

# Основы системного администрирования и сетевых технологий

---

УРОК №8

## Утилита find

Утилита find является одной из самых полезных утилит Linux. Она находит файлы, удовлетворяющие заданному набору параметров, таких как имя файла, дата его модификации, режима доступа, владельца и т.д.

Общий формат команды:

### **find путь –опции**

где путь – имя каталога, с которого необходимо начать поиск (символ «.» служит для обозначения текущего каталога, «/» - корневого каталога, «~» - домашнего каталога текущего пользователя), а опции сообщают команде, какие файлы интересуют пользователя.

Основные опции утилиты find приведены в табл. 1.

Таблица 1. Основные опции утилиты find

Опция	Описание
-name имя_файла	Поиск файлов, имена которых соответствуют заданному шаблону
-print	Вывод полных имен заданных файлов в стандартный поток вывода
-perm режим	Поиск файлов, для которых установлен указанный режим доступа
-prune	Применяется для того, чтобы find не выполняла рекурсивного поиска. Если указана опция –depth, то опция –prune игнорируется
-user владелец	Поиск файлов, принадлежащих указанному пользователю
-group группа	Поиск файлов, принадлежащих указанной группе
-mtime –n +n	Поиск файлов, содержание которых модифицировалось менее чем (-) или более чем (+) n дней назад. Имеются также опции –atime и –ctime, которые позволяют осуществлять поиск файлов по дате последнего чтения и дате последнего изменения файла
-nogroup	Поиск файлов, принадлежащих несуществующей группе

Опция	Описание
-ouser	Поиск файлов, принадлежащих несуществующему пользователю
-newer файл	Поиск файлов, которые созданы позднее, чем указанный файл
-type c	Поиск файлов определенного типа, а именно: b – специальный блочный файл; d – каталог; c – специальный символьный файл; p – именованный канал; l – символическая ссылка; s – сокет; f – обычный файл.
-size n	Поиск файлов, размер которых составляет n единиц. Возможны следующие варианты единиц измерения: b – блок размером 512 байт (установка по умолчанию); c – байт; k – килобайт (1024 байт); w – двухбайтовое слово.
-depth	При поиске файлов сначала просматривается содержимое существующего каталога и лишь затем проверяется запись, соответствующая самому каталогу
-fstype	Поиск файлов, которые находятся в файловой системе определенного типа. Обычно соответствующие сведения хранятся в файле /etc/fstab, который содержит сведения о файловых системах, используемых на компьютере
-mount	Поиск файлов только в текущей файловой системе. Аналогом этой опции является опция -xdev.

Опция	Описание
-exec команда	Выполнение команды интерпретатора shell для всех обнаруженных файлов. Выполняемые команды имеют формат команда {} \; (между символами «{}» и «\;» - пробел)
-ok команда	Аналогична опции -exec, но перед выполнением команды отображается запрос. Эта опция полезна для тестирования команды find.

### Утилита find

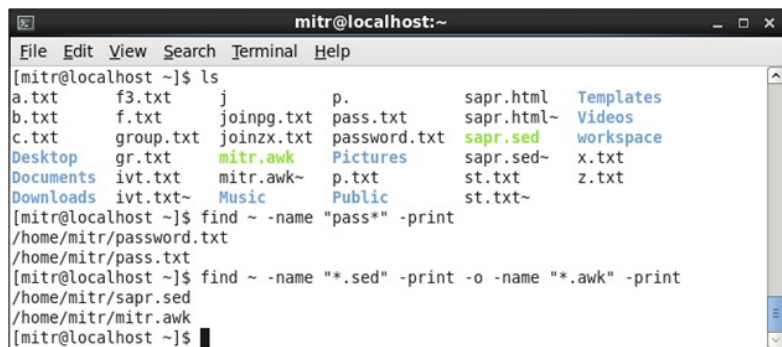
Часто возникает необходимость в поиске файлов, отвечающих сразу нескольким критериям. Поэтому необходим механизм объединения сразу нескольких параметров. Приемы объединения параметров приведены в табл. 2.

Таблица 2. Объединение нескольких параметров поиска

оператор_1 -a оператор_2	Поиск файлов, соответствующих как оператор_1, так и оператор_2.
оператор_1 -o оператор_2	Поиск файлов, соответствующих или оператор_1 или оператор_2
! оператор	Поиск всех файлов, не соответствующих заданному оператору. Символ «!» означает логическое НЕ.
\(выражение\)	Изменение порядка выполнения операторов. В сложных выражениях, прежде всего определяется значение выражения.

Достаточно часто при вызове find забывают указать опцию -print. Без этой опции утилита может не отображать имен файлов.

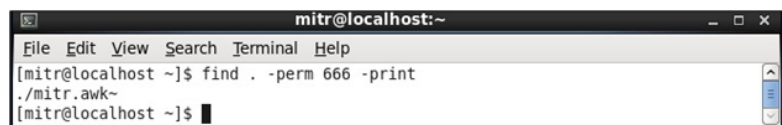
## Поиск файлов по шаблону имени



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ ls  
a.txt      f3.txt      j          p.          sapr.html  Templates  
b.txt      f.txt       joinpg.txt pass.txt    sapr.html~ Videos  
c.txt      group.txt   joinzx.txt password.txt sapr.sed    workspace  
Desktop    gr.txt      mitr.awk   Pictures    sapr.sed~  x.txt  
Documents  ivt.txt     mitr.awk~  p.txt       st.txt     z.txt  
Downloads  ivt.txt~    Music      Public      st.txt~  
[mitr@localhost ~]$ find ~ -name "pass*" -print  
/home/mitr/password.txt  
/home/mitr/pass.txt  
[mitr@localhost ~]$ find ~ -name "*.sed" -print -o -name "*.awk" -print  
/home/mitr/sapr.sed  
/home/mitr/mitr.awk  
[mitr@localhost ~]$
```

Рисунок 1. Поиск файлов по шаблону имени

## Поиск файлов с заданным режимом доступа



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ find . -perm 666 -print  
./mitr.awk~  
[mitr@localhost ~]$
```

Рисунок 2. Вывод полных имен всех файлов текущего каталога с режимом доступа 666 имени

На рис. 1 приведены примеры использования `find` для поиска файлов по шаблону имени. Для поиска по шаблону имени используется опция `-name` (name - имя).

Сначала с помощью команды `ls` просматривается содержимое текущего каталога, который в данном случае совпадает с домашним каталогом текущего пользователя.

Затем с помощью команды

**`find ~ -name "*pass" -print`**

выводятся полные имена всех файлов, имена которых начинаются на «pass». Поиск производится начиная с каталога `~`, то есть с домашнего каталога текущего пользователя.

Далее с помощью команды

**`find ~ -name "*.sed" -print -o -name "*.awk" -print`**

выводятся полные имена файлов, имена которых заканчиваются на «.sed» или (опция `-o`) на «.awk».

Для поиска с помощью `find` файлов с заданным режимом доступа используется опция `-perm` (permission - разрешение).

Найдем все файлы с режимом доступа 666, как это показано на рис. 2.

Если перед значением режима поставить дефис «-» то, будет произведен поиск файлов, для которых установлены все указанные биты разрешений, остальные биты при этом игнорируются.

## Поиск файлов с заданным режимом доступа

Если перед значением режима поставить знак «+», осуществляется поиск файлов, для которых установлен хотя бы один из указанных битов разрешений, при этом остальные биты разрешений игнорируются.

Пусть необходимо найти файлы с правом на запись для группы пользователя. Для этого нужно найти соответствие шаблону режима ....w..... Существует много комбинаций, которые подходят под этот шаблон, например ----w-gwx или g---w-gwx. Конечно, можно попытаться перечислить все возможные комбинации, но можно просто поставить перед восьмеричным значением шаблона знак «-».

Бит права записи в файл для группы пользователя соответствует восьмеричному числу 20 (000 010 000 = 24 = 1610 = 208), поэтому команда

**find ~ -perm -20 -print**

найдет файлы с правами доступа, перечисленными ниже:

gwxgwxgwx

gwxgwxg-x

gW-gW-gW-

gW-gW-g--

gW-gW----

Чтобы найти все файлы, которые можно исполнять (сценарии или программы) остальным пользователям можно задать шаблон режимов доступа ..x..... (26=6410=1008) с помощью команды

**find ~ -perm -100 -print**

## Поиск файлов с заданным режимом доступа

Если же необходимо найти файлы, которые могут исполнять или владелец или группа владельца или остальные, то шаблон режима будет `..x..x..x`, что соответствует восьмеричному ( $20+23+26 = 1+8+64 = 7310=1118$ ). Тогда для поиска таких файлов можно использовать команду

```
find ~ -perm +111 -print
```

## Поиск файлов по имени владельца или группы

Для поиска файлов по имени владельца используется опция `-user`. Пример поиска файлов по имени владельца:

```
find ~ -user "mitr" -print
```

Для поиска файлов по имени группы используется опция `-group`. Пример поиска файлов по имени группы:

```
find ~ -group "mitr" -print
```

Часто, когда учетная запись пользователя удаляется администратором, файлы пользователя еще остаются в системе. Чтобы найти такие файлы используется опция `-nouser`. Пример поиска файлов, собственник которых удален из системы:

```
find ~ -nouser -print
```

Опция `-nogroup` позволяет искать файлы несуществующих групп.

## Поиск файлов по типу

Если необходимо найти только файлы определенного типа, то для их поиска используется опция `-type тип`. Тип файла может быть следующим:

`b` – специальный блочный файл;

`d` – каталог;

`c` – специальный символьный файл;

`p` – именованный канал;

`l` – символическая ссылка;

`s` – сокет;

`f` – обычный файл.

Для того, чтобы найти все каталоги, начиная с домашнего каталога, можно воспользоваться командой

**`find ~ -type d -print`**

## Поиск файлов по размеру

Для поиска файлов по размеру используется опция `-size n`, где `n` – количество единиц в некоторых единицах измерения. Возможны следующие варианты единиц измерения:

`b` – блок размером 512 байт (установка по умолчанию);

`c` – байт;

`k` – килобайт (1024 байт);

`w` – двухбайтовое слово.

При поиске файлов, размеры которых лежат в определенном диапазоне, перед числом можно поставить знаки «-» или «+». Знак «-» означает «меньше чем», а знак «+» - «больше чем». Команда

**`find ~ -size -100c -size +10c -print > size.txt`**

сохраняет в файле `size.txt` полные имена всех файлов, пути которых начинаются с домашнего каталога работающего пользователя и размер которых менее 100 байт, но более 10 байт.



### Опция -depth

Опция `-depth` позволяет организовать поиск таким образом, что сначала проверяются все файлы текущего каталога и рекурсивно все его подкаталоги и только в конце – сам каталог. Опция широко применяется при создании списка файлов, помещаемых в архив с помощью команд `tar` или `cpio`, так как в этом случае сначала записывается образ каталога и лишь после этого задаются права доступа к нему. Это позволяет архивировать те каталоги, для которых у пользователя нет разрешения на запись.

### Опция -prune

Опция `-prune` прекращает поиск, выполняемый командой `find`, на текущем путевом имени. Если текущее путевое имя принадлежит каталогу, то `find` не будет в него заходить для дальнейшего поиска.

## Поиск файлов по времени доступа

Опция `-mtime` применяется при поиске файлов, доступ к которым осуществлялся *x* дней назад. Если аргумент опции снабдить знаком «-», то будут найдены файлы, к которым не было доступа в течении *x* дней. При использовании аргумента со знаком «+» производится поиск файлов, доступ к которым осуществлялся на протяжении последних *x* дней.

Команда

**`find ~ -mtime +2 -print`**

выведет все файлы, доступ к которым осуществлялся в течение последних 2 дней.

## Поиск файлов по времени доступа

Опция `-newer` позволяет найти файлы, доступ к которым осуществлялся в промежутке времени между обновлением двух заданных файлов. Общий формат команды `find` с опцией `-newer` следующий:

**`find -newer старый_файл ! -newer новый_файл`**

Символ «!» является оператором логического отрицания. Он означает, что надо найти файлы новее, чем `старый_файл`, но старше, чем `новый_файл`.

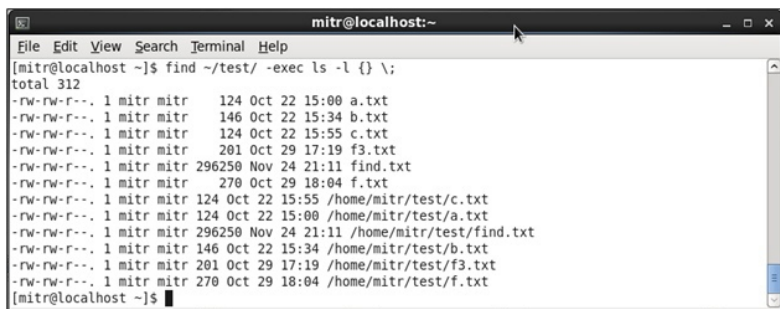
## Выполнение действий с найденными файлами

Для выполнения действий с найденными файлами используются опции `-exec` и `-ok`.

После опции `-exec` следует указать выполняемую команду, после чего ввести фигурные скобки, пробел, слеш и точку с запятой:

**`find ~ -type f -exec ls -l {} \;`**

Пример использования команды приведен на рис. 3. Здесь выполняется поиск обычных файлов, список которых отображается с помощью команды `ls -l`.



```
mitr@localhost:~$ find ~/test/ -exec ls -l {} \;
total 312
-rw-rw-r--. 1 mitr mitr 124 Oct 22 15:00 a.txt
-rw-rw-r--. 1 mitr mitr 146 Oct 22 15:34 b.txt
-rw-rw-r--. 1 mitr mitr 124 Oct 22 15:55 c.txt
-rw-rw-r--. 1 mitr mitr 201 Oct 29 17:19 f3.txt
-rw-rw-r--. 1 mitr mitr 296250 Nov 24 21:11 find.txt
-rw-rw-r--. 1 mitr mitr 270 Oct 29 18:04 f.txt
-rw-rw-r--. 1 mitr mitr 124 Oct 22 15:55 /home/mitr/test/c.txt
-rw-rw-r--. 1 mitr mitr 124 Oct 22 15:00 /home/mitr/test/a.txt
-rw-rw-r--. 1 mitr mitr 296250 Nov 24 21:11 /home/mitr/test/find.txt
-rw-rw-r--. 1 mitr mitr 146 Oct 22 15:34 /home/mitr/test/b.txt
-rw-rw-r--. 1 mitr mitr 201 Oct 29 17:19 /home/mitr/test/f3.txt
-rw-rw-r--. 1 mitr mitr 270 Oct 29 18:04 /home/mitr/test/f.txt
mitr@localhost:~$
```

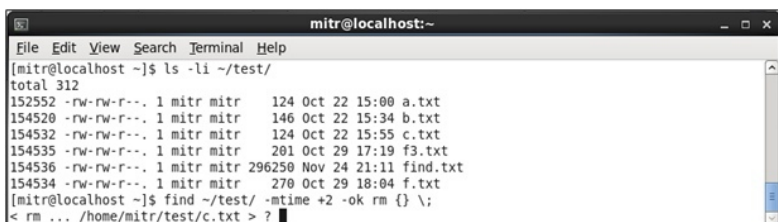
Рисунок 3.  
Пример использования команды  
`find` с опцией `-exec`

## Выполнение действий с найденными файлами



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ find ~/test/ -mtime +2 -exec rm {} \;  
[mitr@localhost ~]$ ls ~/test/  
find.txt  
[mitr@localhost ~]$
```

Рисунок 4. Удаление файлов  
с использованием опции `-exec`



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ ls -li ~/test/  
total 312  
152552 -rw-rw-r-- 1 mitr mitr 124 Oct 22 15:00 a.txt  
154520 -rw-rw-r-- 1 mitr mitr 146 Oct 22 15:34 b.txt  
154532 -rw-rw-r-- 1 mitr mitr 124 Oct 22 15:55 c.txt  
154535 -rw-rw-r-- 1 mitr mitr 201 Oct 29 17:19 f3.txt  
154536 -rw-rw-r-- 1 mitr mitr 296250 Nov 24 21:11 find.txt  
154534 -rw-rw-r-- 1 mitr mitr 270 Oct 29 18:04 f.txt  
[mitr@localhost ~]$ find ~/test/ -mtime +2 -ok rm {} \;  
< rm ... /home/mitr/test/c.txt > ?
```

Рисунок 5.  
Пример использования команды  
`find` с опцией `-ok`

## Команда `xargs`

Чтобы удалить в этом каталоге файлы, которые не обновлялись в этом каталоге в течении 3 дней, можно выполнить команду, представленную на рис. 4.

При перемещении и удалении файлов, найденных с помощью команды `find` желательно быть осторожным. Лучше вместо опции `-exec` использовать опцию `-ok`, которая отличается от `-exec` тем, что просит подтвердить каждую выполняемую операцию. Пример использования опции `-ok` приведен на рис. 5. Для удаления предлагаемого в результате поиска файла необходимо ввести `y`, для предотвращения удаления – `n`.

При наличии опции `-exec` команда `find` передает указанной команде все найденные файлы. Если в системе длина командной строки ограничена, то передаче большого количества имен файлов может появляться сообщение об ошибке. В этом случае можно использовать команду `xargs`, которая будет передавать имена файлов не все сразу, а порциями.

## Команда xargs

Например, следующая команда возвращает список всех файлов, начиная с текущего каталога, а команда xargs выполняет для них команду file, определяющую тип каждого файла:

**find . -type f -print | xargs file**

Подробнее с командой xargs можно познакомиться, используя справочную систему.

## Команда грег

Основной функцией команды грег (global regular expression print – печать глобальных регулярных выражений) является поиск и вывод строк, соответствующих регулярному выражению.

Общий формат команды грег:

**грег [параметры] регулярное\_выражение [файлы]**

В качестве регулярного выражения может быть указана обычная строка. Если файл не задан, то текст берется из стандартного входного потока.

Строку, которая задана в качестве регулярного выражения и состоит из нескольких слов, необходимо заключать в двойные кавычки, в противном случае первое слово будет восприниматься как образец для поиска, а оставшиеся рассматриваться как имена файлов. Если в шаблоне используется имя системной переменной, то оно также заключается в двойные кавычки. Если в состав шаблона входят метасимволы, то шаблон заключается в одинарные кавычки.

Основные опции команды грег представлены в табл. 3.

Команда **ггер**      Таблица 3. Основные опции команды **ггер**

Опция	Описание
-C	Задаёт отображение только числового значения, указывающего, сколько строк соответствует шаблону
-i	Игнорировать регистр символов
-h	Подавляет вывод имен файлов, включающих найденные строки (по умолчанию в выводе <b>ггер</b> каждой строке предшествует имя файла, в котором она содержится)
-l	Отображает только имена файлов, содержащих найденные строки
-n	Задаёт нумерацию выводимых строк
-s	Подавляет вывод сообщений о несуществующих или нетекстовых файлах
-v	Задаёт отображение строк, не соответствующих шаблону
-r	Определяет рекурсивный обход всего дерева указанного каталога.

```

mitr@localhost:~
File Edit View Search Terminal Help
[mitr@localhost ~]$ grep "Маша"
Маша гуляла в лесу.
Маша гуляла в лесу.
Она увидела Колю.
Улыбнулась Маша.
Улыбнулась Маша.
^C
[mitr@localhost ~]$

```

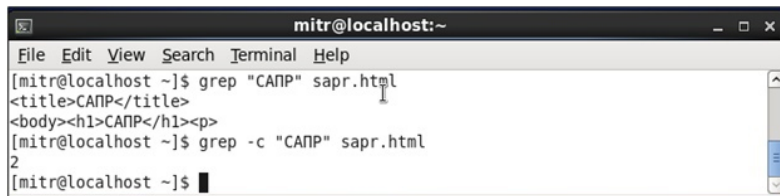
Рисунок 6.  
Обработка **ггер** стандартного  
входного потока

Если не указано имени файла, то команда обрабатывает стандартный ввод, например строки, набранные на клавиатуре, как это показано на рис. 6. При нажатии клавиши **Enter** для перехода на новую строку, **ггер** выводит строки, содержащие шаблон «Маша», отсюда и удвоение этих строк.

Если во всех файлах текущего каталога необходимо найти фразу «sapr vs», то это можно сделать с помощью команды **grep "sapr vs" \***

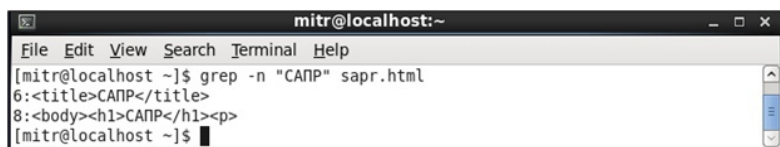
Опция **-c** позволяет определить, сколько строк соответствует заданному шаблону.

## Команда ггер



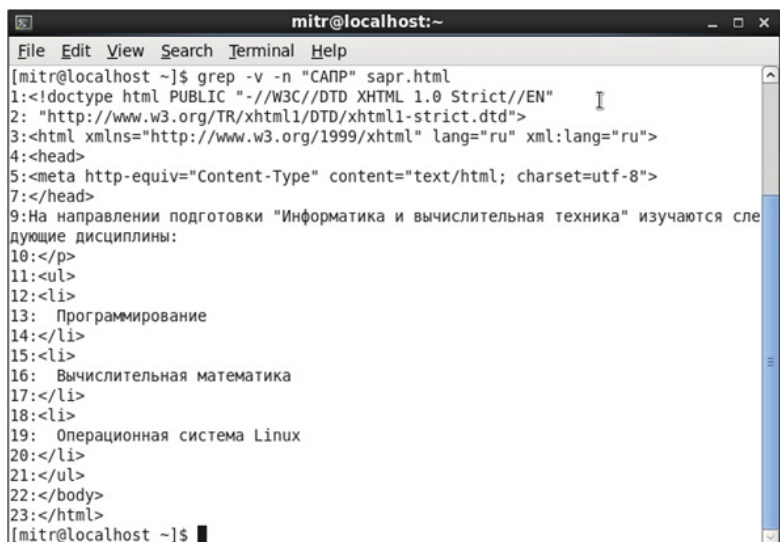
```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ grep "CAПP" sapr.html  
<title>CAПP</title>  
<body><h1>CAПP</h1><p>  
[mitr@localhost ~]$ grep -c "CAПP" sapr.html  
2  
[mitr@localhost ~]$
```

Рисунок 7.



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ grep -n "CAПP" sapr.html  
6:<title>CAПP</title>  
8:<body><h1>CAПP</h1><p>  
[mitr@localhost ~]$
```

Рисунок 8.



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ grep -v -n "CAПP" sapr.html  
1:<!doctype html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
2: "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
3:<html xmlns="http://www.w3.org/1999/xhtml" lang="ru" xml:lang="ru">  
4:<head>  
5:<meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
7:</head>  
9:На направлении подготовки "Информатика и вычислительная техника" изучаются сле  
дующие дисциплины:  
10:</p>  
11:<ul>  
12:<li>  
13: Программирование  
14:</li>  
15:<li>  
16: Вычислительная математика  
17:</li>  
18:<li>  
19: Операционная система Linux  
20:</li>  
21:</ul>  
22:</body>  
23:</html>  
[mitr@localhost ~]$
```

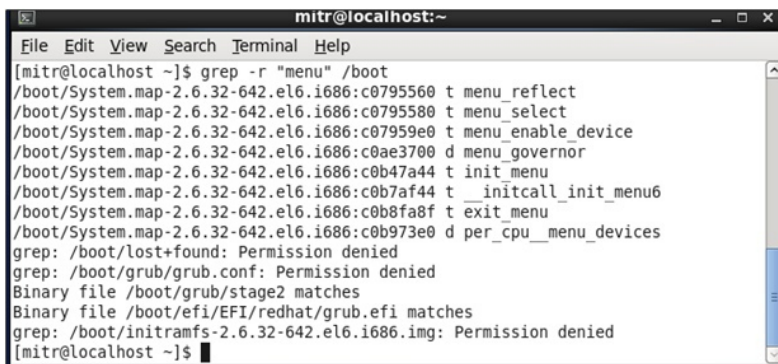
Рисунок 9.

На рис. 7 приведены примеры использования ггер без опции -с и с опцией -с.

С помощью опции -п выводимые строки можно перенумеровать, как это показано на рис. 8. Теперь можно установить, в строках с какими номерами произошло совпадение с шаблоном.

Опция -v используется в тех случаях, когда нужно отобрать строки, не соответствующие шаблону (рис. 9).

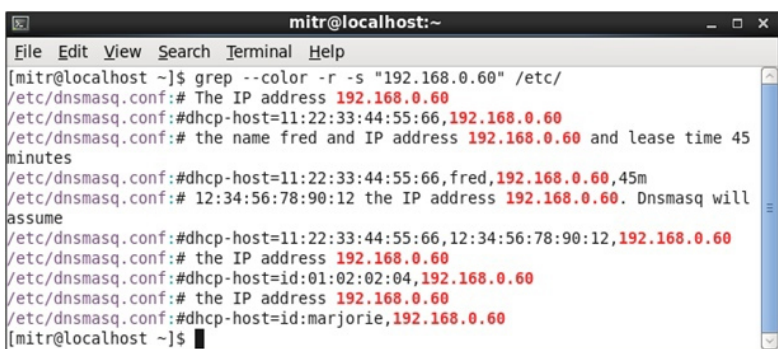
## Команда грег



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ grep -r "menu" /boot  
/boot/System.map-2.6.32-642.el6.i686:c0795560 t menu_reflect  
/boot/System.map-2.6.32-642.el6.i686:c0795580 t menu_select  
/boot/System.map-2.6.32-642.el6.i686:c07959e0 t menu_enable_device  
/boot/System.map-2.6.32-642.el6.i686:c0ae3700 d menu_governor  
/boot/System.map-2.6.32-642.el6.i686:c0b47a44 t init_menu  
/boot/System.map-2.6.32-642.el6.i686:c0b7af44 t __initcall_init_menu6  
/boot/System.map-2.6.32-642.el6.i686:c0b8fa8f t exit_menu  
/boot/System.map-2.6.32-642.el6.i686:c0b973e0 d per_cpu_menu_devices  
grep: /boot/lost+found: Permission denied  
grep: /boot/grub/grub.conf: Permission denied  
Binary file /boot/grub/stage2 matches  
Binary file /boot/efi/EFI/redhat/grub.efi matches  
grep: /boot/initramfs-2.6.32-642.el6.i686.img: Permission denied  
[mitr@localhost ~]$
```

Рисунок 10.

Опция -г позволяет рекурсивно обходить дерево каталогов, начиная с текущего каталога, как это показано на рис. 10.

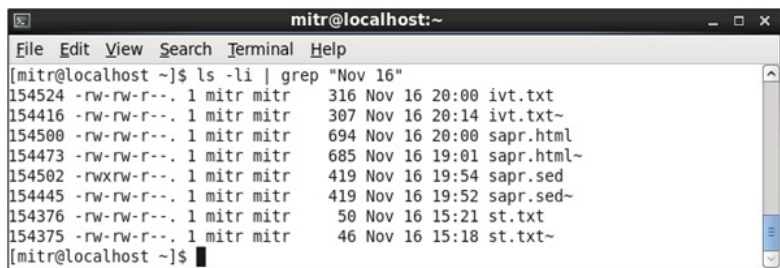


```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ grep --color -r -s "192.168.0.60" /etc/  
/etc/dnsmasq.conf:# The IP address 192.168.0.60  
/etc/dnsmasq.conf:#dhcp-host=11:22:33:44:55:66,192.168.0.60  
/etc/dnsmasq.conf:# the name fred and IP address 192.168.0.60 and lease time 45  
minutes  
/etc/dnsmasq.conf:#dhcp-host=11:22:33:44:55:66,fred,192.168.0.60,45m  
/etc/dnsmasq.conf:# 12:34:56:78:90:12 the IP address 192.168.0.60. Dnsmasq will  
assume  
/etc/dnsmasq.conf:#dhcp-host=11:22:33:44:55:66,12:34:56:78:90:12,192.168.0.60  
/etc/dnsmasq.conf:# the IP address 192.168.0.60  
/etc/dnsmasq.conf:#dhcp-host=id:01:02:02:04,192.168.0.60  
/etc/dnsmasq.conf:# the IP address 192.168.0.60  
/etc/dnsmasq.conf:#dhcp-host=id:marjorie,192.168.0.60  
[mitr@localhost ~]$
```

Рисунок 11.

С помощью регулярных выражений можно определять более сложные критерии поиска строк. При работе с регулярными выражениями шаблон поиска необходимо заключать в одиночные кавычки ' ', что позволит защитить все встречающиеся в нем специальные символы от интерпретатора shell, который в противном случае может перехватить их у грег. Опция --color позволяет выделить цветом вхождения в выводе грег. Пример такого выделения приведен на рис. 11.

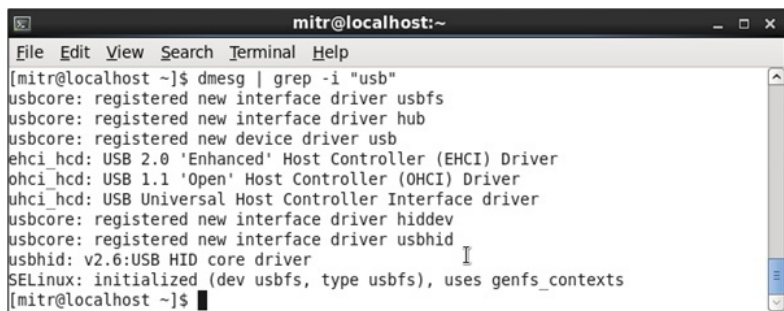
## Использование ггер совместно с другими командами



```
mitr@localhost:~$ ls -li | grep "Nov 16"
154524 -rw-rw-r--. 1 mitr mitr 316 Nov 16 20:00 ivt.txt
154416 -rw-rw-r--. 1 mitr mitr 307 Nov 16 20:14 ivt.txt~
154500 -rw-rw-r--. 1 mitr mitr 694 Nov 16 20:00 sapr.html
154473 -rw-rw-r--. 1 mitr mitr 685 Nov 16 19:01 sapr.html~
154502 -rwxrw-r--. 1 mitr mitr 419 Nov 16 19:54 sapr.sed
154445 -rw-rw-r--. 1 mitr mitr 419 Nov 16 19:52 sapr.sed~
154376 -rw-rw-r--. 1 mitr mitr 50 Nov 16 15:21 st.txt
154375 -rw-rw-r--. 1 mitr mitr 46 Nov 16 15:18 st.txt~
mitr@localhost ~]$
```

Рисунок 12.

Команда ггер часто используется как фильтр для обработки выходных данных других программ. Далее приводятся примеры использования ггер с другими программами. Например, если необходимо найти файлы по дате его последней модификации, то можно воспользоваться командой, показанной на рис. 12.



```
mitr@localhost:~$ dmesg | grep -i "usb"
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
uhci_hcd: USB Universal Host Controller Interface driver
usbcore: registered new interface driver hiddev
usbcore: registered new interface driver usbhid
usbhid: v2.6:USB HID core driver
SELinux: initialized (dev usbfs, type usbfs), uses genfs_contexts
mitr@localhost ~]$
```

Рисунок 13.

Команда ггер очень полезна при просмотре конфигурационных файлов и логов. Частым примером использования команды ггер стал программный канал с командой dmesg, которая выводит сообщения ядра во время загрузки компьютера (рисунок 13).



# Основы системного администрирования и сетевых технологий

---