**Лабораторная работа №3**

**Знакомство с Linux**

В отличие от **Windows *9x***и так далее, в которой ядро системы не отделимо от графического интерфейса пользователя, в операционной системе **Linux** *четко различаются две части*:

* + ***ядро*** ***операционной системы*** — все те программные мо­дули, которые работают с «железом».
  + ***оболочки*** — программы, ко­торые создают интерфейс пользователя.

***Файловая система в Linux***

Чтобы было удобнее пользоваться файлами в различных каталогах, в опера­ционной системе **Linux** введено понятие — корневой каталог, который обо­значается: «**/**». Это та точка, от которой отсчитывается расстояние до любого файла. Сколько бы не было установлено винчестеров в компьютере, корневой каталог всегда один.

Но если корневой каталог **/** является стандартным для всех **Unix**-систем, то на­звания и назначение подкаталогов меняется от дистрибутива к дистрибути­ву. Особенно большая разница между различными ветвями дистрибутивов **Linux**. Далее описано назначение наиболее важных подкаталогов для дистрибутивов **Linux**, разра­ботанных на основе **Red Hat**:

/ — корневой каталог, от которого идет отчет пути ко всем остальным каталогам и файлам.

**/bin** — этот каталог является первым, в котором ищутся файлы команд, введенных с клавиатуры. В нем содержатся основные системные про­граммы (исполняемые файлы, файлы в двоичном коде), такие как общие команды управления файлами, оболочки командной строки и традици­онные **Linux**-программы.

**/boot** — здесь расположены файлы, в которых содержится образ ядра. При запуске компьютера отсюда загружается операционная система.

**/dev** — каталог предназначен для файлов устройств. То есть в каталоге  
находятся специальные файлы, посредством которых операционная сис­тема взаимодействует с системными и физическими устройствами.

**/etc** — в каталоге расположены основные конфигурационные файлы опе­рационной системы, а также подкаталоги с конфигурационными файла­ми прикладных программ.

**/home** — в этом каталоге располагаются все домашние каталоги пользо­вателей, в которые устанавливаются личные конфигурационные файлы и программы.

**/lib** — в каталоге собраны общесистемные библиотеки,

**/lost+found** — служебный каталог, предназначенный, например, для хра­нения фрагментов файлов, которые были обнаружены при проверке фай­ловой системы,

**/mnt** — здесь размещаются каталоги, в которых монтируются внешние файловые системы, в основном это гибкий диск, CD-ROM, различные USB-устройства.

**/root** — домашний каталог администратора (root, суперпользователя).

**/sbin** — каталог для системных файлов программ, применяемых для ад­министрирования системы.

**/tmp** — каталог для временных файлов. Файлы в этом каталоге могут ав­томатически удаляться по прошествии некоторого времени.

**/usr** — в этом каталоге создаются подкаталоги для пользовательских про­грамм, их библиотек и документации.

**/var** — каталог для рабочих и журнальных файлов, которые часто меня­ются, например, здесь можно найти протоколы системных событий, в них создаются временные файлы во время работы принтера и т. д.

В остальном организация файловой структуры традиционна и привычна для пользователей. Например, персональные файлы пользователя располагаются в ка­талоге **home**:

**/home/dima/имя\_файла\_пользователя\_dima**

***Имена файлов в Linux***

Также к существенным отличиям **Linux** от **Windows** относится принцип именования файлов и названия внутренних и внешних устройств.

Имя файла может содержать до 255 любых символов, кроме символа "/" и нулевого кода (**00 Hex**). Расширения имени файла идут через точку в виде трех символов, но на самом деле, они не обязательны.

Кроме того, так как с файлами работает оболочка операционной системы, для которой ряд символов имеет специальное значение, то *не рекоменду­ется* использовать в именах файлов следующие символы:

! @ # $ & ~ % \* ( ) { ] { } < " \ : ; > < ‘ "пробел"

***Принципы именования устройств***

В **Linux** все устройства, с точки зре­ния пользовательских программ, — это файлы. К дисководу, принтеру, сканеру любая прикладная программа обращается как к обычному файлу.

Такой принцип привести к общему знаменателю все особенности конструкции самых разнообразных внешних устройств. Значительно упро­щается процедура интегрирования в операционную систему новых перифе­рийных устройств, которые регулярно выходят из лабораторий и внедряются в производство.

Так как в **Linux** все манипуляции с внешними устройствами сводятся к операци­ям с обычными файлами, то для пользователя архитектура компьютера представ­ляется в виде файловой системы. В одном каталоге находятся устройства ввода-вывода, а в других пользовательские файлы и программы.

***Имена внешних устройств***

Так как любое внешнее устройство отождествляется в **Linux** с файлом, то в имени этого файла имеется цепочка символов, характеризующая его особенности. Причем принято включать в имя устройства название каталога **/dev**. Например, дисковод **А:** называется **/dev/fd0**, дисковод **В:** называется **/dev/fd1**. Но так как в компьютере теперь обычно устанавливается всего один дисковод гибких дисков, то обычно используется универсаль­ное имя: **/dev/floppy**.

Когда модем подключен к **СОМ**-порту компьютера и конфигурация опера­ционной системы настроена автоматически, то программы при обращении к модему используют имя **/dev/modem**. К модему можно обратиться и по имени порта, например, порт **СОМ1** называется **/dev/ttyS0** и т.д.

Имена винчестеров жестко связаны с точка­ми подключения к **IDE**-интерфейсу и типом устройства: первый (*master*) винчестер на первом канале **IDE** называется **/dev/hda**, второй (*slave*) − **/dev/hdb**, первый (*master*) винчестер на втором канале **IDE** называется **/dev/hdc**, второй (*slave*) − **/dev/hdd**.

При ис­пользовании интерфейса **SCSI** первый **SCSI**-диск называется **/dev/sda**, второй − **/dev/sdb**.

В **Linux** разделы винчестера, которые по умолчанию являются в **Windows**, например, дисками **С:** и **D:**, имеют другие имена. Их составляют из имени винчестера и номера раздела на винчестере. Например, если винчестер подключен на первый **IDE**-канал как *Master*, то имена **С:** и **D:** будут звучать так: диск **С** − **/dev/hda1**, диск **D** − **/dev/hda5**. Имя **hdal** говорит о том, что раздел является основным. Таких основных разделов может быть только *четыре*, поэтому возможны только следующие варианты: **/dev/hdal**, **/dev/hda2**, **/dev/hda3** и **/dev/hda4.**

Логический диск **D:** в расширенном разделе получил имя **hda5**. Соответст­венно диск **Е:** будет иметь имя **/dev/hda6**, a **F:** − **/dev/hda7**.

Если винчестер подключен на вторичный канал, то имена разделов будут: диск **С** − **/dev/hdc1**; диск **D** − **/dev/hdc5**.

Если винчестер включен в режиме *Slave*, то имена разделов изменятся: диск **С** − **/dev/hdb1** или **/dev/hdd1**, диск **D** − **/dev/hdb5** или **/dev/hdd5**.

В тех случаях, когда в компьютер установлен винчестер с интерфейсом **serial** **ATA**, имена разделов на нем будут именоваться, например, **/dev/sda1** и **dev/sda2**

***Монтирование файловой системы***

В операционной системе **Linux** есть понятие ***монтирование файловой системы****.* С общей точки зрения любое устрой­ство, дисковый накопитель, прежде чем к нему можно обратиться (читать или записать данные), должно быть смонтировано в какой-либо точке фай­ловой системы. Причем точка монтирования может быть произвольной. Для обеспечения совместимости различных дистрибутивов **Linux** друг с другом и для упрощения взаимодействия пользователей в файловой структуре всегда существует каталог:

**/mnt**

В этом каталоге по традиции монтируются все внешние накопители в виде отдельных каталогов: дисковод гибких дисков − **/mnt/floppy**, привод компакт-дисков − **/mnt/cdrom**.

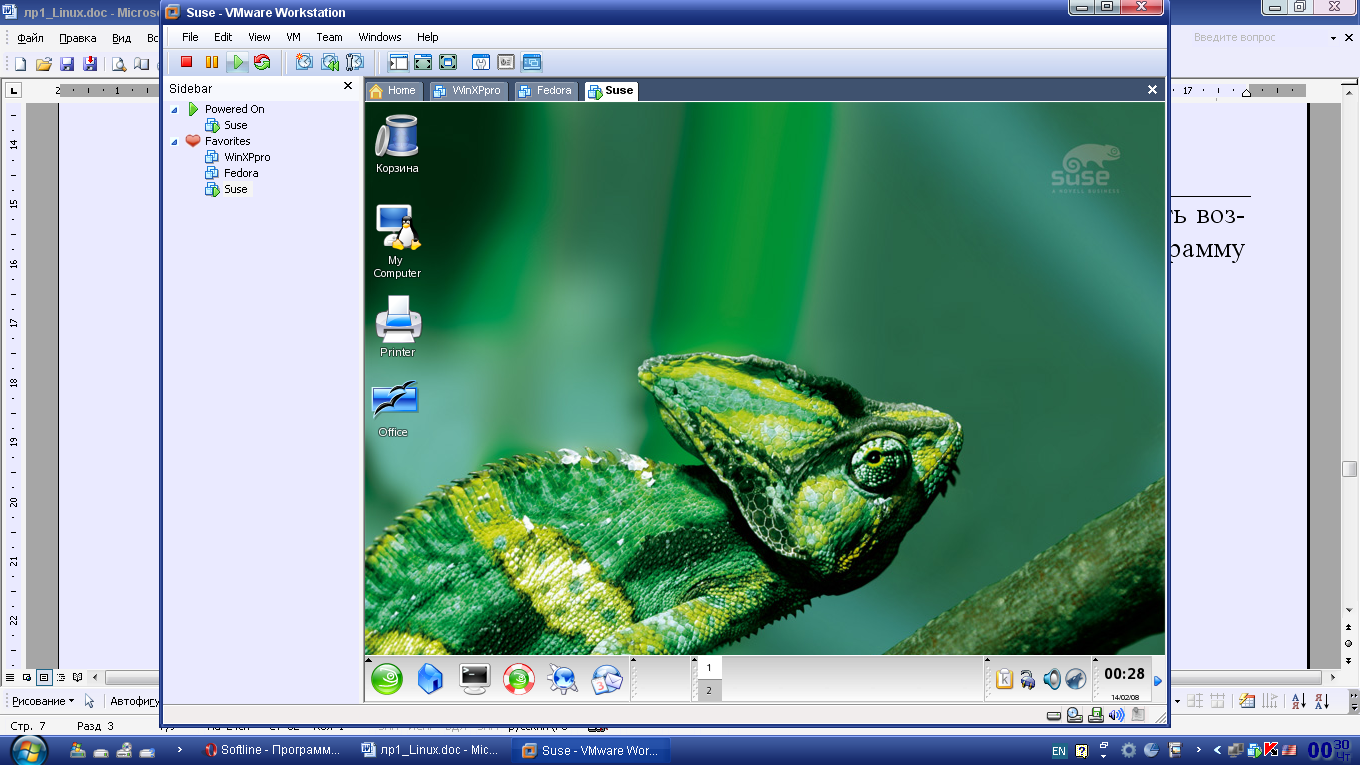
Здесь же стараются монтировать и разделы **Windows**, хотя это необязательно, например, диск **С:** − **/mnt/windows**.

***Виртуальные консоли***

Пусть вы работаете с каким-то документом, а в этот момент надо дать соседу поработать с другой про­граммой. В таком случае достаточно переклю­читься на другую виртуальную консоль и войти в систему под другим име­нем. После этого можно спокойно отойти от компьютера и при этом быть абсолютно спокой­ным за сохранность данных в программе.

Традиционно в **Linux** допускается создание до **63** виртуальных консолей, хотя обычно ис­пользуется **2**−**3**, изредка **5**−**7**. Причем каждая виртуальная консоль может принадлежать одному пользователю, но можно войти в сис­тему и под разными именами. Так как физически к персональному компьютеру подключена одна клавиатура и один монитор, то, чтобы переключаться между виртуальными консолями, используются комбинации клавиш от **Alt+Fl** до **Alt+F12.**

***Виртуальная консоль. Midnight Commander***

Итак, вы запустили операционную систему **Linux**. Перед тем как изучить возможности графической среды, поработаем в консоли (аналог командной строки **Windows**) – нажимаем на пиктограмму  на панели рабочего стола.

В **Linux** также существует аналог файлового менеджера **FAR**, который называется **Midnight Commander**. Для его вызова в консоли наберите

**mc**

Если вы имеете опыт работы с **FAR** в **Windows**, то вы легко сможете работать и с **mc**, поскольку даже основные комбинации клавиш у них совпадают.

***Переключение рабочих столов***

На панели рабочего стола имеются непривычные для пользователя **Windows** кнопки:

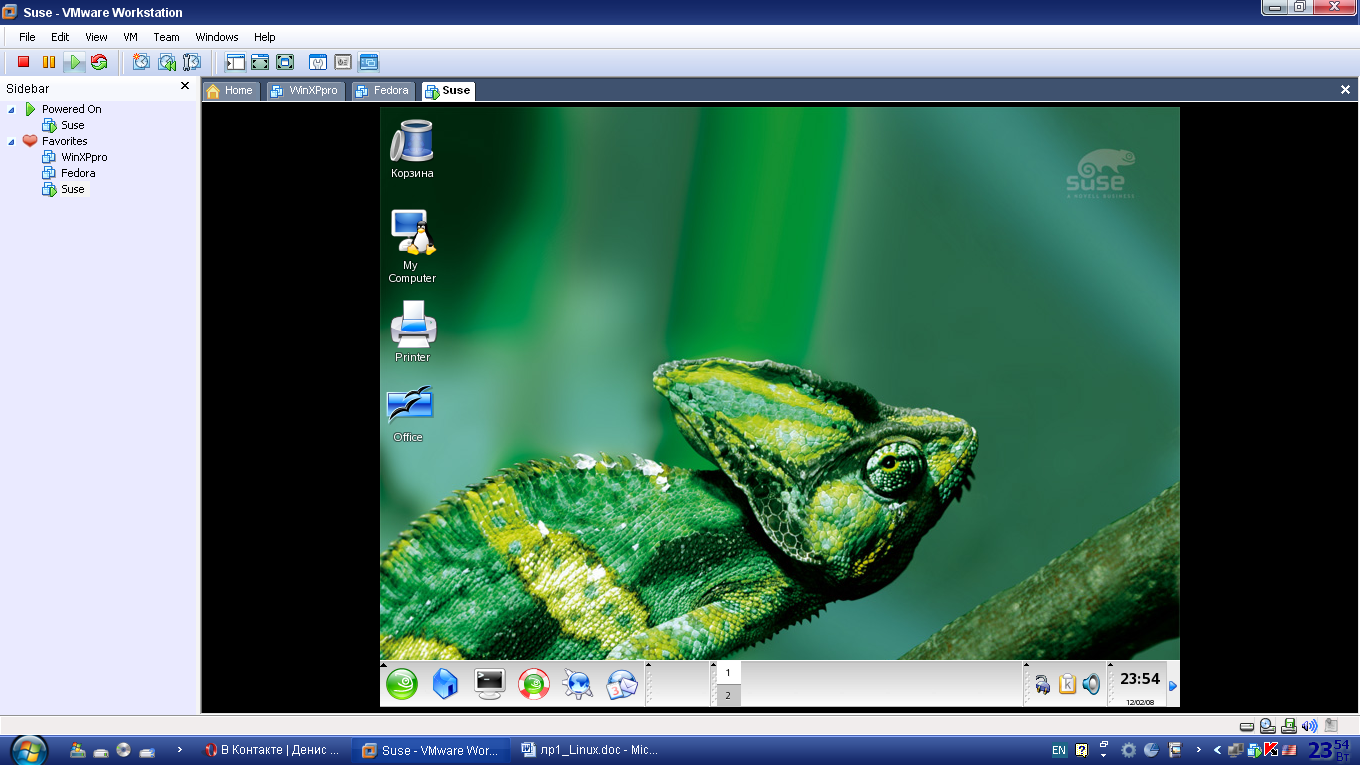


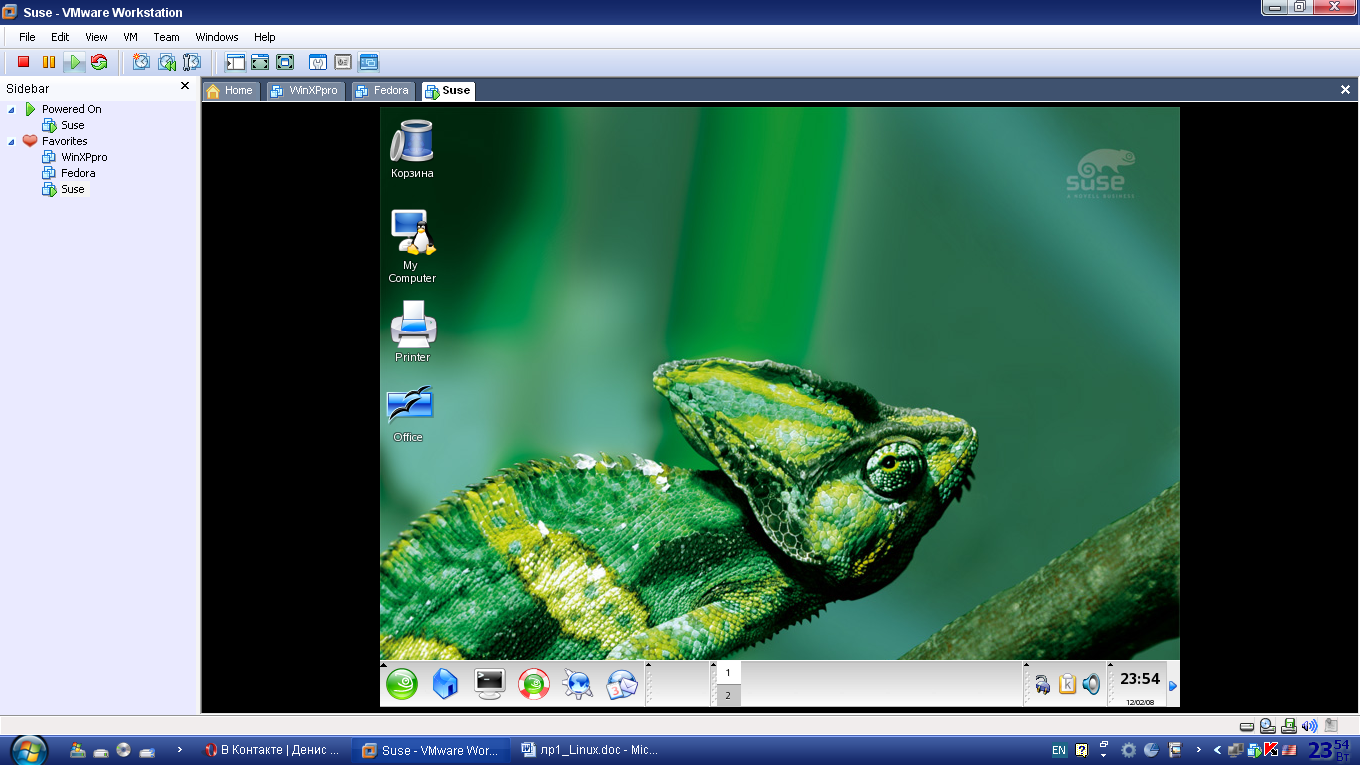
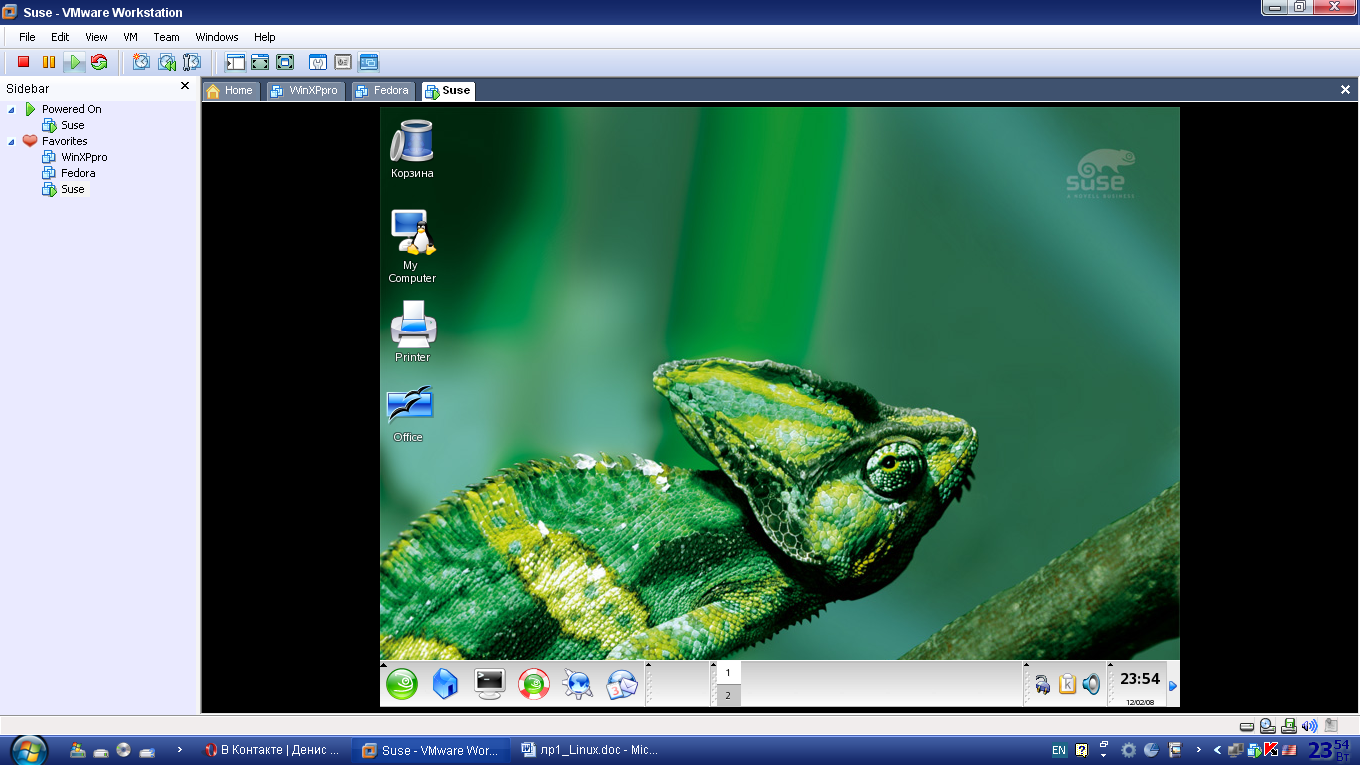
Рис. 2 Панель рабочего стола ВМ **Suse 10.0**

это кнопки переключения рабочих столов, которых по умолчанию два. То есть у пользователя появляется возможность разложить окна запущенных программ по разным экранам. Например, можно работать, в текстовом редакторе на первом рабо­чем столе, на втором — держать активизированные программы для Интернета, а на третьем — игрушку для пятиминутных "перекуров.

**Оболочка bash**

Мы часто говорим, что «пользователь работает с операционной системой», фактически это не верно, поскольку на деле взаимодействие с пользователем организует специальная программа. Существует два вида таких программ − *оболочка*, или **shell**, для работы в командной строке и *графический интерфейс пользователя* **GUI**, организующий взаимодействие с пользователем в графическом режиме.

|  |  |
| --- | --- |
| 256px-Information  256px-Information  256px-Information | Первая оболочка (в первых **UNIX**-системах) имела название **sh**, которое было сокращением от *shell.* Потом были разработаны несколько ее улучшенных вариантов, в частности, ***Bourne shell*** *—* расширенная версия **sh**, написанная Стивом Борном. В рамках проекта **GNU** была создана оболочка **bash**, название которой расшифровывается как *Bourne-again shell,* т. е. «снова оболочка Борна». |

Для начала работы в командной строке выполните команду меню →**Система**→**Терминал**→**Консоль** или нажмите кнопку с изображением терминала на панели **KDE** .

Сама по себе оболочка **bash** не выполняет никаких прикладных задач. Функции ее заключаются в следующем:

* *обеспечивает выполнение всех приложений*: нахождение вызываемых программ, их запуск и организацию ввода/вывода;
* *отвечает за работу с переменными окружения* и выполняет некоторые преобразования (подстановки) аргументов;
* *включает в себя простой язык программирования*, что делает ее мощным инструментом пользователя.

|  |  |
| --- | --- |
| 256px-Information  256px-Information  256px-Information | *Запомните следующие комбинации клавиш*:  **Ctrl+L** − очистить экран и поместить текущую команду в верхней строке экрана;  **Shift+PgUp** и **Shift+PgDown** − позволяют просмотреть несколько страниц экранного вывода;  **Ctrl+C** − прервать выполнение запущенной команды;  **Ctrl+D** − выход из оболочки **bash**. |

***Некоторые команды bash***

**Команда *who.*** Позволяет узнать, кто в данный момент находится в многопользователь­ской системе, какой виртуальный терминал принадлежит вам (так как номер терминала присваивается достаточно произвольно), а также показывает различную статистику по времени работы.

Для получения информации обо всех вошедших в систему пользователях, введите команду:

> **who**

При использовании с командой **who** ключа **-u**, а на более старых системах ключа **-i**, можно получить дополнительно информацию о работе пользова­телей:

> **who -u**

При чтении сообщений команды **who**, следует учитывать, что для пользова­телей, которые работают в текстовом режиме, указывается номер виртуаль­ной консоли, например **tty4**, а для тех, кто использует **X Window** — **:0** (это первый сервер **X Window**).

**Команда *last*** позволяет узнать, какие пользователи и когда входили в сеанс.

**Команда *whoami*** позволяет, узнать имя пользователя, которому принадлежит текущая консоль: **whoami my**

**Команда *su.*** Регистрировать нового пользователя имеет право только администратор. Но переключаться на другую виртуальную консоль и входить в систему, как **root**, каждый раз не только необязательно, но вредно и опасно. Для временного получения прав администратора существует команда **su**. После нажатия клавиши **Enter** будет задан вопрос о пароле администра­тора, аналогично как при входе в систему.

**Подсчет дискового пространства.** Для вывода информации о свободном месте на указанном устройстве ис­пользуется команда:

**df [**-t *тип***] [***путь***]**

В том случае, когда имя устройства не указано, выводится информация о свободном месте на каждой смонтированной файловой системе.

В тех случаях, когда требуется выяснить размеры конкретного каталога, а не общие сведения о разделе, используют команду:

**du [***параметры***] [***файлы***] [***каталоги***]**

Для подсчета объема системных каталогов сначала надо получить права администратора с помощью команды **su**, а по­том ввести команду **du**:

> **du -s** /usr

985M /usr

**Вывод списка файлов.** Для вывода информации о свободном месте на указанном устройстве ис­пользуется команда:

**dir [***параметры***] [***каталог***]**

Правда, команда **dir** введена исключительно ради удобства пользователей MS-DOS. Основная же команда для работы со списками файлов в Unix-системах — это **ls** (сокр. to list — "составлять список"):

**ls [***параметры***][***каталог*/*файл***]**

Если в команде **ls** не указан аргумент, то она выводит содержимое текущего каталога. Если требуется вывести содержимое иного каталога, необходимо указать его имя. Например, команда

**ls** /usr/local

выводит содержимое каталога /usr/local. Также возможен просмотр нескольких каталогов:

**ls** /usr/local /etc/default

выводит содержимое каталога /usr/local и /etc/default.

Для получения подробной информации о файлах используется команда **ls** с ключом **–l**:

**ls –l** /usr/local

выводит подробную информацию о содержимом каталога /usr/local.

Как и в случае команды **dir**, при выводе файлы и каталоги не различаются, поэтому команду **ls** почти всегда используют с рядом ключей. Например, при использовании ключа **-F** команда **ls** помечает в выводимом списке:

* + *ис­полняемые файлы* звездочкой "**\***",
  + *каталоги* - наклонной чертой "**/**",
  + *сим­волические* *ссылки* - символом "**@**" (о них смотрите далее).

При использовании команды **ls** допускается использование маски. Напри­мер, символ "**\***" позволяет следующее:

* **ls** n\* − поиск всех файлов, начинающихся на "**n**"
* **ls** \*n − поиск всех файлов, оканчивающихся на "**n**"
* **ls** n\*n − поиск всех файлов, начинающихся и оканчивающихся на "**n**"

Так как часто пользователя интересует информация обо всех файлах в ката­логе, в том числе включая скрытые файлы, то используется ключ **–а.**

**Команды для работы с каталогами**

Чтобы создать подкаталог, используется команда **mkdir** (*сокр. "make directory" — создать каталог*):

**mkdir** мой\_каталог

С помощью одной команды **mkdir** можно создать целую цепочку каталогов, если использовать ключ **–p**:

**mkdir –p** mydir1/mydir2

Переименовывают каталог, точно так же, как файлы, с помощью команды **mv:**

**mv** мой\_каталог новый\_каталог

Для удаления ненужного каталога служит специальная команда **rmdir**, но если в каталоге уже есть файлы, то их надо удалить заранее с помощью команды **rm**.

**rmdir** каталог

Для передвижения по каталогам используют команду **cd**:

**cd** каталог

Если в качестве имени каталога команде **cd** указать "**..**", то произойдет пе­реход в вышележащий каталог:

**cd ..**

Для перехода в корневой каталог, не используя многократно команду **cd**, можно воспользоваться вариантом:

**cd /**

Из любого каталога можно вернуться в домашний каталог, не используя многократно команду **cd**, с помощью такого варианта:

**cd ~**

Выяснить свое местонахождение в файловой системе позволяет команда **pwd**, которая выводит имя текущего каталога:

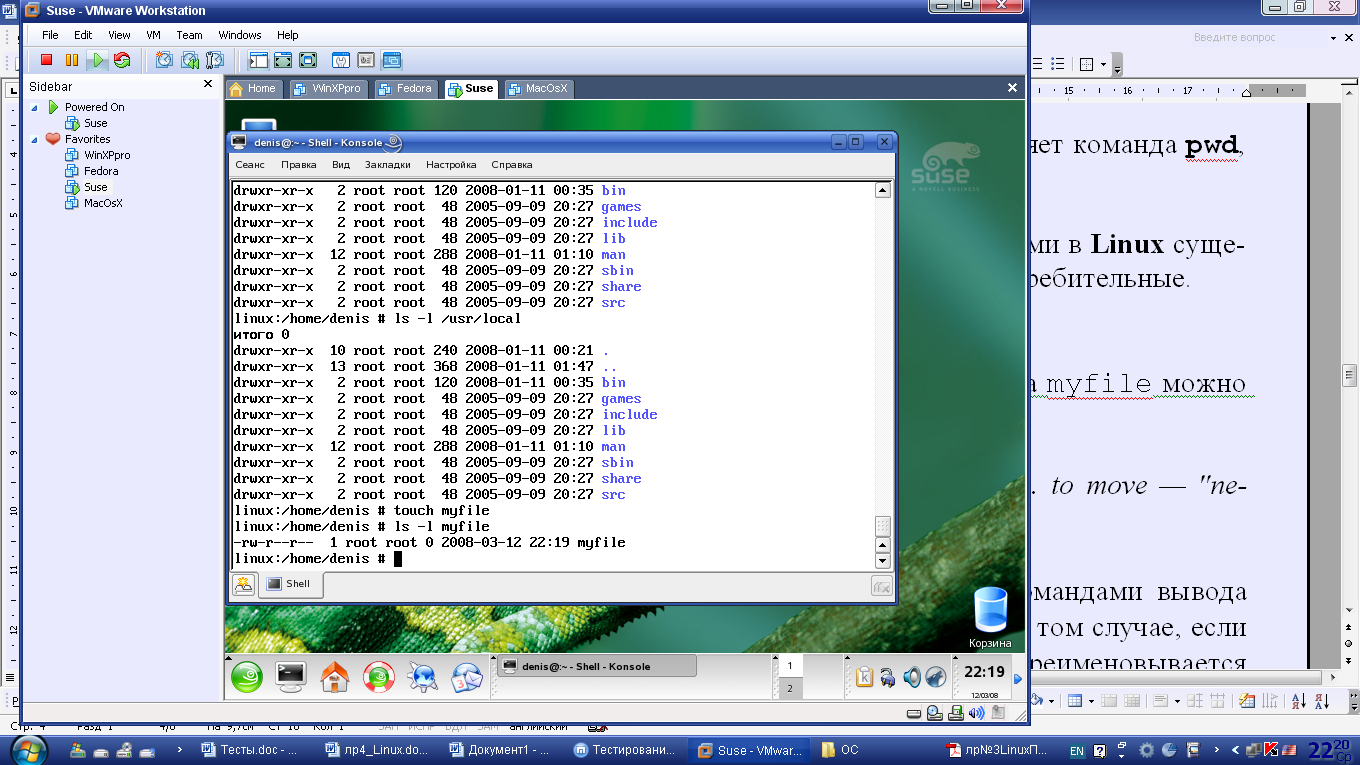
**Команды для работы с файлами.** Для операций с файлами в **Linux** существует множество команд, но мы сей­час рассмотрим наиболее употребительные.

Для создания файла используется команда **touch**, например

**touch** myfile

создает пустой файл myfile. Чтобы убедиться в появлении файла myfile можно использовать команду

> **ls –l** myfile



Время создания

файла

Что означают эти

данные, разберем на следующем занятии

Для удаления файлов используется команда **rm**:

**rm** удаляемый\_файл

Командой **rm** следует пользоваться аккуратно, так как в ряде случаев может не последовать вопроса системы о подтверждении пользователем согласия на удаление.

Для перемещения и переименования файлов применяется команда **mv**, аналогичная команде **rename** из **MS-DOS**:

* + команда **mv** oldName newName переименовывает файл oldNameв newName
  + команда **mv** file1 file2 fileN mydir перемещает указанные файлы file1,file2и fileN в каталог mydir
  + команда **mv** oldDir newDir переименовывает файл oldDirв newDir

При использовании команды **mv** рекомендуется обрамлять ее командами вы­вода списка файлов — **ls**, так как команда **mv** будет выполнена, даже в том случае, если в текущем каталоге уже существует файл с именем, в который переименовывается файл.

Для копирования файлов предназначена команда **cp**, например:

* + команда **cp** myfile yourfile копирует myfile в yourfile
  + команда **cp** file1 file2 fileN mydir копирует указанные файлы file1,file2и fileN в каталог mydir

**Команды для работы с содержимым файлов.** При использовании командной строки очень нужны команды для работы с содержимым файлов. Для этих целей используется множество команд, из которых мы рассмотрим лишь несколько.

Но прежде чем рассматривать процедуры работы с файлами, следует остано­виться на особенностях выполнения команд в **Linux**. В частности, при вводе команд широко используются свойства *стандартного ввода-вывода,* когда вывод одной команды можно передать на исполнение другой. Используя такой прием, можно в одной командной строке объединить любое количе­ство команд посредством четырех символов перенаправления ввода-вывода:

* "**>**" - для перенаправления стандартного вывода;
* "**<**" - для перенаправления стандартного ввода;
* "**>>**" - для добавления стандартного вывода в конец существующего файла;
* "**|**" (символ потока) - для передачи стандартного вывода одной команды  
  на стандартный вход другой.

Самый простой способ создать короткий текстовый файл — это использо­вание команды **echo**, которая предназначена для вывода текста или значе­ния переменных на экран (стандартный вывод):

**echo** "Вывод этого текста на экран"

Чтобы текст был записан в файл, используется перенаправление вывода с помощью символа "**>**":

**echo > myfile** "Этот текст записывается в файл myfile"

Когда файл с таким именем уже существует, то он будет перезаписан. Чтобы новый текст был добавлен в конец файла, используется символ "**>>**":

**echo >>** test "Этот текст добавляется в коней файла test"

Если текст надо распечатать на принтере, то используют символ "**|**":

**echo** "Этот текст отправляется на принтер" **| lр**

Закончим обзор командой **cat**, которой широко пользуются для самых разнообразных целей. Например, для вывода информации из сущест­вующего файла можно ввести следующую команду:

**cat** мой\_файл

С помощью команды **cat** легко создаются файлы, например, вводим коман­ду **cat** и набираем текст на клавиатуре:

**cat >** мой\_файл

Команда **cat** позволяет соединить два файла вместе, например, добавим со­держимое файла **мой\_файл**в файл **новый\_файл**:

**cat** новый\_файл **>>** мой\_файл

**Практические задания**

1. Используйте команду **who –H**. В чем ее особенность?
2. Получите список пользователей, входивших в систему ранее
3. Имя вашего домашнего каталога **~**. Выведете его содержание
4. Выведете содержимое домашнего каталога, включая скрытые файлы
5. Выведите содержимое домашнего и корневого каталогов одной командой
6. Выведите все файлы вашего каталога, начинающиеся на букву D. Объясните показанную информацию
7. Создайте в домашнем каталоге дерево каталогов dir1/dir2/dir3/dir4 одной командой
8. Создайте в каталоге dir2 файл file2
9. Переименуйте dir3 в dirLast
10. Удалите dir4
11. Переместите dirLast в домашний каталог
12. Создайте каталог Toppler, содержащий два файла high11 и low11. Скопируйте оба файла в каталог /tmp
13. Выведите на экран строку «Мне нравится Линукс!» с помощью echo
14. Выведите на экран строку «Мне не нравится Windows!» с помощью cat
15. Создайте новый файл (А) с любым текстом с помощью команды echo. Добавьте в конце этого файла новую запись
16. Убедитесь, что файл создан
17. Создайте новый файл (В) с любым текстом с помощью команды cat. Добавьте в конце этого файла новую запись
18. Выведите его содержимое на экран
19. Объедините эти два файла в один файл (А)
20. Удалите файл (В)