - сигнал приводит к немедленному прекращению работы получившего сигнал процесса.

В отличие от SIGTERM процесс не может блокировать и перехватывать данный сигнал.

SIGSEGV - сигнал посылается процессу, если тот пытается обратиться к неверному адресу памяти.

SIGSTOP - сигнал приводящий к остановке процесса. Для отправки сигнала SIGSTOP активному процессу текущего терминала можно воспользоваться комбинацией клавиш (CTRL-Z).

SIGCONT - сигнал возобновляющий работу остановленного процесса.

SIGUSR1, SIGUSR2 - сигналы определяемые пользователем.

Для того, чтобы отправить процессу сигнал можно использовать команду **kill**. Для того, чтобы процесс мог отправить сигнал другому процессу необходимо чтобы эффективные идентификаторы пользователя у посылающего процесса и у процесса получателя совпадали. Процессы с эффективным идентификатором пользователя равным нулю могут посылать сигналы любым процессам.

Каналы

Часто возникает ситуация когда два процесса последовательно обрабатывают одни и те же данные. Для обеспечения передачи данных от одного процесса к другому в подобных ситуациях используются программные каналы. Программный канал (pipe) служит для установления связи, соединяющей один процесс с другим. Запись данных в канал и чтение из него осуществляются при помощи системных вызовов **write** и **read**, т.е. работа с каналами аналогична работе с файлами. Для создания программного канала используется системный вызов **pipe**. Вызов возвращает два дескриптора файлов, первый из которых открыт для чтения из канала, а второй для записи в канал.

Каналы используются, например, при организации конвейера. При выполнении команды:

```
find /usr/bin -name a* | sort
```

создается канал, команда **find** выводит в него результаты своей работы, а команда **sort** считывает из этого канала данные для сортировки.

Главным недостатком программных каналов является то, что они могут использоваться только для связи процессов имеющих общее происхождение (напр., родительский процесс и его потомок). Другой недостаток ограниченное время существования канала (программные каналы уничтожаются после завершения обращающегося к ним процесса).

Именованные каналы идентичны программным в отношении записи и чтения данных, но они являются объектами файловой системы. Именованный канал имеет имя, владельца и права доступа. Открытие и закрытие именованного канала осуществляется как открытие и закрытие любого файла, но при чтении и записи он ведет себя аналогично каналу.

Для создания именованного канала используется команда **mkfifo**. Если некоторый процесс

открывает именованный канал для записи, то этот процесс блокируется до тех пор, пока другой процесс не откроет этот канал для чтения, и наоборот.

Команды для работы с процессами

nice уровень nice (приоритет).
pgid идентификатор группы процессов.
pid идентификатор процесса.
ppid идентификатор родительского процесса.
rgid, ruid реальные идентификаторы группы и пользователя.
uid реальный идентификатор пользователя.
tty управляющий терминал

Для различных систем параметры и ключевые слова могут сильно различаться. Подробности об использовании ps на конкретной системе можно получить при помощи команды man ps.

kill [-s signal| -signal] pid посылает сигнал указанному процессу. Если значение сигнала опущено, предполагается SIGTERM. signal — символическое имя сигнала без префикса SIG, либо номер сигнала.

Пример:

kill -HUP 172 — послать сигнал SIGHUP процессу с идентификатором 172.

nice [-nice] команда [аргументы]

Выполняет команду с меньшим приоритетом. Если nice не задан, то предполагается 10. Значение nice может быть от -20 (наивысший приоритет) до 20 (наименьший приоритет). Отрицательные числа задаются как -nice. Увеличение приоритета может выполнить только суперпользователь.

Пример:

nice -10 john users — запустить программу john с пониженным приоритетом.

mkfifo [-m режим_доступа] имя

Создает именованный канал с указанным именем и режимом доступа.

tty Выводит имя текущего терминала.

who [am i] Выводит список пользователей работающих в системе.

uname [-amnrsv] Выводит информацию о системе.

uptime Выводит время работы системы и ее среднюю загрузку за последние 5, 10 и 15 минут.

Средства оболочки предназначенные для работы с процессами

Список — последовательность из одного или более конвейеров разделенных операторами

&, && или ||. Более высокий приоритет у операторов && и ||.

Если команда завершается оператором &, то оболочка выполняет ее в фоновом режиме. Если между двумя командами стоит оператор &&, то вторая команда будет выполнена только в том случае, если первая завершится успешно. Если между двумя командами стоит ||, то вторая команда будет выполнена только в том случае, если код завершения первой команды отличен от нуля. Если команды разделены точкой с запятой, то вторая команда будет выполнена после завершения первой, независимо от результата выполнения первой команды.

Оболочка содержит несколько встроенных команд для работы с процессами:

wait [pid]

Ожидает завершения процесса с указанным идентификатором. Если идентификатор опущен, то ожидает завершения всех процессов запущенных оболочкой.

exec команда [аргумент]...

Указанная команда заменяет оболочку и получает в качестве параметров заданные аргументы.

exit [n]

Приводит к завершению оболочки с кодом завершения n. Если аргумент опущен, то код завершения ноль.

trap [действие условие...]

Устанавливает обработчик события. Условие либо EXIT, либо имя сигнала без префикса SIG. EXIT соответствует завершению работы оболочки. Если действие равно "-", то обработчик сбрасывается в значение по умолчанию. Например, после выполнения команды:

trap "echo PRESSED" INT

оболочка будет выводить слово PRESSED после каждого нажатия клавиш CTRL-C. (Нажатие клавиш CTRL-C приводит к посылке сигнала SIGINT процессам подключенным к терминалу).

Самостоятельная работа

1. Запишите в отчет следующую информацию о процессах запущенных пользователем root. Вывод должен быть отсортирован по номеру процесса.

2. Создайте текстовый файл dao.txt, написав в него свою фамилию и имя

3. Создайте в домашнем каталоге именованный канал fifo. Выполните команду

cat /home/dao.txt >fifo

Теперь с другого терминала выполните команду

cat fifo

6. При помощи команд tty, w, uname, uptime выведите в файл отчета имя текущего терминала, информацию о пользователях, работающих в системе, название и версию операционной системы, время работы системы.

7. Установите обработчик сигнала SIGINT. Для этого выполните команду:

trap "echo и получен сигнал SIGINT " INT

Пошлите несколько раз оболочке сигнал SIGINT. Для этого следует нажать клавиши CTRL-C.

Лабораторная работа №5 (1 час) **Процессы в Windows XP**

Процессы мы можем увидеть в диспетчере задач. Откройте диспетчер задач. Нажмите вкладку процессы и изучите содержимое таблицы

CSRSS.EXE

Процесс отвечает за окна консоли, за создание и удаление потоков, а также частично за работу 16-битной среды MS-DOS. Он относится к подсистеме Win32 пользовательского режима (WIN32.SYS же относится к ядру Kernel) и должен всегда выполняться.

EXPLORER.EXE

Пользовательская среда, содержащая такие компоненты, как Панель задач, Рабочий стол и тому подобное. Его практически всегда можно закрывать и снова открывать без каких-либо последствий.

INTERNAT.EXE

Загружает различные выбранные пользователем языки ввода, показывает на панели задач значок >, который позволяет переключать языки ввода. С помощью панели управления возможно без использования данного процесса безо всяких проблем переключать раскладку клавиатуры.

LSASS.EXE

Этот локальный сервер авторизации отвечает за IP-директивы безопасности (интернет- протоколы) и загружает драйвер безопасности. Он запускает процесс, отвечающий за авторизацию пользователей. При успешной авторизации пользователя приложение создаёт и присваивает ему специальный протокол. Все запущенные далее процессы используют этот протокол.

MSTASK.EXE

Отвечает за службу планирования Schedule, которая предназначена для запуска различных приложений в определённое пользователем время.

SMSS.EXE

Диспетчер сеансов запускает высокоуровневые подсистемы и сервисы. Процесс отвечает за различные действия, например запуск Winlogon и Win32 процессов, а также за операции с системными переменными. Когда Smss определяет, что Winlogon или Csrss закрыты, он автоматически выключает систему.

SPOOLSV.EXE

Обеспечивает создание очереди на печать, временно сохраняя документы и факсы в памяти.

SVCHOST.EXE

Этот всеобъемлющий процесс служит хостингом для других процессов, запускаемых с помощью DLL. Поэтому иногда работают одновременно несколько Svchost. С помощью команды > можно вывести на экран все процессы, использующие Svchost.

SERVICES.EXE

Процесс управления системными службами. Запуск, окончание, а также все остальные действия со службами происходят через него.

SYSTEM

Выполняет все потоки ядра Kernel.

SYSTEM IDLE PROCESS

Этот процесс выполняется на любом компьютере. Нужен он, правда, всего лишь для мониторинга процессорных ресурсов, не используемых другими программами.

TASKMGR.EXE

Процесс диспетчера задач, закрывать который крайне не рекомендуется.

WINLOGON.EXE

Отвечает за управление процессами авторизации пользователей.

WINMGTM.EXE

Основной компонент клиентской службы Windows. Процесс запускается одновременно с первыми клиентскими приложениями и выполняется при любом запросе служб. В Windows XP процесс запускается как клиент процесса Svchost.

Попробуйте завершать некоторые процессы. Что при этом происходит?

Лабораторная работа 6 (1 час) **Взаимодействие процессов с помощью pipe**

Понятие о pipe. Системный вызов pipe()

Наиболее простым способом для передачи информации с помощью потоковой модели между различными процессами или даже внутри одного процесса в операционной системе Linux является pipe (канал, труба, конвейер).

Важное отличие pipe от файла заключается в том, что прочитанная информация немедленно удаляется из него и не может быть прочитана повторно.

Pipe можно представить себе в виде трубы ограниченной емкости, расположенной внутри адресного пространства операционной системы, доступ к входному и выходному отверстию которой осуществляется с помощью системных вызовов. В действительности pipe представляет собой область памяти, недоступную пользовательским процессам напрямую, зачастую организованную в виде кольцевого буфера (хотя существуют и другие виды организации). По буферу при операциях чтения и записи перемещаются два указателя, соответствующие входному и выходному потокам. При этом выходной указатель никогда не может

перегнуть входной и наоборот. Для создания нового экземпляра такого кольцевого буфера внутри операционной системы используется системный вызов `pipe()`.

Системный вызов `pipe`

Прототип системного вызова

```
#include <unistd.h>
```

```
int pipe(int *fd);
```

Описание системного вызова

Системный вызов `pipe` предназначен для создания `pip`'а внутри операционной системы.

Параметр `fd` является указателем на массив из двух целых переменных. При нормальном завершении вызова в первый элемент массива – `fd[0]` – будет занесен файловый дескриптор, соответствующий выходному потоку данных `pip`'а и позволяющий выполнять только операцию чтения, а во второй элемент массива – `fd[1]` – будет занесен файловый дескриптор, соответствующий входному потоку данных и позволяющий выполнять только операцию записи.

Возвращаемые значения

Системный вызов возвращает значение 0 при нормальном завершении и значение -1 при возникновении ошибок.

В процессе работы системный вызов организует выделение области памяти под буфер и указатели и заносит информацию, соответствующую входному и выходному потокам данных, в два элемента таблицы открытых файлов, связывая тем самым с каждым `pip`'ом два файловых дескриптора. Для одного из них разрешена только операция чтения из `pip`'а, а для другого – только операция записи в `pip`. Для выполнения этих операций мы можем использовать те же самые системные вызовы `read()` и `write()`, что и при работе с файлами. Естественно, по окончании использования входного или/и выходного потока данных, нужно закрыть соответствующий поток с помощью системного вызова `close()` для освобождения системных ресурсов. Необходимо отметить, что, когда все процессы, использующие `pipe`, закрывают все ассоциированные с ним файловые дескрипторы, операционная система ликвидирует `pipe`. Таким образом, время существования `pip`'а в системе не может превышать время жизни процессов, работающих с ним.

Прогон программы для `pipe` в одном процессе

Достаточно яркой иллюстрацией действий по созданию `pip`'а, записи в него данных, чтению из него и освобождению выделенных ресурсов может служить программа, организующая работу с `pip`'ом в рамках одного процесса, приведенная ниже:

```
/* Программа 05-2.c, иллюстрирующая работу с pip'ом в рамках одного процесса */
```

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int main(){
    int fd[2];
    size_t size;
    char string[] = "Hello, world!";
    char resstring[14];
    /* Попробуем создать pipe */
    if(pipe(fd) < 0){
        /* Если создать pipe не удалось, печатаем об этом сообщение
        и прекращаем работу */
        printf("Can't create pipe\n");
        exit(-1);
    }
    /* Пробуем записать в pipe 14 байт из нашего массива, т.е. всю
    строку "Hello, world!" вместе с признаком конца строки */
    size = write(fd[1], string, 14);
    if(size != 14){
        /* Если записалось меньшее количество байт, сообщаем об
        ошибке */
        printf("Can't write all string\n");
        exit(-1);
    }
    /* Пробуем прочитать из pip'a 14 байт в другой массив, т.е. всю
    записанную строку */
    size = read(fd[0], resstring, 14);
    if(size < 0){
        /* Если прочитать не смогли, сообщаем об ошибке */
        printf("Can't read string\n");
        exit(-1);
    }
    /* Печатаем прочитанную строку */
    printf("%s\n",resstring);
    /* Закрываем входной поток*/
    if(close(fd[0]) < 0){
        printf("Can't close input stream\n");
    }
}
```

```

}
/* Закрываем выходной поток*/
if(close(fd[1]) < 0){
    printf("Can't close output stream\n");
}
return 0;
}

```

Программа 05-2.c, иллюстрирующая работу с pip'ом в рамках одного процесса

Самостоятельная работа

1. Наберите программу, откомпилируйте ее и запустите на исполнение.
2. Проанализируйте результаты

Лабораторная работа № 7 (2 часа) Изучение файловой системы Linux

1 Общие сведения

Linux — многопользовательская, многозадачная операционная система с разделением времени. В любой момент в системе выполняется множество процессов, каждый процесс принадлежит некоторому пользователю. Пользователь это объект обладающий определенными правами в системе. Каждый пользователь идентифицируется уникальным идентификатором пользователя (**UID — user identifier**). Пользователю присваиваются имя и пароль. Пользователь с **UID 0 (root)** обладает неограниченными правами. Кроме того каждый пользователь входит в одну или несколько групп. Принадлежность к группе добавляет пользователю определенные права в системе. Каждая группа идентифицируется уникальным идентификатором группы (**GID — group identifier**). Информация о пользователях хранится в файле /etc/passwd. Каждая строка файла содержит информацию об одном пользователе: регистрационное имя, зашифрованный пароль, UID, GID, полное имя, домашний каталог, командную оболочку. **Командная оболочка** (командный интерпретатор, **shell**) — средство интерактивного взаимодействия с системой. **Домашний каталог** — каталог в котором хранятся файлы пользователя. При входе пользователя в систему этот каталог становится текущим для оболочки.

2 Файловая система

Файловая система — это структура, с помощью которой ядро операционной системы организует и представляет пользователям ресурсы памяти системы. Сюда относится память на различного рода носителях информации. Емкость и количество носителей различно в разных системах. Ядро объединяет эти ресурсы в единую иерархическую структуру, которая начинается в каталоге / и разветвляется, охватывая произвольное число подкаталогов.

Цепочка имен каталогов, через которые необходимо пройти для доступа к заданному файлу, вместе с именем этого файла называется путевым именем файла (pathname). Путевые имена могут быть полными или относительными. В любой момент каждый процесс привязан к некоторому текущему каталогу. Относительные имена интерпретируются с текущего каталога.

Файловое дерево может быть произвольного размера. Однако существуют определенные ограничения зависящие от конкретной операционной системы. Как правило имя каталога не должно содержать более 256 символов, а в определении одного пути не должно быть более 1023 символов.

В ОС Linux существует восемь типов файлов:

Обычный файл— это просто последовательность байтов. Обычный файл может содержать выполняемую программу, главу книги, графическое изображение и т.п.

Каталоги -могут содержать файлы любых типов в любых сочетаниях. Специальные имена . и ..обозначают соответственно сам каталог и его родительский каталог.

Файлы устройств - позволяют программам взаимодействовать с аппаратными средствами и периферийными устройствами системы. При конфигурировании ядра к нему добавляются те модули, которые знают, как взаимодействовать с каждым из устройств системы. За всю работу по управлению конкретным устройством отвечает специальная программа, называемая драйвером устройства.

Драйверы устройств образуют стандартный коммуникационный интерфейс, который выглядит как обычный файл. Когда ядро получает запрос к байт-ориентированному или блок-ориентированному файлу устройства, оно просто передает этот запрос соответствующему драйверу устройства.

Каждому типу устройств системы может соответствовать несколько файлов устройств.

Поэтому файлы устройств характеризуются двумя номерами: старшим и младшим. Старший определяет драйвер, а младший конкретное устройство.

Доменные гнезда (sockets) Linux — это соединения между процессами, которые позволяют им взаимодействовать, не подвергаясь влиянию других процессов. Доменные гнезда Linux локальны для конкретного хост-компьютера. Обращение к ним осуществляется через объект файловой системы, а не через сетевой порт.

Именованные каналы - также как и доменные гнезда обеспечивают взаимодействие двух несвязанных процессов, выполняемых на одной машине.

Жесткие ссылки — это скорее не тип файла, а его дополнительное имя. У каждого файла имеется как минимум одна ссылка. Как правило, это имя, под которым он был создан. Добавлением ссылки создается псевдоним файла. Ссылку невозможно отличить от имени файла, к которому она присоединена: в ОС Linux они идентичны. Linux подсчитывает количество ссылок, указывающих на каждый файл, и не освобождает блоки данных файла до тех пор, пока не удалит его последнюю ссылку.

Символические ссылки— обеспечивают возможность указывать вместо путевого имени файла имя ссылки. Символическая ссылка содержит путь к файлу, на который она ссылается.

Имена файлов могут состоять из любых символов, за исключением слэша и символа с кодом ноль. Максимальная длина имени файла определяется конкретной системой. Для каждого файла определен владелец этого файла и группа владельца данного файла. Для каждого файла определяются права доступа владельца файла, группы, всех остальных. Есть три типа прав доступа: чтение, запись, выполнение/поиск. Изменить права доступа к файлу может только владелец и суперпользователь (root).

3 Перенаправление потоков

В ОС *LINUX* существует три стандартных потока: поток ввода, поток вывода и поток стандартного протокола (поток ошибок).

Стандартный ввод (*STDIN*) — это просто то место, откуда система ожидает получение данных. Обычно — это клавиатура, хотя это может быть командный сценарий или программа. Когда вы изменяете стандартное значение, вы вызываете перенаправление ввода.

Аналогично, стандартный вывод (*STDOUT*) — то место, которое подразумевается операционной системой для вывода информации, обычно — экран монитора. Можно так же перенаправить вывод, *STDOUT*, по усмотрению любой команды или командного сценария. Цепочка событий от *STDIN* до *STDOUT* выглядит примерно так:

стандартный ввод *STDIN* ⇔ команда Linux ⇔ стандартный вывод *STDOUT*

STDIN часто упоминается как *fd0*, или дескриптор 0, *STDOUT* упоминается как *fd1*. Есть также стандартное устройство вывода ошибок (*STDERR*), которому система сообщает о любых ошибках в выполнении программ или командного сценария. Стандартно — это также терминал.

Существуют всего четыре символа перенаправления стандартного ввода/вывода:

- Символ > используется для перенаправления стандартного вывода команды в файл. Например, `cat > file` запишет выводимую командой `cat` информацию в файл `file`.
- Символ < используется для переназначения стандартного ввода команды. Например, при выполнении команды `cat < file` в качестве стандартного ввода `cat` используется не клавиатура, а файл `file`.
- Символ >> используется для дописывания стандартного вывода команды в конец существующего файла. Например, `cat >> file` допишет выводимую командой `cat` информацию в конец файла `file`.
- Символ | является символом потока. Он используется, когда вы хотите передать стандартный вывод одной команды на стандартный ввод другой.

Примеры:

`cat > filename` – перенаправление вывода в файл `filename` (если этот файл существует, то его прежнее содержимое будет утеряно);

`cat >> filename` – добавить содержимое вывода к содержимому файла `filename`;

`cat < filename` – сформировать стандартный ввод из содержимого файла `filename`.

`cat /home/qqq/Desktop/file_1 | wc -w > /home/qqq/Desktop/num_names` – содержимое файла `file_1` передается в распоряжение команды `wc -w` (счетчик слов), которая выводит количество слов в файле `num_names`, находящийся на рабочем столе (`/home/qqq/Desktop/`).

`ls | wc -l`

Те же действия можно организовать так:

`ls > buffer`

`wc -l < buffer`

`rm -f buffer`

Команда `ls` выводит в файл `buffer` список файлов текущего каталога, а команда `wc -l` считает количество строк в этом файле, команда `rm -f buffer` удаляет файл с именем `buffer`.

Таким образом, первые две команды позволяют посчитать количество файлов в текущем каталоге.

4 Основы работы с командным интерпретатором

слово — последовательность символов, воспринимаемая интерпретатором как одна единица.

имя — слово состоящее только из алфавитно-цифровых символов и знаков подчеркивания, начинающееся с буквы или знака подчеркивания. Также называется идентификатором.

Метасимвол — символ, который, не будучи заключен в кавычки, разделяет слова. Один из следующих:

| & ; () < > space tab

управляющий оператор — слово выполняющее функции управления. Один из следующих:

|| && ; ; ; () | <newline>

Зарезервированные слова — это слова имеющие специальное значение для интерпретатора.

Следующие слова являются зарезервированными:

! case do done elif else esac fi for function if

in select then until while { } time [[]]

Простая команда это последовательность из необязательного присвоения значения переменной с последующими словами и перенаправлениями, прерываемая управляющим оператором.

Первое слово определяет выполняемую команду. Последующие слова передаются команде в качестве аргументов.

[VAR=val] command argument ...

Возвращаемое значение простой команды — код завершения или 128+n если команда была прервана по сигналу n.

Конвейер — последовательность из одной или более команд, разделенных символом |. Формат конвейера следующий:

```
[time [-p]] [!] command [ | command2 ... ]
```

Стандартный вывод `command` подключается к стандартному вводу команды `command2`. Это подключение производится до выполнения любых перенаправлений.

Если конвейеру предшествует зарезервированное слово `!`, то код завершения конвейера равен логическому отрицанию кода завершения последней команды. Иначе код завершения конвейера равен коду завершения последней команды. Интерпретатор ожидает завершения всех команд до того как вернет значение.

Если конвейеру предшествует зарезервированное слово `time`, то после завершения выполнения конвейера будет выведена информация о времени выполнения конвейера и о затраченном времени процессора в режимах пользователя и системы.

Каждая команда в конвейере выполняется как отдельный процесс (т.е. в подоболочке).

5 Переменные окружения

У каждого процесса имеется область памяти называемая программным окружением (`program environment`) — это набор строк, заканчивающихся нулевым символом. Эти строки называются переменными окружения. Каждая строка имеет вид: имя переменной = значение. Имя переменной может состоять из алфавитно-цифровых символов и знака подчеркивания. Цифра не может быть первым символом имени. Присвоение значения переменной в оболочке производится следующим образом:

Имя = Значение

Для того, чтобы значение переменной передавалось процессам порождаемым оболочкой, следует использовать встроенную команду `export`. Следующие две команды помечают переменные `VAR` и `TST` как экспортируемые и присваивают переменной `TST` значение `/usr/doc`:

```
export VAR
export TST=/usr/doc
```

Для того, чтобы просмотреть значения переменных окружения можно использовать команду `set`, которая выводит значения всех переменных окружения.

Для того, чтобы получить значение переменной, перед ее именем указывается знак доллара. Такое выражение будет заменяться интерпретатором на значение переменной. Например, команда `echo` выводит в стандартный вывод свои аргументы, следующее выражение:

```
echo TST=$TST
выведет на экран TST=/usr/doc (при условии, что значение переменной TST – /usr/doc).
```

6 Основные команды для работы с файлами

cd [каталог]

Меняет текущий каталог на указанный. Если параметр опущен, то текущим становится домашний каталог.

ls [-alFR] [файл ...]

Выводит список файлов в указанном (или текущем) каталоге. Ключ `-a` заставляет выводить все файлы, ключ `-l` служит для вывода подробной информации о файлах, ключ `-F` приводит к тому, что к именам каталогов добавляется символ `/`, к именам ссылок `@`, к именам выполняемых файлов `*`. При использовании ключа `-R` выводится список файлов не только указанного каталога, но и его подкаталогов.

touch файл ...

Меняет время доступа и изменения файла. Если файл не существовал, то он будет создан.

mkdir каталог

Создает каталог.

rmdir каталог

Удаляет каталог.

cp [-rp] файл1 файл2

cp [-rp] файл ... каталог

Копирует один файл в другой или копирует файлы в указанный каталог. Ключ `-R` предназначен для копирования каталогов, ключ `-r` позволяет сохранять владельцев файлов, режим доступа и время доступа и изменения.

rm [-r] файл ...

Удаляет файлы. Ключ `-r` позволяет удалять каталоги.

mv файл1 файл2

```
mv file ... directory
```

Перемещает один файл в другой или перемещает файлы в заданный каталог.

cat [файл ...]

Объединяет содержимое указанных файлов и выводит на стандартный вывод.

find путь выражение

Команда предназначена для поиска файлов. Находит файлы для которых значение выражения "истина". Для определения выражений могут использоваться следующие примитивы:

-name шаблон -возвращает значение истина если файл соответствует шаблону.

-nouser -возвращает значение истина, если идентификатор пользователя не определен.

-nogroup -возвращает значение истина, если идентификатор группы не определен.

-perm режим возвращает значение истина, если файл имеет заданный режим доступа. Режим может задаваться как в символьной форме (`-`, `+`, `=`), так и в числовой.

-type тип значение истина, если файл имеет указанный тип. (`b` – блок-ориентированное устройство, `c` – байт-ориентированное устройство, `d` – каталог, `f` – регулярный файл, `p` – канал).

-**links n** значение истина, если файл имеет указанное число ссылок.
 -**user пользователь** значение истина если файл принадлежит указанному пользователю.
 -**group группа** значение истина если файл принадлежит указанной группе.
 -**size [+|-] размер[c]** истина, если размер файла в блоках (байтах, если используется размерс) равен (больше, если используется +; меньше, если используется -) заданному.
 -**atime [+|-]d** истина, если доступ к файлу производился между (d-1)*24 и d*24 часов назад (+ более d*24 часов назад, - менее (d-1)*24 часов назад). Аналогично -mtime для времени изменения содержимого файлов и -ctime для времени изменения статуса файлов.
 -**exec программа [аргументы]**; исполнение программы для каждого найденного файла. Имя программы и аргументы состоящие только из двух символов {} будут заменены именем найденного файла. Заключительному знаку ; должен предшествовать \.
 -**ok программа [аргументы]**; аналогична предыдущей команде, но выводит запрос на подтверждение.
 -**print** выводит имя найденного файла.
 Параметры могут объединяться следующим образом:
 (выражение), группировка выражений
 !выражение, отрицание выражений
 выражение1 [-a] выражение2, логическое И
 выражение1 -o выражение2, логическое ИЛИ.

Самостоятельная работа

1. Войдите в систему под выданной пользовательской учетной записью.
2. Создайте в домашнем каталоге при помощи команды touch файл отчета с именем lab1.txt.
3. При помощи команды set просмотрите значения переменных окружения.
4. Выведите в файл lab1.txt значения переменных окружения PATH, LANG, HOME.
5. Просмотрите полученный файл при помощи команды less.
6. Последовательно перейдите в каталоги /bin, /usr, /etc, /usr/bin. Выполните в каждом каталоге команду ls с различными ключами (-a, -l, -F, -R). Если вывод команды ls не умещается на экране, то можно воспользоваться командой less (напр. ls -l | less).
7. Перейдите обратно в домашний каталог (команда cd).
8. Изучите команды uname и date. Просмотрите справку об этих командах (man uname, man date).
9. Сравните вывод команд date и LANG=C date.
10. Добавьте в конец файла отчета, используя перенаправление вида >>, информацию выводимую командами uname -a и date.
11. Создайте в домашнем каталоге подкаталоги test1 и test2.
12. Скопируйте файл /home/labs/text.txt в каталог test2. Добавьте в файл отчета вывод команды ls -R.
13. Переместите файл text.txt из каталога test2 в каталог test1. Снова добавьте в файл отчета вывод команды ls -R.
14. Изучите команду find. Добавьте в файл отчета все подкаталоги каталога /usr/share содержащие в своем имени сочетание букв "ru" (используйте параметры -type и -name).
15. Воспользуйтесь командой gmdir для удаления каталогов созданных при выполнении пункта 12.
16. Добавьте в конец файла отчета две строки. Первая должна содержать текст: "Лабораторная работа No ". Вторая должна содержать Ваши имя и фамилию.

Лабораторная работа № 8 (2 часа) Работа с файлами и директориями в Linux

Войдите в графическую консоль linux.

Создание, редактирование и сохранение файлов и папок

Создадим новый текстовый файл.

1. Зайдите в папку **Documents**
2. В правой части файл менеджера щелкните на любом свободном месте правой клавишей мыши и вызовите контекстное меню.
3. Выберите пункт **Создать**. В появившемся окне выберите пункт **Текстовый файл...** В появившемся окне введите любое имя (например, **Текст пример**) и нажмите кнопку **ОК**.
4. В папке **Documents** (Документы) появился новый файл.
5. Давайте попробуем отредактировать его. Щелкните по названию файла. Файл менеджер сам выберет программу, которая может редактировать текстовые файлы и откроет ее. В данном случае это программа Kwrite. Она предназначена для простейшего редактирования текстовых файлов.
 В текстовом редакторе наберите:

Я очень долго добирался от Тифлиса до Киева.

В Киев поезд пришел к вечеру. Был широкий разгар весны, цвели каштаны, на Куполах Владимирского собора горел горячий блеск заката, нарядно шумел Крещатик. И тем беднее и опустошеннее показалась мне комнатка, где жили мама и сестра Галя.

Прошло больше двух лет с тех пор, как я уехал из Киева в Одессу, а потом в Тифлис. За это время мама и Галя постарели, но стали спокойнее.

6. Сохраним результаты нашей работы. Выберите в меню команду **Файл** и в ней — команду **Сохранить**.
7. Закройте текстовый редактор.
Вы также можете создать текстовый файл непосредственно из текстового редактора.

Копирование и перемещение файлов и папок

Рассмотрим несколько примеров работы с файлами и папками.

Допустим, вы закончили работу над очередным проектом и теперь несколько рабочих файлов, которые вы создали в процессе подготовки проекта необходимо разместить в таком месте, чтобы вы впоследствии могли легко их найти.

Для этого такие материалы лучше располагать в отдельной папке с названием, хорошо отражающим ее содержимое. Совершенно очевидно, что отчетные материалы лучше всего хранить в папке **Проекты**.

Создайте папку **Проекты**. Щелкните правой кнопкой мыши в правой части окна файл менеджера и в контекстном меню выберите команду **Создать** и потом **Папку**.

Введите имя новой папки **Проекты А** и в ней создайте две папки с файлами проектов – **Проект 1** и

Проект 2.

В папке **Проект 1** создайте три файла а1, а2, а3 со следующими текстами:

A1

Несколько дней лил, не переставая, холодный дождь. В саду шумел мокрый ветер. В четыре часа дня мы уже зажигали керосиновые лампы, и невольно казалось, что лето окончилось навсегда и земля уходит все дальше и дальше в глухие туманы, в неуютную темень и стужу.

A2

При каждой возможности я посылал маме деньги и все время мучился, что денег мало и доходят они с перерывами. Но мама не жаловалась. Я убедился, что характер у нее действительно был стоический

A3

По ночам часто плакал во сне Фунтик – маленькая рыжая такса. Приходилось вставать и закутывать его теплой шерстяной тряпкой. Фунтик благодарил сквозь сон, осторожно лизал руку и, вздохнув, засыпал. Темнота шумела за стенами плеском дождя и ударами ветра, и страшно было подумать о тех, кого, может быть, застигла эта ненастная ночь в непроглядных лесах.

Переместим эти файлы в папку **Проект 2**

Прежде всего, эти три файла надо выделить. Для этого нажмите клавишу **Ctrl** и, не опуская ее, щелкните левой клавишей мышки по каждому из этих файлов.

Теперь в меню **Правка** выберите команду **Вырезать**. Вы увидите, что изображения двух выделенных файлов потускнели.

В левой панели экрана раскройте папку **Отчеты** (щелкните по стрелке рядом с ее названием - все входящие в нее папки появятся в виде списка ниже).

Откройте папку **Проект 2**, щелкнув по ней мышкой.

Опять выберите меню **Правка** и команду **Вставить 3 файла**.

Три файла переместятся на новое место.

Обратите внимание, что три файла пропали из папки **Проект 1** и появились в папке **Проект 2**. Команда **Вырезать** всегда переносит файлы, т.е. удаляет их из старого места и размещает в новом.

Если вы хотите скопировать файлы, т.е. сохранить их на старом месте после переноса на новое, воспользуйтесь в меню **Правка** командой **Копировать**.

Команда **Копировать** чаще всего используется, когда вы переносите файлы на внешние носители, чтобы перенести их на другой компьютер.

Скопируем три выделенных файла в папку **Проект 1.**, используя совершенно другую технику. Она называется **«перетяни и брось»**.

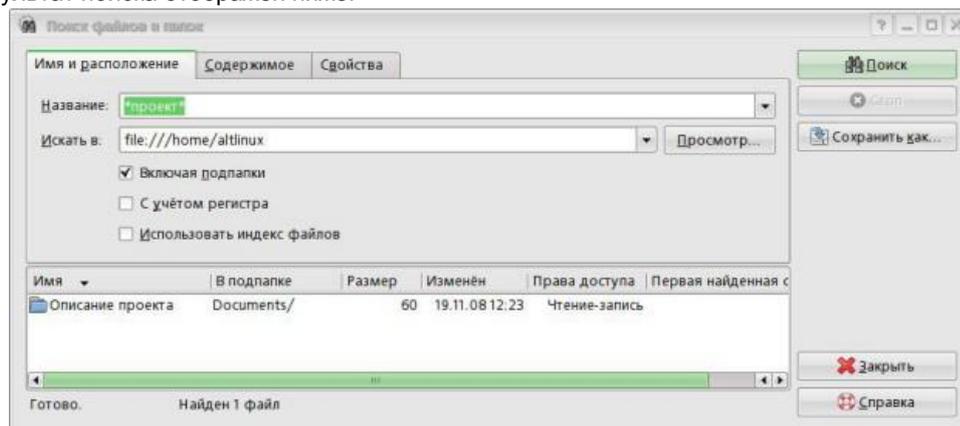
Зайдите в папку **Проект 2.**, выделите все файлы, щелкните по выделенным файлам левой кнопкой мыши и не отпускайте ее, перетащите файлы на название папки **Проект 1** в левой части окна. И опустите кнопку мыши. Появится диалоговое окно, в котором вам будет предложено скопировать или переместить файлы. Выберите команду **Копировать сюда** и все три файла благополучно будут скопированы в эту папку.

Навигация в файловой системе. Поиск файлов и папок

Давайте попробуем найти папку **Описание проекта**. Допустим, мы забыли полное название файла, но помним, что в его названии было слово **«проект»**.

Запустим **Поиск файлов и папок**.

Сформируем в строке **Название** следующий запрос: «*проект*». Это значит, что в названии файла и слева и справа от слова проект может быть любое количество символов. Результат поиска отображен ниже.



Попробуем теперь найти файл по содержащейся в нем информации.

Прежде всего необходимо очистить предыдущий запрос. Далее перейдите на вкладку **Содержимое**. Наберите в строке **Содержит текст** поисковый запрос – слово **каштаны**. Нажмите на кнопку Поиск.

В окне с результатами поиска появится список найденных файлов, среди них и созданный нами раньше файл **Текст пример**, который содержит фразу: «**цвели каштаны**».

Лабораторные работа №9 (2 часа) **Работа с файлами и директориями в Windows**

1. Создать на Рабочем столе каталог **МОЙ КАТАЛОГ**, а в нем подкаталоги **ДОКУМЕНТЫ** и **ПРОГРАММЫ**.
2. Перейти на устройство **C:** в подкаталог **МОИ ДОКУМЕНТЫ**.
3. Сделать копию любого файла в **МОЙ КАТАЛОГ** при помощи **контекстного меню**.
4. Сделать копию любого файла из папки **C:\МОИ ДОКУМЕНТЫ** в **МОЙ КАТАЛОГ** в подкаталог **ДОКУМЕНТЫ** при помощи команд **Копировать** и **Вставить**.
5. Сделать копию любого файла из папки **C:\МОИ ДОКУМЕНТЫ** в **МОЙ КАТАЛОГ**, подкаталог **ПРОГРАММЫ** при помощи буксировки мыши.
6. Назвать скопированные файлы другими русскими именами.
7. Задать второму файлу атрибут только для чтения, а третьему - скрытый.
8. Показать скрытые файлы.
9. Скрытый файл удалить.
10. Весь каталог **ПРОГРАММЫ** скопировать на Рабочий стол, а из папки **МОЙ КАТАЛОГ** удалить.
11. Найти файлы (через поиск файлов) с именами **READ** или **README**.
12. Скопировать один файл через контекстное меню в папку **ДОКУМЕНТЫ**.
13. Показать для него атрибуты, а затем изменить по своему желанию.
14. Не показывать скрытые файлы.
15. Создать копию **МОЙ КАТАЛОГ**.
16. Показать работу преподавателю.

Лабораторная работа № 10 (2 часа) **Работа с программой fdisk.**

Технические средства Эмулятор программы **fdisk**

Запустите эмулятор программы **fdisk**.

Если у вас большой жёсткий диск - программа выдаст предупреждающее сообщение, выберете "Y" на предложение работать с большими дисками.

В главном меню 4 пункта:

- 1 - Создание раздела
- 2 - Установка активного раздела
- 3 - Удаление раздела
- 4 - Информация о дисках

Нажмите "4", а затем "Enter" - и получите информацию о текущем разбиении.

Нажмите "Esc" для возврата в главное меню.

Удаление существующих разделов

Если ваш диск уже был конфигурирован, то вначале придется удалить предыдущее разбиение. Чтобы сделать это, выберете

пункт 3 главного меню. На экране появится:

Delete DOS Partition or Logical DOS Drive

(Удаление разделов DOS или логических дисков)

Current fixed disk drive: 1

(Текущий привод: 1)

Choose one of the following:

(Выберите один из вариантов:)

1. Delete Primary DOS Partition

(1. Удалить первичный раздел DOS)

2. Delete Extended DOS Partition

(2. Удалить расширенный раздел DOS)

3. Delete Logical DOS Drive(s) in the Extended DOS

(3. Удалить логические диски из расширенного раздела DOS)

4. Delete Non-DOS Partition

(4. Удалить не-DOS раздел)

Порядок ваших действий:

- Удалите не DOS раздел(если он существует)
- Удалите логические диски из расширенного раздела DOS (если этот раздел существует), причем вам при удалении нужно будет несколько раз подтвердить свои намерения и ввести метку диска.
- Удалите сам расширенный раздел DOS (если он у вас был), причем при удалении необходимо будет несколько раз подтвердить свои намерения.
- Удалите первичный раздел DOS, причем вам при удалении необходимо будет несколько раз подтвердить свои намерения и ввести метку диска.

Теперь предыдущее разбиение удалено, и вы можете создать новое.

Создание нового разбиения

После удаления предыдущего разбиения нужно создать новое. Для этого надо выбрать опцию 1 из главного меню.

На экране появится сообщение:

Create DOS partition or Logical DOS Drive

(Создание разделов DOS или логических дисков)

Current fixed disk drive: 1

(Текущий привод: 1)

Choose one of the following:

(Выберите один из вариантов:)

1. Create Primary DOS Partition

(1. Создать первичный раздел DOS)

2. Create Extended DOS Partition

(2. Создать расширенный раздел DOS)

3. Create Logical DOS Drive(s) in Extended DOS Partition

(3. Создать логические диски в расширенном разделе DOS)

Вначале нужно создать первичный раздел DOS. Выберите пункт 1. Вам будет задан вопрос:

Do you wish to use the maximum available size for a Primary DOS Partition and make the partition active (Y/N.....)? []

(Вы хотите использовать для первичного раздела DOS все дисковое пространство и сделать этот раздел активным?)

Если вы ответите положительно, то на всем вашем винчестере будет создан один логический диск, с которого будет проводиться загрузка операционной системы.

В этом случае разбиение жесткого диска закончено. Если же вы хотите разбить свой винчестер на несколько логических, то надо ответить отрицательно. На экран будет выведен запрос:

Enter partition size in Mbytes or percent of disk space (%) to create a Primary DOS Partition.....: []

(Введите размер первичного раздела DOS в Мбайтах или в процентах от дискового пространства.)

После того как вы создадите первичный раздел нужно будет создать расширенный раздел DOS. (**Create Extended DOS Partition**)

На экране появится сообщение:

Enter partition size in Mbytes or percent of disk space (%) to create an Extended DOS Partition.....: []

(Введите размер расширенного раздела DOS в Мбайтах или в процентах от дискового пространства.)

Если вам не нужно оставлять место для работы с другой операционной системой(не MS-DOS), то вы должны указать все пространство на диске, оставшееся свободным, что и предлагается по умолчанию.

После того как вы создали расширенный раздел DOS нужно разбить его на логические диски. (**Create Logical DOS Drive(s) in Extended DOS Partition**)

Enter Logical drive size in Mbytes or percent of disk space (%)....[]

(Введите размер логического диска в Мбайтах или процентах от дискового пространства.)

Если вы хотите создать два логических (один-первичный раздел DOS, другой-полный размер расширенного раздела DOS),то нужно согласиться с предлагаемым по умолчанию максимальным размером. В противном случае введите ваш размер и повторите эту операцию для следующего логического диска.

После разбиения винчестера на разделы и логические диски нужно установить активный раздел (именно с него будет загружаться компьютер).

Установка активного раздела

Для загрузки с винчестера необходимо, чтобы первичный раздел DOS был активным. Выберите опцию 2 из главного меню и введите номер раздела с которого будет осуществляться загрузка:

Enter the number of the partition you want to make active.....: []

(Введите номер раздела, который вы хотите сделать активным)

При выводе на экран информации о разбиении активный раздел помечается буквой А.

После выполнения этих действий разбиение жесткого диска закончено. Осталось только отформатировать получившиеся логические диски.

Примечание:

Если ни один из разделов не сделан активным, то загрузиться с винчестера невозможно. При этом каких-либо сообщений не выводится.

Лабораторная работа № 11 (2 часа) Изучение состава внешних устройств на компьютере и их настроек.

Технические средства: Персональный компьютер, программа **Sysinfo.exe**, программа **checkit.exe**,

1. Загрузить программу
2. Нажать клавишу F1 и изучить инструкцию, необходимую для работы с данной программой.
3. После ознакомления необходимо проанализировать системные конфигурации ПЭВМ.
4. В отчет занести сводку о системе (F10=>Система=>Сводка по системе...).
5. По результатам работы составить отчет.
6. В выводе отразить ответ на вопрос: Для чего предназначена программа SYSINFO.EXE?
7. Загрузить диагностическую программу checkit.exe.
8. Нажать клавишу F1 и изучить инструкцию по работе программы.
9. Протестируйте системные ресурсы ПК (После запуска программы в начальный момент времени она начинает тестировать систему).
10. Произведите диагностику аппаратных средств ПК (клавиатуру, мышь и др.)
11. Результаты тестирования отразите в отчете.
12. Вывод должен содержать ответ на вопрос: Какая диагностическая программа производит тестирование в среде Windows? Есть ли различия с программой checkit.exe?

Лабораторная работа № 12. (2 часа) Файлы устройств. Аппарат прерываний. Добавление/удаление устройств

Технические средства Виртуальная машина с ОС Linux

1. Проверьте значение параметра **/proc/sys/net/ipv4/icmp_echo_ignore_all**, который отвечает за широковещательный **ping**. Значение 0 этого параметра означает, что ядро отвечает на ping. Значение 1 – ядро игнорирует ping

```
#cat /proc/sys/net/ipv4/icmp_echo_ignore_all
0
```

2. Убедитесь, что ваш компьютер отвечает на команду ping с другого компьютера. Нажмите Ctrl-C для завершения ping.

Пример:

```
[admin@lmaster4r2 ~]$ ping 192.168.128.115
PING 192.168.128.115 (192.168.128.115) 56(84) bytes of data.
64 bytes from 192.168.128.115: icmp_seq=1 ttl=128 time=0.718 ms
64 bytes from 192.168.128.115: icmp_seq=2 ttl=128 time=0.322 ms
64 bytes from 192.168.128.115: icmp_seq=3 ttl=128 time=0.255 ms
64 bytes from 192.168.128.115: icmp_seq=4 ttl=128 time=0.270 ms

--- 192.168.128.115 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.255/0.391/0.718/0.190 ms
[admin@lmaster4r2 ~]$
```

3. Измените параметр на "1".

```
#echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all
#cat /proc/sys/net/ipv4/icmp_echo_ignore_all
1
```

4. Еще раз протестируйте команду ping и убедитесь, что ваша система не отвечает на ping.
5. Перезагрузите компьютер

```
#reboot
6. Убедитесь, что значение параметра стало прежним(0).
```

```
#cat /proc/sys/net/ipv4/icmp_echo_ignore_all
0
```

7. Отредактируйте файл /etc/sysctl.conf от имени root, вставьте еще одну строку в конце файла:
net.ipv4.icmp_echo_ignore_all=1

8. Повторите пункт 5. или обновите параметры ядра без перезагрузки командой

```
#sysctl -p
```

9. Проверьте значение параметра

```
#cat /proc/sys/net/ipv4/icmp_echo_ignore_all
1
```

10. Отредактируйте файл /etc/sysctl.conf от имени root и удалите только что вставленную строку.

11. Вновь обновите параметры ядра

```
#sysctl -p
```

Лабораторная работа № 13 (2 часа)

Семейство протоколов TCP/IP

Сокеты в Linux

Создание сокета. Системный вызов socket()

Для создания сокета в операционной системе служит системный вызов `socket()`. Для транспортных протоколов семейства TCP/IP существует два вида сокетов: UDP-сокеты – сокет для работы с датаграммами, и TCP сокет – потоковый сокет. Однако понятие сокета (см. лекцию 14, раздел "Полные адреса. Понятие сокета (socket)") не ограничивается рамками только этого семейства протоколов. Рассматриваемый интерфейс сетевых системных вызовов (`socket()`, `bind()`, `recvfrom()`, `sendto()` и т. д.) в операционной системе UNIX может применяться и для других стеков протоколов (и для протоколов, лежащих ниже транспортного уровня).

При создании сокета необходимо точно специфицировать его тип. Эта спецификация производится с помощью трех параметров вызова `socket()`. Первый параметр указывает, к какому семейству протоколов относится создаваемый сокет, а второй и третий параметры определяют конкретный протокол внутри данного семейства.

Второй параметр служит для задания вида интерфейса работы с сокетом – будет это потоковый сокет, сокет для работы с датаграммами или какой-либо иной. Третий параметр указывает протокол для заданного типа интерфейса. В стеке протоколов TCP/IP существует только один протокол для потоковых сокетов – TCP и только один протокол для датаграммных сокетов – UDP, поэтому для транспортных протоколов TCP/IP третий параметр игнорируется.

В других стеках протоколов может быть несколько протоколов с одинаковым видом интерфейса, например, датаграммных, различающихся по степени надежности.

Для транспортных протоколов TCP/IP мы всегда в качестве первого параметра будем указывать предопределенную константу `AF_INET` (Address family – Internet) или ее синоним `PF_INET` (Protocol family – Internet).

Второй параметр будет принимать предопределенные значения `SOCK_STREAM` для потоковых сокетов и `SOCK_DGRAM` – для датаграммных.

Поскольку третий параметр в нашем случае не учитывается, в него мы будем подставлять значение 0.

Ссылка на информацию о созданном сокете помещается в таблицу открытых файлов процесса подобно тому, как это делалось для `pip`'ов и FIFO (см. семинар 5). Системный вызов возвращает пользователю файловый дескриптор, соответствующий заполненному элементу таблицы, который далее мы будем называть дескриптором сокета. Такой способ хранения информации о сокете позволяет, во-первых, процессам-детям наследовать ее от процессов-родителей, а, во-вторых, использовать для сокетов часть системных вызовов, которые уже знакомы нам по работе с `pip`'ами и FIFO: `close()`, `read()`, `write()`.

Адреса сокетов. Настройка адреса сокета. Системный вызов bind()

Когда сокет создан, необходимо настроить его адрес. Для этого используется системный вызов `bind()`. Первый параметр вызова должен содержать дескриптор сокета, для которого производится настройка адреса. Второй и третий параметры задают этот адрес.

Во втором параметре должен быть указатель на структуру `struct sockaddr`, содержащую удаленную и локальные части полного адреса.

Указатели типа `struct sockaddr *` встречаются во многих сетевых системных вызовах; они используются для передачи информации о том, к какому адресу привязан или должен быть привязан сокет.

Системные вызовы sendto() и recvfrom()

Для отправки датаграмм применяется системный вызов `sendto()`. В число параметров этого вызова входят:

- дескриптор сокета, через который отсылается датаграмма;
- адрес области памяти, где лежат данные, которые должны составить содержательную часть датаграммы, и их длина;
- флаги, определяющие поведение системного вызова (в нашем случае они всегда будут иметь значение 0);

- указатель на структуру, содержащую адрес сокета получателя, и ее фактическая длина.

Системный вызов возвращает отрицательное значение при возникновении ошибки и количество реально отосланных байт при нормальной работе. **Нормальное завершение системного вызова не означает, что датаграмма уже покинула ваш компьютер!** Датаграмма сначала помещается в системный сетевой буфер, а ее реальная отправка может произойти после возврата из системного вызова. Вызов `sendto()` может блокироваться, если в сетевом буфере не хватает места для датаграммы.

Для чтения принятых датаграмм и определения адреса получателя (при необходимости) служит системный вызов `recvfrom()`. В число параметров этого вызова входят:

- Дескриптор сокета, через который принимается датаграмма.
- Адрес области памяти, куда следует положить данные, составляющие содержательную часть датаграммы.
- Максимальная длина, допустимая для датаграммы. Если количество данных датаграммы превышает заданную максимальную длину, то вызов по умолчанию рассматривает это как ошибочную ситуацию.
- Флаги, определяющие поведение системного вызова (в нашем случае они будут полагаться равными 0).
- Указатель на структуру, в которую при необходимости может быть занесен адрес сокета отправителя. Если этот адрес не требуется, то можно указать значение `NULL`.
- Указатель на переменную, содержащую максимально возможную длину адреса отправителя. После возвращения из системного вызова в нее будет занесена фактическая длина структуры, содержащей адрес отправителя. Если предыдущий параметр имеет значение `NULL`, то и этот параметр может иметь значение `NULL`.

Системный вызов `recvfrom()` по умолчанию блокируется, если отсутствуют принятые датаграммы, до тех пор, пока датаграмма не появится. При возникновении ошибки он возвращает отрицательное значение, при нормальной работе – длину принятой датаграммы.

Пример программы UDP-клиента

Рассмотрим, наконец, простой пример программы 15–16-1.с. Эта программа является UDP-клиентом для стандартного системного сервиса `echo`. Стандартный сервис принимает от клиента текстовую датаграмму и, не изменяя ее, отправляет обратно. За сервисом зарезервирован номер порта 7. Для правильного запуска программы необходимо указать символьный IP-адрес сетевого интерфейса компьютера, к сервису которого нужно обратиться, в качестве аргумента командной строки, например: `a.out 192.168.253.12`

Ниже следует текст программы

```
/* Простой пример UDP клиента для сервиса echo */
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <string.h>
#include <stdio.h>
#include <errno.h>
#include <unistd.h>
int main(int argc, char **argv)
{
    int sockfd; /* Дескриптор сокета */
    int n, len; /* Переменные для различных длин и
                количества символов */
    char sendline[1000], recvline[1000]; /* Массивы
                для отсылаемой и принятой строки */
    struct sockaddr_in servaddr, cliaddr; /* Структуры для
                адресов сервера и клиента */
    /* Сначала проверяем наличие второго аргумента в
                командной строке. При его отсутствии ругаемся и прекращаем
                работу */
    if(argc != 2){
        printf("Usage: a.out <IP address>\n");
        exit(1);
    }
    /* Создаем UDP сокет */
    if((sockfd = socket(PF_INET, SOCK_DGRAM, 0)) < 0){
        perror(NULL); /* Печатаем сообщение об ошибке */
        exit(1);
    }
    /* Заполняем структуру для адреса клиента: семейство
                протоколов TCP/IP, сетевой интерфейс – любой, номер порта
                по усмотрению операционной системы. Поскольку в структуре
                содержится дополнительное не нужное нам поле, которое
```

```

должно быть нулевым, перед заполнением обнуляем ее всю */
bzero(&cliaddr, sizeof(cliaddr));
cliaddr.sin_family = AF_INET;
cliaddr.sin_port = htons(0);
cliaddr.sin_addr.s_addr = htonl(INADDR_ANY);
/* Настраиваем адрес сокета */
if(bind(sockfd, (struct sockaddr *) &cliaddr,
sizeof(cliaddr)) < 0){
    perror(NULL);
    close(sockfd); /* По окончании работы закрываем
дескриптор сокета */
    exit(1);
}
/* Заполняем структуру для адреса сервера:
семейство протоколов TCP/IP, сетевой интерфейс – из аргумента
командной строки, номер порта 7. Поскольку в
структуре содержится дополнительное не нужное нам
поле, которое должно быть нулевым, перед заполнением
обнуляем ее всю */
bzero(&servaddr, sizeof(servaddr));
servaddr.sin_family = AF_INET;
servaddr.sin_port = htons(7);
if(inet_aton(argv[1], &servaddr.sin_addr) == 0){
    printf("Invalid IP address\n");
    close(sockfd); /* По окончании работы закрываем
дескриптор сокета */
    exit(1);
}
/* Вводим строку, которую отошлем серверу */
printf("String => ");
fgets(sendline, 1000, stdin);
/* Отсылаем датаграмму */
if(sendto(sockfd, sendline, strlen(sendline)+1,
0, (struct sockaddr *) &servaddr,
sizeof(servaddr)) < 0){
    perror(NULL);
    close(sockfd);
    exit(1);
}
/* Ожидаем ответа и читаем его. Максимальная
допустимая длина датаграммы – 1000 символов,
адрес отправителя нам не нужен */
if((n = recvfrom(sockfd, recvline, 1000, 0,
(struct sockaddr *) NULL, NULL)) < 0){
    perror(NULL);
    close(sockfd);
    exit(1);
}
/* Печатаем пришедший ответ и закрываем сокет */
printf("%s\n", recvline);
close(sockfd);
return 0;
}

```

Наберите и откомпилируйте программу. Перед запуском **"узнайте у своего системного администратора"**, запущен ли в системе стандартный UDP-сервис echo и если нет, попросите стартовать его. Запустите программу с запросом к сервису своего компьютера, к сервисам других компьютеров. Если в качестве IP-адреса указать несуществующий адрес, адрес выключенной машины или машины, на которой не работает сервис echo, то программа бесконечно блокируется в вызове `recvfrom()`, ожидая ответа. Протокол UDP не является надежным протоколом. Если датаграмму доставить по назначению не удалось, то отправитель никогда об этом не узнает!

Пример программы UDP-сервера

Поскольку UDP-сервер использует те же самые системные вызовы, что и UDP-клиент, мы можем сразу приступить к рассмотрению примера UDP-сервера (программа 15–16-2.с) для сервиса echo.

```

/* Простой пример UDP-сервера для сервиса echo */
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <string.h>

```

```

#include <stdio.h>
#include <errno.h>
#include <unistd.h>
int main()
{
    int sockfd; /* Дескриптор сокета */
    int clien, n; /* Переменные для различных длин
    и количества символов */
    char line[1000]; /* Массив для принятой и
    отсылаемой строки */
    struct sockaddr_in servaddr, cliaddr; /* Структуры
    для адресов сервера и клиента */
    /* Заполняем структуру для адреса сервера: семейство
    протоколов TCP/IP, сетевой интерфейс – любой, номер порта
    51000. Поскольку в структуре содержится дополнительное не
    нужное нам поле, которое должно быть нулевым, перед
    заполнением обнуляем ее всю */
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(51000);
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    /* Создаем UDP сокет */
    if((sockfd = socket(PF_INET, SOCK_DGRAM, 0)) < 0){
        perror(NULL); /* Печатаем сообщение об ошибке */
        exit(1);
    }
    /* Настраиваем адрес сокета */
    if(bind(sockfd, (struct sockaddr *) &servaddr,
    sizeof(servaddr)) < 0){
        perror(NULL);
        close(sockfd);
        exit(1);
    }
    while(1) {
        /* Основной цикл обслуживания*/
        /* В переменную clien заносим максимальную длину
        для ожидаемого адреса клиента */
        clien = sizeof(cliaddr);
        /* Ожидаем прихода запроса от клиента и читаем его.
        Максимальная допустимая длина датаграммы – 999
        символов, адрес отправителя помещаем в структуру
        cliaddr, его реальная длина будет занесена в
        переменную clien */
        if((n = recvfrom(sockfd, line, 999, 0,
        (struct sockaddr *) &cliaddr, &clien)) < 0){
            perror(NULL);
            close(sockfd);
            exit(1);
        }
        /* Печатаем принятый текст на экране */
        printf("%s\n", line);
        /* Принятый текст отправляем обратно по адресу
        отправителя */
        if(sendto(sockfd, line, strlen(line), 0,
        (struct sockaddr *) &cliaddr, clien) < 0){
            perror(NULL);
            close(sockfd);
            exit(1);
        } /* Уходим ожидать новую датаграмму*/
    }
    return 0;
}

```

Листинг 15-16.2. Программа 15–16-2.с . Простой пример UDP-сервера для сервиса echo.

Наберите и откомпилируйте программу. Запустите ее на выполнение. Модифицируйте текст программы UDP-клиента (программа 15–16-1.с), заменив номер порта с 7 на 51000. Запустите клиента с другого виртуального терминала или с другого компьютера и убедитесь, что клиент и сервер взаимодействуют корректно.

Пример программы TCP-клиента

Рассмотрим пример – программу 15–16-3.с. Это простой TCP-клиент, обращающийся к стандартному системному сервису echo. Стандартный сервис принимает от клиента текстовую датаграмму и, не изменяя ее, отправляет обратно. За сервисом зарезервирован номер порта 7. Заметим, что это порт 7 TCP – не путать с портом 7 UDP из примера в разделе "Пример программы UDP-клиента"! Для правильного запуска программы необходимо указать символьный IP-адрес сетевого интерфейса компьютера, к сервису которого требуется обратиться, в качестве аргумента командной строки, например:

```
a.out 192.168.253.12
```

Для того чтобы подчеркнуть, что после установления логического соединения клиент и сервер могут обмениваться информацией неоднократно, клиент трижды запрашивает текст с экрана, отправляет его серверу и печатает полученный ответ. Ниже представлен текст программы.

```
/* Простой пример TCP-клиента для сервиса echo */
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <string.h>
#include <stdio.h>
#include <errno.h>
#include <unistd.h>
void main(int argc, char **argv)
{
    int sockfd; /* Дескриптор сокета */
    int n; /* Количество переданных или прочитанных
            символов */
    int i; /* Счетчик цикла */
    char sendline[1000],rcvline[1000]; /* Массивы
            для отсылаемой и принятой строки */
    struct sockaddr_in servaddr; /* Структура для
            адреса сервера */
    /* Сначала проверяем наличие второго аргумента в
            командной строке. При его отсутствии прекращаем
            работу */
    if(argc != 2){
        printf("Usage: a.out <IP address>\n");
        exit(1);
    }
    /* Обнуляем символьные массивы */
    bzero(sendline,1000);
    bzero(rcvline,1000);
    /* Создаем TCP сокет */
    if((sockfd = socket(PF_INET, SOCK_STREAM, 0)) < 0){
        perror(NULL); /* Печатаем сообщение об ошибке */
        exit(1);
    }
    /* Заполняем структуру для адреса сервера: семейство
            протоколов TCP/IP, сетевой интерфейс – из аргумента
            командной строки, номер порта 7. Поскольку в структуре
            содержится дополнительное не нужное нам поле,
            которое должно быть нулевым, перед заполнением обнуляем
            ее всю */
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(51000);
    if(inet_aton(argv[1], &servaddr.sin_addr) == 0){
        printf("Invalid IP address\n");
        close(sockfd);
        exit(1);
    }
    /* Устанавливаем логическое соединение через
            созданный сокет с сокетом сервера, адрес которого мы занесли
            в структуру */
    if(connect(sockfd, (struct sockaddr *) &servaddr,
            sizeof(servaddr)) < 0){
        perror(NULL);
        close(sockfd);
        exit(1);
    }
    /* Три раза в цикле вводим строку с клавиатуры, отправляем
```

```

ее серверу и читаем полученный ответ */
for(i=0; i<3; i++){
    printf("String => ");
    fflush(stdin);
    fgets(sendline, 1000, stdin);
    if( (n = write(sockfd, sendline,
    strlen(sendline)+1)) < 0){
        perror("Can't write\n");
        close(sockfd);
        exit(1);
    }
    if ( (n = read(sockfd,recvline, 999)) < 0){
        perror("Can't read\n");
        close(sockfd);
        exit(1);
    }
    printf("%s", recvline);
}
}
/* Завершаем соединение */
close(sockfd);
}

```

Листинг 15-16.3. Программа 15–16-3.с . Простой пример TCP-клиента для сервиса echo.

Наберите и откомпилируйте программу. Перед запуском "**узнайте у своего системного администратора**", запущен ли в системе стандартный TCP-сервис echo и, если нет, попросите это сделать. Запустите программу с запросом к сервису своего компьютера, к сервисам других компьютеров. Если в качестве IP-адреса указать несуществующий адрес или адрес выключенной машины, то программа сообщит об ошибке при работе вызова connect() (правда, возможно, придется подождать окончания timeout'a). При задании адреса компьютера, на котором не работает сервис echo, об ошибке станет известно сразу же. Протокол TCP является надежным протоколом. Если логическое соединение установить не удалось, то отправитель будет знать об этом.

Лабораторная работа № 14 (2 часа) Установка ОС WinXP

Технические средства Виртуальная машина

Подготовка дискового пространства

Реальными параметрами, является процессор с частотой не менее 300 Мгц и объем оперативной памяти не менее 128 мегабайт.

Немаловажным этапом окажется и предварительная подготовка жесткого диска перед установкой Windows XP. Если имеется операционная система линейки Win9x/ME, оптимальным вариантом будет создание дополнительного раздела для XP, которая, как и Windows 2000, "умеет" загружаться с логического диска. При отсутствии необходимого программного инструментария для работы с дисковыми разделами (Paragon Partition Manager, Acronis Partiton Expert): в процессе установки Windows XP предложит услуги по созданию нового раздела . Необходимо учесть, что для нужд этой системы придется выделить 1,5 гигабайта на жестком диске.

Выбор файловой системы NTFS или FAT32

Кроме этого, XP позволяет осуществлять форматирование созданного или имеющегося разделов в файловые системы FAT32 или NTFS. Впрочем, конвертация FAT32 в NTFS осуществляется без потери информации посредством встроенной утилиты convert, достаточно открыть меню Выполнить кнопки Пуск, где ввести команду **CONVERT X: /FS:NTFS** (в данном случае X: является разделом с установленной Windows XP).

Безусловным преимуществом FAT32 является тот факт, что эта файловая система быстрее и требует меньше памяти для работы. Если система работает только с FAT32, то в память не грузятся драйверы и сервисы, необходимые для NTFS. Еще одно преимущество FAT32 - доступ к диску посредством загрузочной дискеты, созданной средствами Windows 9x/ME. Проблема доступа к NTFS-разделу также решается при помощи соответствующих приложений (пример - российская разработка "NTFS for Win98" от компании Парагон). Однако, главным доводом в пользу NTFS является устойчивость этой файловой системы:

NTFS содержит две копии аналога FAT, именуемые MFT (Master File Table). В отличие от FAT, MFT больше напоминает таблицу базы данных. Если оригинал MFT поврежден в случае аппаратной ошибки (bad-сектор), то система при следующей загрузке использует копию MFT, и автоматически создаёт новый оригинал, но уже с учётом повреждений. Помимо этого, NTFS обладает встроенными средствами шифрования файлов, что обеспечивает определённую уверенность в сохранности данных (если система не будет переустанавливаться). Кроме того: в NTFS возможно сжатие отдельных каталогов и файлов, в отличие от DriveSpace, который позволял сжимать только диски целиком. Очень удобная возможность: компрессия "на лету", например, больших графических BMP-файлов, с целью экономии дискового пространства, причём для пользователя этот процесс будет абсолютно "прозрачным".

Установка Windows XP

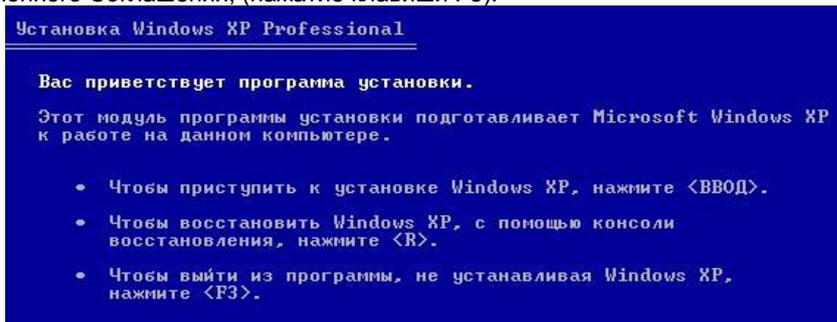
Не стоит бояться установки Windows XP: на самом деле все гораздо проще, чем думают некоторые пользователи.

Ниже приведены несколько скриншотов с пояснениями для тех, кто впервые устанавливает Windows XP. Для начала следует задать в BIOS приоритет загрузки с CD-ROM, и поместить диск с дистрибутивом XP в лоток привода.

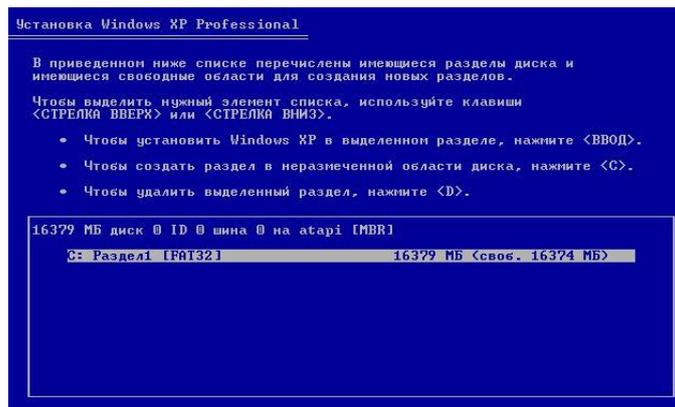
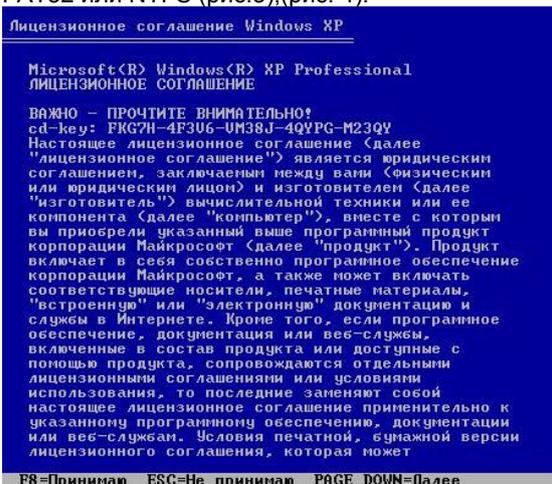
На первом этапе программа установки появится сообщение о начальном копировании необходимых файлов, причем, этот процесс может длиться довольно долго - несколько минут (рис.1).



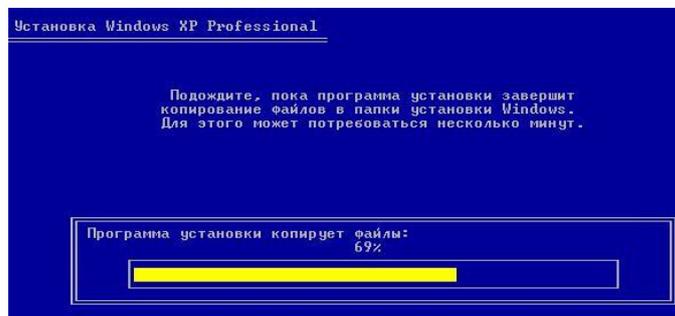
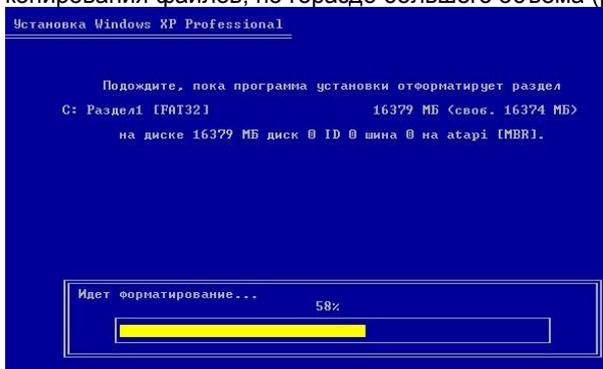
Далее нажимаем **ENTER** для начала процесса инсталляции системы (рис.2). Следующий шаг - принятие Лицензионного Соглашения, (нажатие клавиши F8).



Далее необходимо выбрать раздел для установки (можно и на ЛОГИЧЕСКИЙ диск, поскольку Windows 2000/XP загружаются из таких разделов) с последующим форматированием в файловые системы FAT32 или NTFS (рис.3),(рис. 4).



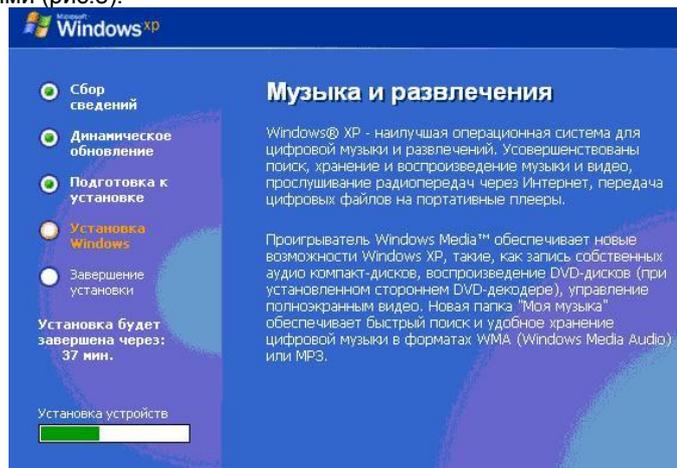
Процесс форматирования довольно длителен, после чего наступает шаг для еще одного копирования файлов, но гораздо большего объема (рис.5),(рис. 6).



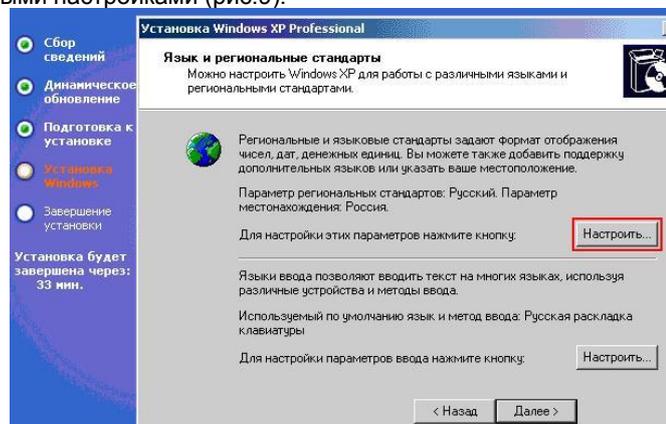
По окончании копирования файлов программа установки предложит перезагрузить машину - не забудьте вновь зайти в настройки BIOS или вынуть (рис.7) CD-диск из CD-привода).



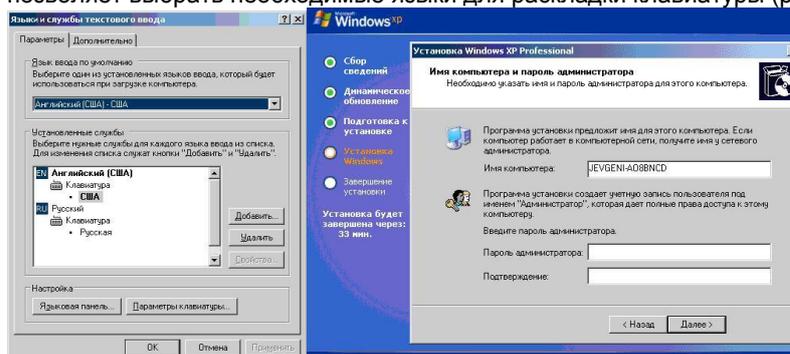
После перезагрузки ОС начинается следующая фаза – графическая фаза установки Windows XP с рекламой и возможностями (рис.8).

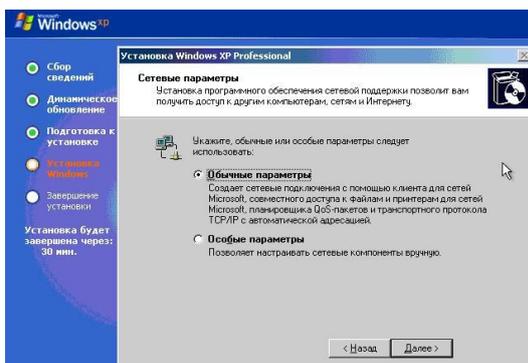


Процесс установки длится более получаса, после этого выходит окно, приглашающее определиться с региональными и языковыми настройками (рис.9).



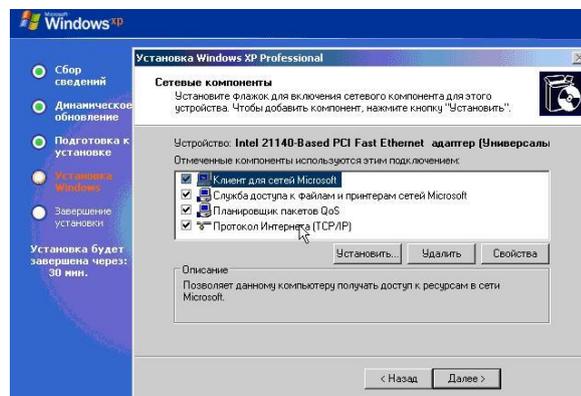
В первой секции этого окна посредством кнопки Настроить выбираются необходимые региональные параметры, и язык для программ, не поддерживающих UNICODE - задав параметр РУССКИЙ, обеспечивается работа с русскими программами в англоязычной системе. Вторая секция (Языки ввода) позволяет выбрать необходимые языки для раскладки клавиатуры (рис.10),(рис. 11).





Далее задается имя для компьютера (по умолчанию программа установки предлагает свое сгенерированное имя) и, при желании, задать пароль администратора.

Затем следует определиться с сетевыми параметрами: типичные или особые. Для домашних условий не важно, что вы выберете - "Особые" параметры установятся те же самые, что и "Обычные" по умолчанию (рис. 12), (рис. 13).



Остается указать имя рабочей группы (или оставить по умолчанию WORKGROUP), после чего система продолжит копирование файлов, а участие пользователя сведется к перезагрузке машины и заполнению полей учетной записи пользователя (или нескольких таких записей, если на компьютере будет работать не один человек).

Лабораторная работа № 15 (2 часа) Установка операционной системы Linux

Технические средства Виртуальная машина с ОС Linux

Средства для выполнения работы:

- **аппаратные:** компьютер с установленной ОС Windows XP.
- **программные:** приложение VM (VirtualBox); виртуальная машина VM-2; установочный образ ОС openSUSE.iso.

Теоретические сведения

Linux (Линукс) — UNIX-совместимое ядро операционной системы, разработка которого была начата финским студентом Линусом Торвалдсом в 1991 году.

Как таковое ядро ОС **Linux** способно управлять аппаратным обеспечением и не предоставляет инструментов пользователю для решения его задач. Поэтому ОС **Linux** распространяются в виде дистрибутивов.

Дистрибутив операционной системы — это форма распространения системного программного обеспечения. Наличие дистрибутивов вызвано тем, что форма программного обеспечения, используемая для его распространения, почти никогда не совпадает с формой программного обеспечения работающей системы.

Дистрибутив обычно содержит программы для начальной инициализации системы (инициализация аппаратной части, загрузка урезанной версии системы и запуск программы-установщика), программу-установщик (для выбора режимов и параметров установки) и набор специальных файлов, содержащих отдельные части системы (так называемые *пакеты*). Программа установки позволяет также произвести первичную настройку системы.

На сегодняшний день существует большое количество различных дистрибутивов ОС **Linux** (более 100), ориентированных на различных пользователей. К наиболее популярным относятся такие дистрибутивы, как: **PCLinuxOS, Ubuntu, openSUSE, Fedora, Mint, Sabayon, Mandriva, Debian, MEPIS, Damn Small 1**.

Характеристики ОС семейства Linux — многозадачность, многопользовательский доступ, свопирование оперативной памяти на диск, загрузка выполняемых модулей по требованию, совместное использование выполняемых программ, общие библиотеки, динамическое кэширование диска, возможность запуска исполняемых файлов других ОС, поддержка различных форматов файловых систем, работа на различных аппаратных платформах и т.д.

OpenSUSE — дистрибутив рассчитан, в первую очередь, на настольные системы и ориентирован на широкий круг пользователей от новичков до профессионалов, - любопытствующих, мигрирующих с других ОС и давних приверженцев **Linux** и свободного программного обеспечения. Удобный и дружелюбный интерфейс, родная языковая среда наряду с самыми свежими версиями пользовательских и системных программ и пакетов призваны окончательно развеять миф о недоступности и неудобстве **Linux** в повседневной работе.

Разрабатывается французской компанией. Дистрибутив основан на **Red Hat Linux**, с большим количеством дополнений. Содержит большое множество программного обеспечения. Включает в себя новую и интуитивно понятную программу установки с графическим интерфейсом, переведенную на большое количество языков; фирменные средства настройки системы; поддержку автоматического мониторинга/демонтирования носителей информации.

Установка **OpenSUSE** на компьютер отличается простотой и включает в себя следующие этапы: загрузка компьютера с установочного диска, принятие лицензионного соглашения, выбор языка, установка параметров безопасности, подготовка жесткого диска, выбор пакетов для установки, ввод данных пользователя и его пароля.

К одной из самых сложных частей процесса установки относится подготовка жесткого диска, которая в случае неправильных действий пользователя может привести к потере данных. Установщик ОС **OpenSSE** может автоматически использовать все свободное пространство на жестком диске, весь диск целиком или позволяет пользователю самому выполнить подготовку диска к установке.

Подготовка жесткого диска к установке сводится к выделению места для устанавливаемой ОС, созданию разделов в выделенном пространстве и их форматированию. Специалисты по ОС семейства **Linux** рекомендуют создавать отдельные разделы для размещения ядра ОС и файлов загрузчика ОС, файлов пользователей, файла подкачки.

Как правило в случае выбора автоматического режима подготовки диска, установщик самостоятельно создает перечисленные выше разделы.

Самостоятельная работа Установите ОС OpenSUSE в VM VirtualBox.

1. Запустите приложение виртуальных машин **VirtualBox**.
2. Запустите созданную ранее виртуальную машину VM-2 кнопкой **Запустить** и перейдите в окно VM, активизировав его щелчком мыши.
3. Выберите язык установки *русский* (см. Рисунок 3):
 - откройте список поддерживаемых языков (клавиша **F2**);
 - выберите в списке нужный язык (*русский*) с помощью клавиш управления курсором;
 - подтвердите выбор клавишей **ENTER**.

Лабораторная работа № 16 (2 часа) Изменение пользовательских интерфейсов и настроек Linux

Технические средства Виртуальная машина с ОС Linux

Выполните следующие действия:

1. Запустите Linux и введите свой пароль
2. Найдите основные элементы интерфейса и вспомните их:
 - рабочий стол (рабочая область экрана, на которой отображаются окна (в том числе диалоговые), значки и меню)
 - панель задач KDE (по умолчанию отображаемая в нижней части экрана панель, на которой

находится кнопка  - аналогичная кнопке **Пуск** в ОС Windows. С помощью кнопок на панели задач можно переключать запущенные программы



- ярлыки (графические элементы, ссылки на любой элемент, доступный на компьютере или в сети, такой как программа, файл, папка, диск, веб-страница, принтер или другой компьютер)

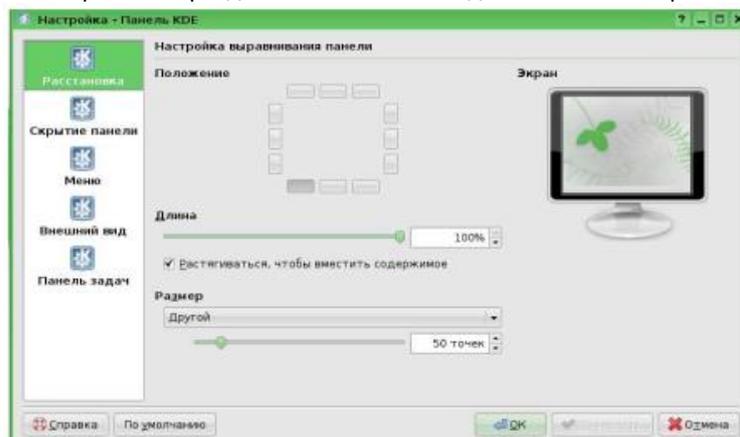
- кнопка  позволяющая запустить приложения (программы), выполнить различные действия
- системный лоток - область уведомления (область на панели задач справа). Здесь отображается текущее время, а также значки, обеспечивающие быстрый доступ к



некоторым программам, таким как программы изменения громкости звука и управления питанием

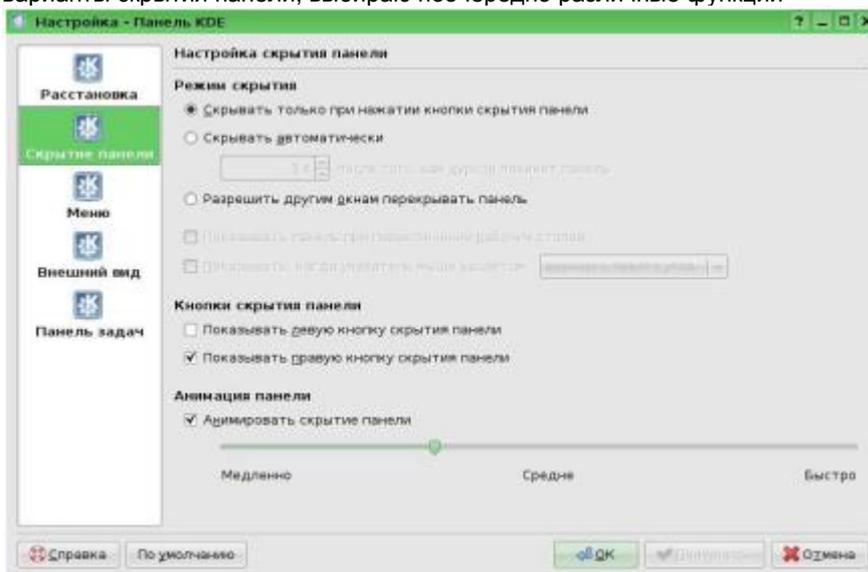
Настройка панели задач

1. Щелкните правой кнопкой мыши на панели задач
2. Выберите пункт контекстного меню **Настроить панель...**
3. Щелкните раздел **Расстановка**. Здесь можно настроить размер и положение панели.

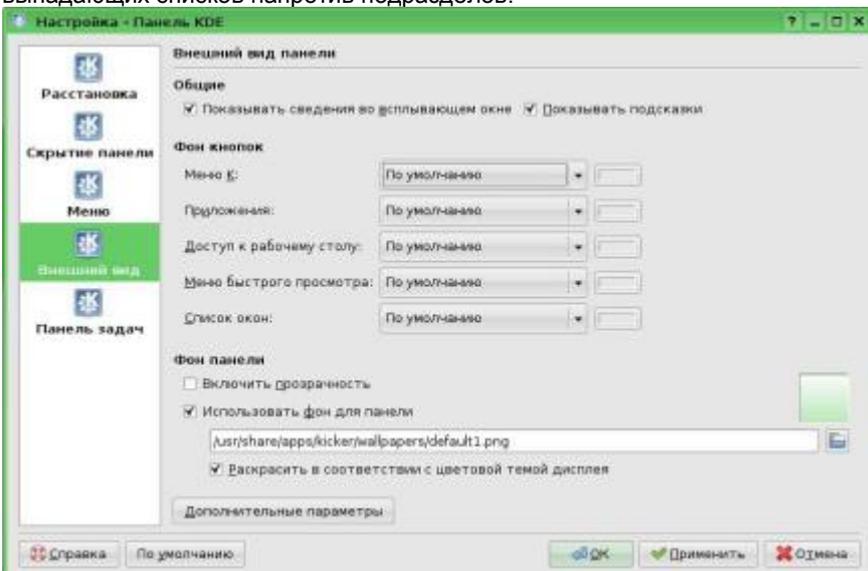


В области под названием **Положение** находятся 12 маленьких кнопок, расположенных в форме квадрата. Каждая из них соответствует определённому положению панели.

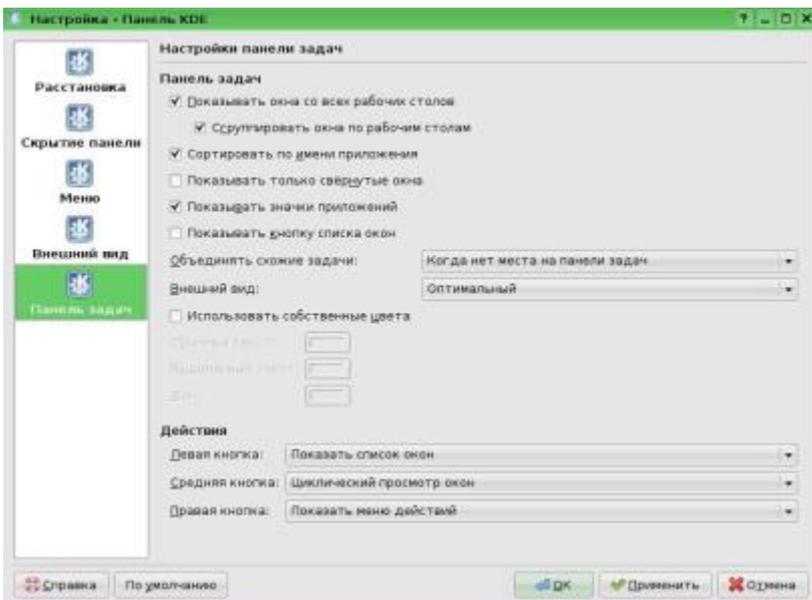
4. Нажмите на любую из кнопок. Вы увидите в окне предварительного просмотра справа, как изменится положение панели. Учтите, что свободное место используется эффективнее, если панель расположена горизонтально, то есть вдоль верхней или нижней границы экрана.
5. В области под названием **Длина** отрегулируйте минимальную длину панели с помощью ползунка и счётчика
6. В области под названием **Размер** задайте высоту панели. Размер панели может быть **Крошечный**, **Маленький**, **Средний**, **Большой** или **Другой**.
7. Выберите **Другой** и измените размер панели, указав высоту с помощью ползунка или счётчика.
8. Щелкните раздел **Скрытие панели**. Рассмотрите различные варианты скрытия панели, выбирая поочередно различные функции



9. Щелкните раздел **Внешний вид**. Попробуйте изменить внешний вид панели, изменяя цвета и прозрачность, выбирая необходимые данные из выпадающих списков напротив подразделов.



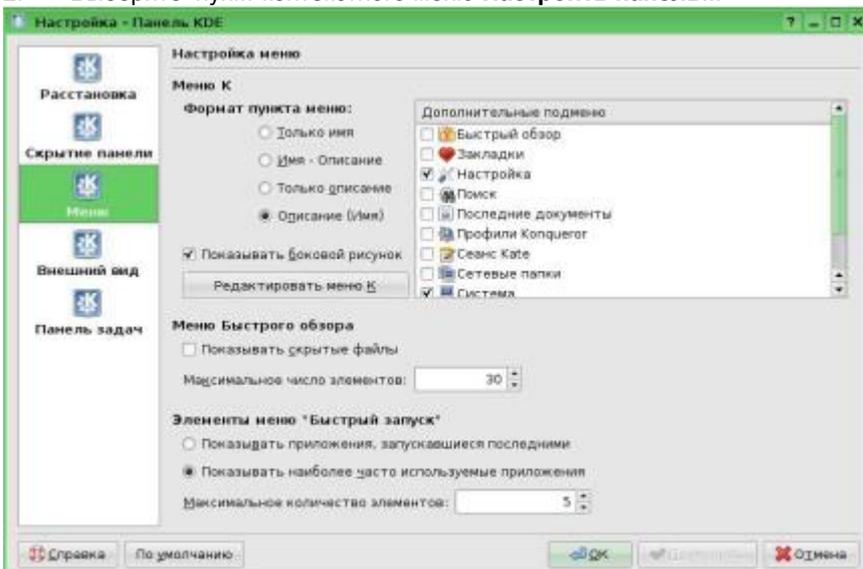
10. Щелкните раздел **Панель задач**. Добавляя или убирая галочки, ознакомьтесь с функциями настройки панели задач. Не забывайте нажимать кнопку **Применить**, чтоб просмотреть изменения.



Внимание! Вы всегда можете вернуть все настройки назад, щелкнув кнопку **По Умолчанию** или **Меню приложений**



1. Лево́й кнопкой мыши щелкните кнопку  . При этом откроется К-меню.
2. В К-меню поочередно подводите указатель мыши к пунктам Приложений и ознакомьтесь с перечнем программ, которые установлены на вашем компьютере. Вы также можете настроить К-меню самостоятельно. Для этого:
 1. Щелкните правой кнопкой мыши на панели задач
 2. Выберите пункт контекстного меню **Настроить панель...**



3. Щелкните раздел **Меню**. В разделе К-меню можно настроить функциональность меню К. Параметры **Дополнительные подменю** определяют, какие дополнительные пункты будет содержать К-меню. Ознакомьтесь с представленной ниже информацией и выберите нужные разделы для вашего главного меню.

- **Подменю Закладки** позволяет перейти по любой закладке Konqueror.
- **Подменю Поиск** предоставляет средства поиска среди локальных файлов и в Интернете.
- **Подменю Профили Konqueror** содержит профили программы Konqueror.
- **Подменю Сетевые папки** открывает окно программы Konqueror, показывающее настроенные сетевые папки.
- **Подменю Настройка** позволяет загрузить любой из модулей Центра управления KDE непосредственно из меню К.
- **Подменю Система** печати содержит несколько пунктов для управления принтерами, доступными в вашей системе.
- **Подменю Быстрый обзор** позволяет быстро перемещаться по папкам файловой системы.

- **Подменю Последние документы** содержит список последних открывавшихся документов и автоматически запускает программу для просмотра или редактирования выбранного документа.
- **Подменю Система** используется для быстрого доступа к некоторым часто посещаемым местам, включая домашнюю папку и корзину. Оно может оказаться удобнее отдельных пунктов в К-меню.
- **Подменю Терминальные сеансы** содержит пункты для загрузки разного типа терминалов (интерфейсов командной строки).

Самостоятельная работа.

- Начать сеанс работы в ОС Linux
- Ввести логин и пароль
- Настроить рабочий стол KDE: панель задач, главное меню. Изменить внешний вид. Вернуться к настройкам «По умолчанию»
- Разместить ярлыки часто используемых приложений на рабочем столе.
- Завершить сеанс, перезагрузить компьютер.

Лабораторная работа № 17 (2 часа) Установка программного обеспечения в Windows

Программы и функции в составе Windows позволяют многое, но может потребоваться установить и другие программы.

Способ добавления программы зависит от того, где находятся файлы ее установки. Обычно программы устанавливаются с компакт-диска или DVD, через Интернет или с сетевого диска.

Установка программы с компакт-диска или DVD

- Вставьте диск в компьютер и следуйте инструкциям на экране. Введите пароль администратора или подтверждение пароля, если появится соответствующий запрос.
- Многие программы, устанавливаемые с компакт-диска или DVD, пытаются автоматически запустить мастер установки программы. В этих случаях появляется диалоговое окно **«Автозапуск»**, и можно запустить мастер.
- Если установка программы не начинается, ознакомьтесь с информацией документации программы. Возможно, там есть инструкции по установке программы вручную. Если невозможно получить информацию о программе, можно поискать на диске и открыть файл установки программы, обычно **Setup.exe** или **Install.exe**.

Установка программы через Интернет

1. В веб-обозревателе щелкните ссылку на программу.
2. Выполните одно из действий.
 - Чтобы сразу начать установку программы, щелкните команду Открыть или Выполнить и следуйте инструкциям на экране. Введите пароль администратора или подтверждение пароля, если появится соответствующий запрос.
 - Чтобы начать установку позднее, щелкните команду **Сохранить** и загрузите на компьютер файл установки. Чтобы затем установить программу, дважды щелкните файл и следуйте инструкциям на экране. Это безопасный способ, так как файл установки можно сначала проверить на наличие вирусов.

Примечание.

При загрузке и установке программ через Интернет удостоверьтесь, что издатель программы и веб-узел, предоставляющий программу, заслуживают доверия.

Установка программы с сетевого диска

Если компьютер входит в сеть (такую как внутренняя сеть предприятия), предоставляющую программы, которые можно добавлять, то программы можно устанавливать через панель управления.

1. Откройте окно **«Получение программ»**, нажав кнопку **Пуск** и выбрав пункты **Панель управления, Программы и Получение программ**.
2. Выберите программу в списке и нажмите команду **Установить**.
3. Следуйте инструкциям на экране. Введите пароль администратора или подтверждение пароля, если появится соответствующий запрос.

Самостоятельная работа

1. Ознакомиться с методическими рекомендациями
 2. Установить программу по заданию преподавателя
 3. Проверить работоспособность установленной программы
 4. Деинсталлировать программу
 5. Составить отчет
- <http://windows.microsoft.com/ru-RU/windows-vista/Install-a-program>

Лабораторная работа № 18 (2 часа) Установка и настройка прикладного программного обеспечения в Linux

Технические средства Виртуальная машина с ОС Linux

Установка пакета

Обновите информацию о пакетах в **Synaptic**, чтобы узнать о последних доступных версиях, выбрав в меню **Редактирование** — **Получить сведения о пакетах** или нажав **Получить сведения** на панели инструментов.

1. Отметьте пакет для установки - **aiksaurus-data**:

Двойной щелчок мыши на названии пакета в списке пакетов

или

Нажмите правой кнопкой мыши на пакете и выберите **Отметить для установки** в контекстном меню.

или

Выделите пакет и меню **Пакет** — **Отметить для установки**. Если установка пакетов требует дополнительных изменений, вас спросят о подтверждении. Чтобы отметить дополнительные изменения, нажмите **Применить**.

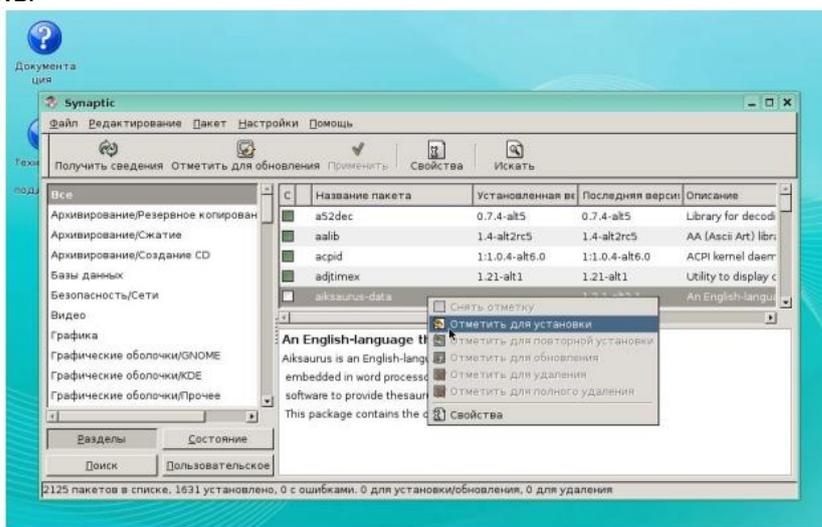


Рис. 1. Выбор пакета для установки

2. Примените отмеченные изменения, чтобы установить пакеты, нажав кнопку **Применить** на панели инструментов или выбрав в меню **Редактирование** — **Внести отмеченные изменения**.

3. Проверьте итоговые изменения, после запроса о подтверждении. Чтобы продолжить установку подтвердите изменения, нажав **Применить**.

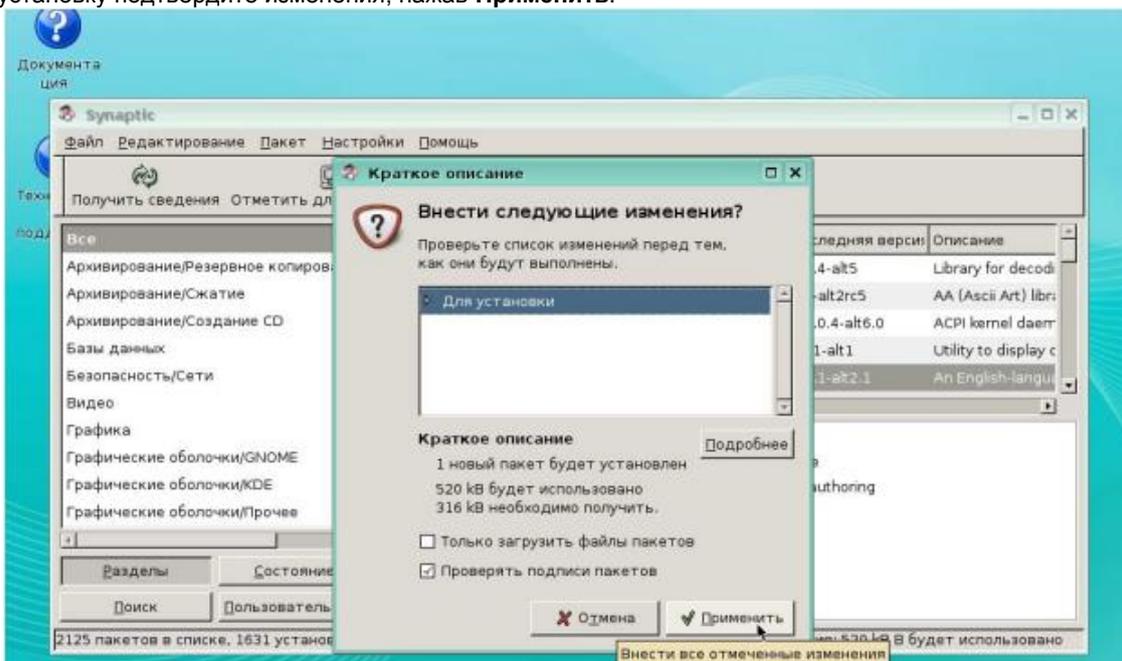


Рис. 2. Краткое описание.

4. Во время проведения изменений вы увидите строку состояния. Подождите, пока изменения будут применены. Это может занять некоторое время в зависимости от количества изменений. После этого вы вернётесь в основное меню.

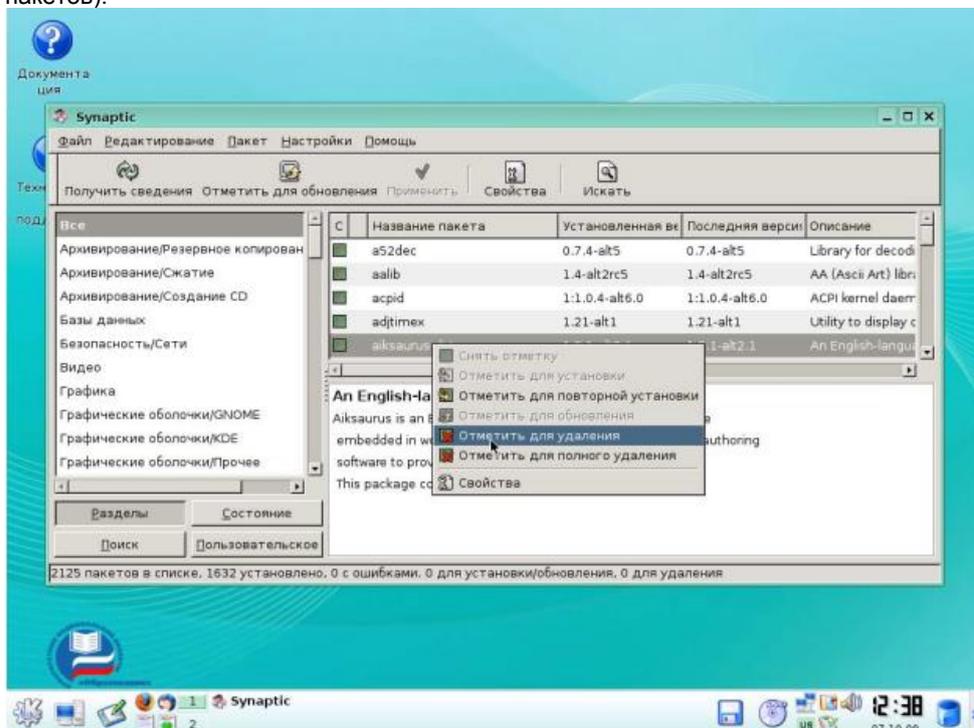
Удаление пакета

Удалите пакет **aiksaurus-data**

1. Зайдите в **Synaptic**

2. Отметьте пакет для удаления, нажав правой кнопкой мыши на пакете и выбрав **Отметить для удаления** в контекстном меню.

3. Примените отмеченные изменения, чтобы удалить пакеты (так же, как при установке пакетов).



4. Вас спросят о подтверждении. Проверьте итоговые изменения, которые будут применены, после чего подтвердите изменения.

5. Во время проведения изменений вы увидите строку состояния. Когда изменения будут применены, вы вернётесь в основное меню.

. Самостоятельная работа

1. Установите пакеты: **nfs-server**, **nfs-clients** и **samba-client**, если они не установлены (Пакеты находятся в группе Сети/Прочее).

2. Найдите в Synaptic пакет для архивирования zip, воспользовавшись поиском. Посмотрите версию пакета. Установите его, если он не установлен.

Лабораторная работа № 19 (2 часа)

Файловые оболочки (файловые менеджеры, командиры) в Windows

1. Общие сведения

В настоящее время существует большое количество файловых оболочек, однако все они унаследовали концепцию и, в той или иной степени, интерфейс от первоначального варианта Norton Commander (NC) 1986 года выпуска. Часть из них прямо копирует NC (FAR, WinNavigator), часть использует те же функциональные элементы, но представленные «в стиле windows» (Windows/Total Commander и др.). Сам же NC был и остается средоточием стандартных решений.

Мы будем рассматривать только современные оболочки для Windows и только некоторые из основных их возможностей.

Преимущества файловых оболочек состоят в следующем:

- наличие двух панелей, что позволяет сразу видеть результаты копирования; как известно из практики, именно эта операция часто сопряжена с трудностями и большим числом промахов;
- наличие одновременно с панелями командной строки;
- возможность делать комментарии к файлам, выводимые на панель для всех файлов, а не только для тех, куда подведена мышь;
- возможность раскрашивать панели и их содержимое;
- возможность сохранения настроек и др.

Эффект работы с файловыми оболочками малозаметен при небольшом количестве прodelываемых за сеанс операций с файлами и запусками приложений, но отлично проявляется в противоположных ситуациях. Прежде всего это сказывается в отсутствии лишнего на экране и минимизированном количестве простых, но утомляющих действий.

Современные оболочки обладают огромным спектром возможностей, освоение которых – дело практики. **Изучив возможности, наиболее соответствующие вашим личным запросам, и настроив под них оболочку, можно сэкономить массу сил и времени.**

В данной работе рассматриваются две наиболее популярных и доказавших свою эффективность многофункциональных оболочки:

- Far Manager, чей интерфейс почти в точности повторяет интерфейс NC и скорее относится к текстовым;
- Total Commander, выдержанный в значительной степени в стиле Windows.

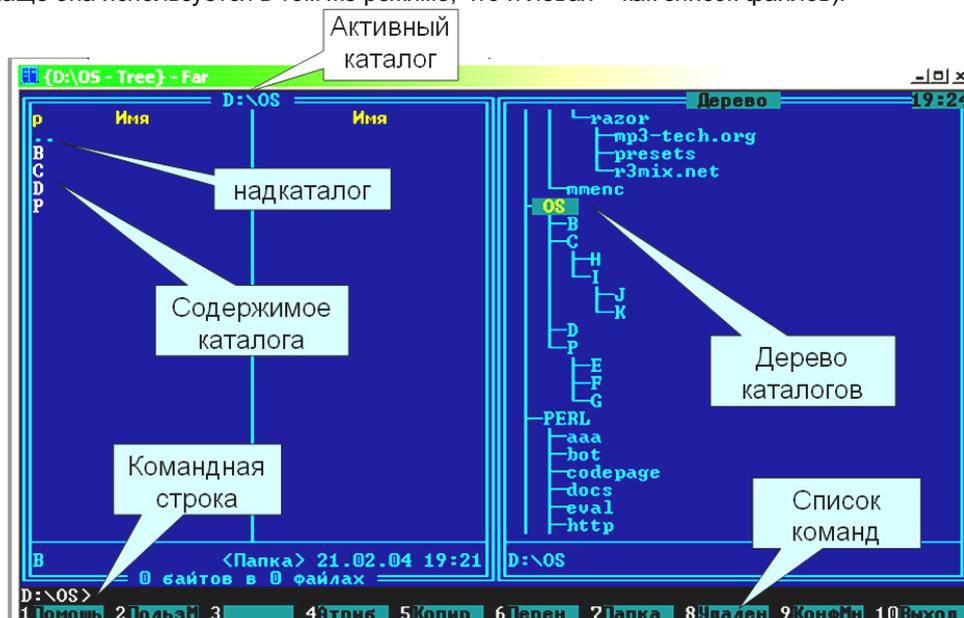
Оба менеджера обладают очень широким диапазоном настроек и функций, подавляющая часть которых является общей для них. Однако реализации с точки зрения пользователя каких-то возможностей одинаковы, каких-то – различны. Кроме того, для многих действий предлагается несколько вариантов выполнения (например, пункт меню и комбинация клавиш). Поэтому в качестве упражнения можно опробовать способы действий, предлагаемые одним менеджером, на другом менеджере.

В практической деятельности целесообразно выбрать по вкусу один из менеджеров как основной, не пожалеть некоторых усилий на хорошее освоение его основных возможностей и использовать его как рабочий инструмент.

2. Файловая оболочка Far Manager

Внешний вид Far Manager представлен на рисунке ниже.

Имеет место классический двухпанельный интерфейс (в данном случае правая панель отображает дерево каталогов, чаще она используется в том же режиме, что и левая – как список файлов).



Свойства FAR можно установить, нажав правую кнопку мыши в самой верхней полосе окна.

Панель управления FAR оформлена как *выпадающее меню* и появляется при нажатии левой или правой кнопки мыши на верхней двойной рамке окна. В других менеджерах это меню присутствует постоянно.

Операции могут производиться над файлом или каталогом, находящимся под курсором, либо над группой выделенных файлов и каталогов (для краткости назовем вышеупомянутое *объектом*).

Основные операции и соответствующие им функциональные клавиши

Копирование объекта в текущий каталог противоположной панели	F5 либо «перетаскивание» мышкой
То же, но с удалением исходного объекта («перемещение»)	F6
Создание каталога в текущем каталоге	F7
Удаление объекта	F8
Просмотр файла (для каталога – подсчет размера)	F3
Редактирование файла (для каталога – редактирование атрибутов)	F4
Выбор отображаемого диска в левой/правой панели	Alt-F1/Alt-F2
Поиск файла/каталога по имени (с использованием символов подстановки)	Alt-F7
Редактирование файла по имени (если указать имя несуществующего файла, такой будет создан)	Shift-F4
«Быстрый поиск» по началу имени (только Far)	Удерживая Alt, начать набирать имя
Выделение объекта\снятие выделения	INS, правая кнопка мышки, «+» на цифровой клавиатуре (по имени, с использованием символов подстановки)
Определение размера каталога	F3
Редактирование атрибутов файла/каталога	Files/File Attributes

Практически любое действие также можно выполнить через выпадающее меню.

Другие операции, повышающие эффективность работы

- 1. Отображение скрытых файлов и папок:**
 - Options; Panel Settings; опция show hidden/system files;
 - сокрытие файлов – форма их защиты от случайных изменений, поэтому делать их видимыми следует только при необходимости и достаточном профессионализме работы с компьютером – обратите внимание на пояснение к опции: только для экспертов.
- 2. Выделение группы файлов:**
 - Gray + (+ на цифровой части клавиатуры); задать маску¹ группы; ОК;
 - с выделенной группой работают так же, как с отдельным файлом (переместить, копировать, удалить).
- 3. Снятие выделения группы:** аналогично выделению, но с заменой Gray + на Gray -.
- 4. Создание и просмотр комментариев к файлам и каталогам:**
 - Files; Describe Files; написать комментарий к файлу или каталогу; Enter;
 - вывести комментарии на панель: Left/Right (для нужной панели); Descriptions;
 - комментарии записываются в файл `descript.ion` по умолчанию это скрытый файл; это текстовый файл, с которым, отобразив его на панели, можно работать обычным образом (например, просматривать, редактировать, печатать).

• Некоторые важные возможности

В Far имеется «временный диск» (опция «временная» или «temporagy» в списке дисков), позволяющий облегчить копирование файлов при работе с несколькими дисками/каталогами.

Far позволяет работать с сетью Microsoft Windows, заменяя собой иконку «сетевое окружение» («сеть» в списке дисков).

Важной особенностью Far является работа с процессами ОС Windows. При выборе «Список процессов» («Task List») в панели выводится полный список процессов с указанием PID, приоритета, объема занимаемой памяти и т.п. По F3 можно получить более полную информацию о процессе.

Правильно настроенные файловые оболочки автоматически работают с архивами, интерпретируя их как каталоги и выполняя архивацию и разархивацию при копировании файлов.

3. Файловая оболочка Total commander

Total commander – наследник оболочки Windows Commander.

Первая версия Windows Commander появилась в 1993 году, последняя – в *уточнить год и версию* году. Оболочка тесно интегрируется с ОС Windows. Имеет богатый легко настраиваемый интерфейс.

Обладает рядом дополнительных возможностей, например, для файла или каталога можно создать кнопку быстрого доступа, перенеся его мышью на панель инструментов.

В отличие от Far, не поддерживает списка процессов.

Основные операции и соответствующие им функциональные клавиши

Практически совпадают с приведенными для Far Manager. Отличия:

Определение размера файла/каталога	Files/Properties или Ctrl/L
Редактирование атрибутов файла/каталога	Files/Change Attributes

Другие операции, повышающие эффективность работы

- 1. Отображение скрытых файлов и папок:**
 - Configuration; Options; Display; File display; опция show hidden/system files;
 - сокрытие файлов – форма их защиты от случайных изменений, поэтому делать их видимыми следует только при необходимости и достаточном профессионализме работы с компьютером – обратите внимание на пояснение к опции: только для экспертов.
- 2. Выделение группы файлов:**
 - Gray + (+ на цифровой части клавиатуры); задать маску группы; ОК;
 - с выделенной группой работают так же, как с отдельным файлом (переместить, копировать, удалить).
 - **Снятие выделения группы:** аналогично выделению, но с заменой Gray + на Gray -.
 - Сопоставить некоторый цвет файлам с заданным расширением (например, сиреневый – exe, синий – doc, txt, морская лазурь – jpg):
 - Configuration; Options; Color; Define Color by file type; Add; Specify file type;
 - задать тип файла в виде *.<расширение> (все файлы с заданным расширением); ОК;
 - в появившейся палитре выбрать цвет; ОК; ОК; ОК.
- 3. Переименование группы файлов:**
 - выделить файлы; File; Multi-Rename Tool;
 - [N] в поле *Rename mask: file name* соответствует старому имени файла, [E] – старому расширению;
 - добавить символы к имени (или расширению) можно, приписав их слева или справа от конструкции [N] (или [E]);
 - заменить последовательность символов в имени на другую: задать исходную и желаемую последовательность на панели Search&Replace;
 - к имени можно добавит дату и время создания файла;

- если группа файлов должна иметь одинаковые имена, отличающиеся только символом-номером, можно добавить этот номер с помощью кнопки Counter, причем счетчик можно установить по своему усмотрению (Define counter).
- 4. Создание и просмотр комментариев к файлам и каталогам:**
 - File; Edit Comment или комбинация Ctrl/Z; написать комментарий к файлу или каталогу (опцию OEM/DOS убрать); F2OK;
 - вывести комментарии на панель: Show Comments;
 - комментарии записываются в файл `descript.ion` (или `files.bbs`, в зависимости от опции на панели Configuration; Options; Operation; File comments); по умолчанию это скрытый файл; это текстовый файл, с которым, отобразив его на панели, можно работать обычным образом (например, просматривать, редактировать, печатать).
 - 5. Назначить редактор для редактирования документов:**
 - найти по имени исполняемый файл нужного редактора (файл с расширением `.exe`) и записать **полное имя** этого файла; например, такой файл для редактора UltraEdit-32 – `uedit32.exe`; для записи полного имени надо знать его местонахождение;
 - Configuration; Options; Edit/View; Editor for F4; ввести полное имя файла редактора; OK;
 - при размещении курсора на имени некоторого файла и нажатии клавиши F4 файл будет открываться для редактирования в окне назначенного редактора.
 - 6. Вынесение на панель инструментов значков каталогов и файлов и работа с ними:**
 - установить курсор на нужном каталоге (файле) и мышкой перетащить его на панель; на панели создается значок, соответствующий каталогу (файлу);
 - вид значка определяется по умолчанию; при желании можно изменить его на один из предлагаемых или создать собственный значок и сослаться на него: вызвать контекстное меню нажатием на имеющемся значке правой кнопки мыши; Change; Icon file (задать имя каталога со значками);
 - удалить значок можно обычным образом через контекстное меню;
 - клик мыши на значке каталога приводит к открытию каталога;
 - клик мыши на значке файла приводит к его открытию с помощью приложения, связанного с этим файлом, или выполнению, если это исполняемый файл.
 - 7. Создание вкладок и работа с ними:**
 - показать вкладку, даже если она одна: Configuration; Options; Folder Tabs; отметить первую опцию; появится полоса вкладок с вкладкой, соответствующей текущему каталогу;
 - создать новую вкладку: установить курсор мыши на имени нужного каталога и перетащить его на полосу вкладок (или Ctrl/T);
 - закрыть вкладку: щелкнуть на ней правой кнопкой мыши и в контекстном меню выбрать одно из действий в группе *Close ...* (или Ctrl/W);
 - переход между вкладками: щелчок на нужной вкладке (или Ctrl/Tab);
 - сохранить вкладки в файле: щелкнуть правой кнопкой на какой-либо вкладке и в контекстном меню выбрать одно из действий в группе *Save ...* (сохранить вкладки одной или обеих панелей);
 - загрузить вкладки из файла: щелкнуть правой кнопкой на какой-либо вкладке и в контекстном меню выбрать одно из действий в группе *Load...* (загрузить вкладки, добавив к имеющимся, или заменить).

Последние два действия очень помогают, если в определенные периоды времени приходится работать с определенными каталогами. Например, у меня (E.C.) есть два файла вкладок: `moi_kursy_osen.tab` и `moi_kursy-vesna.tab`.

8. Создание новых кнопок

- Configuration; Button Bar; Add; выбрать команду, назначаемую на кнопку: нажать кнопку с лупой рядом со строкой Command, выделить команду, OK; название команды появится в строке Command;
- в строке Icon выбрать иконку для кнопки – из готовых или из собственных картинок (файлов с расширением `.ico`).

Таким образом можно создать, например, кнопки, переносящие имя файла в буфер обмена: команды `cm_CopyNames_ToClip` (только имя) или `cm_CopyFullNames_ToClip` (полное имя). Для переноса имени в буфер достаточно поставить курсор на нужный файл и нажать кнопку.

9. Непосредственное изменение файла настроек

- открыть файлы настроек: Configuration; Change Settings Files Directly;
- настройки хранятся в файлах `wincmd.ini` (основные настройки) и `wscx_ftp.ini` (настройки ftp-соединений);
- по именам можно найти местоположение этих файлов (Alt/F7; имя).

Одно из применений этого пункта – простая замена файла настроек на желаемый вместо кропотливой поэлементной настройки менеджера. Так, в каталоге лабораторной работы содержится мой (E.C.) файл конфигурации; если он приходится по вкусу, то можно заменить им текущий файл конфигурации. Надо только исправить пути к редакторам, учитывая их расположение на Вашем компьютере.

- 10. Показ на панели миниатюр (thumbnails) файлов и каталогов вместо строки имен:** Ctrl/Shift/F1; для графических файлов выводится миниатюра содержимого; возврат в обычный режим – та же комбинация.

Еще ряд полезных возможностей содержится в файле крутые советы_tab.doc в каталоге с лабораторной работой. Пока этот файл плохо отформатирован, из-за чего и не включен в текст работы.

Самостоятельная работа

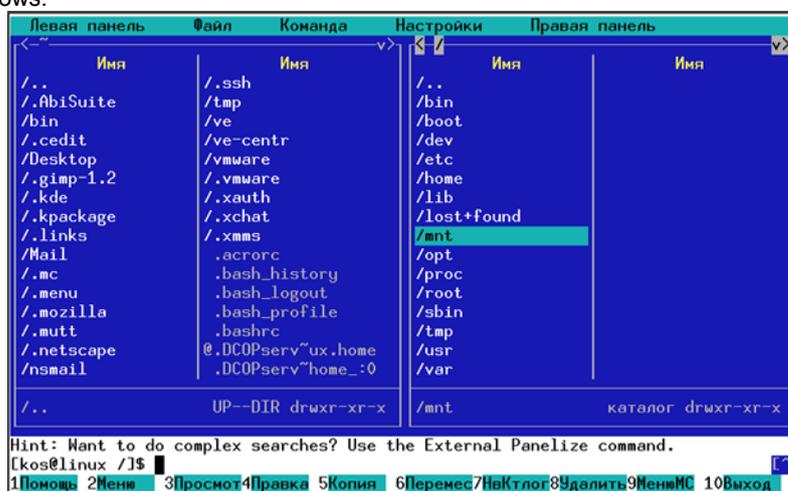
1. С помощью одной из файловых оболочек создать заново такую же систему каталогов, что и в лабораторной работе 13 (на каком-либо другом диске). Получится две копии.
2. Открыть каталоги верхнего уровня каждой из копий соответственно в левой и правой панели. Сверить обе копии, опускаясь до подкаталогов самого нижнего уровня.
3. Создать на любом из дисков каталог CURRENT.
4. Создать текстовый файл total.txt. Записать в него все полезные действия по работе с файловым менеджером (по мере выполнения работы).
5. Найти каталог с 20 – 30 файлами с разными расширениями и несколькими подкаталогами.
6. Создать некоторый временный каталог. Скопировать в него содержимое каталога из п. 5 (чтобы не испортить исходный каталог).
7. Раскрасить имена файлов разных типов в разные цвета.
8. Выделить все файлы какого-то типа и переименовать их, например, добавив к имени цифру 1.
9. Выделить все файлы с расширением bak (резервные копии) и удалить их.
10. Задать описания свойств файлов.
11. Сделать комментарии к каталогам и файлам.
12. Вынести на верхнюю панель иконки для 2 – 3 каталогов. Убедиться, что это помогает быстро перейти в нужный каталог.
13. Создать 2 – 3 вкладки на одной из панелей.
14. Сохранить созданный набор вкладок в файле.
15. Изменить набор вкладок.
16. Восстановить прежние вкладки, загрузив их из файла.
17. Создать кнопку для копирования в буфер обмена полного имени файла.
18. Сохранить файл total.txt. Переименовать с помощью менеджера в total.doc. Открыть в редакторе Word, отформатировать и сохранить как документ Word.

Лабораторная работа № 20 (2 часа) Файловые менеджеры в Linux Файловая оболочка Midnight Commander

Хотя для управления файловой системой и вообще для работы с файлами можно использовать такие команды операционной системы, как pwd, ls, cd, mv, mkdir, rmdir, cp, rm, cat, more и т. д., гораздо удобнее делать большую часть работы по обслуживанию файловой системы с помощью программы Midnight Commander, которая наглядно представляет все выполняемые действия, облегчая тем самым выполнение указанных операций.

Midnight Commander (или просто mc) - это программа, которая позволяет просмотреть структуру каталогов и выполнить основные операции по управлению файловой системой. Другими словами, это файловый менеджер.

Для того, чтобы запустить Midnight Commander, надо набрать в командной строке оболочки mc и нажать <Enter>. Если программа не запустилась, надо найти, где расположен исполняемый файл с именем mc, воспользовавшись командой find / name mc, после чего указать в командной строке полный путь, например, у меня это /usr/bin/mc. После запуска вы увидите голубой экран, очень напоминающий экран программы Norton Commander для MS-DOS или программы FAR E. Рошалья, которая широко используется в DOS-окне под Windows.



Почти все пространство экрана при работе с Midnight Commander занято двумя "панелями", отображающими списки файлов двух каталогов. Над панелями расположена строка меню, причем к выбору команд в этом меню можно переключиться по клавише <F9> или с помощью мыши (если сразу после запуска mc вы не видите строки меню, не огорчайтесь - видна или нет строка меню, определяется настройками программы).

Самая нижняя строка представляет собой ряд экранных кнопок, каждая из которых ассоциирована с одной из функциональных клавиш <F1> - <F10>. Можно считать эту строку подсказкой по использованию функциональных клавиш, а можно и непосредственно запускать соответствующие команды, щелкая мышкой по экранной кнопке. Отображение строки с экранными кнопками можно отключить, если вы хотите сэкономить пространство экрана (об этом будет рассказано позже, когда будем говорить о настройках программы). Такая экономия оказывается оправданной по двум причинам. Во-первых, вы достаточно быстро запомните назначение этих 10 клавиш, и подсказка вам становится не нужна (а мышкой по этим клавишам щелкать не всегда удобно). Во вторых, если даже вы забыли, какая именно клавиша вам нужна для выполнения необходимого в данный момент действия, вы всегда можете воспользоваться меню **Файл** главного меню программы (только помните, что выход в главное меню осуществляется по клавише <F9>). Через меню **Файл** обеспечивается возможность выполнения любой операции из числа ассоциированных с функциональными клавишами кроме <F1> и <F9>.

Вторая снизу строка на экране - это командная строка программы Midnight Commander (точнее - командная строка текущей оболочки shell), где можно ввести и выполнить любую команду системы. Выше нее (но под панелями) может отображаться поле "полезных советов" (hint4s), которое можно убрать, соответствующим образом отрегулировав настройки программы.

Каждая панель состоит из заголовка, списка файлов какого-либо каталога и строки мини-статуса (последняя может быть не видна, это тоже задается настройками программы). В заголовке панели указан полный путь к каталогу, содержимое которого отображается в панели, а также три экранных кнопки - "<", "v" и ">", которые используются для управления программой с помощью мыши (эти кнопки не работают, если вы запустили mc в эмуляторе терминала). В строке "мини-статуса" отображаются некоторые данные о том файле или каталоге, на который в данный момент указывает подсветка (например, размер файла и права доступа к нему).

Одна из панелей является текущей (активной), о чем свидетельствует подсветка имени каталога в заголовке панели и подсветка одной из ее строк. Соответственно, в той оболочке, из которой была запущена программа Midnight Commander, текущим является каталог, отображаемый в активной панели. В этом каталоге и выполняются почти все операции. Операции типа копирования (<F5>) или переноса файла (<F6>) используют каталог, отображаемый на второй панели, в качестве целевого каталога (в который осуществляется копирование или перенос).

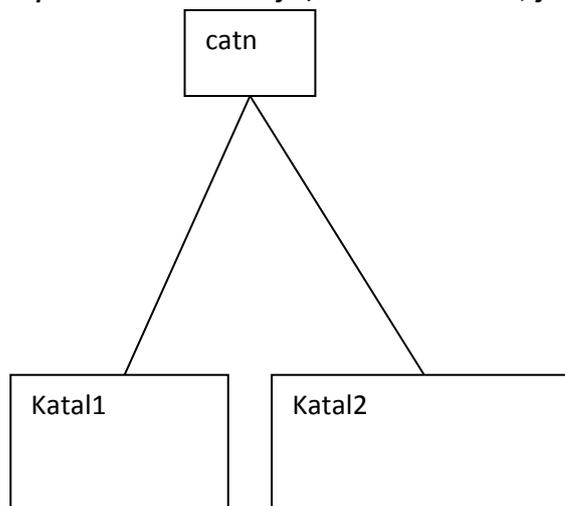
В активной панели одна строка выделена подсветкой.

При работе с программой Midnight Commander практически в любой момент можно обратиться к интерактивной подсказке, вызов которой осуществляется нажатием клавиши <F1>. Подсказка организована как гипертекст, т. е. в ее тексте встречаются гипертекстовые ссылки на другие ее разделы. Такие ссылки выделены голубым фоном.

Самостоятельная работа

Используя программу Midnight Commander выполнить следующие задания:

1. Создать следующую структуру каталогов
 2. В каталоге catn, где n –ваш номер в списке создайте файл f1 с текстом: ***Когда во власти смелых устремлений берусь я в камне воплотить черты и молотом осуществить мечты, ударом мощным властно правит гений.***
 3. Создать файл f2 с текстом: ***Но есть иной властитель побуждений недостижимой нам высоты- великий прародитель красоты и несравненных на земле творений.***
 4. Скопировать файлы f1, f2 в каталог katal1
 5. Перейти в каталог katal2 и создать в нем символические ссылки на файлы из каталога katal1
 6. Вывести списки каталогов с отображением индексных дескрипторов
 7. Посмотреть содержимое созданных файлов
 8. В каталоге catn создать жесткие ссылки на файл f1. Записать значения дескрипторов
 9. Установить для созданных файлов следующие права: автор- читать, изменять, выполнять; члены группы- читать, изменять; прочие пользователи- читать.
1. **Продемонстрировать полученные результаты преподавателю**
 2. Удалить файлы из каталога katal1
 3. Удалить каталог katal1
 4. Удалить всю структуру каталогов



Лабораторная работа № 21 (2 часа) Исследование эффективности работы архиватора

7-Zip — свободный файловый архиватор с высокой степенью сжатия данных. Поддерживает несколько алгоритмов сжатия и множество форматов данных, включая собственный формат 7z с высокоэффективным алгоритмом сжатия LZMA. Программа разработана в 1999 году и является бесплатной, а также имеет открытый исходный код, большая часть которого свободно распространяется на условиях лицензии GNU LGPL, за исключением кода декомпрессора unRAR, который имеет ограничения. Основной платформой является Windows, где доступны две версии программы: с графическим интерфейсом и версия для командной строки. Консольная версия была портирована сообществом разработчиков для систем стандарта POSIX под общим названием **r7zip**. Портированные версии для других систем, так же как и оригинальная программа **7-zip**, доступны на сайте системы SourceForge.

7-Zip использует многопоточность и позволяет задействовать для сжатия, в зависимости от алгоритма или формата различное количество потоков. При создании архивов, в которых файлы могут сжиматься независимо друг от друга (например, ZIP), программа может использовать до 8 потоков одновременно. Для алгоритма сжатия LZMA архиватор одновременно может использовать до 2 потоков. Невозможность использования большего их количества объясняется последовательным характером непрерывного сжатия. Алгоритм сжатия LZMA2 не обладает этим недостатком.

При сжатии в формате 7z используются также специальные фильтры-нормализаторы. Так, для более оптимального сжатия 32-битного x86 кода используются нормализующие конвертеры BCJ и BCJ2. Программа также имеет оптимизирующий дельта-конвертер для некоторых типов мультимедиаданных, например несжатых 24-битных изображений.

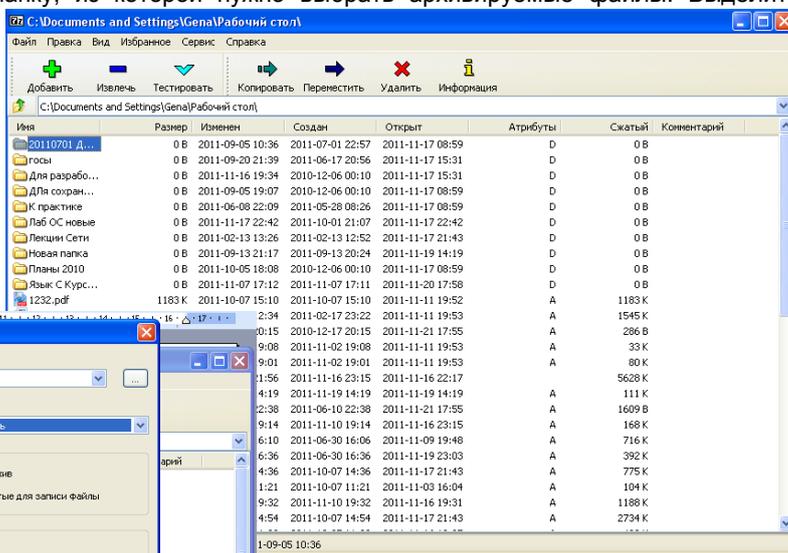
После запуска архиватора появляется окно:

В адресной строке выбрать папку, из которой нужно выбрать архивируемые файлы. Выделить нужные файлы

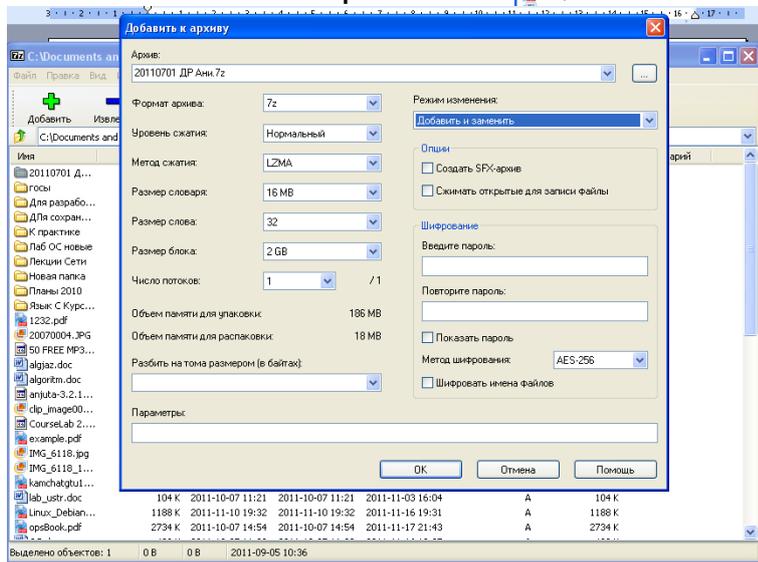
Для создания архива нужно нажать кнопку **Добавить**. Появится окно:

Выберете формат архива и уровень сжатия, число потоков и нажмите на клавишу **ОК**.

Аналогично можно упаковать папку.



Самостоятельная работа



Упакуйте с помощью архиватора 7-zip три папки:

- с текстовыми файлами
- с файлами изображений
- с мультимедийными файлами

В каждом случае используйте следующие уровни сжатия:

- нормальный
- максимальный
- скоростной

По результатам заполните таблицу:

Папка	Размер папки до сжатия	Размер архива при уровне сжатия			Коэффициент сжатия		
		нормальный	скоростной	максимальный	Нормальный	Скоростной	Максимальный

Коэффициент сжатия вычисляется по формуле:

$$K = \frac{\text{объем архива}}{\text{объем папки}} * 100\%$$

Лабораторная работа № 22 (2 часа) Изучение различных способов входа в систему

Windows XP Professional имеет более надежный набор режимов запуска, чем предыдущие версии, и может эффективно помочь в обнаружении проблем с установкой и настройками. Вы можете просмотреть список режимов запуска, нажав F8 во время загрузки.

Операционная система Windows XP Professional, предложит девять способов «поднять» компьютер и заставить его работать.

- **Safe Mode (Безопасный режим).** Запускает Windows XP Professional с минимальным набором драйверов (мышь, монитор, клавиатура, основная память и системные сервисы по умолчанию) и в режиме VGA.
- **Safe Mode With Networking (Безопасный режим с загрузкой сетевых драйверов).** Запускает Windows XP Professional в безопасном режиме, но с возможностью установки сетевого соединения.
- **Safe Mode With Command Prompt (Безопасный режим с поддержкой командной строки).** Запускает Windows XP Professional в безопасном режиме, но вместо рабочего стола на экране появляется окно команд.
- **Enable Boot Logging (Включить протоколирование загрузки).** Запускает Windows XP Professional и регистрирует все драйверы и сервисы, которые были или не были успешно загружены. Список сохраняется в файле под названием ntblog.txt, расположенном в каталоге C:\Windows.
- **Enable VGA Mode (Включить режим VGA).** Запускает Windows XP Professional с основным VGA-драйвером. Эта опция позволяет пользователям восстановить систему после некорректной установки видеодрайвера.
- **Last Known Good Configuration (Загрузка последней удачной конфигурации).** Запускает Windows XP Professional, используя информацию журнала, сохранившуюся после последнего успешного запуска системы. Этот способ не решает проблем с аппаратными устройствами и изменения, внесенные после последней загрузки, будут потеряны.
- **Debugging Mode (Режим отладки).** Режим отладки запускает Windows XP Professional, пересылая информацию об отладке на присоединенный соседний компьютер.
- **Selective Startup (Выборочный запуск).** Этот режим не указан в списке, открываемом при нажатии F8, а включается с помощью утилиты System Configuration (о которой мы поговорим позже в разделе «Утилита System Configuration»).
- **Normal Mode (Обычный режим).** Запускает Windows XP Professional в нормальном режиме.

Каждый из этих режимов полезен по-своему. Но, скорее всего, вы будете пользоваться только четырьмя режимами.

Безопасный режим

Безопасный режим является одним из главных инструментов еще со времен Windows 95. В этом режиме загружается минимальный набор драйверов. Основные функции Windows сохраняются, но вы узнаете режим Safe Mode, поскольку система обращается к VGA-режиму экрана (640x480, 16 цветов). Единственными доступными в безопасном режиме устройствами являются клавиатура, мышь и жесткий диск. Этот режим используется тогда, когда компьютер не может корректно загружаться, например, после инсталляции некачественной программы.

Безопасный режим с загрузкой сетевых драйверов

Этот режим похож на безопасный режим и позволяет установить сетевое соединение. Его можно использовать, если вам требуется попасть в сеть для решения проблемы, связанной с Windows, например, найти в сети файл или воспользоваться Удаленным помощником (Remote Assistance).

Безопасный режим с поддержкой командной строки

В двух предыдущих режимах присутствует то качество Windows, которое делает эту систему столь популярной, а именно графический пользовательский интерфейс (GUI). Однако в безопасном режиме с поддержкой командной строки вы не сможете на него положиться. Вместо этого вы будете работать в окне команд и использовать команды DOS для достижения своих целей.

Этот режим можно запустить с установочного диска Windows XP Professional, и он называется консоль восстановления (Recovery Console). Для старта консоли восстановления загрузите компьютер с установочного компакт-диска и нажмите клавишу «R» в начале процесса установки. Более того, полезно сделать консоль восстановления частью инсталляции Windows XP Professional. Для этого вставьте компакт-диск в CD- или DVD-дисковод, выберите Start\Run (Пуск\Выполнить), введите d:\i386\winnt32.exe/cmdcons (где d: — обозначение CD-дисковода) и затем щелкните на ОК.

Складывается впечатление, что этот режим доставляет много хлопот. Однако если вы знаете имена и местоположение файлов, которые вызывают проблемы, то можете найти их и уничтожить без всякого сожаления.

Загрузка последней удачной конфигурации

Система Windows XP Professional может вернуть компьютер к тому времени, когда все работало нормально. Выбрав этот стартовый режим, вы, скорее всего, потеряете все приложения, которые пытались проинсталлировать, но заодно исчезнет и маленький злодей, вызвавший неприятности. В отличие от System Restore, на контрольно-пропускном пункте Last Known Good Configuration вы не создадите исходной конфигурации для восстановления системы. Вместо этого компьютер просмотрит реестр и выберет точку для восстановления из имеющихся конфигураций. Чаще всего это срабатывает, но можно и не получить ожидаемых результатов.

Дело в том, что ваше понимание того, что такое хорошая конфигурация, и точка зрения Windows являются субъективными и не обязательно совпадают. Например, если у вас возникли проблемы с приложением или драйвером, но система Windows работает, то ОС сочтет такую конфигурацию хорошей. Для вас же она будет плохой, так как не дает ожидаемых результатов.

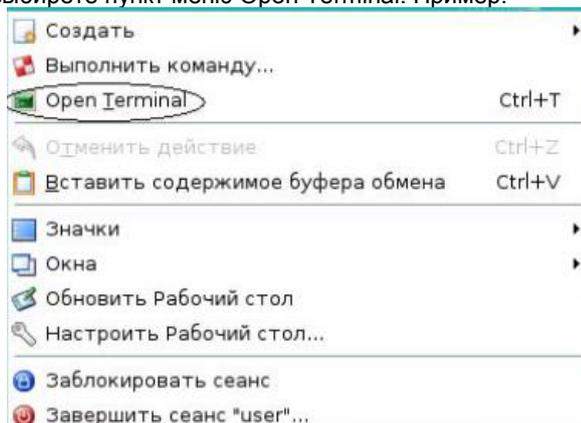
Самостоятельная работа

Загрузите систему в безопасном режиме, в безопасном режиме с поддержкой командной строки, в нормальном режиме и наблюдайте за изменениями в работе Windows

Лабораторная работа № 23 (2 часа) Изменение учетных записей и прав доступа к файлам. Система авторизации и управление правами доступа пользователей

Технические средства Виртуальная машина с ОС Linux

1. Зарегистрируйтесь в графической подсистеме пользователем user и запустите программу konsole. Это делается так - установив мышку на рабочем столе нажмите на мышке правую клавишу, выберите пункт меню Open Terminal. Пример:



2. Из графической консоли переключитесь на пользователя root с помощью команды su -, введите пароль root.



Пример:

3. Просмотрите список пользователей в файле /etc/passwd командой cat /etc/passwd.

```
[root@lmaster4r2 ~]# cat /etc/passwd
root:x:0:0:System Administrator:/root:/bin/bash
bin:x:1:1:bin:/:/dev/null
daemon:x:2:2:daemon:/:/dev/null
adm:x:3:4:adm:/var/adm:/dev/null
lp:x:4:7:lp:/var/spool/lpd:/dev/null
mail:x:8:12:mail:/var/spool/mail:/dev/null
news:x:9:13:news:/var/spool/news:/dev/null
tomcat:x:118:134:Apache Tomcat:/usr/share/tomcat5:/bin/sh
admin:x:500:500:~/home/admin:/bin/bash
user:x:501:100:~/home/user:/bin/bash
dictd:x:119:36:~/var/lib/dictd:/dev/null
student2:x:503:503:~/home/student2:/bin/bash
[root@lmaster4r2 ~]#
```

Пример:

4. Просмотрите с помощью программы getent подробные сведения об учетной записи пользователей root и user. Пример:

```
[root@lmaster4r2 ~]# getent passwd root
root:x:0:0:System Administrator:/root:/bin/bash
[root@lmaster4r2 ~]# getent passwd user
user:x:501:100::/home/user:/bin/bash
[root@lmaster4r2 ~]#
```

5. Просмотрите список групп в файле /etc/group. Пример:

```
[root@lmaster4r2 ~]# cat /etc/group
root:x:0:
bin:x:1:root
daemon:x:2:root
sys:x:3:root,bin,adm
adm:x:4:root
tty:x:5:
disk:x:6:root
...|
conshelp:x:135:
admin:x:500:
dictd:x:36:
student2:x:503:
[root@lmaster4r2 ~]#
```

6. Просмотрите содержимое файла /etc/shadow. Он не содержит паролей пользователей, т.к. используется альтернативный shadow.

```
[root@lmaster4r2 ~]# cat /etc/shadow
root:x:14494:0:0:
bin:!:14494:0:0:
daemon:!:14494:0:0:
adm:!:14494:0:0:
lp:!:14494:0:0:
gdm:!:14494:0:0:
ldap:!:14494:0:0:
exim:!:14494:0:0:
apache:!:14494:0:0:
nobody:!:14494:0:0:
[root@lmaster4r2 ~]#
```

7. У вас есть пользователь user, просмотрите содержимое файла /etc/tcb/user/shadow. Пример:

```
[root@lmaster4r2 ~]# cat /etc/tcb/user/shadow
user:$2a$08$1LbvQfIN2mphHalMU3j00eZS3l9Worpr93yZz4FywxyLIcSn8gw7a:14529:0:0:
[root@lmaster4r2 ~]#
```

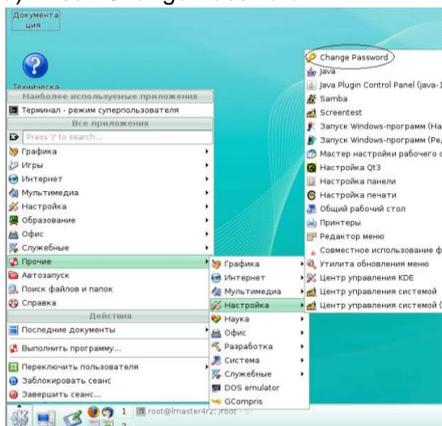
Это и есть зашифрованный пароль пользователя user

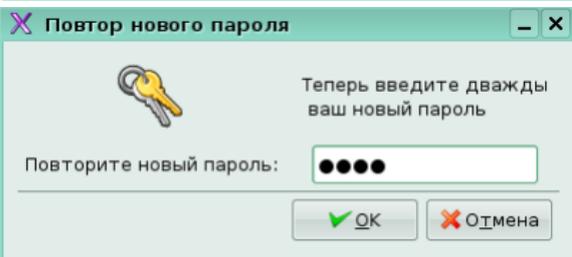
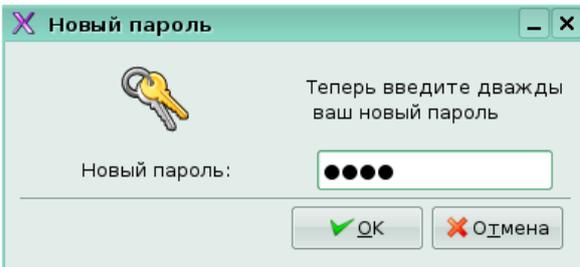
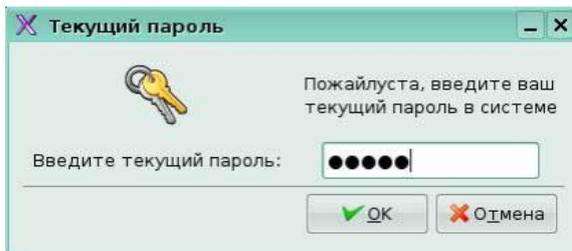
8. Закройте графическую консоль.

9. Из графической подсистемы запустите программу Change Password.

10. Измените текущему пользователю user пароль. Пример.

а) вызов Change Password:





11. Переключитесь с помощью клавиш `ctrl-alt-f1` в текстовый режим и зарегистрируйтесь пользователем `root`. Пример:

```

Линукс Мастер

Welcome to ALT Linux 4.0 Junior (HeyTeacher) / tty1
lmaster4r2 login: root
Password:
Last login: Mon Oct 12 15:31:52 MSD 2009 from localhost on tty1
[root@lmaster4r2 ~]#
  
```

14. Создайте пользователей `student1`, `student2`, в качестве пароля для обоих укажите `password`, определите `uid`, `gid` пользователя `student1` с помощью команды `id`, переключитесь на пользователя `student1` с помощью команды `su`, убедитесь в этом с помощью команды `whoami`, вернитесь в `root`. Пример:

```

[root@lmaster4r2 ~]# useradd student1
[root@lmaster4r2 ~]# useradd student2
[root@lmaster4r2 ~]# passwd student1
passwd: updating all authentication tokens for user student1.

You can now choose the new password or passphrase.

Enter new password:
Weak password: not enough different characters or classes for this length.
Re-type new password:
passwd: all authentication tokens updated successfully.
[root@lmaster4r2 ~]# passwd student2
passwd: updating all authentication tokens for user student2.

You can now choose the new password or passphrase.

Enter new password:
Weak password: not enough different characters or classes for this length.
Re-type new password:
passwd: all authentication tokens updated successfully.
[root@lmaster4r2 ~]# id student1
uid=502(student1) gid=502(student1) groups=502(student1)
[root@lmaster4r2 ~]# su - student1
[student1@lmaster4r2 ~]$ whoami
student1
[student1@lmaster4r2 ~]$ exit
  
```

15. Переключитесь на пользователя student2 и создайте два файла file1.txt и file2.txt.

Пример:

```
[root@lmaster4r2 ~]# su - student2
[student2@lmaster4r2 ~]$ touch file1.txt
[student2@lmaster4r2 ~]$ touch file2.txt
```

16. Просмотрите содержимое домашней директории пользователя student2, определите права у файлов file1.txt и file2.txt. Пример:

```
[student2@lmaster4r2 ~]$ ls -l
итого 12
drwx----- 2 student2 student2 4096 Окт 11 13:45 Desktop
drwx----- 2 student2 student2 4096 Окт 11 13:20 Documents
-rw-r--r-- 1 student2 student2  0 Окт 12 15:54 file1.txt
-rw-r--r-- 1 student2 student2  0 Окт 12 15:54 file2.txt
drwx----- 5 student2 student2 4096 Окт 12 14:30 tmp
[student2@lmaster4r2 ~]$
```

17. Измените владельца файла file1.txt на user. Пример:

```
[root@lmaster4r2 student2]# chown user /home/student2/file1.txt
[root@lmaster4r2 student2]# ls -l /home/student2/file1.txt
-rw-rw-r-- 1 user student2 0 Окт 12 17:01 /home/student2/file1.txt
```

18. Измените группу файла на student1. Пример:

```
[root@lmaster4r2 student2]# chgrp student1 /home/student2/file1.txt
[root@lmaster4r2 student2]# ls -l /home/student2/file1.txt
-rw-rw-r-- 1 user student1 27 Окт 12 17:28 /home/student2/file1.txt
```

19. Проверьте доступ пользователя user к файлу /home/student2/file1.txt, владельцем которого он является. Пример:

```
[user@lmaster4r2 ~]$ touch /home/student2/file1.txt
touch: невозможно выполнить touch для '/home/student2/file1.txt': Отказано в доступе
```

20. Узнайте почему нет прав и дайте доступ к файлу /home/student2/file1.txt.

Пример:

```
[root@lmaster4r2 student2]# ls -ld /home/student2
drwx----- 15 student2 student2 4096 Окт 12 15:54 /home/student2
```

Причина отказа в доступе в том, что нет доступа пользователя user к директории, где лежит файл /home/student2/file1.txt. Пример предоставления доступа к директории:

```
[root@lmaster4r2 student2]# chmod go+rwx /home/student2
[root@lmaster4r2 student2]# ls -ld /home/student2
drwxrwxrwx 15 student2 student2 4096 Окт 12 15:54 /home/student2
```

Проверим доступ:

```
[user@lmaster4r2 ~]$ touch /home/student2/file1.txt
[user@lmaster4r2 ~]$ █
```

Доступ есть!

21. Измените права доступа к файлам file1.txt и file2.txt (разрешить группе изменять файл)

```
[user@lmaster4r2 ~]$ chmod g+w /home/student2/file1.txt
[user@lmaster4r2 ~]$ ls -l /home/student2/file1.txt
-rw-rw-r-- 1 user student2 27 Окт 12 17:15 /home/student2/file1.txt
```

22. Удалите файлы file1.txt и file2.txt самостоятельно.

23. Завершить сессию shell.

Лабораторная работа № 24 (2 часа)

Написание командного файла и анализ его работы

Пакетные файлы

1. В любом текстовом редакторе (Блокнот, Ncedit и т.п.) создать пакетный файл с именем <ваше имя>.bat, который создает на диске C: два каталога Primer1 и Primer2
2. Запустите созданный файл на выполнение и проверьте результат его работы (на диске C: должны появиться два каталога, с именами указанными выше)
3. Откройте каталог Primer1 и создайте в нем два текстовых файла f1 и f2. В файл f1 запишите свои фамилию, имя, отчество, а в файл f2 запишите сегодняшнюю дату.
4. Измените, пакетный файл таким образом, чтобы он копировал файлы f1 и f2 из каталога Primer1 в каталог Primer2. Строки создания каталогов нужно отключить, но не удалять.
5. Запустите созданный файл на выполнение и проверьте результат его работы.
6. Измените, пакетный файл таким образом, чтобы он удалял файл f1 из каталога Primer1 после копирования.
7. Покажите работу преподавателю.
8. Запустите созданный файл на выполнение и проверьте результат его работы
9. Измените, пакетный файл таким образом, чтобы он копировал два любых файла по заданию преподавателя из корневого каталога в каталог Primer2
10. Продемонстрируйте работу преподавателю
11. Запишите в отчет текст пакетного файла и прокомментируйте каждую его строку

Лабораторная работа № 25 (2 часа)

Изучение файловой системы Linux

(Файловые системы и диски)

Цель работы: получение практических навыков работы с файловыми системами, дисками и самостоятельной работы с документацией команд.

Поясняющая информация

Команды POSIX для работы с ФС и дисками (должны быть во всех операционных системах)

df - выводит информацию о подмонтированных дисках

Команды и конфигурационные файлы в LINUX

fdformat - форматирование гибкого диска

mformat - создает файловую систему MS-DOS

fsck - проверка файловой системы

mkfs - создание файловой системы (форматирование)

mkswap - создание раздела подкачки

swapon - активизация раздела подкачки

fdisk - разбивка диска

mount - монтирование файловых систем

umount - размонтирование файловых систем

df - выводит информацию о подмонтированных дисках

/etc/fstab - файл для описания подключаемых файловых систем

Команды для работы с Windows

Chkdsk - вывод на экран отчета о состоянии диска

Chkntfs - просмотр или задание планирования автоматической проверки системы для томов файловых систем FAT, FAT32 или NTFS при запуске компьютера.

Cipher - отображение или изменение шифрования папок и файлов на томах NTFS

Compact - вывод сведений или изменение уплотнения файлов и каталогов в разделах NTFS
Convert - преобразование томов с файловой системой FAT и FAT32 в тома с файловой системой NTFS.

Defrag - доиск и объединение фрагментированных файлов

DiskPart - Программа DiskPart.exe — это работающий в текстовом режиме командный интерпретатор, который позволяет управлять объектами (дисками, разделами или томами) с помощью сценариев или команд, вводимых с командной строки.

Format - Форматирование диска

Fsutil (поддерживаются только с версии Windows 5.1) - является служебной программой командной строки, которая используется для выполнения связанных задач файловых систем FAT и NTFS.

Подкоманды для нее:

behavior - запрашивает, изменяет, включает или отключает настройки для создания имен файлов с длиной 8.3 символа, для возможности отображения расширенных символов в имена файлов длиной 8.3 символа в томах NTFS, для обновления штампа времени последнего доступа к томам NTFS, частоты событий, записанных в системных журнал, и количества места на диске, зарезервированного зоной основной таблицы файлов.

dirty - запрос установки «грязного» бита тома. Задает «грязный» бит тома. Когда установлен «грязный» бит тома, autochk автоматически проверит том на наличие ошибок при следующей перезагрузке компьютера.

file - поиск файла по идентификатору безопасности, запрос файла в размещенных диапазонах, установка короткого имени файла, допустимой длины данных или нулевых данных для файла.

fsinfo - перечисляет все диски, запрашивает тип диска, сведения о томе, специальные сведения о томе NTFS или статистику файловой системы.

hardlink - создает жесткую ссылку.

objectid Обычно используется специалистами по технической поддержке. Управляет идентификаторами объектов, которые используются Windows XP для отслеживания объектов, таких как файлы и каталоги.

quota - управляет дисковыми квотами в томах NTFS

reparsepoint - работа с точками монтирования

sparse - управление разреженными файлами.

usn - управление журналом изменения , в котором сохраняется архив всех изменений файлов в томе.

volume - размонтирование тома и отображение свободного места на диске.

Label - служит для создания, изменения или удаления метки тома (т. е. имени) диска.

Mountvol - служит для создания, удаления и получения списка точек подключения тома.

Subst - устанавливает соответствие пути в файловой системе заданному диску.

Vol - отображает метку тома диска и серийный номер, если они существуют

Самостоятельная работа

Задание 1

1. Запустите Linux.
2. Составьте справочник для выше приведенных команд (на русском языке), расписав какие параметры для чего нужны.
3. Поработайте с этими командами.

Задание 2

1. Запустите Windows
2. Составьте справочник для выше приведенных команд (на русском языке), расписав какие параметры для чего нужны.
3. Поработайте с этими командами.

Лабораторная работа № 26 (2 часа) Изучение структуры реестра Windows

Технические средства Виртуальная машина с ОС Windows

Реестр и его роль

Реестр (registry) представляет собой иерархически организованную базу данных, которую ОС использует для запуска приложений, хранения системных правил, профилей пользователей и прочих настроек и т.д. Также реестр используется практически всеми приложениями для хранения своих настроек.

Впервые реестр был использован в Windows 95, с тех пор он стал быстрее и лучше защищен от всевозможных ошибок.

Хотя реестр организован довольно понятным образом, он представляет собой сложную и обширную структуру.

Реестр можно рассматривать как записную книжку Windows - как только системе нужна какая-то информация, она ищет ее в реестре. Реестр очень обширен, и дать однозначное его определение невозможно. Кратко и достаточно точно можно сказать, что *реестр - компонент операционной системы компьютера, который в иерархической базе данных хранит важнейшие установки и информацию о приложениях, системных операциях, пользовательской и аппаратной конфигурациях.*

История реестра

Изначально каждая программа хранила нужные для себя настройки и данные своим собственным способом, как правило – в виде файлов собственной структуры. Поэтому при переносе программы с одной машины на другую достаточно было настроить эти файлы (как правило – прописать нужные имена дисков и каталоги, т.к. каждая машина имела свои диски). С другой стороны, эта процедура – тогдашний эквивалент инсталляции - порой была очень сложной. В Windows 3.x была сделана попытка упорядочить формат и способ хранения конфигурационных файлов. В частности, всем им было предписано носить расширение .ini.

Реестр был создан потому, что с файлами INI, которые появились в Windows 3.x, пользователям приходилось думать, какой файл INI за что отвечает и как изменить в нужном файле ту или иную настройку. Часто было трудно выяснить месторасположение таких файлов для нужной программы. Кроме того, отсутствовал способ определения того, какая программа связана с конкретным INI- файлом. С другой стороны, редактировать такие файлы было намного легче, чем реестр. Одним из самых важных различий между файлами INI и реестром являлось расположение файлов – INI- файлы принадлежали программе, в то время, как реестр – часть Windows. Именно поэтому в наше время невозможно «просто перенести» программу – требуется записать ее настройки в реестр, что, собственно, и составляет процедуру инсталляции программы.

Где расположены файлы реестра

В ОС Windows 9x реестр хранится в двух файлах: **System.dat** и **User.dat**. Эти два файла находятся в папке с Windows. Если на данном компьютере несколько пользователей, то система создает несколько файлов **User.dat**.

В Windows NT (2000/XP) и Vista (Vista/7) есть специальный каталог SYSTEM32\CONFIG, хранящий в виде защищенных файлов разделы реестра.

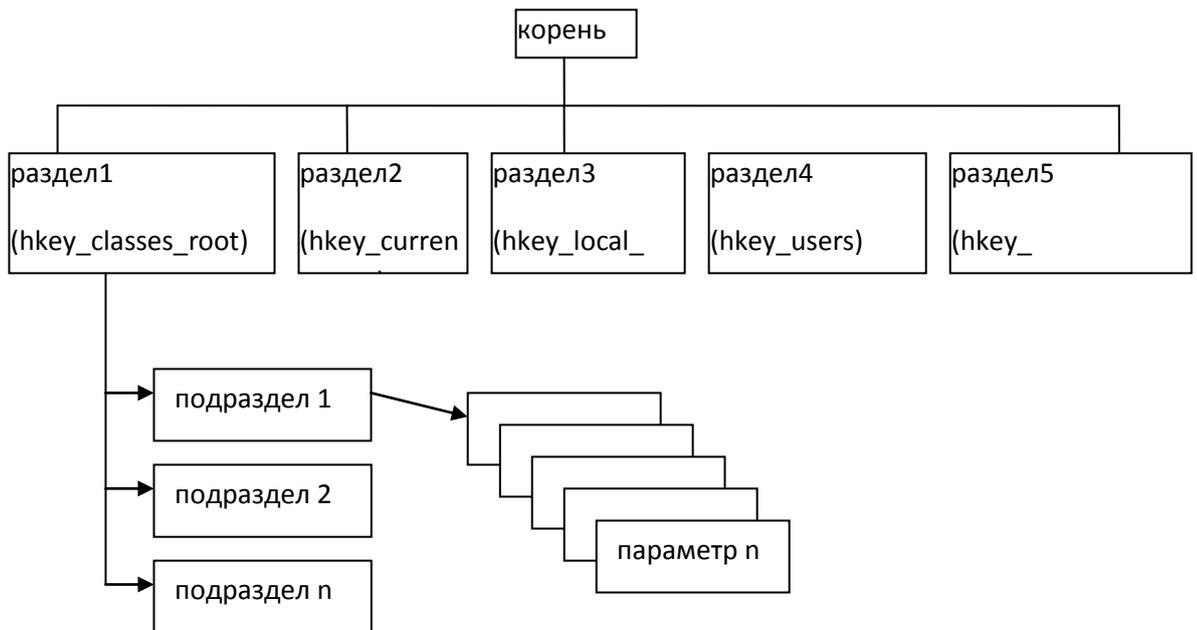
Роль реестра

Реестр можно рассматривать как записную книжку Windows - как только системе нужна какая-то информация, она ищет ее в реестре. Реестр очень обширен, и дать однозначное его определение невозможно. Кратко и достаточно точно можно сказать, что *реестр - компонент операционной системы компьютера, который в иерархической базе данных хранит важнейшие установки и информацию о приложениях, системных операциях, пользовательской и аппаратной конфигурациях.*

Архитектура реестра

В целом реестр очень напоминает файловую систему с той разницей, что вместо файлов на нижнем уровне содержатся параметры.

Информация, хранящаяся в иерархической базе данных реестра, собрана в разделы (key), которые содержат один или более подразделов (subkey). Каждый подраздел содержит параметры (value):



Возможность создавать вложенные подразделы позволяет группировать параметры. В результате получается древовидная структура, которую можно просмотреть в Редакторе реестра (Registry editor, RegEdit). Каждый раздел (ветвь) соответствует определенному типу информации о пользователе, аппаратном обеспечении, приложении и т.д.

Изменяя тот или иной параметр, можно управлять работой Windows, защитить компьютер от нежелательных пользователей и просто настраивать внешний вид Windows.

В частности, в разделе

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

содержится список параметров. Имена этих параметров не играют роли для системы, а значения представляют собой имена исполняемых файлов, которые следует запускать всякий раз при запуске системы. Добавив туда свой параметр, можно заставить систему запускать свою программу.

Редактирование реестра

RegEdit - это программа которая позволяет редактировать файлы реестра. Запустить ее можно из командной строки, либо через меню Start->Run. В левой части окна отображается вся иерархическая структура реестра, в правой – параметры, существующие в текущем разделе. Следует отметить, что параметры могут находиться даже в корне реестра.

Файлы реестра

Regedit позволяет импортировать и экспортировать часть реестра в файлы. Эта возможность, например, может быть использована для создания резервных копий, либо для переноса ПО с одной машины на другую. Структура этих файлов такова:

```

REGEDIT4
[раздел реестра]
"параметр"="строковое значение"
  
```

Если параметр имеет тип dword, то соответствующая строка должна иметь вид:

"параметр"=dword:00000000 (где вместо 00000000 надо задать нужное значение).

А если тип параметра двоичный, то формат строки:

"параметр"=hex:00,00,00,00 (где через запятую указываются значения байтов в шестнадцатеричном виде).

Следует обратить внимание, что в конце файла (*.reg) обязательно должна быть пустая строка.

Кроме того, следует учитывать, что в значении строкового параметра перед символами "кавычки" и "обратный слеш" (\) должен добавляться символ "обратный слеш". А параметр "(По умолчанию)" обозначается символом "@" (без кавычек).

Пример задания параметра

Чтобы присвоить параметру "(По умолчанию)" значение

"C:\Program Files\Accessories\WORDPAD.EXE" "%1", надо записать:

@="C:\Program Files\Accessories\WORDPAD.EXE" \"%1\""

Созданный reg-файл следует запустить на выполнение (с файлами типа *.reg по умолчанию ассоциирован редактор реестра REGEDIT.EXE).

Пример файла, содержащего параметр для автоматического запуска драйвера мыши

REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]

"Gnetmous"="G:\genius\gnetmous.exe"

Для того чтобы полностью удалить раздел (ключ) реестра с помощью reg-файла, надо перед именем раздела поставить "-" (без кавычек).

В Windows 2000/XP/Vista/7 используются файлы реестра 5-й версии (в windows NT – 4-й). Они отличаются поддержкой Unicode (каждый символ занимает два байта вместо одного) и заголовком «Windows Registry Editor Version 5.00» вместо «REGEDIT4». Файлы 4-й версии по-прежнему используются и поддерживаются.

В «свежеустановленной» системе размер реестра составляет примерно 12 – 15Мб, увеличиваясь со временем до 20 – 25 Мб. Поскольку реестр содержит чисто текстовую информацию, можно оценить количество параметров, учитывая, что один параметр занимает около 100 байт. В силу этого не существует полного описания реестра (не следует также забывать, что каждая программа заносит в реестр что-то свое). Однако, есть множество программ, называемых «твикерами» или «настройщиками», предназначенных для удобного редактирования некоторых системных параметров реестра. Эти программы содержат достаточно подробное описание изменяемых параметров и рекомендации по настройке. Наиболее обширной на сегодняшний день является система Xteq X-setup, позволяющая модифицировать около 1000 различных значений. Умелое ее использование способно обеспечить в отдельных случаях (на сильно «замусоренной системе») 20 – 30% прироста производительности системы. Чаще же всего использование «улучшалонок» реестра приводит к появлению всевозможных проблем, в силу чего рекомендовать такие программы к применению не стоит.

Ремонт реестра

В ряде случаев возникает острая необходимость в исправлении реестра системы. Например, популярные вирусы категории «смс-блокеры» (такой вирус запускается при старте системы, блокирует работу компьютера, доступ к меню и диспетчеру задач (см. ниже) и требует выслать смс на определенный сотовый номер), лечатся именно таким образом. Первые, примитивные версии этих вирусов, просто записывались в вышеупомянутый раздел Run, соответственно, лечение состояло в поиске и удалении «лишней» записи. Современные вирусы блокируют редактор реестра, поэтому приходится либо загружать компьютер со специального восстановительного диска на базе Linux (например, Dr. Web LiveCD), либо снимать жесткий диск, подключать его к другому ПК и редактировать реестр, используя команду «Загрузить куст» (Load Hive) редактора реестра, либо получать удаленный доступ к реестру посредством все того же редактора (эта мера популярна в Vista).

Самостоятельная работа

- 1.1. Создайте в реестре параметр, благодаря которому при каждой загрузке ОС будет автоматически запускаться Microsoft Word.
- 1.2. Создайте список всех параметров из раздела автозапуска.

Лабораторная работа № 27 (2 часа) Настройка клиентов для работы в сети Сетевые ресурсы. Клиентская часть

Целью данного практического задания является установка имени компьютера, настройка сетевого интерфейса и клиентской части NFS, DNS.

1. Установите имя компьютера как stationN.example.com, где N – номер, который вам скажет преподаватель.

Для начала введите команду hostname для определения текущего имени компьютера.

Пример:

```
#hostname  
localhost.localdomain
```

Это задание можно выполнить двумя способами.

Способ 1:

а) Текстовым редактором от пользователя root изменить параметр HOSTNAME в файле /etc/sysconfig/network.

Пример:

```
HOSTNAME=station10.example.com
```

б) Перезапустить систему.

Пример:

```
#reboot
```

в) ввести команду для определения имени компьютера

```
#hostname  
station10.example.com
```

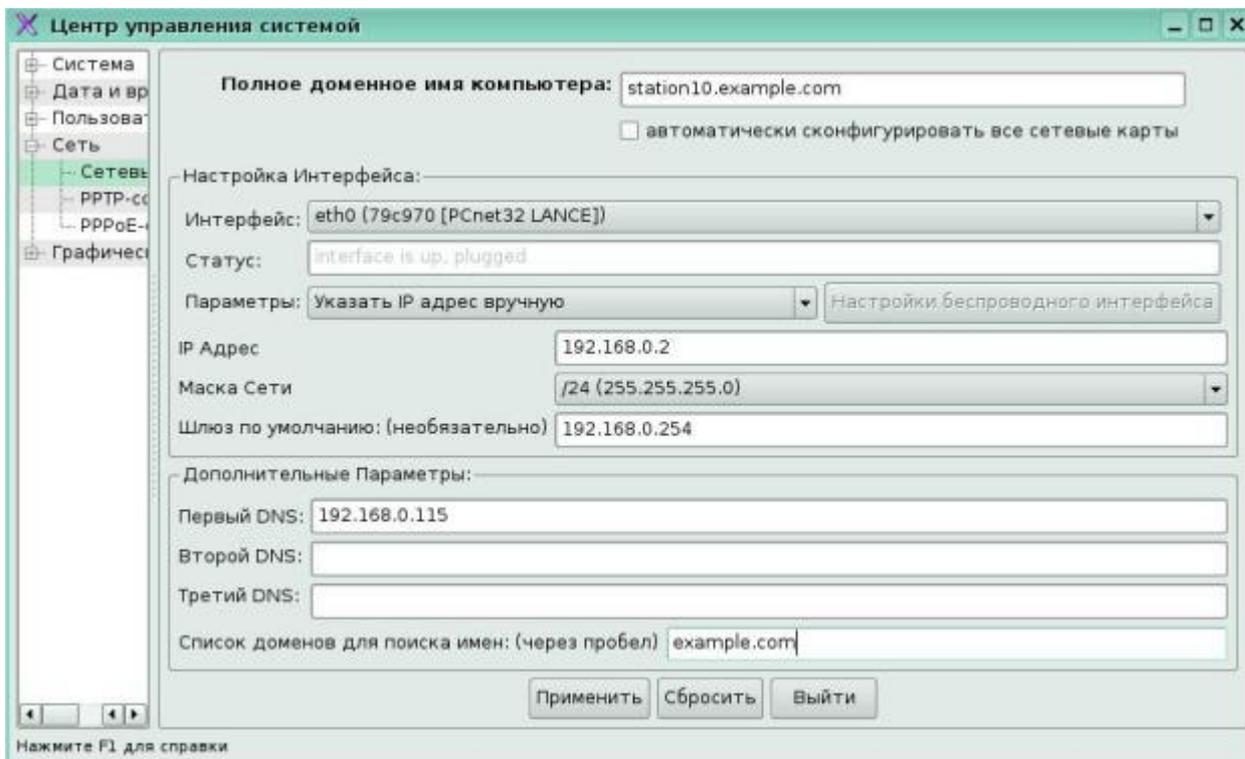
Способ 2:

а) Запустить программу **Центр Управления Системой**

б) В разделе **Сеть -> Сетевые интерфейсы** в поле **Полное доменное имя**

компьютера прописать значение stationN.example.com, где N – номер, который вам скажет преподаватель.

Пример:



2. Установите возможность выхода вашего компьютера в другую сеть.

Сценарий: Существуют две различные сети, 192.168.0.0/24 и 192.168.1/24. Ваша система в 192.168.0.0/24 сети. Роутер имеет адреса 192.168.0.254 и 192.168.1.254. Для того чтобы ваша система могла взаимодействовать с сетью 192.168.1.0/24.

Это задание можно выполнить двумя способами.

Способ 1:

а) в текстовом редакторе нужно в файле /etc/sysconfig/network прописать параметр GATEWAY=192.168.0.254 и перезапустить сетевую службу

```
#mcedit /etc/sysconfig/network
```

```
GATEWAY=192.168.0.254
```

Выйти из редактора с сохранением изменений

2)перезапустить службу network.

Пример:

```
#service network restart
```

Способ 2:

Или запустить программу **Центр управления системой** и в настройке для **Сетевых соединений** в поле **Шлюз по умолчанию** прописать значение **192.168.0.254**.

3. У вашего основного DNS сервера адрес 192.168.0.254. Настройте вашу систему для использования этого DNS сервера.

Для настройки использования DNS сервера по адресу 192.168.0.254 нужно в текстовом редакторе в файл /etc/resolv.conf добавить строку:

```
nameserver 192.168.0.254
```

Проверить работу DNS сервера нужно командой

```
#host server1.example.com
```

где server1.example.com – DNS сервер .

Лабораторная работа № 28 (2 часа) Запуск Windows – приложений в среде Linux (WINE - Запускаем Windows-приложения на Linux)

Wine - не эмулятор, и поэтому ему не требуется время для эмуляции - Wine запускает приложения так, как это делает Windows. Вы сможете запускать практически все приложения, используемые вами в Windows и настроить их по своему вкусу, то есть загрузить собственные шрифты и т.д.

Wine - это свободное программное обеспечение, позволяющее пользователям UNIX-подобных систем исполнять 16 и 32-битные приложения Microsoft Windows.

Установка Wine:

Воспользуемся терминалом суперпользователя (ТОЛЬКО для установки):

```
sudo -s
```

Добавим список репозитория, ключ, и установим Wine:

```
RELEASE=`lsb_release -cs`  
elif [[ $RELEASE = "lucid" || $RELEASE = "karmic" || $RELEASE = "jaunty" || $RELEASE = "intrepid" || $RELEASE =  
"hardy" ]]; then  
wget -q http://wine.budgetdedicated.com/apt/sources.list.d/$RELEASE.list -O /etc/apt/sources.list.d/winehq.list  
wget -q http://wine.budgetdedicated.com/apt/387EE263.gpg -O- | apt-key add -  
apt-get update && apt-get install -y wine cabextract
```

Примечание: если вы используете не Ubuntu, а, например, Linux Mint 9, то выражение «[http://wine.budgetdedicated.com/apt/sources.list.d/\\$RELEASE.list](http://wine.budgetdedicated.com/apt/sources.list.d/$RELEASE.list)» замените на «<http://wine.budgetdedicated.com/apt/sources.list.d/lucid.list>», иначе при выполнении команды будет подставлено значение «isadora», а не «lucid», и установка не пройдет.

Затем создадим в домашней директории папку `~/.wine/`, запустив конфигуратор Wine:

```
winecfg
```

Пока нам там делать ничего не нужно, закроем.

Установка и удаление приложений:

Для установки приложения Windows воспользуемся файловым браузером winefile:

```
winefile
```

Затем переходим в нужный каталог нашей файловой системы Linux либо виртуального диска `C:/` и двойным щелчком запускаем нужный исполняемый файл. Многие приложения после установки требуют перезапуска системы, но мы работаем в виртуальной среде, и поэтому компьютер перезагружать не следует. Нужно только перезапустить Wine (при этом завершить остальные работающие приложения до перезагрузки):

```
wineboot
```

Если установка производится из файла с расширением `*.msi`, то воспользуемся командой:

```
msiexec /i example.msi
```

где `example.msi` - название вашего файла.

Для удаления приложения Windows нужно выполнить следующее:

Переходим в «Приложения» → «Wine» → «Программы» → «<Папка с программой>» → «Uninstall».

Если же после установки приложения, его папка не появилась в «Приложения» → «Wine» → «Программы», то воспользуемся терминалом. Например, чтобы запустить `example.exe`, находящийся на виртуальном диске `C:/` в директории `Program Files/Random`, переходим в соответствующую директорию и запускаем его:

```
cd ~/.wine/drive_c/"Program Files"/Random example.exe
```

Общие настройки:

Приложения:

По умолчанию установлен Windows XP, который можно изменить на другую версию. Причем можно настроить так, чтобы каждое приложение запускалось с параметром определенной системы.

Библиотеки:

По умолчанию все библиотеки в Wine используются свои.

Графика:

Для некоторых приложений может потребоваться включить или выключить вершинные и пиксельные шейдеры, может потребоваться эмуляция виртуального рабочего стола. Попробуйте изменять параметры этого раздела сами.

Вид и интеграция: Во вкладке «Графика», по умолчанию стоят галочки «разрешить менеджеру окон управлять окнами Wine» и «разрешить менеджеру окон декорировать окна», оформление win-окон будет таким же, как и у любых других приложений. При желании вы можете управлять размерами системных шрифтов и цветом десктопа.

Диски: Здесь можно добавлять диски, используя для них любую директорию вашей файловой системы.

Аудио: При работающем звуке ничего изменять не рекомендуется.

Помните!

При запуске Wine желательно завершить все работающие приложения и отключить Compoz (эффекты).

Самостоятельная работа

1. Установить Windows приложения по заданию преподавателя в Wine
2. Проверить работоспособность установленных приложений

6.1.2. Время выполнения лабораторной работы (см. таблицу ниже)

Название лабораторной работы	Время выполнения
Лабораторная работа №1. Знакомство с операционной системой Linux	1 час
Лабораторная работа № 2. Знакомство с текстовым редактором Vim	1 час
Лабораторная работа №3. Процессы в ОС Linux	1 час
Лабораторная работа № 4. Управление процессами Linux	1 час
Лабораторная работа № 5. Процессы в Windows	1 час
Лабораторная работа № 6. Организация процессов через pipe	1 час
Лабораторная работа №7. Изучение файловой системы Linux	2 часа
Лабораторная работа №8. Работа с файлами и директориями в Linux	2 часа
Лабораторная работа №9. Работа с файлами и директориями в Windows	2 часа
Лабораторная работа №10. Создание разделов диска с помощью программы fdisk	2 часа
Лабораторная работа №11. Изучение состава внешних устройств на компьютере и их настроек.	2 часа
Лабораторная работа №12. Файлы устройств. Аппарат прерываний	2 часа
Лабораторная работа №13. Семейство протоколов TCP/IP. Сокеты в Linux	2 часа
Лабораторная работа №14. Установка операционной системы Windows	2 часа
Лабораторная работа №15. Установка операционной системы Linux	2 часа
Лабораторная работа №16. Изменение пользовательских интерфейсов и настроек в Linux	2 часа
Лабораторная работа №17. Установка программного обеспечения в Windows	2 часа
Лабораторная работа №18. Установка программного обеспечения в Linux	2 часа
Лабораторная работа №19. Файловые менеджеры в Windows	2 часа
Лабораторная работа №20. Файловые менеджеры в Linux	2 часа
Лабораторная работа №21. Исследование эффективности работы архиваторов	2 часа
Лабораторная работа №22. Изучение различных способов входа в систему	2 часа
Лабораторная работа №23. Изменение учетных записей и прав доступа к файлам	2 часа
Лабораторная работа №24. Написание командного файла и анализ его работы	2 часа
Лабораторная работа №25. Изучение файловой системы Linux	2 часа
Лабораторная работа №26. Изучение структуры реестра Windows	2 часа
Лабораторная работа №27. Настройка клиента для работы в сети	2 часа
Лабораторная работа №28. Запуск Windows-приложений в среде Linux	2 часа

6.1.3. Перечень объектов контроля и оценки

Наименование объектов контроля и оценки	Основные показатели оценки результата	Оценка
Лабораторная работа №1. Знакомство с операционной системой Linux		
У1. Умение использовать средства операционных систем для решения практических задач	Правильно выполненная лабораторная работа	1 балл
З1. Знание основных функций операционных систем	Работа с «горячими» клавишами при выполнении лабораторной работы	
Лабораторная работа № 2. Знакомство с текстовым редактором Vim		
У1. Умение использовать средства операционных систем для решения практических задач	Созданные два текстовых файла и сохраненные под своим именем	1 балл
Лабораторная работа №3. Процессы в ОС Linux		
У2. Умение использовать сервисные средства, предоставляемые с операционной системой	Отчет о дополнительных сведениях, которые были представлены командой top	1 балл
З1. Знание основных функций операционных систем	Выведение сведений о процессах в динамике в ОС Linux	
З3. Знание принципов построения операционных систем		
Лабораторная работа № 4. Управление процессами Linux		
У1. Умение использовать средства операционных систем для решения практических задач	Правильно выполненная лабораторная работа (созданный файл отчета имя текущего терминала, информацию о пользователях, работающих в системе, название и версию операционной системы, время работы системы)	1 балл
З3. Знание принципов построения операционных систем	Работа с командами fifo, tty, w, uname, uptime и обработчиком сигнала SIGINT.	
Лабораторная работа № 5. Процессы в Windows		
У2. Умение использовать сервисные средства, предоставляемые с операционной системой	Отчет о основных компонентах клиентской службы Windows	1 балл
З1. Знание основных функций операционных систем	Принцип работы с диспетчером задач	
З3. Знание принципов построения операционных систем		
Лабораторная работа № 6. Организация процессов через pipe		
У2. Умение использовать сервисные средства, предоставляемые с операционной системой	Набранная и откомпилированная программа. Отчет с анализом результата запуска программы.	1 балл
З2. Знание машинно-независимых свойств операционных систем	Организацию процессов через pipe	
Лабораторная работа №7. Изучение файловой системы Linux		
У2. Умение использовать сервисные средства, предоставляемые с операционной системой	Выполнять команды в файловой системе Linux	1 балл
З4. Знание сопровождения операционных систем	Отчет о работе с файловой системой Linux	
Лабораторная работа №8. Работа с файлами и директориями в Linux		
У1. Умение использовать средства операционных систем для решения практических задач	Работать с файлами и каталогами в ОС Linux	1 балл
З1. Знание основных функций операционных систем	Основные команды для работы с файлами и каталогами в ОС Linux	
Лабораторная работа №9. Работа с файлами и директориями в Windows		
У1. Умение использовать средства операционных систем для решения практических задач	Работать с файлами и каталогами в ОС Windows	
З1. Знание основных функций операционных систем	Основные команды для работы с файлами и каталогами в ОС Windows	
Лабораторная работа №10. Создание разделов диска с помощью программы fdisk		
У1. Умение использовать средства операционных систем для решения практических задач	Создавать разделы диска с помощью программы fdisk	1 балл
Лабораторная работа №11. Изучение состава внешних устройств на компьютере и их настроек.		
У4. Умение подключать к операционной системе новые сервисные средства	Установленные программы на ПК: Sysinfo.exe, checkit.exe	1 балл
З4. Знание сопровождения операционных систем	Sysinfo.exe, checkit.exe	
Лабораторная работа №12. Файлы устройств. Аппарат прерываний		
У4. Умение подключать к операционной системе новые сервисные средства	Добавление и удаление файлов устройств	1 балл
З4. Знание сопровождения операционных систем	Аппаратные прерывания устройств	
Лабораторная работа №13. Семейство протоколов TCP/IP. Сокеты в Linux		
У5. Умение решать задачи операционных систем	Работать с протоколом TCP/IP и сокетами Linux	1 балл
З4. Знание сопровождения операционных систем	Подключение к сети при помощи протокола TCP/IP	
Лабораторная работа №14. Установка операционной системы Windows		
У5. Умение решать задачи операционных систем	Устанавливать ОС Windows на ПК	1 балл
З4. Знание сопровождения операционных систем	Установленная ОС Windows на ПК	

Лабораторная работа №15. Установка операционной системы Linux		
У5. Умение решать задачи операционных систем	Устанавливать ОС Linux на ПК	1 балл
34. Знание сопровождения операционных систем	Установленная ОС Linux на ПК	
Лабораторная работа №16. Изменение пользовательских интерфейсов и настроек в Linux		
34. Знание сопровождения операционных систем	Изменение пользовательских интерфейсов и настроек в Linux	1 балл
Лабораторная работа №17. Установка программного обеспечения в Windows		
У2. Умение использовать сервисные средства, поставляемые с операционной системой	Устанавливать ПО в Windows	1 балл
34. Знание сопровождения операционных систем	Установленное ПО в Windows	
Лабораторная работа №18. Установка программного обеспечения в Linux		
У2. Умение использовать сервисные средства, поставляемые с операционной системой	Устанавливать ПО в Linux	1 балл
34. Знание сопровождения операционных систем	Установленное ПО в Linux	
Лабораторная работа №19. Файловые менеджеры в Windows		
У2. Умение использовать сервисные средства, поставляемые с операционной системой	Установленный файловый менеджер в Windows	1 балл
34. Знание сопровождения операционных систем	Принцип установки файлового менеджера в Windows	
Лабораторная работа №20. Файловые менеджеры в Linux		
У2. Умение использовать сервисные средства, поставляемые с операционной системой	Установленный файловый менеджер в Linux	1 балл
34. Знание сопровождения операционных систем	Принцип установки файлового менеджера в Linux	
Лабораторная работа №21. Исследование эффективности работы архиваторов		
У2. Умение использовать сервисные средства, поставляемые с операционной системой	Архивированные файлы	1 балл
Лабораторная работа №22. Изучение различных способов входа в систему		
У3. Умение использовать различные операционные системы	Вход в различные ОС (Windows и Linux)	1 балл
31. Знание основных функций операционных систем	Знание принципов входа в различные ОС (Windows и Linux)	
Лабораторная работа №23. Изменение учетных записей и прав доступа к файлам		
У3. Умение использовать различные операционные системы	Изменение учетной записи и прав доступа к файлам в разных ОС (Windows и Linux)	1 балл
32. Знание машинно-независимых свойств операционных систем	Знание принципа изменения учетной записи и прав доступа к файлам в разных ОС (Windows и Linux)	
Лабораторная работа №24. Написание командного файла и анализ его работы		
У5. Умение решать задачи операционных систем	Создавать командные файлы	1 балл
34. Знание сопровождения операционных систем	Отчет о работе созданного командного файла	
Лабораторная работа №25. Изучение файловой системы Linux		
У2. Умение использовать сервисные средства, поставляемые с операционной системой	Отчет о работе с файловыми системами, дисками и самостоятельной работы с документацией команд. (Windows и Linux)	1 балл
У3. Умение использовать различные операционные системы		
32. Знание машинно-независимых свойств операционных систем	Знать о работе с файловыми системами, дисками и самостоятельной работы с документацией команд.	
Лабораторная работа №26. Изучение структуры реестра Windows		
У5. Умение решать задачи операционных систем	Изменение настроек реестра Windows	1 балл
33. Знание принципов построения операционных систем	Знание возможности и необходимость изменения настроек реестра Windows	
Лабораторная работа №27. Настройка клиента для работы в сети		
У1. Умение использовать средства операционных систем для решения практических задач	установка имени компьютера, настройка сетевого интерфейса и клиентской части NFS, DNS.	1 балл
34. Знание сопровождения операционных систем	Знание принципа установки имени компьютера, настройки сетевого интерфейса и клиентской части NFS, DNS.	
Лабораторная работа №28. Запуск Windows-приложений в среде Linux		
У3. Умение использовать различные операционные системы	Установка Wine из приложения Windows в ОС Linux	1 балл
34. Знание сопровождения операционных систем	Знание принципа установки Wine из приложения Windows в ОС Linux	

За правильное выполнение лабораторной работы выставляется положительная оценка – 1 балл.

За неправильное выполнение лабораторной работы выставляется отрицательная оценка – 0 баллов.

6.2. Вопросы к зачету (1 семестр)

1. Операционные системы. Эволюция операционных систем.
2. Особенности современного этапа развития ОС. Требования к современным ОС.
3. Операционные системы. Классификация современных операционных систем.
4. Архитектура ОС. Ядро и вспомогательные модули ОС. Аппаратная зависимость и переносимость ОС.
5. Сетевые операционные системы. Структура сетевой ОС.
6. ОС для рабочих групп и ОС для сетей масштаба предприятия (корпоративные ОС).
Признаки корпоративных ОС.
7. Назначение и функции ОС. ОС для автономного компьютера. Функциональные компоненты ОС автономного компьютера.
8. Программное обеспечение ЭВМ. Классификация программного обеспечения.
9. Базовое программное обеспечение: операционная система, операционные оболочки, системные утилиты.
10. Основные команды в cmd для работы с каталогами в командной строке.
11. Командная строка cmd. Работа с файлами: создание, редактирование, удаление. Текстовый редактор edit.
12. Операционные оболочки. Операционная оболочка Fag. Основные приёмы работы. Функциональные клавиши, их назначение.
Операционные системы семейства Windows. Общий обзор операционных систем.
13. История возникновения ОС. Основные системные требования. Изменение настроек Windows.
14. Пакеты прикладных программ. Интегрированный пакет Microsoft Office.
15. Проблемы безопасности операционных систем. Свойства безопасной системы. Защитные механизмы операционных систем.
16. Операционные системы семейства Windows. Файловая система ОС WINDOWS. Работа в программе Проводник. Клавиши быстрого доступа.
17. Сервисное программное обеспечение: программы-драйверы, программы-упаковщики, программы оптимизации, программы восстановления информации и др.
18. Архиваторы: Виды, возможности, назначение, приемы работы.
19. Процессы и потоки. Управление процессами. Планирование процессов и потоков.
20. Ввод-вывод и файловая система (ФС). Задачи ФС по управлению файлами и устройствами.
21. Логическая организация файловой системы. Файловая система. Файл, типы файлов. Иерархическая структура файловой системы. Имена файлов. Атрибуты файлов.
22. Вирусы, их классификация. Антивирусные программы. Примеры антивирусных программ.
23. Операционная система Unix. Обзор ОС семейства UNIX. Основные системные требования. Разновидности Unix.
24. Операционная система Linux. Дистрибутивы Linux. История возникновения ОС Linux. Файловая система Linux. Структура каталогов. Виды графических оболочек.
25. Операционная система Linux. Работа в программе Konqueror. Создание структуры каталогов.
Работа с ППП в ОС Linux. Текстовый редактор OpenOffice.org writer. Работа с электронными таблицами OpenOffice.org Calc. Создание презентаций в программе OpenOffice.org Impress. Сравнительная сложность с MS Office.
- 26.

27. ОС Linux. Работа с терминалом UNIX. Распределение знаков. Операции с каталогами в командной строке.
28. ОС Linux. Командная строка Работа с файлами: создание, редактирование, удаление. Текстовый редактор vi, ed.
29. ОС Linux. Файловый менеджер MC. Работа с панелями. Создание собственной иерархической структуры каталогов.

6.4. Вопросы к дифференцированному зачету по дисциплине «Операционные системы и среды» (2 семестр)

1. Понятие операционной системы.
2. Типы операционных систем.
3. Назначение и функции операционной системы.
4. Понятие программного интерфейса, его назначение.
5. Виды программных интерфейсов
6. Командный интерфейс. Виды реализации.
7. WIMP- интерфейс. Виды реализации.
8. SILK- интерфейс. Виды реализации.
9. Классификация операционных систем.
10. Упрощенная архитектура типовой микроЭВМ.
11. Логическая структура микропроцессорной системы.
12. Программное прерывание и его обработка.
13. Последовательность действий при обработке прерываний.
14. Задание. Определение, смысл.
15. Планирование заданий.
16. Понятие операционного окружения. Состав, назначение.
17. Утилиты расширения функциональности ОС.
18. Процесс. Определение, смысл.
19. Диспетчеризация процессов.
20. Способы выбора процесса для диспетчеризации.
21. Организация ввода-вывода.
22. Управление вводом-выводом, как функция ОС.
23. Многоуровневая организация подсистемы ввода-вывода.
24. Механизм разделения центральной памяти.
25. Функции ОС по управлению памятью.
26. Организация защиты памяти в ЭВМ.
27. Виртуальная память. Механизм реализации.
28. Сегментная организация памяти.
29. Механизм реализации страничной адресации памяти.
30. Этапы связывания адресов памяти.
31. Файловая система ОС.
32. Логическая организация файловой системы.
33. Иерархическая структура файловой системы.
34. Файловые операции.
35. Имена файлов. Общие принципы создания.
36. Шаблон имени файла.
37. Конкатенация файлов.
38. MS DOS. Правила создания имени файла.
39. Windows. Правила создания имени файла.
40. Монтирование файловой системы.
41. Атрибуты файлов.
42. Логическая организация файлов.
43. Защищенность операционных систем.

44. Отказоустойчивость операционных систем.
45. Структура ОС MS DOS.
46. Модули в составе MS DOS.
47. Назначение модуля IO.SIS.
48. Структура ОС MS Windows.
49. Технология Drag and Drop.
50. Рабочий стол, его элементы.
51. Виды окон Windows.
52. Фрагментация диска.
53. Microsoft Management Console.
54. Структура ОС MS Linux.
55. Межпроцессорное взаимодействие в различных ОС.
56. Загрузка операционной системы. Принципы и основные понятия.
57. Загрузка операционной системы Linux.
58. Загрузка операционной системы Windows.
59. Интерфейс пользователя. Приглашение системы.
60. Ввод команд.
61. Запуск и выполнение команд.
62. Работа с дисками в MS DOS.
63. Метка диска.
64. Работа каталогами в MS DOS.
65. Работа файлами в MS DOS.
66. Работа с дисками в Windows.
67. Работа с каталогами в Windows.
68. Работа файлами в Windows.
69. Пакетные командные файлы.
70. Рекурсивный вызов командных файлов.
71. Файл AUTOEXEC.BAT.
72. Основные команды пакетных файлов.
73. Принципы работы с ОС Ubuntu.
74. Операционная оболочка NC. Назначение.
75. Работа с окнами NC.
76. Назначение функциональных клавиш NC.
77. Элементы основного меню NC.
78. Операционная оболочка Far. Назначение.
79. Работа с окнами Far.
80. Назначение функциональных клавиш Far.
81. Элементы основного меню Far.
82. Установление кодировок в FAR.
83. Технология сравнения каталогов в FAR.
84. Подключение плагинов Far.
85. Назначение архиваторов.
86. Профили архивации.
87. Разница и преимущества форматов RAR и ZIP.
88. Типы лицензий на использование WinRAR.
89. Самораспаковывающийся архив.
90. Коэффициент сжатия, в чём его смысл.
91. Оболочка DOS Shell.
92. Технология создания в оболочке DOS Shell каталога.
93. Технология создания в оболочке DOS Shell файла.
94. Назначение операционной системы Ubuntu.
95. Эмуляторы ОС.
96. Технология виртуализации.
97. Сетевые ОС.
98. Серверные технологии.
99. Администрирование сервера.
100. Перспективы развития ОС.

6.5. Билеты для экзамена по дисциплине «Операционные системы и среды» (3 семестр)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Понятие операционной системы. Эволюция операционных систем
2. Файловая система. Основные функции.
3. Задача. Показать на ПК и объяснить работу следующих команд MS DOS **cd**; **cls**; **date**; **dir**.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 2

1. Назначение и функции операционной системы.
2. Работа с файлами в Total Commander.
3. Задача. Файловый менеджер FAR.
Создайте в текущей папке FAR дерево папок и файлов, изображенное на рисунке. При этом в файл определение1.txt запишите определение понятия «файл», а в файл определение2.txt - логическое определение понятия «папка», в файл адрес.txt – путь к файлу определение2.txt от корневой папки диска.



ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 3

1. Классификация операционных систем в зависимости от особенностей использованного алгоритма управления процессором.
2. Файловый менеджер Total Commander. Типы файловых менеджеров.
3. Задача. Файловый менеджер Total Commander
 1. Выполнить следующие действия с объектами в Total Commander (копирование и перемещение объектов выполнить 2 способами):
 - a. создать папку со своим именем в папке D:\Students\«название группы»;
 - b. найти все текстовые файлы на диске C: в подкаталоге Windows;
 - c. скопировать первые 5 найденных файлов в свою созданную папку;
 - d. вывести на экран общие сведения о своей папке и об одном из скопированных файлов;
 2. Привести пример применения опций расширенного копирования файлов.
 3. Произвести поиск файлов в каком либо каталоге, используя различные примеры шаблонов.
 4. Заархивировать исходный файл, содержащий только текст с помощью архиваторов ZIP и RAR, сделать вывод о степени сжатия этими архиваторами. Сделать такую же последовательность действий для исходного файла, содержащего кроме текста изображение. Сделать выводы.

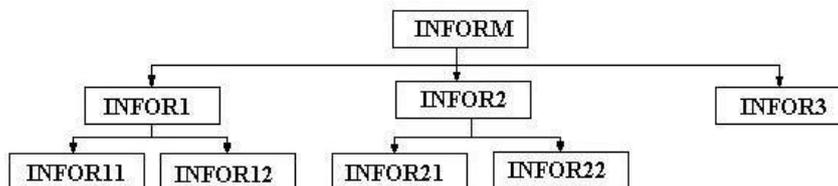
ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 4

1. Интерфейс пользователя. Виды интерфейсов.
2. Файловая система. Типы файлов.
3. Задача. Работа с файловой структурой MS DOS
Имеется следующее описание файловой структуры: «На диске C содержатся две папки GROUP и WORK и один файл расписание.txt. Папка GROUP содержит в себе две папки КОЛЛЕДЖ и ОГУ, а также один файл студент.doc. Папка WORK содержит папку WORK1. Папка WORK1 содержит файл works.exe.» Изобразите согласно этому описанию дерево папок и файлов описанной файловой структуры.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 5

1. Стандартные сервисные программы поддержки интерфейса пользователя с ОС.
2. Интерфейс файлового менеджера Total Commander.
3. Задача. Работа с файловой структурой MS DOS.

В корневом каталоге диска D создать папку «Экзамен», в этой папке создать дерево папок

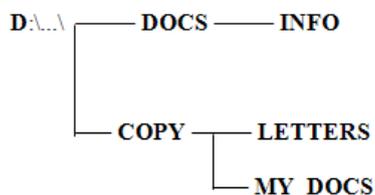


1. Выведите дерево данного каталога со всеми его ветвями.
2. Очистите экран.
3. Создать текстовый файл **text11.txt** в каталоге **INFOR11**. В текстовом файле написать определение операционной системы MS DOS.
4. Создать текстовый файл. **text12.txt** в каталоге **INFOR12**. В текстовом файле написать определение Операционной системы.
5. Создать текстовый файл. **text21.txt** в каталоге **INFOR21**. В текстовом файле написать определение Программного обеспечения.
6. Создать текстовый файл. **text22.txt** в каталоге **INFOR22**. В текстовом файле написать определение Файла.
7. Скопировать файлы **text11.txt**, **text12.txt** в каталог **INFOR3**.
8. Переместить файлы **text21.txt**, **text22.txt** в каталог **INFOR3**

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 6

1. Основные понятия безопасности. Классификация угроз.
2. Иерархическая структура файловой системы.
3. Задача. Работа с файловой структурой MS DOS.

Создайте следующую структуру каталогов в своей папке:



1. Используя команду **Дерево папок** убедитесь в том, что структура каталогов создана правильно.
2. Создайте в каталоге **INFO** текстовые файлы **name.txt** и **group.txt**. В первом файле запишите свои имя и фамилию, во втором – название специальности, курс и номер группы.
3. Допишите в файл **name.txt** дату своего рождения.
4. Используя копирование, создайте в каталоге **INFO** файл **info.txt**, объединяющий содержимое файлов **name.txt** и **group.txt**, просмотрите получившийся файл.
5. Скопируйте файл **info.txt** в каталог **LETTERS**.
6. Переименуйте файл **info.txt** в каталоге **LETTERS** в файл **info1.txt**.
7. Скопируйте файлы **name.txt** и **group.txt** в каталог **MY_DOCS**.
8. Очистите экран
9. Просмотрите текущую версию операционной системы на данном компьютере, время и дату.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 7

1. Логическая организация файловой системы.
2. Основные функции файлового менеджера Total Commander.
3. Задача. В программной оболочке FAR-manager выполнить задание:
 1. Создать файл визитка.txt, содержащий бланк визитки, по следующему образцу:

ФАМИЛИЯ: _____
ИМЯ: _____
ОТЧЕСТВО: _____
ДОМАШНИЙ АДРЕС: _____
ТЕЛЕФОН: _____

2. Внесите изменения в файл визитка.txt (см. п.1), дополнив в него графы визитки:

E-MAIL _____

ХОББИ _____

Создайте файлы с содержимым, приведенным в таблице

Имя файла	Содержимое файла
Пример1.txt	Файл – это поименованная область данных на диске или других носителях информации
Пример2.txt	«ОБЩЕПРИНЯТЫЕ РАСШИРЕНИЯ В ИМЕНИ ФАЙЛА» txt doc com exe pas for
Пример3.txt	***** *FAR-manager – * * это * * программная * * оболочка. * *****

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №8

1. Понятие прерывания. Виды прерываний
2. Физическая организация файловой системы.
3. Задача. Показать на ПК и объяснить работу следующих команд MS DOS:
md; ver; time; dir.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 9

1. Понятие процесса. Состояния процесса
2. Работа с файловым менеджером Total Commander: работа с файлами, каталогами, дисками.
3. Задача. Работа с общесистемными командами MS DOS.
 1. Вызовите справку команды data и time.
 2. Посмотрите текущую дату и время компьютера.
 3. Измените приглашение системы на: текущий диск, текущее время, знак равенства
 4. Посмотрите версию DOS компьютера.
 5. Вызовите справку команды prompt.
 6. Очистите экран.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 10

1. Понятие процесса
2. Файловые операции, контроль доступа к файлам.
3. Задача. Файловый менеджер Total Commander.

Задание на обслуживание файловой структуры в Total Commander.

 1. Создайте папку «Третий урок по ТС » в ней создайте папки «ПРАКТИКА1 в ТС», «ПРАКТИКА2 в ТС», «ПРАКТИКА3 в ТС», «ПРАКТИКА4 в ТС» в них файл «Работа1 с корзиной», «Работа2 с корзиной», «Работа3 с корзиной», «Работа4 с корзиной».
 2. Удалите файл «Работа 1 с корзиной», а затем папку «ПРАКТИКА 1 в ТС» в корзину.
 3. Удалите файл «Работа 2 с корзиной», а затем папку «ПРАКТИКА 2 в ТС» без предварительного помещения в корзину.
 4. Уничтожение файла или папки. Уничтожить файл «Работа3 с корзиной».
 5. Копирование папки или файла двумя способами копирования файла (папки):
 - 1) копирование с использованием двух панелей Total Commander.
 - 2) копирование с использованием дерева папок.

Скопировать папку «ПРАКТИКА3 в ТС» с диска E (F,D) на диск C. Скопировать файл визитка1 на диск C.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 11

1. Прерывание
2. Примеры файловых систем.
3. Задача. Создание масок в файловом менеджере Total Commander.

Организуите процесс поиска на вашем компьютере с помощью файлового менеджера Total Commander файлов, начинающихся с латинской буквы s, имеющие в имени третью букву l; всех файлов, имеющих расширение txt; всех файлов, заканчивающихся на «o» и имеющих любое расширение.

Составьте маску для одновременного поиска на одном из дисков вашего компьютера файлов *config.txt*, *comag.exe*, *controg.txt*.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 12

1. Планирование процессов
2. Файловая система. Структура файла.
3. Задача. Работа в операционной системе MS DOS.

Создать программу из команд MS DOS и отладить ее на компьютере:

1. Очистить экран монитора
2. На диске C: в корневом каталоге создать каталог «Ekzamen1»
3. В этом каталоге создать новую папку со своей фамилией латинскими буквами (например: Petrov)
4. Перейти в каталог «Ekzamen1»
5. Просмотреть содержимое этого каталога (вывод оглавления)
6. Узнать версию операционной системы
7. Узнать время компьютера

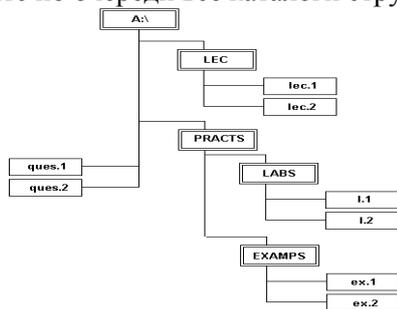
ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 13

1. Алгоритмы планирования процессов
2. Отказоустойчивость файловых и дисковых систем.
3. Задача. Показать на ПК и объяснить работу следующих команд MS DOS:
rd; md; ver; cd; cd..; copy; format.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 14

1. Виды программного обеспечения (системное и прикладное).
2. Служебные программы.
3. Задача. Работа с файловой структурой MS DOS.

Создайте по очереди все каталоги структуры в соответствии с рисунком.



1. В файле lec.1 запишите формат команд DOS и их назначение для работы с файлами.
2. В файле lec.2 запишите формат команд DOS и их назначение для работы с каталогами.
3. В файл **ques.1** запишите формат следующих команд и их назначение: вход в каталог, выход в наддиректорию, просмотр дерева каталогов, очистка экрана.
4. В файл **ques.2**: копирования файлов, смена текущего диска.
5. Файлы **l.1** и **l.2** создайте путем объединения файлов **lec.1** и **ques.1**, **lec.2** и **ques.2**, соответственно.
6. Файлы **ex.1** и **ex.2** создайте путем копирования файлов **lec.1** и **lec.2**.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 15

1. Классификация ресурсов.
2. Базовые технологии безопасности.
3. Задача. Файловый менеджер Total Commander.
 1. Выполнить следующие действия с объектами в Total Commander (копирование и перемещение объектов выполнить 2 способами):
 - а. создать папку со своим именем в папке C:\Students\«название группы»;
 2. Создать в меню «Запуск» команду, запускающую текстовый процессор MS Word, графический редактор Paint.
 3. Вызвать из главного меню вкладку Поиск файлов.
 - А) Найдите на диске С все файлы, с любым именем и расширением, содержащие слово «Практическая», созданные сегодня или вчера, размер которых не превышает 2 КБ, с любым атрибутом.
 - Б) Пользуясь расширенным поиском найдите следующие файлы и сохраните маски в качестве шаблонов:
 - «Шаблон 1» - все файлы с расширением jpeg, созданный на прошлой неделе, размер которых не превышает 100 КБ, атрибут архивный.
 - «Шаблон 2» - все файлы начинающиеся на букву Р с расширением doc, содержащие слово «информатика».
 - «Шаблон 3» - все файлы последняя буква которых «а» с любым расширением, созданные сегодня, атрибут только для чтения.
 - «Шаблон 4» - все файлы вторая и четвертая буква которых а, с расширением doc, содержащие слово «Файл».
 - «Шаблон 5» - Найти файлы с расширением doc на диске С:, содержащие слово «работа», с атрибутом только для чтения, созданные в этом месяце.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 16

1. Интерфейс программной оболочки FAR. Основные команды.
2. Файловая система FAT
3. Задача. Работа с файловой структурой MS DOS.
 1. Создайте на диске С: в директории USER директорию (1), имя которой = Вашей фамилии.
 2. Создайте в директории USER директорию (2), имя которой состоит из сегодняшней даты (например: 12января2012года).
 3. В директории (1) создайте текстовый файл ss.txt, в котором наберите свою фамилию, имя, отчество. Там же создайте файлы f1.doc, f2.doc, f3.ttt, f4.ttt с любым содержимым.
 4. Скопируйте файл ss.txt в директорию (2) под именем aa.txt.
 5. Объедините файлы ss.txt и aa.txt в один, поместите его в директории (1) под именем ac.txt.
 6. Переименуйте файл ac.txt в ac1.txt.
 7. Просмотрите содержимое файла ac1.txt с помощью двух разных команд.
 8. Скопируйте все файлы с расширением .doc в директорию (2).
 9. Поместите в директории (1) файл report.txt с результатом сравнения файлов ss.txt и ac1.txt.
 10. Просмотрите содержимое корневого каталога в широком формате.
 11. Измените, вид системного приглашения, восстановите прежний вид

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 17

1. Определение семафоров. Реализация семафоров.
2. Планирование процессов. Понятие очереди.
3. Задача. Файловый менеджер Far Manager.

Задание на обслуживание файловой структуры (создание, копирование, перемещение, действия с группой файлов) в Far Manager

 1. Создайте папку «Третий урок по Far » в ней создайте папки «ПРАКТИКА1 в FAR», «ПРАКТИКА2 в FAR», «ПРАКТИКА3 в FAR», «ПРАКТИКА4 в FAR» в них файл «Работа1 с корзиной», «Работа2 с корзиной», «Работа3 с корзиной», «Работа4 с корзиной».
 2. Копирование папки или файла двумя способами копирования файла (папки):

- 1) копирование с использованием двух панелей FAR-manager;
- 2) копирование с использованием дерева папок.
3. Скопировать папку «ПРАКТИКА3 в FAR» с диска E (F,D) на диск C. Скопировать файл визитка1 на диск C.
4. **Переименование папки или файла.** Переименовать файл «Работа4 с корзиной» в файл «Переименованный файл». Переименовать папку «ПРАКТИКА4 в FAR» в папку «Переименованная папка».
5. **Перемещение папки или файла.** Переместить файл «Переименованный файл» в папку со своей фамилией.
6. **Действия с группой файлов и папок.** Группу файлов «визитка1», «Пример 1», переместить в папку «Переименованная папка».

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 18

1. Типы операционных систем.
2. Интерфейс пользователя Windows
3. Задача. Операционная система MS DOS

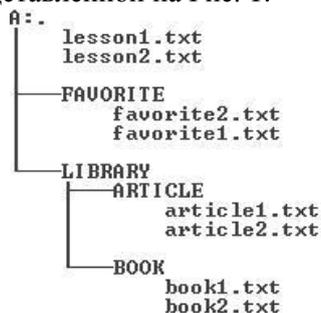
Создайте в текущей папке MS DOS дерево папок и файлов, изображенное на рисунке. При этом в файл *определение1.txt* запишите определение понятия «файл», «файловая система», а в файл *определение2.txt* - определение понятия «папка», в файл *адрес.txt* – путь к файлу *определение2.txt* от корневой папки диска.



ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 19

1. Файловая система NTFS.
2. Инструментальные средства (утилиты).
3. Задача. Работа с файловой структурой Far Manager.

1. Создайте структуру папок файловым менеджером FAR Manager в соответствии со структурой папок и файлов, представленной на Рис. 1.



2. Текстовые файлы lesson1.txt и lesson2.txt создайте путем ввода текста с клавиатуры.
3. Файлы favorite1.txt и favorite2.txt создайте методом копирования файлов lesson1.txt и lesson2.txt.
4. Файлы article1.txt и article2.txt создайте методом копирования группы файлов и переименования их после копирования в папке A:\LIBRARY\ARTICLE.
5. Файлы book1.txt и book2.txt переместите в папку A:\LIBRARY\BOOK из папки A:\FAVORITE, где создайте их методом копирования и вставки текста из других файлов
6. Определите размер каталога LIBRARY.
7. Используя выделение в группу по шаблону, определите, сколько файлов с расширением bat записаны в корневом каталоге жесткого диска, какой суммарный размер они имеют. Определите аналогичные характеристики для файлов с расширением txt.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 20

1. История развития операционных систем.
2. Основные функциональные клавиши Total Commander.
3. Задача. Архивирование файлов программой WinRAR.
 1. Создайте в своей рабочей папке (папке с вашей группой) следующие папки: папку со своей фамилией, в ней папки **Архивы**.
 2. Найдите на компьютере файлы трех типов **.doc, .bmp, .exe**, и скопируйте их в папку **Архивы**.
 3. Заархивируйте графический файл и сравните размеры обоих файлов.
 4. Сравните размеры исходного файла и архивного. Данные внесите в **таблицу 1**.
 5. Заархивируйте файл типа .doc и сравните размеры обоих файлов. Данные внесите в **таблицу 1**.
 6. Заархивируйте файл типа .exe и сравните размеры обоих файлов. Данные внесите в **таблицу 1**.
 7. На основании данных таблицы сделайте вывод о преимуществах архивирования файлов различных типов.

Формат архива	Имя файла и его расширение	Исходный размер	Размер после архивации	$K = \frac{\text{Размер архива}}{\text{Размер файла в архиве}} \cdot 100\%$ Процент качества.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 21

1. Классификация операционных систем
2. Конфигурирование системы. Реестр
3. Задача. Работа с масками в Far Manager
 1. Создайте на рабочем диске следующие файлы (работа в FAR-manager):
baby.txt, lady.txt, sandy.exe
 2. Для каждого из данных файлов необходимо составить маску, которая найдет данный файл.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 22

1. Программные оболочки FAR.
2. Файловая структура MS DOS.
3. Задача. Программная оболочка Far Manager.

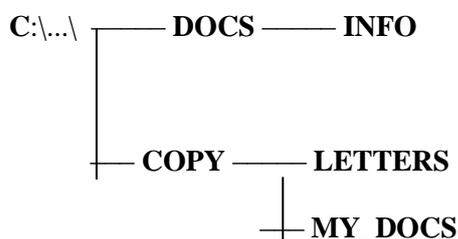
В программной оболочке Far Manager выполнить задание:

 1. Запустите файловый менеджер FAR и разверните его, если необходимо, на весь экран.
 2. Создайте следующую структуру каталогов в своей папке:
 D:\..\ \ — DOCS — INFO
 |
 — COPY — LETTERS
 |
 — MY_DOCS
 3. Используя панель Дерево папок, убедитесь в том, что структура каталогов создана правильно. Вернитесь к виду панели Средний
 4. Используя поиск, найдите на диске C: файлы и папки, имена которых начинаются с буквосочетания inf.
 5. Используя Панель информации, определите, сколько свободного места осталось на диске C:.
 6. Используя различные режимы сортировки, определите файл с самой поздней датой создания, файл с наибольшим размером. Запишите эту информацию.
 7. Просмотрите режимы панелей Краткий, Средний, Полный, Широкий. Чем они отличаются друг от друга.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 23

1. Структура операционной системы MS-DOS
2. Структура оперативной памяти
3. Задача. Файловый менеджер Total Commander.

Создайте следующую структуру каталогов в своей папке:



1. Используя команду **Дерево папок**, убедитесь в том, что структура каталогов создана правильно. Вернитесь к первоначальному виду панели.
2. Создайте в каталоге **INFO** текстовые файлы **name.txt** и **group.txt**. В первом файле запишите свои имя и фамилию, во втором – название специальности, курс и номер группы.
3. Допишите в файл **name.txt** дату своего рождения.
4. Используя копирование, создайте в каталоге **INFO** файл **info.txt**, объединяющий содержимое файлов **name.txt** и **group.txt**, просмотрите получившийся файл.
5. Скопируйте файл **info.txt** в каталог **LETTERS**.
6. Переименуйте файл **info.txt** в каталоге **LETTERS** в файл **info1.txt**.
7. Используя выделение, скопируйте файлы **name.txt** и **group.txt** в каталог **MY_DOCS**.
8. Скройте от просмотра файлы в каталоге **INFO** (**Файл- атрибуты файла**)
9. Используя поиск, найдите на диске **C:** файлы и папки, имена которых начинаются с буквосочетания **inf**.
10. Используя поиск, найдите файлы на диске **C:** в которых записана Ваша фамилия.
11. Определите, сколько свободного места осталось на диске **C:**.
12. Используя различные режимы сортировки, определите файл с самой поздней датой создания, файл с наибольшим размером.
13. Используя выделение в группу по шаблону, определите, сколько файлов с расширением **bat** записаны в корневом каталоге жесткого диска, какой суммарный размер они имеют. Определите аналогичные характеристики для файлов с расширением **txt**. Запишите эту информацию. Если возможно, скопируйте файлы с расширением **bat** в каталог **LETTERS** на диске **C:**.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 24

1. Состав основных компонентов операционной системы.
2. Интерфейс пользователя MS DOS. Приглашение системы. Ввод, запуск и выполнение команд
3. Задача. Файловый менеджер Total Commander
 1. Создайте в своей папке 5 папок одной командой Математика, Информатика, Программирование, История, География (Математика | Информатика | Программирование | История | География)
 2. Создайте одной командой вложенные друг в друга папки Папка1, Папка2, Папка3 (Папка1/Папка2/Папка3).
 3. Создайте в этой же папке 23Pi 5 различных текстовых файлов (Fail1, Fail2, Fail3, fail4, Fail5).
 4. Измените их расширение первого файла на *скрытый*, второго на *системный*, третьего только для чтения, четвертого архивный) .
 5. Изменить время создания файлов на 01.09.2006.
 6. Создать архив из этих файлов с именем «*Фамилия.rar*».
 7. Создайте новую папку «Распакованные» и распакуйте в нее свой архив.
 8. Переименуйте 5 ранее созданных файла в форму «*Задание_N_Фамилия*», где *N* - порядковый номер файла, *Фамилия* – ваша фамилия.
 9. Переименуйте файлы таким образом, чтобы в имени файлов остались первые 9 символов.

10. Создайте комментарий к каждому файлу.
11. Разбейте ваш архив «*Фамилия.tar*» на части по 10 Кб.
12. Найдите все архивные файлы на диске D(Сохраните в шаблоны масок) .
13. Отобразите в правой панели свою папку в виде эскизов.
14. Создать маску поиска для видео файлов, назвать «Поиск видео»(маску сохранить в качестве шаблона).
15. Создать маску поиска для графических файлов, назвать «Поиск рисунков»(маску сохранить в качестве шаблона).
16. Создать 5 текстовых файлов. Переименовать инструментом «Групповое переименование», таким образом, чтобы в названии файлов присутствовало название каталога.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 25

1. Загрузка операционной системы MS-DOS
 2. Настройка файлового менеджера Total Commander
 3. Задача. Файловый менеджер Total Commander.
- В программной оболочке Total Commander выполнить задание

1. Создайте в своем каталоге папку Проба.
2. За одно нажатие клавиши F7, в нем создайте каталог Учеба, с подкаталогами Информатика и Программирование.
3. Скопируйте в каталог Программирование 3 любых файла Блокнот с текстом в нем.
4. Скопируйте данный текст одного из файлов в документ Word
5. Сохраните документ в каталоге Учеба под именем Prog.doc.
6. Переместите файл Prog.doc в каталог Информатика.
7. Переименуйте 3 данных файлы с текстами в имена: Example1, Example2, Example3.
8. В каталоге Программирование создайте каталог Example.
9. Переместите в него файлы из каталога Программирование.
10. Каталог Example переместите в каталог Учеба.
11. Удалите пустой каталог.
12. Создайте маску для выделения только текстовых файлов.
13. Создайте маску для выделения документов Word.
14. Создайте маску для выделения известных Вам архивных файлов.
15. Создайте маску для выделения изображений.
16. Создайте маску для выделения файлов с именем, начинающимся с символов "Pro".

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 26

1. Система прерываний
2. Понятие ресурса. Распределение ресурсов.
3. Задача. Работа с файловым менеджером Far Manager.

1. Уберите с экрана, а затем восстановите последовательно левую и правую панели.
2. Установите для правой панели режим отображения списка каталогов и файлов в полной форме.
3. Последовательно просмотрите оглавления всех каталогов и подкаталогов на диске C:.
4. Выведите на левой панели оглавление своего каталога (папка 23ACU2) и сделайте эту панель активной.
5. Создайте в своем каталоге подкаталоги SUBDIR и SUBWORK
6. Создайте в подкаталоге SUBDIR файл NOTE.TXT с текстом: «Переход на другую панель - TAB»
7. Скопируйте файл NOTE.TXT в каталог SUBWORK.
8. Создайте в SUBDIR файлы FILE1.TXT, FILE2.txt, FILE3.txt с текстом: «Убрать панели – комбинация клавиш », «Оглавление другого диска -Alt +F1 «Печать на принтер- F5, PRN».
9. Скопируйте эти файлы в подкаталог SUBWORK (группой).
10. Переименуйте файлы FILE1.txt в FILE.doc.
11. Переименуйте файлы FILE2.txt и FILE3.txt в соответствующие файлы с расширением doc (группой)
12. Просмотрите NOTE.txt.
13. Отредактируйте NOTE.txt.
14. Перейдите в корневой каталог.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 27

1. Основные задачи ОС.
2. Общесистемные команды MS-DOS
3. Задача. Операционная система MS DOS.

Создать программу из команд MS DOS и отладить ее на компьютере:

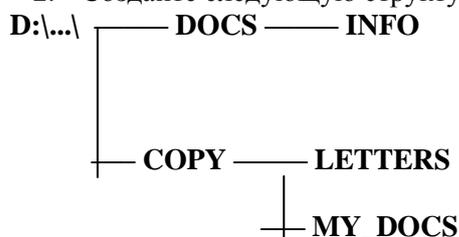
1. Очистить экран монитора
2. На диске C: в корневом каталоге создать каталог «Ekzamen2»
3. Просмотреть оглавление (вывод оглавления).
4. Скопировать один файл из корневого диска в новую папку «Ekzamen2»
5. Перейти в каталог «Ekzamen2»
6. Просмотреть содержимое этой папки.
7. Узнать дату на компьютере.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 28

1. Понятие виртуальной памяти.
2. Работа с архивными файлами в Total Commander
3. Задача. Программная оболочка FAR-manager.

В программной оболочке FAR-manager выполнить задание:

1. Запустите файловый менеджер FAR и разверните его, если необходимо, на весь экран.
2. Создайте следующую структуру каталогов в своей папке:



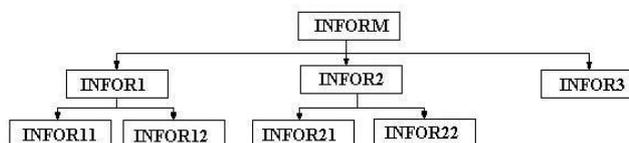
3. Создайте в каталоге INFO текстовые файлы name.txt и group.txt. В первом файле запишите свои имя и фамилию, во втором – название специальности, курс и номер группы
4. Допишите в файл name.txt дату и своего рождения.
5. Используя копирование, создайте в каталоге INFO файл info.txt, объединяющий содержимое файлов name.txt и group.txt, просмотрите получившийся файл.
6. Переименуйте файл info.txt в файл info1.txt.

Используя **Панель информации**, определите, сколько свободного места осталось на диске C:.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 29

1. Прикладное программное обеспечение
2. Организация виртуальной оперативной памяти.
3. Задача. Работа с файловой структурой MS DOS.

В корневом каталоге диска D создать папку «Экзамен», в этой папке создать дерево папок



1. Выведите дерево данного каталога со всеми его ветвями.
2. Очистите экран.
3. Создать текстовый файл **text11.txt** в каталоге **INFOR11**. В текстовом файле написать определение операционной системы MS DOS.
4. Создать текстовый файл. **text12.txt** в каталоге **INFOR12**. В текстовом файле написать определение Операционной системы.
5. Создать текстовый файл. **text21.txt** в каталоге **INFOR21**. В текстовом файле написать определение Программного обеспечения.
6. Создать текстовый файл. **text22.txt** в каталоге **INFOR22**. В текстовом файле написать определение Файла.
7. Скопировать файлы **text11.txt**, **text12.txt** в каталог **INFOR3**.
8. Переместить файлы **text21.txt**, **text22.txt** в каталог **INFOR3**

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 30

1. Понятие программного интерфейса, его назначение.
2. Диспетчер задач в Windows.
3. Задача. Работа с общесистемными командами MS DOS.
 1. Вызовите справку команды data и time.
 2. Посмотрите текущую дату и время компьютера.
 3. Измените приглашение системы на: текущий диск, текущее время, знак равенства
 4. Посмотрите версию DOS компьютера.
 5. Вызовите справку команды prompt.
 6. Очистите экран.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 31

1. Алгоритмы распределения ресурсов на основе очередей.
2. Структура программного обеспечения.
3. Задача. Работа с файловой структурой MS DOS.
 1. Запустить операционную систему MS – DOS.
 2. Создать текстовый файл в корневом каталоге MS-DOS.txt и записать в нем определение операционной системы MS DOS.
 3. Перейти с диска C на диск D.
 4. На диске D создать каталог с вашей фамилией.
 5. В каталоге с вашей фамилией создать подкаталог с вашим именем.
 6. Скопировать с диска файл MS – DOS.txt в каталог:
 - а) с вашей фамилией;
 - б) с вашим именем.
 7. Переименовать файл MS-DOS.txt на диске C в файл с названием samtron.doc.
 8. Вывести на экран содержимое каталога на диске D
 9. Очистите экран

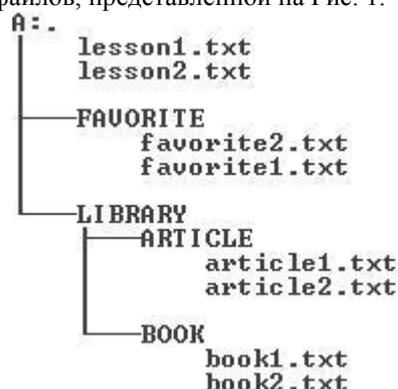
ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 32

1. Механизм обработки прерываний.
2. Классификация операционных систем в зависимости от особенностей областей использования.
3. Программная оболочка Far Manager.
 1. В корневом каталоге диска создать каталог Addons.
 2. В созданном каталоге создать каталоги :
 1. ШКОЛА
 2. ЦВЕТЫ
 3. ПОЭЗИЯ
 4. МУЗЫКА
 3. В созданном вами каталоге создайте подкаталог:
 1. ЛИТЕРАТУРА
 2. РОЗА
 3. ПУШКИН
 4. РОМАНС
 4. В созданном вами подкаталоге создайте файл с указанным названием следующего содержания:
 1. **School** Буквы разные писать
 2. **Rose Миллион**, миллион, миллион алых роз
 3. **Pushkin Я** вас любил: любовь ещё, быть может,
 4. **Lyric** Мир полон звуков, звуки все - мы сами
 5. В каталог скопировать с диска C по 2 файла с расширениями:

I.	ШКОЛА	в каталог I	...txt
II	ЦВЕТЫ	в каталог II	...bat
III	ПОЭЗИЯ	в каталог III	...hlf
IV	МУЗЫКА	в каталог IV	...reg
 6. Скопированные на диск C файлы переименовать соответственно:
 - I. *matemat.txt , inform.txt*
 - II. *pion.bat, astra.bat*
 - III. *blok.hlf, bunin.hlf*
 - IV. *motsart.reg, bach.reg*

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 33

1. Планирование заданий: мультипрограммирование
2. Основные команды операционной системы MS DOS.
3. Задача. Работа с файловой структурой Far Manager.
 - a. Создайте структуру папок файловым менеджером FAR Manager в соответствии со структурой папок и файлов, представленной на Рис. 1.

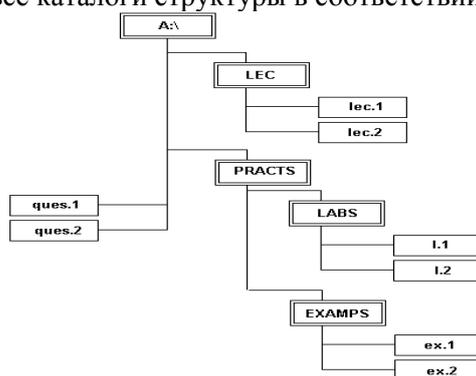


- b. Текстовые файлы lesson1.txt и lesson2.txt создайте путем ввода текста с клавиатуры.
- c. Файлы favorite1.txt и favorite2.txt создайте методом копирования файлов lesson1.txt и lesson2.txt.
- d. Файлы article1.txt и article2.txt создайте методом копирования группы файлов и переименования их после копирования в папку A:\LIBRARY\ARTICLE.
- e. Файлы book1.txt и book2.txt переместите в папку A:\LIBRARY\BOOK из папки A:\FAVORITE, где создайте их методом копирования и вставки текста из других файлов
- f. Определите размер каталога LIBRARY.
- g. Используя выделение в группу по шаблону, определите, сколько файлов с расширением bat записаны в корневом каталоге жесткого диска, какой суммарный размер они имеют. Определите аналогичные характеристики для файлов с расширением txt.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 34

1. Основные понятия планирования процессов.
2. Основными функциями управления ресурсами.
3. Задача. Работа с файловой структурой MS DOS.

Создайте по очереди все каталоги структуры в соответствии с рисунком 1.

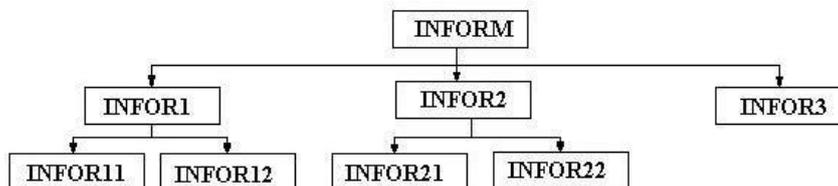


1. В файле lec.1 запишите формат команд DOS и их назначение для работы с файлами.
2. В файле lec.2 запишите формат команд DOS и их назначение для работы с каталогами.
3. В файл **ques.1** запишите формат следующих команд и их назначение: вход в каталог, выход в наддиректорию, просмотр дерева каталогов, очистка экрана.
4. В файл **ques.2**: копирования файлов, смена текущего диска.
5. Файлы **l.1** и **l.2** создайте путем объединения файлов **lec.1** и **ques.1**, **lec.2** и **ques.2**, соответственно.
6. Файлы **ex.1** и **ex.2** создайте путем копирования файлов **lec.1** и **lec.2**.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 35

1. Понятие ресурса. Типы ресурсов.
2. Понятие прерывания. Приоретизация.
3. Задача. Работа с файловой структурой MS DOS.

В корневом каталоге диска D создать папку «Экзамен», в этой папке создать дерево папок



1. Выведите дерево данного каталога со всеми его ветвями.
2. Очистите экран.
3. Создать текстовый файл **text11.txt** в каталоге **INFOR11**. В текстовом файле написать определение операционной системы MS DOS.
4. Создать текстовый файл. **text12.txt** в каталоге **INFOR12**. В текстовом файле написать определение Операционной системы.
5. Создать текстовый файл. **text21.txt** в каталоге **INFOR21**. В текстовом файле написать определение Программного обеспечения.
6. Создать текстовый файл. **text22.txt** в каталоге **INFOR22**. В текстовом файле написать определение Файла.
7. Скопировать файлы **text11.txt**, **text12.txt** в каталог **INFOR3**.
8. Переместить файлы **text21.txt**, **text22.txt** в каталог **INFOR3**

7. Шкала оценки образовательных достижений

Процент результативности (правильных ответов)	Оценка уровня подготовки	
	балл (отметка)	вербальный аналог
90 ÷ 100	5	отлично
80 ÷ 89	4	хорошо
70 ÷ 79	3	удовлетворительно
менее 70	2	неудовлетворительно

8. Перечень материалов, оборудования и информационных источников, используемых в аттестации

1. Карпов В.Е., Коньков К. А. Основы операционных систем. Курс лекций. Учебное пособие. М.,ИНТУИТ.РУ..
2. Карпов В.Е., Коньков К. А. Основы операционных систем. Практикум. Учебное пособие. М.,ИНТУИТ.РУ..
3. Карповский В.А. Методические указания к проведению лабораторных работ по дисциплине «Операционные системы, среды и оболочки». Владимир, 2009
4. Кипрушкин С.А., Соловьев А.В. Основы работы в Linux, Петрозаводск, 2009
5. Лунев А.В. «Лабораторный практикум по курсу «Операционные системы»», Нижний Новгород, 2003
6. Неделько В.М. Системное и прикладное программное обеспечение для ОС Linux. Лабораторный практикум,Новосибирск, 2005.
7. Операционная система (LINUX + KDE) и программное обеспечение для сжатия и архивирования файлов (ARK). <http://www.spohelp.ru/pages/3-uchebnyy-tsentr>
8. Лабораторные работы по курсу "Операционные системы", http://fet.aics.ru/os_lab.html
9. <http://ami.nstu.ru/~vms/method6/Method1.HTM>
10. http://raaar.ru/unix/linux_uch/MT_LINUXa.html
11. <http://asu105.narod.ru/labrab/os/>
12. <http://ipm.kstu.ru/os/lab/>