

Национальный исследовательский университет «МИЭТ»

# РАБОТА В КОМАНДНОМ ИНТЕРПРЕТАТОРЕ BASH В ОС LINUX.

Козлов А.В.

Москва – 2012г.

# Краткое содержание

- ▣ Типы данных в Linux, относительные и абсолютные пути, дерево каталогов.
- ▣ Командный интерпретатор: виды КИ, команды, списки, общие и групповые символы.
- ▣ , процессы, фильтры, сортировка, переменные окружения, псевдонимы;
- ▣ Файловая система: объекты ФС, права доступа на объекты; управление правами доступа;
- ▣ Поиск и сортировка данных: команды `find` и `sort`;
- ▣ Сжатие и архивирование данных: команды `tar`, `compress`, `gzip`, `gunzip`, `zcat`;
- ▣ Системные процессы: родительские и дочерние процессы, управление процессами, приоритетный и фоновый режимы заданий;

# Вход в систему в графическом интерфейсе

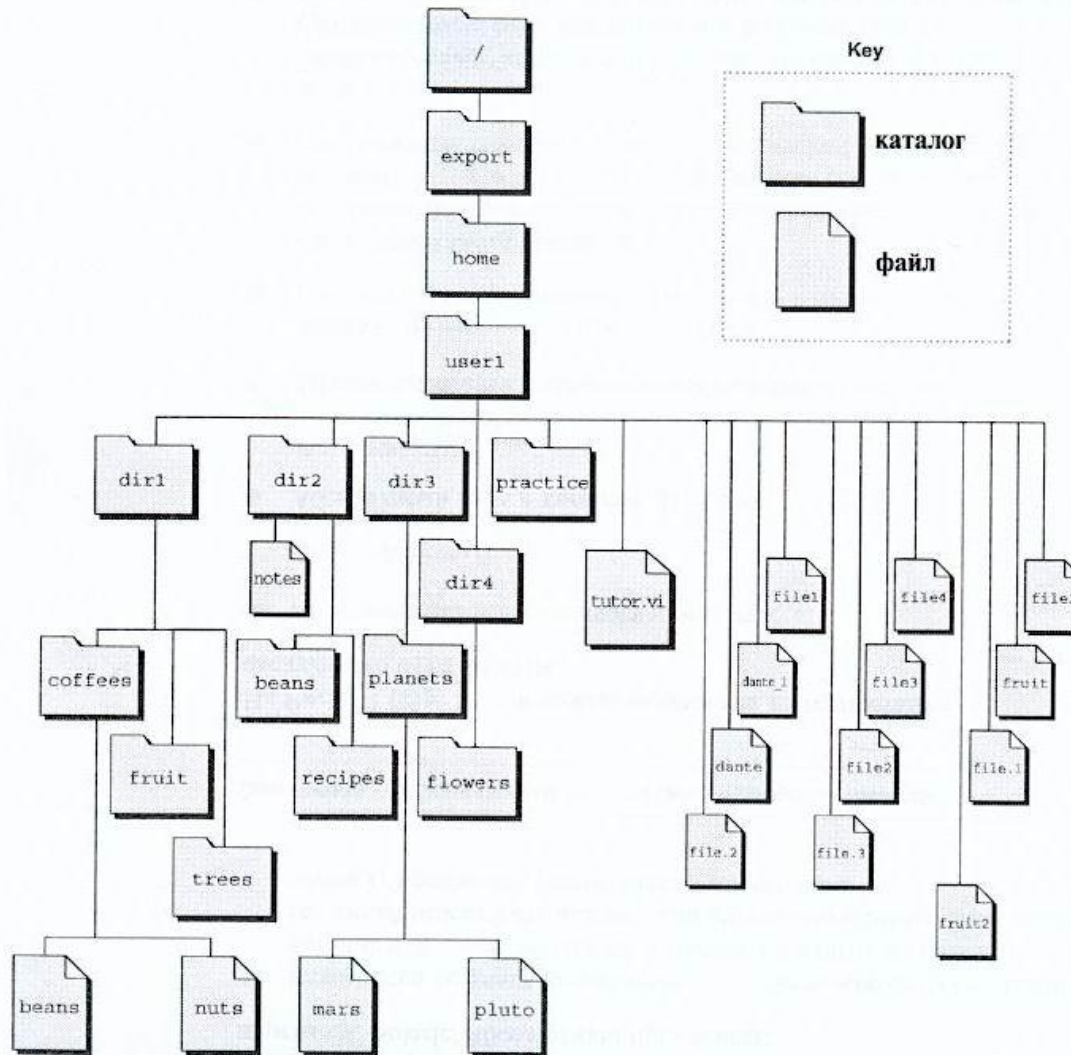


# Типы данных в ОС Linux

- ▣ - файл (поименованная область на диске);
- ▣ - директория;
- ▣ - ссылка;
- ▣ - файл устройства.

Типы файлов: блочные и символьные.

# Дерево каталогов (файловая система)



/  
/boot  
/etc  
/bin  
/mnt  
/sbin  
/usr  
/home  
и т.д.

Рис. 3-1 Дерево каталогов

# Понятия абсолютного и относительного путей

- ▣ Абсолютный путь всегда начинается с корневого раздела, например, доступ к файлу 1:

`/home/pasha/1`

- ▣ Относительный путь – путь к объекту, относительно текущего местоположения, например, если текущим местоположением является каталог `pasha`, то:

1

# Разновидности командных интерпретаторов

- Bourne Shell (BSH) – Стив Боурн (АТ&Т);
- Bourne again Shell (BASH);
- C Shell (CSH) – Билл Джой (Беркли);
- Z Shell (ZSH);
- PDK Shell (PDKSH);
- Korn Shell (KSH) – Дэвид Корн (АТ&Т);
- TC Shell (TCSH).



# Команды командного интерпретатора

- ▣ *Интерфейсом командного интерпретатора является интерфейс командной строки, в которой вводится команда совместно с опциями, параметрами и действиями.*
- ▣ *Опция – это предваряемый дефисом символ, состоящий из одной буквы, который модифицирует работу команды.*
- ▣ *Параметр – это слово, которое вводится в командной строке после команды, а при наличии опций – после опций.*



# Команды командного интерпретатора

- ▣ команда -опции параметры
- ▣ ls - отображение объектов файловой системы, например, `ls -la /home`;
- ▣ `ls -F` Выводит список имен файлов и каталогов, ставя после имени каталога косую черту;
- ▣ `ls -R` Выводит содержимое рабочего каталога со всеми его подкаталогами
- ▣ cp – копирование данных, например,  
`cp -i /home/pasha/1 /home/pash/old`
- ▣ mv – переименование или перемещение объекта файловой системы, например:  
`mv -i 1 2` - переименование объекта 1 в объект 2 в текущей директории

# Команды командного интерпретатора

`mv -i 1 /home/pasha/old` – перемещение объекта 1 в директорию /home/pasha/old

- ▣ `pwd` – отображение абсолютного пути текущего местоположения пользователя, например:
- ▣ `pwd`
- ▣ `/home/pasha` - возврат абсолютного пути командой `pwd`

# Команды командного интерпретатора

- ▣ `rm` – удаление объектов файловой системы, например:

```
rm -i 1 2 3 4 5 6 7
```

- ▣ `rmdir` – удаление пустых каталогов (частный случай команды `rm`)

```
rmdir /home/pasha/old
```

- ▣ `cd` – переход из одного каталога в другой (change directory), например:

```
cd /home
```

# Команды командного интерпретатора

- ▣ `mkdir` - создание новой директории
- `mkdir TR` - создание директории TR
- `ls TR` - проверка на наличие TR
- `TR` - директории TR создана

# Групповые символы командного интерпретатора

Групповые символы командного интерпретатора	Групповые символы командного интерпретатора
*	Устанавливает соответствие с любым набором символов в именах файлов
?	Устанавливает соответствие с любым отдельным символом в именах файлов

Отообразим все объекты в текущей директории

```
[antonk@lab4236sl ~]$ ls *
1  3.txt  BMT_dio.cmd  c.c      te cplot.phy
1.txt 4.txt  BMT_mdr.dat  d.c      tecswb.pref
2  5.txt  BMT_mdr.grd  dessis.par  vnc2swf-0.5.0.tar.gz
2.txt BMT.log a.c          pyvnc2swf-0.8.2.tar.gz
3  BMT_des.cmd b.c        script
DBtest:
genesis_layout_antonk.X-2005.10 inspect_macro.par
inspect_layout_antonk.X-2005.10
Desktop:
TCL_License.terms  bin  tcad
temp:
a.c b.c c.c d.c list
```

Отообразим все объекты в текущей директории, которые начинаются на «BMT»

```
[antonk@lab4236sl ~]$ ls BMT*
BMT.log BMT_des.cmd BMT_dio.cmd BMT_mdr.dat BMT_mdr.grd
```

# Групповые символы командного интерпретатора

Два примера отображения данных об объектах , которые начинаются на «bmt» и далее имеют: один произвольный символ или знак вопроса.

```
[antonk@lab4236s ~]$ ls -l bmt?
```

```
-rw-rw-r-- 1 antonk antonk  0 Sep 21 15:11 bmt1  
-rw-rw-r-- 1 antonk antonk  0 Sep 21 15:11 bmt2  
-rw-rw-r-- 1 antonk antonk  0 Sep 21 15:11 bmt3  
-rw-rw-r-- 1 antonk antonk 42 Sep 21 15:11 bmt?
```

```
[antonk@lab4236s ~]$ ls -l bmt\?
```

```
-rw-rw-r-- 1 antonk antonk 42 Sep 21 15:11 bmt?
```

```
[antonk@lab4236s ~]$
```

# Стандартные потоки Linux

- ▣ *Входной поток данных в ОС Linux называется стандартным вводом, а выходной поток данных – стандартным выводом.*
- ▣ *Существует также отдельный выходной поток данных, зарезервированный исключительно для сообщений об ошибках, называемый стандартным выводом сообщений об ошибках.*



# Символы перенаправления командного интерпретатора в Linux

Символы перенаправления	Результат применения
>	Перенаправляет стандартный вывод в файл или на устройство. При этом, если файл не существует, он создается, если же файл существует, он перезаписывается
>!	Восклицательный знак приводит к перезаписи файла, если он существует. Этот символ отменяет действие опции, которая запрещает перезапись существующих файлов
<	Перенаправляет стандартный ввод из файла или устройства в программу
>>	Перенаправляет стандартный вывод в файл или на устройство, добавляя выводимые данные к концу файла
2>	Перенаправляет стандартный вывод сообщений об ошибках в файл или на устройство
2>>	Перенаправляет и добавляет стандартный вывод сообщений об ошибках в файл или на устройство
2>&1	Перенаправляет стандартный вывод сообщений об ошибках на стандартный ввод
>&	Перенаправляет стандартный вывод сообщений об ошибках в файл или на устройство
&	Перенаправляет по каналу стандартный вывод сообщений об ошибках на вход другой команды

# Пример использования перенаправления в файл

```
[antonk@lab4236sl temp]S ls
```

```
a.c b.c c.c d.c
```

```
[antonk@lab4236sl temp]S ls -l
```

```
total 16
```

```
-rw-rw-r-- 1 antonk antonk 0 Oct 31 19:19 a.c
```

```
-rw-rw-r-- 1 antonk antonk 0 Oct 31 19:19 b.c
```

```
-rw-rw-r-- 1 antonk antonk 0 Oct 31 19:19 c.c
```

```
-rw-rw-r-- 1 antonk antonk 0 Oct 31 19:19 d.c
```

```
[antonk@lab4236sl temp]S ls > list
```

```
[antonk@lab4236sl temp]S cat list
```

```
b.c
```

```
d.c
```

```
list
```

# Переменные окружения и псевдонимы

*Переменной окружения* называется имя, которое ссылается на определенную область памяти, где временно хранятся данные.

*Псевдоним* – индификатор (обычно с коротким именем), с помощью которого удобно обращаться к часто применяемым командам.

```
LANG=ru_RU.UTF-S
LM_LICENSE_FILE=7326@82.179.185.68
ISEDDB=/home/antonk/DBtest/
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
HOME=/home/antonk
HISTSIZE=2
LOGNAME=antonk
CVS_RSH=ssh
QTDIR=/usr/lib64/qt-3.3/lib
SSH_CONNECTION=82.179.185.70 1209 82.179.185.125 22 LESSO-
PEN=/usr/bin/lesspipe.sh %s
ISEROT=/syn/SYNOPSYS/TCAD61
STROOT_LIB=/root/STDB
STDB=/root/STDB
G_BROKEN_FILENAMES=1
```

```
[antonk@lab4236s ~]$ cat > rrr
```

```
1223333
```

```
[antonk@lab4236s ~]$ cat rrr
```

```
1223333
```

```
[antonk@lab4236s ~]$ set -o noclobber
```

```
[antonk@lab4236s ~]$ ls -ld /home/antonk > rrr
```

```
bash: rrr: cannot overwrite existing file
```

```
[antonk@lab4236s ~]$ set +o noclobber
```

```
[antonk@lab4236s ~]$ ls -ld /home/antonk > rrr
```

```
[antonk@lab4236s ~]$ cat rrr
```

```
drwx----- 27 antonk antonk 4096 Sep 22 11:53 /home/antonk
```

Назначение новых переменных: команда **set**

Перезапись файлов, защищенных от записи, может быть осуществлена путем изменения переменной **noclobber**

# Переменные окружения и псевданимы

Вывод информации о системных переменных

```
[antonk@lab4236s ~]$ set -o
```

```
allexport    off
braceexpand  on
errexit      off
errtrace     off
functrace    off
hashall      on
histexpand   on
ignoreeof    off
interactive-comments on
keyword      off
monitor      on
noclobber    off
```

Назначение новой переменной и проверка действия в интерпретаторе BASH

```
[antonk@lab4236s ~]$ MY_VAR=5
[antonk@lab4236s ~]$ set | grep MY_VAR
MY_VAR=5
[antonk@lab4236s ~]$ echo $MY_VAR
5
[antonk@lab4236s ~]$
```

Назначение псевданима lh

```
[antonk@lab4236s ~]$ alias lh='ls -ld /home/antonk'
[antonk@lab4236s ~]$ lh
drwx----- 27 antonk antonk 4096 Sep 22 14:22 /home/antonk
[antonk@lab4236s ~]$
```



# Программный канал. Конвейерная передача.

Программный канал | (pipe, символ вертикальной черты), помещается между двумя командами и связывает стандартный поток вывода предыдущей команды со стандартным потоком ввода следующей команды.

```
[antonk@lab4236s ~]$ ls -la /home/antonk | head -5 > yyyu
[antonk@lab4236s ~]$ cat yyyu

total 1713212
-rw-rw-r-- 1 antonk antonk 109 Sep 22 13:45 !
drwx-----27 antonk antonk 4096 Sep 22 14:22 ..
drwxr-xr-x 34 root root 4096 Sep 18 09:23 ..
-rw----- 1 antonk antonk 491 Sep 18 09:29 .ICEauthority

[antonk@lab4236s ~]$
```

```
[antonk@lab4236s ~]$ ps -ef | grep antonk | tail -4 | sort -nr -k 2
antonk 18889 14588 0 14:34 pts/2 00:00:00 sort -nr -k 2
antonk 18888 14588 0 14:34 pts/2 00:00:00 tail -4
antonk 18887 14588 0 14:34 pts/2 00:00:00 grep antonk
antonk 18886 14588 0 14:34 pts/2 00:00:00 ps -ef

[antonk@lab4236s ~]$
```

- 1) Расширенный вывод сведений (включая скрытые объекты) для /home/antonk ;
  - 2) Передача данных на head (чтение 5-ти верхних строк) ;
  - 3) Перенаправление в новый файл уууу .
- 
- 1) Отображение системных процессов в форматированном выводе;
  - 2) Передача на tail (отображение последних 4-х строк списка) ;
  - 3) Числовая обратная сортировка по 2-му столбцу (по PID) .

# Программный канал. Конвейерная передача.

- 1) Поиск в домашней директории пользователя всех файлов, имя которых начинается с «r»
- 2) Поиск в домашней директории пользователя всех файлов, имя которых начинается с «r», передача на фильтр `grep`, чтобы убрать все строки, содержащие 64.

```
[antonk@lab4236s ~]$ find ~ -name 'r*' -type f
```

```
/home/antonk/222/lab9/tt/r
```

```
/home/antonk/.gstreamer-0.10/registry.x86_64.xml
```

```
/home/antonk/.gstreamer-0.8/registry-x86_64.xml /home/antonk/r
```

```
[antonk@lab4236s ~]$ find ~ -name 'r*' -type f | grep -v 64
```

```
/home/antonk/222/lab9/tt/r /home/antonk/r
```

```
[antonk@lab4236s ~]$
```

# Уровни безопасности

- ▣ Первый уровень безопасности связан с аккаунтом пользователя, т.е. с возможностью регистрации пользователя в операционной системе.
- ▣ Второй уровень безопасности связан с доступом пользователя к данным (объектам Linux – файлам и каталогам) операционной системы.



# Расширенный вывод сведений об объекте файловой системе

Рассмотрим полную информацию о файле по расширенному выводу команды ls с опцией -l слева направо.



# Категории пользователей и права доступа на объект

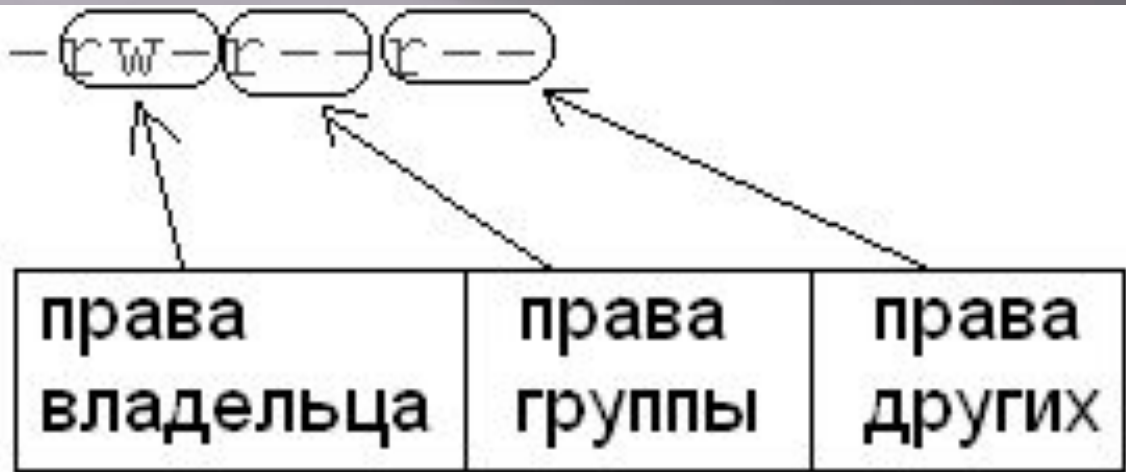
Категории пользователей:

- Пользователь (user, u);
- Группа пользователей (group, g);
- Другие (other, o).

Права доступа на объект:

- Чтение (read, r);
- Создание или модификация (write, w);
- Исполнение (executive, x).

# Права доступа на объекты файловой системы



r      w      x  
↑      ↑      ↑  
чтение запись выполнение

# Назначение прав доступа на объект с использованием «восьмеричного вида»

Восьмеричный вид	Двоичный вид	Символьный вид	Права доступа, разрешено
0	000	---	Нет никаких прав
1	001	--x	Только выполнение
2	010	-w-	Только запись
3	011	-wx	Только запись и выполнение
4	100	r--	Только чтение
5	101	r-x	Только чтение и выполнение
6	110	rw-	Только чтение и запись
7	111	rxw	Полные права

# Назначение прав доступа на объект с использованием «восьмеричного вида»

```
[antonk@fargo laba2]$ ls -l laba2.cmd
-rw-r--r--  1 antonk  user          1810 Apr 22 13:37 laba2.cmd
[antonk@fargo laba2]$
```

```
[antonk@fargo laba2]$ chmod 760 laba2.cmd
[antonk@fargo laba2]$ ls -l laba2.cmd
-rwxrw----  1 antonk  user          1810 Apr 22 13:37 laba2.cmd
[antonk@fargo laba2]$
```

```
[antonk@fargo laba2]$ cd ..
[antonk@fargo 111]$ ls -l
total 4
drwxr-xr-x  2 antonk  user          4096 Jul 13 18:00 laba2
[antonk@fargo 111]$ chmod -R 760 laba2/
[antonk@fargo 111]$ ls -l
total 4
drwxrw----  2 antonk  user          4096 Jul 13 18:00 laba2
[antonk@fargo 111]$ cd laba2/
[antonk@fargo laba2]$ ls -l
total 28
-rwxrw----  1 antonk  user          1079 Apr 22 13:37 channel1.plx
-rwxrw----  1 antonk  user          1175 Apr 22 13:37 gauss2.plx
-rwxrw----  1 antonk  user          1810 Apr 22 13:37 laba2.cmd
-rwxrw----  1 antonk  user         12352 Apr 22 13:37 laba2.log
[antonk@fargo laba2]$
```

# Назначение прав доступа на объект с использованием «СИМВОЛЬНОГО ВИДА»

```
[antonk@fargo laba2]$ ls -l
total 28
-rwxrw---- 1 antonk user 1079 Apr 22 13:37 channel1.plx
-rwxrw---- 1 antonk user 1175 Apr 22 13:37 gauss2.plx
-rwxrw---- 1 antonk user 1810 Apr 22 13:37 laba2.cmd
-rwxrw---- 1 antonk user 12352 Apr 22 13:37 laba2.log
[antonk@fargo laba2]$ chmod u-x laba2.log
[antonk@fargo laba2]$ ls -l
total 28
-rwxrw---- 1 antonk user 1079 Apr 22 13:37 channel1.plx
-rwxrw---- 1 antonk user 1175 Apr 22 13:37 gauss2.plx
-rwxrw---- 1 antonk user 1810 Apr 22 13:37 laba2.cmd
-rw-rw---- 1 antonk user 12352 Apr 22 13:37 laba2.log
[antonk@fargo laba2]$
```

Из примера видно, что атрибуты файла laba2.log с rwxrw---- изменились на rw-rw---- .



# Смена пользователя и группы

```
[root@fargo laba2]# ls -l
total 28
-rwxrw---- 1 antonk user 1079 Apr 22 13:37 channel1.plx
-rwxrw---- 1 antonk user 1175 Apr 22 13:37 gauss2.plx
-rwxrw---- 1 antonk user 1810 Apr 22 13:37 laba2.cmd
-rw-rw---- 1 antonk user 12352 Apr 22 13:37 laba2.log
[root@fargo laba2]# chown kras laba2.cmd
[root@fargo laba2]# ls -l
total 28
-rwxrw---- 1 antonk user 1079 Apr 22 13:37 channel1.plx
-rwxrw---- 1 antonk user 1175 Apr 22 13:37 gauss2.plx
-rwxrw---- 1 kras user 1810 Apr 22 13:37 laba2.cmd
-rw-rw---- 1 antonk user 12352 Apr 22 13:37 laba2.log
[root@fargo laba2]#
```

Изменение администратором группы пользователей для файла laba2.cmd

`chgrp kras laba2.cmd`



# Поиск файлов `find`

`find` путь выражение(я) действие(я)

Таблица 6 – Выражения команды `find`.

Выражение	Описание
<code>-name</code> <u>имя файла</u>	Поиск файлов по имени. Могут использоваться метасимволы <code>^</code> , <code>*</code> , <code>?</code> , <code>[ ]</code> .
<code>-size</code> [ <code>+</code>   <code>-</code> ] <code>n</code>	Поиск объектов, размер которых меньше чем <code>-n</code> и больше чем <code>+n</code> или равны <code>n</code> . <code>n</code> задается в 512-байтных блоках.
<code>-atime</code> [ <code>+</code>   <code>-</code> ] <code>n</code>	Поиск файлов, к которым производился доступ более чем <code>+n</code> дней назад, менее чем <code>-n</code> дней назад или <code>n</code> дней.
<code>-mtime</code> [ <code>+</code>   <code>-</code> ] <code>n</code>	Поиск файлов, которые были изменены более чем <code>+n</code> дней назад, менее чем <code>-n</code> дней или ровно <code>n</code> дней.
<code>-user</code> <u>loginID</u>	Поиск файлов, которые принадлежат пользователю <u>loginID</u> .
<code>-type</code>	Поиск по типу объекта, например: <code>f</code> (файл) или <code>d</code> (директория).
<code>-perm</code>	Поиск файлов, которые имеют соответствующие права доступа.

# Действия с командой find

Таблица 7 – Действия с командой find.

Действие	Описание
-exec команда { } \;	Автоматическое выполнение указанной команды на каждый найденный файл. Перед знаком обратной черты обязательно должен быть указан пробел.
-ok команда { } \;	Ручное выполнение указанной команды (интерактивная форма) на каждый найденный файл. Перед знаком обратной черты обязательно должен быть указан пробел.
-print	Дает команду выводить каждый найденный файл на экран. Команда по умолчанию.

# Пример с find

Задача: найти в домашней директории пользователя файлы, начинающиеся на «bmt»

```
[antonk@lab4236s ~]$ find ~ -name 'bmt*' -type f
/home/antonk/bmt_link
/home/antonk/BMT/bmt2_des.log
/home/antonk/BMT/bmt2_des.pit
/home/antonk/bmt?
/home/antonk/bmt1
/home/antonk/bmt2
/home/antonk/bmt3
```

```
[antonk@lab4236s ~]$ find ~ -name 'bmt*' -type f -print
/home/antonk/bmt_link
/home/antonk/BMT/bmt2_des.log
/home/antonk/BMT/bmt2_des.plt ...
```

```
[antonk@lab4236s ~]$ find . -name 'bmt*' -type f -maxdepth 1
./bmt_link
./bmt?
./bmt1
./bmt2
./bmt3
```

```
[antonk@lab4236s ~]$ find ~ -name copy -type d -print
/home/antonk/copy
```

# Пример с find

```
[antonk@lab4236s ~]$ touch in ip it ir ikt
```

```
[antonk@lab4236s ~]$ ls -li*
```

```
-rw-rw-r-- 1 antonk antonk 0 Oct 14 12:54 ikt
-rw-rw-r-- 1 antonk antonk 0 Oct 14 12:54 in
-rw-rw-r-- 1 antonk antonk 0 Oct 14 12:54 ip
-rw-rw-r-- 1 antonk antonk 0 Oct 14 12:54 ir
-rw-rw-r-- 1 antonk antonk 0 Oct 14 12:54 it
```

```
[antonk@lab4236s ~]$ find ~ -name 'i*' -exec rm {} \;
```

```
[antonk@lab4236s ~]$ ls -li*
```

```
ls: i*: No such file or directory
```

```
[antonk@lab4236s ~]$ mkdir WWW
```

```
[antonk@lab4236s ~]$ find ~ -name 'bmt*' -ok cp {} ~/WWW\;
```

```
< cp ... /home/antonk/bmt_link > ? y
< cp ... /home/antonk/BMT/bmt2_des.log > ? y
< cp ... /home/antonk/BMT/bmt2_des.plt > ? y
< cp ... /home/antonk/bmt? > ? y
< cp ... /home/antonk/bmt1 > ? y
< cp ... /home/antonk/bmt2 > ? y
< cp ... /home/antonk/bmt3 > ? y
< cp ... /home/antonk/WWW/bmt_link > ? y
```

```
[antonk@lab4236s ~]$ touch in ip it ir ikt
```

```
[antonk@lab4236s ~]$ ls -li*
```

```
-rw-rw-r-- 1 antonk antonk 0 Oct 14 13:03 ikt
-rw-rw-r-- 1 antonk antonk 0 Oct 14 13:03 in
-rw-rw-r-- 1 antonk antonk 0 Oct 14 13:03 ip
-rw-rw-r-- 1 antonk antonk 0 Oct 14 13:03 ir
-rw-rw-r-- 1 antonk antonk 0 Oct 14 13:03 it
```

```
[antonk@lab4236s ~]$ find ~ -name 'i*' -ok rm {} \;
```

```
< rm ... /home/antonk/ir > ? y
< rm ... /home/antonk/ip > ? n
< rm ... /home/antonk/it > ? y
< rm ... /home/antonk/ikt > ? y
< rm ... /home/antonk/in > ? y
```

```
[antonk@lab4236s ~]$ ls -li*
```

```
-rw-rw-r-- 1 antonk antonk 0 Oct 14 13:03 ip
```

# Команда `grep`

- ▣ `grep -опции шаблон имя_файлов`
- ▣ `команда | grep -опции шаблон`
  
- ▣ Например,
- ▣ `grep -n root passwd group`                   или
- ▣ `ps -ef | grep -n root`



# Опции команды `grep`

Таблица 1 – Параметры команды `grep`

Параметр	Описание
<code>-i</code>	Задаёт режим игнорирования регистра, символы верхнего и нижнего регистра считаются идентичными.
<code>-l</code>	Перечисляет только имена файлов, которые содержат найденные строки.
<code>-n</code>	Задаёт простановку относительного номера к каждой строке в файле.
<code>-v</code>	Меняет условие поиска на обратные для выдачи только тех строк, которые не соответствуют шаблону.
<code>-c</code>	Задаёт вывод только количества строк, которые содержат шаблон.
<code>-w</code>	Задаёт поиск выражения как слова.

# Метасимволы в регулярных выражениях

Метасимвол	Значение	Пример	Результат
^	Знак начала строки	'^шаблон'	Выбор всех строк, которые начинаются с «шаблон»
\$	Знак конца строки	'шаблон\$'	Выбор всех строк, которые заканчиваются на «шаблон»
.	Любой один символ	' <u>ш</u> ... <u>н</u> '	Выбор строк, которые содержат символ «ш» и через четыре символа символ «н»
[ ]	Соответствует одной позиции в шаблоне	' <u>[Шш]</u> аблон'	Выбор строк, которые содержат «Шаблон» и «шаблон»



# Метасимволы в регулярных выражениях

[ <u>^</u> ]	Соответствует одному символу не в диапазоне поиска	‘ <u>[^a-p]</u> аблон’	Выбор строк, которые не содержат символы в диапазоне от «а» до «р» и находящихся перед « <u>аблон</u> »
*	Соответствует любому количеству символов	‘ <u>[a-я]</u> *’	Выбор алфавитных символов нижнего регистра

# Примеры с grep

```
[antonk@lab4236s ~]$ grep root/etc/group/etc passwd
```

```
/etc/group:root:x:0:root
/etc/group:bin:x:1:root,bin,daemon
/etc/group:daemon:x:2:root,bin,daemon
/etc/group:sys:x:3:root,bin,adm
/etc/group:adm:x:4:root,adm,daemon
/etc/group:disk:x:6:root
/etc/group:wheel:x:10:root
/etc/group:pkcs11:x:101:root
/etc/passwd:root:x:0:0:root:/root:/bin/bash
/etc/passwd:operator:x:11:0:operator:/root:/sbin/nologin
```

```
[antonk@lab4236s ~]$ ls -l | grep bmt
```

```
-rw-rw-r-- 1 antonk antonk      0 Sep 21 15:11 bmt1
-rw-rw-r-- 1 antonk antonk      0 Sep 21 15:11 bmt2
-rw-rw-r-- 1 antonk antonk      0 Sep 21 15:11 bmt3
-rw-rw-r-- 1 antonk antonk    42 Sep 21 15:11 bmt?
-rw-rw-r-- 1 antonk antonk   192 Sep 30 15:35 bmt_link
```

```
[antonk@lab4236s ~]$ ls -l | grep ^d
```

```
drwxr-xr-x 2 antonk antonk 4096 Dec 22 2008 1111111
drwxr-xr-x 11 antonk antonk 4096 Dec 22 2008 222
drwxr-xr-x 4 antonk antonk 4096 Dec 22 2008 333
drwxr-xr-x 2 antonk antonk 4096 Rug 14 2008 BMT
drwxr-xr-x 3 antonk antonk 4096 Oct 15 15:53 Desktop
drwxrwxr-x 2 kras kras 4096 Dec 9 2008 Marsh
drwxrwxr-x 2 antonk antonk 4096 Oct 14 13:18 WWW
drwxrwxr-x 2 antonk antonk 4096 Oct 14 13:17 copy
drwxrwxr-x 3 antonk antonk 4096 Oct 3 15:34 new
```

# Команда egrep

egrep -опции шаблон имя\_файла(ов)  
команда | egrep -параметры шаблон

Например,

egrep -n 'root | admin' passwd group   или  
ps -ef | egrep '(r | g)+oot'

# Метасимволы команды egrep

Таблица 3 – Расширенный набор метасимволов команды egrep

Метасимвол	Описание	Пример	Результат
+	Соответствует одному или более предшествующих символов	' <u>[a-я]</u> + <u>аблон</u> '	Выбор одного или более символов нижнего регистра, которые находятся в диапазоне от «а» до «я» и за которыми следует « <u>аблон</u> »
<u>v п</u>	Соответствует одному из «v» или «п»	'яблоко груша'	Выбирается яблоко и/или груша
( )	Символы групп	'(1 2)+' '(к р)+ <u>орова</u> '	Выбор одного или более появлений. Например, «1» или «2» и «корова» или « <u>ророва</u> »

# Примеры с egrep

Осуществим поиск всех строк в файле `/etc/passwd`, в которых шаблон состоит из буквы «m», далее одной из перечисленных: «a», «y», «g», затем другие символы по строке.

Затем отобразим все строки файла `/etc/passwd`, которые включают одно из двух выражений: «daemon:/sbin» или «adm:/sbin».

```
[root@lab4236s antonk]# egrep 'm(a|y|g)'/etc/passwd

mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/sbin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash
amanda:x:33:6:Amanda user:/var/lib/amanda:/bin/bash
cyrus:x:76:12:Cyrus IMAP Server:/var/lib/imap:/bin/bash
mailman:x:41:41:GNU Mailing List Manager:/usr/lib/mailman:/sbin/nologin
st266084:x:705:705:Eidelman:/home/2009_2010/ekt47/st266084:/bin/bash

[root@lab4236s antonk]# egrep '(daemon:|adm:)/sbin'/etc/passwd

daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
```

# Команда fgrep

Обрабатывает все символы и любые метасимволы регулярных выражений, указанные в командной строке, как обычные символы.

`fgrep -опции шаблон имя_файла(ов)`

2 метасимвола:

? – один произвольный символ;

\$ - признак окончания регулярного выражения.



# Примеры с fgrep

В следующем примере найдем все строки, содержащие звездочку «\*».

```
[root@lab4236s antonk]# fgrep '*' /etc/syslog.conf
#kern.*                /dev/console
*.info;mail.none;news.none;authpriv.none;cron.none /var/log/messages
authpriv.*            /var/log/secure
mail.*                -/var/log/maillog
*.emerg                *
local7.*              /var/log/boot.log
```

# Команда `sort`

*Сортировка* – изменения порядка представления выдаваемой информации в соответствии с выбранными критериями.

`sort` -опции `-k номер_поля имя_файла(ов)`  
команда | `sort -параметр -k номер_поля`

Например,

`sort -n -k 3 group passwd` или

`ps -ef | sort -n -k 3`

# Описание параметров sort

Таблица 4 – Описание некоторых параметров команды sort

Параметр	Описание
-n	Производит числовую сортировку
(+ -)n	Начинает (+n) или заканчивает (-n) сортировку поля, следующего за полем n.
-r	Меняет порядок сортировки
-f	Игнорирует регистр символов
-d	Используется словарная сортировка. Сравниваются только буквы, цифры и пробелы, все остальные символы игнорируются.
-o <u>имя файла</u>	Сохранение результатов в файл <u>имя файла</u>
-b	Игнорирует начальные пробелы при определении начальной и конечной позиции ограниченного ключа сортировки.

# Описание параметров `sort`

Параметр	Описание
<code>-b</code>	Игнорирует начальные пробелы при определении начальной и конечной позиции ограниченного ключа сортировки.
<code>-t СИМВОЛ</code>	Использует «СИМВОЛ» как разделитель полей. Если параметр <code>-t</code> не указан, то по умолчанию в качестве разделителей полей используются символы пробела и табуляции.

# Примеры с sort

Обратная числовая сортировка всех файлов расширенного вывода данных, имена которых начинаются на буквы l и r, в домашней директории пользователя antonk по числу ссылок (по второму полю).

```
[antonk@lab4236s ~]$ ls -l ~/l* ~/p* | sort -rn -k 2
-rwxr-xr-x 2 antonk antonk 123 Oct 24 2008 /home/antonk/p.pl_link
-rwxr-xr-x 2 antonk antonk 123 Oct 24 2008 /home/antonk/p.pl
lrwxrwxrwx 1 antonk antonk 7 Sep 30 13:10 /home/antonk/link_rc->/etc/rc
lrwxrwxrwx 1 antonk antonk 7 Sep 30 11:09 /home/antonk/link2 -> /etc/re
lrwxrwxrwx 1 antonk antonk 7 Sep 30 11:06 /home/antonk/link -> /etc/re
```

То же действие, но с сохранением результатов в файл spisok

```
[antonk@lab4236s ~]$ ls -l ~/l* ~/p* | sort -rn -k 2 -o spisok
[antonk@lab4236s ~]$ cat spisok
-rwxr-xr-x 2 antonk antonk 123 Oct 24 2008 /home/antonk/p.pl_link
-rwxr-xr-x 2 antonk antonk 123 Oct 24 2008 /home/antonk/p.pl
lrwxrwxrwx 1 antonk antonk 7 Sep 30 13:10 /home/antonk/link_rc -> /etc/rc
lrwxrwxrwx 1 antonk antonk 7 Sep 30 11:09 /home/antonk/link2 -> /etc/rc
lrwxrwxrwx 1 antonk antonk 7 Sep 30 11:06 /home/antonk/link -> /etc/rc
[antonk@lab4236s ~]$ sort -nr -k 7 spisok > spisok_sort
[antonk@lab4236s ~]$ cat spisok_sort
lrwxrwxrwx 1 antonk antonk 7 Sep 30 13:10 /home/antonk/link_rc -> /etc/rc
lrwxrwxrwx 1 antonk antonk 7 Sep 30 11:09 /home/antonk/link2 -> /etc/rc
lrwxrwxrwx 1 antonk antonk 7 Sep 30 11:06 /home/antonk/link -> /etc/rc
-rwxr-xr-x 2 antonk antonk 123 Oct 24 2008 /home/antonk/p.pl_link
-rwxr-xr-x 2 antonk antonk 123 Oct 24 2008 /home/antonk/p.pl
```

# Команда `cmp`

Символьное сравнение двух текстовых файлов до первого символа различия. Возвращает номер строки и номер различающегося символа.

`cmp имя файла1 имя файла2`

```
[antonk@fargo laba2]$ cat r1
1
2
3
[antonk@fargo laba2]$ cat r2
1
4
3
[antonk@fargo laba2]$ cmp r1 r2
r1 r2 differ: char 3, line 2
```



# Команда diff

Команда diff – аналог стр. diff выводит все различия двух текстовых файлов при сравнении.

```
[antonk@fargo laba2]$ cat r1
1
2
3
[antonk@fargo laba2]$ cat r2
1
4
3
[antonk@fargo laba2]$ diff r1 r2
2c2
< 2
---
> 4
```

Таблица 5 – Некоторые опции команды diff

Параметр	Описание
<code>-i</code>	Игнорирует регистр символов, например, «P» эквивалентно «p».
<code>-c</code>	Создает подробный список различий.

# Команда WC

- ▣ Подсчет строк, слов, символов.
- ▣ `wc -опция(и) имя_файла(ов)`
- ▣ `команда | wc -опция имя_файлов`

Команда	Описание
l	Вывод количества строк
w	Вывод количества слов
c	Вывод количества байт
m	Вывод количества символов

```
[antonk@lab4236s ~]$ wc yuyu
```

```
5 38 243 yuyu
```

```
[antonk@lab4236s ~]$ ls -l www | wc
```

```
1 9 48
```

# Команда ps

Таблица 9 – Описание полей данных при использовании команды ps

Столбец	Описание
UID	Имя владельца процесса
PID	Уникальный идентификатор процесса
PPID	Уникальный идентификатор родительского процесса
C	Использование процессорного времени. Используется при выводе данных по команде top .
STIME	Время запуска процесса ( <u>Месяц</u> <u>День</u> )
TTY	Терминал, который использовался при запуске процесса (y)
TIME	Суммарное время работы процесса.
CMD	Имя команды, которая породила процесс.

# Команда `pgrep`

Таблица 10 – Параметры команды `pgrep`

Параметр	Описание
<code>-x</code>	Выводятся PID, полностью <u>совпадающие</u> с шаблоном.
<code>-n</code>	Выводятся только более <u>новые</u> PID, чем тот, что соответствует шаблону. Выводится ID процесса, соответствующего шаблону.
<code>-U</code>	Отображаются PID, <u>которые</u> принадлежат заданному пользователю. Требуется задать только имя пользователя.
<code>-l</code>	Наряду с PID процесса выводится его имя.

# Сигналы для управления процессами

Таблица 11 – Перечень некоторых сигналов.

Тип	Имя	Номер	Генерирующее условие
Отсутствие сигнала		0	Выход по команде exit или при достижении конца программы.
Разрыв	SIGHUP	1	Разрыв линии.
Прерывание терминала	SIGINT	2	Нажатие клавиши прерывания (например, Ctrl-C).
Прекращение	SIGQUIT	3	Нажатие клавиши прекращения (обычно, Ctrl-Shift-\ или Ctrl-Shift- ).
Уничтожение	SIGKILL	9	Самый сильный сигнал для процесса. <u>Применять в крайнем случае.</u>
Завершение программы	SIGTERM	15	Сигнал воздействия на процесс по умолчанию.
Стоп	SIGTSTP	20	Нажатие клавиши приостановки (обычно Ctrl-Z).

# Команда tar

Таблица 12 – Параметры команды tar

Функции	Определение
c	Создание нового tar-файла. Запись начинается с начала файла.
t	Просмотр содержимого tar-файла. Разархивирования не происходит.
x	Извлечение заданных файлов из tar-архива.
f	Используется для архивации в файл или из файла.
v	Выводятся комментарии (verbose mode).
r	Заданные файлы дописываются в конец ленточного архива.
u	Производится обновление архива. Если заданных файлов в архиве нет, или в архиве уже хранятся их обновленные версии, то они дописываются в конец архива.
e	При первой ошибке происходит немедленный выход.
h	Указание интерпретировать символические таким же образом, как обычные файлы и каталоги.

m	Используется при извлечении файлов из архива. Временем последней модификации становится момент извлечения.
o	При извлечении файлов из архива владельцем становится тот, кто выполнил команду tar .
w	Позволяет задать <u>текст приглашения</u> .



# Контрольные вопросы

- ▣ Назначение командного интерпретатора;
- ▣ Понятие абсолютного и относительного путей;
- ▣ Программный канал – конвейерная передача;
- ▣ Перенаправление стандартного ввода на устройство или в файл.
- ▣ Назовите уровни безопасности ОС Linux;
- ▣ Какие категории пользователей применяются Linux;
- ▣ Поясните принцип назначения прав доступа на объекты файловой системы в восьмеричной и символьной системах.
- ▣ Дайте определение понятиям фильтрация, сортировка и поиск ОФС;
- ▣ С помощью примеров поясните смысл работы команд `grep`, `sort`, `find`, `wc`.
- ▣ Назначение команды `ps`. Наиболее применяемые опции. Приведите примеры с `ps`.
- ▣ Команда `rgrep`. Пример с `rgrep`.
- ▣ Назначение команды `tar`. Создайте архив с помощью `tar`.
- ▣ Осуществите просмотр и извлеките данные из архива `tar`.

# Рекомендуемая литература

- ▣ В.В. Ермак, А.В. Козлов, В.Ю. Савченко / ОС Linux для разработчиков и пользователей ПО САПР БИС // Учебное пособие – М.МИЭТ, 2011. – 220с.:ил.
- ▣ Марк Г. Собелл / Практическое руководство по Red Hat Linux: Fedora Core и Red Hat Enterprise Linux // 2-е изд., : пер. с англ. – М.: «Издательский дом “Вильямс”», 2005. – 1072 с.: ил. – Парал. Тит. англ.;
- ▣ Петерсен Р. / Linux: Полное руководство // пер. с англ. 3-е изд., доп. – «Киев: ВНУ, Ирина», 2000. – 796 с. – УДК 004.451 + (библиотека МИЭТ; шифр 004.451 П-293).