

Лабораторная работа № 16

Тема: "Интерфейс командной строки ОС Linux"

Цель:

- ознакомиться с командным интерпретатором;
- изучить базовые команды операционной системы Linux

Порядок выполнения работы:

1. Изучить теоретические сведения
2. Выполнить задания
3. Ответить на контрольные вопросы
4. Оформить отчет

Основные теоретические сведения

Командная строка — это способ организации интерфейса, в котором каждая строка, введенная пользователем — это команда системе, которую та должна выполнить. Термин «командная строка» происходит от того, что команды вводятся обычно в одну строку, которая завершается нажатием клавиши «ввод» (Enter).

Чтобы получить командную строку, пользователь должен войти в систему и запустить программу, которая будет принимать его команды и передавать их на выполнение — командную оболочку (её ещё называют **интерпретатор командной строки**, просто **оболочка**, по-английски «shell»).

Получить командную строку можно многими способами. Самый простой и универсальный — зарегистрироваться на одной из первых шести виртуальных консолей: после входа в систему запустится команда оболочка и появится **приглашение командной строки**. Не выходя из графической среды можно получить командную строку при помощи **любого эмулятора терминала** — они перечислены в главном меню в разделе «Терминалы». Для пользователей графической среды KDE команда строка доступна также по нажатию *F2* (функция «ввести команду»).

Формат команд в ОС LINUX следующий:

имя команды [аргументы] [параметры] [метасимволы]

Имя команды может содержать любое допустимое имя файла;

аргументы - одна или несколько букв со знаком минус (-); параметры - передаваемые значения для обработки; метасимволы интерпретируются как специальные операции. В квадратных скобках указываются необязательные части команд.

По ходу изучения операционной системы UNIX вам часто будет требоваться информация о том, что делает та или иная команда или системный вызов, какие у них параметры и опции, для чего предназначены некоторые системные файлы, и каков их формат и т.д. Для этого существует команда **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 команде **pwd**. Пролистать страницу полученного описания, если оно все не поместилось на экране, можно нажав клавишу <пробел>. Для прокрутки одной строки воспользуйтесь клавишой <Enter>. Вернуться на страницу назад позволит одновременное нажатие клавиш <Ctrl> и . Выйти из режима просмотра информации можно нажатием клавиши <q>.

Для получения точной информации о разбиении на разделы, форме указания номера раздела и дополнительных возможностях утилиты **man** наберите команду **man man**.

Системные команды

Ведите команду **echo**, которая выдает на экран свои аргументы:

echo good morning

На экране появится приветствие "good morning" – аргумент команды echo.

who [am i] - получение информации о работающих пользователях.

В квадратных скобках указываются аргументы команды, которые можно опустить. Ответ представляется в виде таблицы, которая содержит следующую информацию:

- идентификатор пользователя;
- идентификатор терминала;
- дата подключения;
- время подключения.

date - вывод на экран текущей даты и текущего времени.

cal [[месяц]год] - календарь; если календарь не помещается на одном экране, то используется команда **cal год | more** и клавишей пробела производится постраничный вывод информации.

tty - сообщение имени специального файла стандартного вывода, соответствующего терминалу пользователя.

cat <имя файла> - вывод содержимого файла на экран.

Команда **cat > text.1** создает новый файл с именем **text.1**, который можно заполнить символьными строками, вводя их с клавиатуры. Нажатие клавиши **Enter** создает новую строку. Завершение ввода - нажатие **Ctrl - d**.

Команда **cat text.1 > text.2** пересыпает содержимое файла **text.1** в файл **text.2**. Слияние файлов осуществляется командой **cat text.1 text.2 > text.3**.

ls [-alrstu] [имя] - вывод содержимого каталога на экран. Если аргумент не указан, выдается содержимое текущего каталога.

rm <имя файла> - удаление файла (файлов).

Команда **rm text.1 text.2 text.3** удаляет файлы **text.1**, **text.2**, **text.3**. Другие варианты этой команды - **rm text.[123]** или **rm text.[1-3]**.

wc [имя файла] - вывод числа строк, слов и символов в файле.

clear - очистка экрана.

mc - вызов файлового менеджера (программы - оболочки) **Midnight Commander**, аналогичного **Norton Commander**.

sort [-dr] - сортировка входных файлов и вывод результата на экран.

find / _name [имя файла] - Поиск определенного файла на всем диске с ОС Linux

Команды для работы с директорией

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

cd имя_директории,

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

Просмотреть содержимое текущей или любой другой директории можно, воспользовавшись командой **ls** (от **list**). Если ввести ее без параметров, эта команда распечатает вам список файлов, находящихся в текущей директории.

Если же в качестве параметра задать полное или относительное имя директории:

ls имя_директории,

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

ls -a или **ls -a имя_директории**

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

mkdir имя_директории, где **имя_директории** - полное или относительное имя создаваемой директории. У команды **mkdir** имеется набор опций, описание которых можно посмотреть с помощью утилиты **man**.

Для копирования файлов может быть использована команда **cp** (сокращение от **copy**). Команда **cp** умеет копировать не только отдельный файл, но и набор файлов, и даже целиком директорию вместе со всеми входящими в нее поддиректориями (рекурсивное копирование).

Для удаления файлов или директорий применяется команда **rm** (сокращение от *remove*). При удалении несколько регулярных файлов, вид команды **rm** выглядит следующим образом:

rm файл1 файл2 ... файлN, где файл1, файл2, ... файлN - полные или относительные имена регулярных файлов, которые вы хотите удалить. Вместо непосредственно имен файлов могут использоваться их шаблоны.

Если вы хотите удалить одну или несколько директорий вместе с их содержимым (рекурсивное удаление), то к команде добавляется опция **-r**:

rm -r дир1 дир2 ... дирN.

Командой удаления файлов и директорий следует пользоваться с осторожностью. Удаленную информацию восстановить невозможно.

Для переименования файла или его перемещения в другой каталог применяется команда **mv** (сокращение от *move*). Для задания имен перемещаемых файлов в ней тоже можно использовать их шаблоны.

Примеры использования метасимволов:

\$echo *

\$ls F*E

\$ls */memo

\$who | wc - подсчет количества работающих пользователей командой **wc** (*word count* - счет слов);

\$cat text.1 > text.2 - содержимое файла **text.1** пересыпается в файл **text.2**;

\$mail student < file.txt - электронная почта передает файл **file.txt** всем пользователям, перечисленным в командной строке;

\$cat text.1, text.2 - просматриваются файлы **text.1** и **text.2**;

\$cat text.1 >> text.2 - добавление файла **text.1** в конец файла **text.2**;

\$cc primer.c & - трансляция СИ - программы в фоновом режиме. Имя выполняемой программы по умолчанию **a.out**.

\$cc -o primer.o primer.c - трансляция СИ-программы с образованием файла выполняемой программы с именем **primer.o**;

\$rm text.* - удаление всех файлов с именем **text**;

\${cat text.1; cat text.2} | lpr - просмотр файлов **text.1** и **text.2** и вывод их на печать;

Наряду с перенаправлением выходных данных существует способ перенаправить входные данные. Если во время выполнения некоторой команды требуется ввести данные с клавиатуры, можно положить их заранее в файл, а затем перенаправить стандартный ввод этой команды с помощью знака "меньше" – "**<**" и следующего за ним имени файла с входными данными. Другие варианты перенаправления потоков данных можно посмотреть в *UNIX Manual* для командного интерпретатора.

Символы "**<**" и "**>**" можно менять местами в командной строке, но они оба нужны. Команды, которые выглядят похоже, на самом деле могут совершенно по-разному обрабатываться операционной системой.

Команды создания файла.

Чтобы создавать файлы, и просматривать их содержимое на экране можно воспользоваться командой **cat**. Если набрать ее в виде

cat имя_файла

то на экран выведется все его содержимое.

Внимание! Не пытайтесь рассматривать на экране содержимое директорий – все равно не получится! Не пытайтесь просматривать содержимое неизвестных файлов, особенно если вы не знаете, текстовый он или бинарный. Вывод на экран бинарного файла может привести к непредсказуемому поведению вашего терминала.

Если даже ваш файл и текстовый, но большой, то все равно вы увидите только его последнюю страницу. Большой текстовый файл удобнее рассматривать с помощью утилиты

more (описание ее использования вы найдете в UNIX Manual). Команда *cat* интересна с другой точки зрения.

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

cat файл1 файл2 ... файлN

то система выдаст на экран их содержимое в указанном порядке. Вывод команды *cat* можно перенаправить с экрана терминала в какой-нибудь файл, воспользовавшись символом перенаправления выходного потока данных — знаком "больше" — ">".

Команда **cat файл1 файл2 ... файлN > файл_результата**

сольет содержимое всех файлов, чьи имена стоят перед знаком ">", воедино в файл_результата — конкатенирует их (от слова concatenate и произошло ее название).

Большой текстовый файл удобнее рассматривать с помощью утилиты **more**.

Например, команда

\$ cat pat.file выполняет те же действия, что и команда **\$ cat < pat.file**

Однако на самом деле это различные команды. В первом случае **pat.file** — это *аргумент* команды **cat**. С точки зрения Linux аргументы — это переменные, используемые в командной строке. В данном случае получилось так, что аргумент — это файл. Во втором же примере файл **pat.file** играет роль стандартного ввода для команды **cat**.

Перенаправление может использоваться как для стандартного вывода, так и для стандартного ввода. Команда **cat** может использоваться для копирования файлов следующим образом:

\$ cat kevin.report > pat.report

В этом примере стандартным вводом служит файл **kevin.report**, а стандартным выводом — файл **pat.report**. Содержимое файла **kevin.report** не изменяется.

Для того чтобы дописать данные в конец существующего файла, используется специальный символ перенаправления. В следующем примере содержимое файла **kevin.report** дописывается в конец файла **pat.report**:

\$ cat kevin.report » pat,report

touch — команда Unix, предназначенная для установки времени последнего изменения файла или доступа в текущее время. Также используется для создания пустых файлов.

touch [-acfhm] [-t [[CC]YY]MMDDhhmm.[SS]] файл

-a Устанавливает время последнего доступа к файлу. Время последнего изменения не устанавливается, если явно не задан ключ **-m**

-c Указывает утилите не создавать файл, если он не существует, при этом никаких сообщений об ошибке показано не будет.

-f Пытается обновить информацию о времени, даже если права доступа файла не позволяют делать.

-h Указывает утилите не изменять данные о файле, если он задан символьской ссылкой.

-m Устанавливает время последней изменения файла.

-r file Использовать значения времени из файла, заданного аргументом **file**.

-t time Устанавливает время последнего изменения и доступа в соответствии с указанным форматом **time**.

Формат даты, указанный в ключе **-t** задается в соответствии с шаблоном **[[CC]YY]MMDDhhmm[.SS]**:

CC — первые две цифры года (век).

YY — последние две цифры года.

Если параметр CC не задан и значение YY находится в пределах 69 и 99, то тогда CC устанавливается равным 19,

в противном случае используется 20.

MM — двузначный номер месяца.

DD — двузначный номер дня.

hh — значение часов даты.

mm — значение минут даты.

SS — значение секунд даты.

Таблица основных команд Linux и их аналоги в DOS

Команда Linux	Действие	Аналог в DOS
ls, dir	Вывод на экран информации о файлах и каталогах	dir
cd	Смена текущего каталога	cd
pwd	Текущий каталог	-
cp	Копирование файлов в другой каталог и/или с другим именем	copy
mv	Перенос файла и переименование его	ove, rename
rm	Удаление файла	del
mkdir	Создание каталога	mdir, md
rm	Удаление каталога	rm, rmdir
man	Вывод помощи (по командам ОС)	help
more	Вывод содержимого файла на экран с паузами после заполнения каждого экрана	more
cat filename	Вывод на экран содержимого файла nbsp	type
cat > Newfilename	ВПеренаправление содержимого файла в новый файл/создание нового файла с заданным в другом файле содержимым	-
Ctrl+D	Окончание перенаправления	-
grep	Поиск строк в файле, соответствующих запросу	-
chmod	Смена атрибутов файла	attrib
chown	Смена владельца файла	-
df	Вывод информации о наличии свободного места на диске	dir
find	Поиск и вывод на экран имен файлов, соответствующих заданным критериямnbsp	-
file	Вывод информации о файле	-
wc	Подсчет слов или знаков в файле	-

Опции команд

Большинство команд имеет многочисленные опции. Опция вводится с помощью "--" (тире). Например:

dir-l

показывает список файлов в текущем каталоге, но в расширенном формате (по умолчанию формат вывода "короткий"). Если надо ввести более одной опции, то можно их объединить, эти варианты эквивалентны:

dir -l -a или dir -la

Любая из двух команд покажет мне список файлов в текущем каталоге в расширенном формате (опция -l), и включит все файлы в список ,также и скрытые (опция -a). Большинство часто используемых опций обозначены одной буквой, как и в UNIX. Но есть и новый стиль, который выглядит так: dir --help

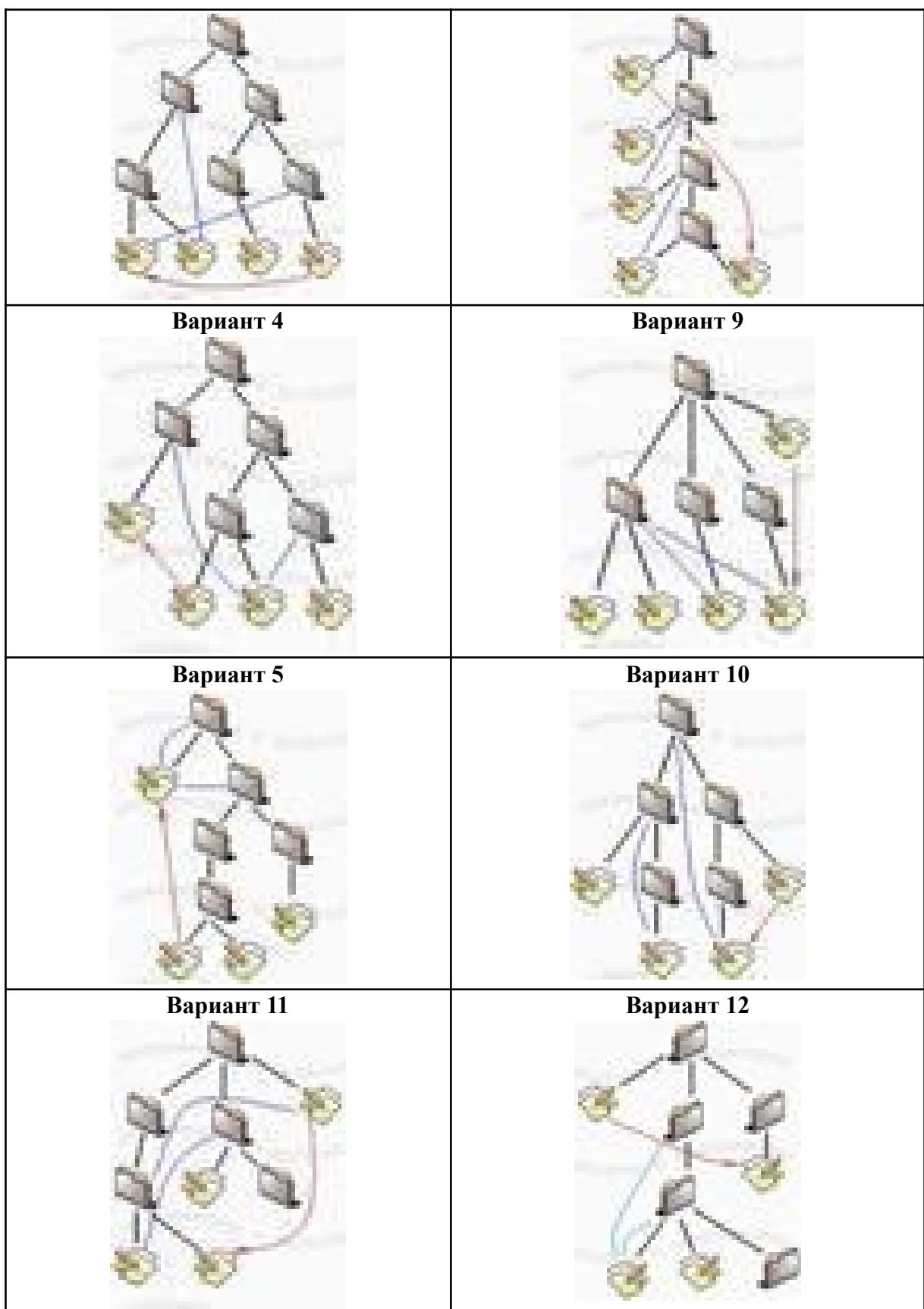
Это - опция длиной более одного символа, и она должна начинаться с двойного тире. Вышеуказанная команда показывает краткую помощь для команды dir, включая список всех опций. Так как их очень много (более одного экрана), лучше ввести эту команду так: dir --help | more

Практическое задание

1. Войти в систему
2. Создать пользователя - со своим именем
3. Назначить пароль и войти под этим пользователем
4. Создайте свою домашнюю структуру каталогов с помощью команды `mkdir`, используя название отделения, своей группы и свою фамилию, например `/home/otdelenie_CT/ЛГРУППЫ...` в соответствии с вариантом, полученным у преподавателя (см. таблицу 1).
- 5. Для всех вариантов выполнить следующие действия:**
- 5.1 В домашней директории создать файл `hw` командой `touch`
- 5.2 Записать в него строку «Hello word! I am»
- 5.3 Вывести содержимое файла `hw` на экран
- 5.4 Создать с помощью команды `cat` новый текстовый файл.
- 5.5 Изучить справку к команде `ls`, выполнить ее с различными ключами. Привести в отчете команду `ls` и все ключи и провести их анализ.
- 5.6 Научиться с помощью команд `cp`, `rm`, `mkdir`, `mv` создавать, копировать, удалять переименовывать файлы, создавать собственные поддиректории просматривать их содержимое и перемещаться по логическому дереву файловой системы.
- 5.8 Посмотреть свой `PATH`, используя команду `echo $PATH`
- 5.9 Изучить системные команды такие как `cal`, `du`, `date`, `df`, `free` и др.

Таблица 1

Вариант 1	Вариант 6
Вариант 2	Вариант 7
Вариант 3	Вариант 8



Контрольные вопросы:

1. С помощью какой команды и как можно переместить файл в другой каталог?
2. Куда вы переходите, выполнив команду cd без параметров?
3. Как посмотреть, какие еще параметры можно задать команде ls?
4. Как осуществить просмотр подкаталогов и их содержимого ?
5. Как осуществить просмотр скрытых файлов в домашнем каталоге?