

Лабораторная работа №5

Интерфейс командной строки ОС Unix

ЦЕЛЬ РАБОТЫ: Установка одной из версий ОС *Unix* в виртуальной машине, как второй гостевой ОС. Изучение команд *Unix* - подобных операционных систем.

Создайте текстовый файл, и все исследования по работе заносите в него.

Можно воспользоваться оператором перенаправления ">".

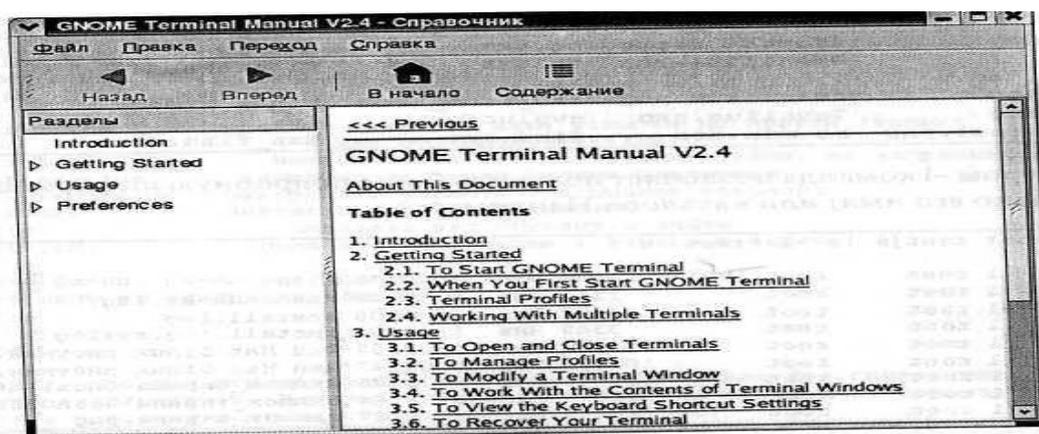
В *Linux* проще всего переключиться в текстовый режим комбинацией клавиш **Alt-Ctrl-F1**. **Alt-Ctrl-F7** - обратное переключение в графический режим. Фактически, комбинации от **Alt-Ctrl-F1** до **Alt-Ctrl-F6** загружают 6 отдельных приглашений на вход в систему в текстовом режиме, тогда как **Alt-Ctrl-F7** - в графическом режиме.

При изучении системы команд *Unix* - подобных ОС необходимо помнить, что многие команды *Windows* и *MS-DOS* совпадают по имени и частично по функциям с командами *Unix*. Это такие команды как: **cd, copy, date, echo, find, for, goto, if, mkdir, more, rmdir, sort**.

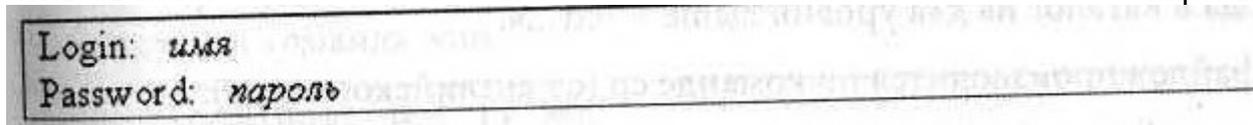
Помощь по начальным сведениям о *Unix* - подобных ОС можно получить, например, на следующих сайтах: linuxrsp.ru; linuxgid.ru;

Приглашение к вводу команды в *Unix* может выглядеть по-разному:
- это приглашение для суперпользователя (root), вошедшего в систему,
\$ или **[имя@localhost имя]\$** - для обычных пользователей.

Помощь по командам *Unix* можно получить, набрав «man» и через пробел - имя команды, например: **Man gnome**. В общем виде вызванный справочник выглядит, как показано ниже.



Вход в систему производится в диалоге, когда система запрашивает имя пользователя и его пароль.



После идентификации пользователя и определения его прав появляется приглашение.

```
[имя@localhost имя]$
```

если имя не совпадает с именем суперпользователя (root), и

```
#
```

Или

```
[root@localhost root]#
```

если в систему вошел суперпользователь.

Выход из системы может производиться по-разному. Для выхода из системы служит команда **logout**, по которой прекращается сеанс работы с данным пользователем, но система не завершает свою работу. Прекратить текущий сеанс работы можно также, нажав одновременно три клавиши **Ctrl+Alt+Backspace**.

Для полного завершения работы нажимается **Ctrl+Alt+Del**. Об окончании работы системы говорит строка:

```
The system is halted.
```

Получение информации о файлах происходит по команде «ls» (от list). Без параметров эта команда выводит на экран информацию о файлах, содержащихся в текущем каталоге.

```
[user@localhost user]$ ls
100307-1.doc          4Nautilus.png    bc.txt             floppy             vi
123.txt              5Nautilus.png    du.txt             Kirichenko
who_and_du.doc       2Nautilus.png    6Nautilus.png     end_find_.doc     Linux 2.doc
3Nautilus.png        7autilus.png     evolution          ls
4Gnome-terminal.png bc man.doc        find.txt           man_find.doc
```

С параметром **-l** команда позволяет получить более подробную информацию о файле (если указано его имя) или каталоге. Например:

```
[root@localhost root]# ls -l /root
итого 548
-rw-r--r--  1 root   root           0 Мар  8 09:30 1.txt
-rw-r--r--  1 root   root        1217 Янв  5 2005 anaconda-ks.cfg
-rw-r--r--  1 root   root       20088 Янв  5 2005 install.log
-rw-r--r--  1 root   root       3348 Янв  5 2005 install.log.syslog
-rw-r--r--  1 root   root      174080 Мар  8 09:57 Red Hat Linux рисунки.doc
-rw-r--r--  1 root   root      160891 Мар  8 09:47 Red Hat Linux рисунки.sxw
-rw-r--r--  1 root   root       54669 Мар  8 09:26 снимок_экрана-Gnome-help.png
-rw-r--r--  1 root   root       16873 Мар  8 09:48 снимок_экрана-OpenOffice.org
-rw-r--r--  1 root   root       97769 Мар  8 09:27 снимок_экрана.png
```

Смена каталога производится по команде **cd имя_каталога**. Эта команда позволяет сделать текущим каталог, имя которого указано в команде. Имя каталога должно содержать путь к нему. Для перехода в каталог более

высокого уровня служит команда «**cd..**». Для перехода в каталог на два уровня выше - «**cd...**».

Копирование файлов производится по команде **cp** (от английского **copy**):
cp [опции] исходный_объект целевой_объект.

Опция **-R** распространяет действие команды на подкаталоги исходного объекта. Справка руководства по этой команде дает следующую информацию.

```
$ man cp
cp <file1> <file2> - copy one to other.
```

Определение наличия свободного места на диске производится по команде **df**. Эта команда позволяет определить:

- общий объем раздела,
- размер занятого информацией дискового пространства,
- объем свободного места на диске,
- сведения о точках монтирования и т.д.

```
[root@localhost root]# df
Файловая система    1К-блоков  Испол  Доступно  Исп%  Подключено к
/dev/hda3            4214068  1685328  2314676   43%    /
/dev/hda2             101107    9275    86611   10%    /boot
none                 256916     0    256916   0%     /dev/shm
```

Удаление файлов производится по команде «**rm**»:

rm [опции] файлы_или_каталоги

Для определения состава пользователей, работающих в данный момент в системе, используется команда **who**.

```
[user@localhost user]$ who
user      :0          Mar 10 18:26
user      pts/0        Mar 10 18:27 (:0.0)
```

В данный момент в системе работает один пользователь - **user**. Если пользователей работает несколько, то кому принадлежит данный терминал можно определить по команде **whoami**.

```
[user@localhost user]$ whoami
user
```

Команды **shell** обычно выполняются группами, например, выполнение команды и проверка, выполнена ли она. Выполните последовательно две команды: войдите, например, в папку **bin** и выведите ее содержимое.

```
[root@localhost root]# cd /bin
[root@localhost bin]# ls -l
итого 4828
-rwxr-xr-x 1 root root 2644 Фев 25 2003 arch
-rwxr-xr-x 1 root root 92444 Фев 6 2003 ash
-rwxr-xr-x 1 root root 492968 Фев 6 2003 ash.static
-rwxr-xr-x 1 root root 10456 Янв 25 2003 aumix-minimal
lrwxrwxrwx 1 root root 4 Янв 5 2005 awk -> gawk
-rwxr-xr-x 1 root root 10848 Фев 18 2003 basename
-rwxr-xr-x 1 root root 626028 Фев 11 2003 bash
lrwxrwxrwx 1 root root 4 Янв 5 2005 bash2 -> bash
```

Файлы и процессы, являются центральными понятиями операционной системы **Unix**. Файловая подсистема управляет файлами, размещает записи файлов в отведенные для них места, управляет свободным пространством, доступом к файлам и поиском данных для пользователей.

Работа с файлами ведется с помощью команд. Команда представляет собой имя исполняемого файла (двоичного или текстового, так называемого **скрипта**, написанного на одном из специальных командных языков) или имя внутренней команды самого процессора. При активизации каждой такой команды операционная система создает *процесс*. Процессы взаимодействуют с подсистемой управления файлами и с аппаратными средствами, используя для этого совокупность специальных команд, таких как **open** (открыть файл на чтение или запись), **close**, **read**, **write**, **stat** (запросить атрибуты файла), **chown** (изменить запись с информацией о владельце файла) и **chmod** (изменить права доступа к файлу).

Подсистема управления процессами ядра ОС отвечает за синхронизацию процессов, их взаимодействие, распределение памяти и планирование выполнения процессов. По характеру выполнения процессы могут быть фоновыми и привилегированными. Любой запускаемый процесс по умолчанию будет выполняться как привилегированный (**foreground**). Это значит, что такой процесс постоянно связан с терминалом ЭВМ и делает невозможным выполнение еще каких-либо действий с системой, пока не завершится.

Фоновый процесс (**background**) после запуска освобождает терминал и позволяет перейти к другой задаче, не дожидаясь его завершения. Фоновая обработка наиболее пригодна для процессов, которые долго выполняются. Программы, выполняющиеся в виде фоновых процессов, называются *демонами* (**daemon**). В любой момент времени в системе существуют десятки процессов, которые были запущены при старте операционной системы, вызваны ядром для обслуживания каких-либо событий, добавлены пользователем при запуске какой-либо задачи.

Обычно большинство процессов находится в состоянии ожидания - сна (**sleep**), не мешая остальным и дожидаясь сигнала для активизации. Кроме того, в системе можно найти процессы, закончившие работу, но еще не получившие разрешения на выгрузку из основной памяти, - эти процессы называются *зомби*. В ядре операционной системы находится таблица процессов, каждая запись которой описывает состояние одного из процессов.

Для того чтобы узнать, какие процессы в данный момент существуют, используется команда **ps**.

```
[root@localhost root]# ps
  PID TTY          TIME CMD
 4194 pts/0    00:00:00 bash
 4244 pts/0    00:00:00 ps
```

Создание пустого файла производится с помощью знака `>`, перенаправляющего стандартный вывод с экрана в файл.

```
[user@localhost user]$ > 1.txt
[user@localhost user]$ ls -l
итого 1304
-rw-rw-r--  1 user  user      31744 Map 10 18:39 100307-1.doc
-rw-rw-r--  1 user  user         0 Map 10 17:23 123.txt
-rw-rw-r--  1 user  user         0 Map 10 18:42 1.txt
```

Отличие файловой системы **Unix** от файловой системы **Windows** заключается в том, что в **Unix** отсутствует такое понятие, как логическое устройство. При указании пути к файлу в **Unix** имя устройства не упоминается. Дерево каталогов **Unix** «растет» из одного корня. Корневой каталог имеет предопределенное имя `/` (слэш). Этот же символ применяется и для разделения подкаталогов. Полный путь к файлу в **Unix** выглядит следующим образом:

/каталог1/каталог2/каталог3/.../файл.

Физически разные компоненты дерева каталогов **Unix** могут размещаться на разных дисках, но логически они принадлежат одной древовидной структуре с одним корневым узлом. Для объединения файловых систем различных устройств в одну структуру используется операция *монтирования*.

Сущность этой операции заключается в том, что каждое физическое устройство можно рассматривать, как свою собственную файловую систему (файловую систему устройства) с корневым каталогом `/`. Если этот раздел диска объявлен в операционной системе, как корневой раздел (`root`), его каталог становится корневым каталогом всей файловой системы (файловой системы ЭВМ). Файловые системы остальных устройств должны быть смонтированы в каталогах файловой системы ЭВМ.

Операция монтирования связывает корневой каталог монтируемого раздела (устройства) с выбранным каталогом файловой системы ЭВМ - точкой монтирования. В результате монтирования корневой каталог файловой системы устройства получает имя каталога, являющегося точкой монтирования, благодаря чему файловая система устройства «привязывается» к файловой системе ЭВМ в точке монтирования.

Большинство команд **Unix** имеют формат: `$ <команда> <ключи> <параметры>`. Параметрами обычно являются *имена файлов*. Все ключи начинаются с дефиса. Например:

`$ ls -l a.out`

Если используется два и более однобуквенных ключа, большинство команд позволяют объединять их. Например, две приведенные команды идентичны:

\$ ls -lg a.out

\$ ls -l -g a.out

Некоторые ключи требуют *наличия* параметра. В этом случае параметр дается после ключа, причем последний нельзя объединять с другим ключом. Есть несколько команд, которые не соответствуют этим правилам. Ниже дается перечень *наиболее* употребительных команд.

Команда	Пояснения
logout	Завершить сеанс работы с UNIX
ls	Дать перечень файлов, находящихся в текущем каталоге. Эквивалент команды DIR, имеющейся во многих операционных системах. Команда ls -l дает более подробную информацию, включая размер файлов, их принадлежность и дату создания
rm файлы	Стереть (удалить) один или несколько файлов (аналог DOS-команды DELETE). Например, команда rm file1 file2 file3 удаляет три файла: file1, file2, file3. Команда rm -i перед удалением каждого файла просит Вас подтвердить свое намерение
mv старое-имя новое-имя	Переименовать (переместить) файл из <i>старое-имя</i> в <i>новое-имя</i>
cp файл1 файл2	Копировать <i>файл1</i> в новый файл с именем <i>файл2</i>

Команда	Пояснения
more файл	Вывести текстовый файл на экран в постраничном режиме. Для вывода следующей страницы нужно нажать клавишу пробел. Полезно запомнить фразу «show me more of this file» («покажи мне этот файл дальше»)
grep образец файл	Показать все строки в файле, отвечающие, образцу. Поиск осуществляется с учетом регистра. Команда grep используется для поиска конкретных текстовых строк в файлах. Например, если <i>phones</i> – перечень номеров телефонов и имен, то команда \$ grep «John Johnson» <i>phones</i> находит в этом списке номер телефона Джона Джонсона. Обратите внимание: образец поиска взят в кавычки
grep -i образец файл	Аналогична предыдущей команде, но регистр не учитывается: прописные и строчные буквы воспринимаются одинаково
pwd	Показать текущий каталог
cd каталог	Изменить текущий каталог
mkdir каталог	Создать новый каталог без файлов с именем каталог
rmdir каталог	Стереть (удалить) каталог с именем каталог. Этот каталог должен быть пуст, т.е. в нем не должно быть файлов
man команда	Вывести на экран справку ОС UNIX по команде команда.
uuencode окончательное-имя <файл>выходной-файл	Создать текстовую версию двоичного файла, годную для пересылки электронной почтой. <i>Окончательное-имя</i> – это имя, которое файл получает после распаковки; <i>файл</i> – имя кодируемого файла; <i>выходной-файл</i> – закодированный файл, который подлежит отправке по электронной почте. Символы > и < обозначают стандартный ввод и стандартный вывод
uudecode закодированный-файл	Команда, обратная команде uuencode
compress файл	Сжать файл так, чтобы он занимал меньше места в памяти. В результате получается двоичный файл с тем же именем, что и исходный, и суффиксом .z . Исходный файл удаляется. Сжатый файл нельзя пересылать по электронной почте, т.к. он двоичный, но его можно преобразовать в текстовый командой uuencode . Подобную же операцию в UNIX выполняет gzip
uncompress файл	Получить исходный файл из сжатого файла. Подобную же операцию в UNIX выполняет gunzip
tar	Эта команда предназначалась для создания архивов магнитных лент (tape archive), но используется и для создания архивов файлов (объединений, включающих несколько файлов). Подобные архивы встречаются в Internet. Структура этой команды довольно необычна. Поэтому вместо ее описания ниже приведены три примера команды tar . В этих примерах файл.tar – архив, созданный командой tar . В первых двух командах работа идет с архивом, полученным извне: \$ tar cf file.tar list \$ tar tf file.tar В последнем случае мы сами создадим архив: \$ tar xf file.tar

Задание1 Составьте аналогичное описание для команд **cat, more, chown, chgrp, who, du.**

Задание2 Определите структуру каталогов на Вашей ЭВМ.

Задание 3. Найдите справочные данные для **Linux**-команд, название которых совпадает с командами **MS-DOS**. Попробуйте их выполнить.

Сформулируйте, чем они отличаются.

Задание 4. Создайте файл и посмотрите его метаданные, продемонстрируйте “права” созданного файла. Какие действия возможны с данным файлом.