



Поиск в Linux с помощью команды find

🕒 Обновлено: 15.12.2023 📅 Опубликовано: 25.07.2016

Утилита `find` представляет универсальный и функциональный способ для поиска в [Linux](#). Данная статья является шпаргалкой с описанием и примерами ее использования.

[Синтаксис](#)

[Примеры](#)

[Поиск по имени](#)

[По дате](#)

[По типу файла](#)

[По правам](#)

[По содержимому](#)

[С сортировкой по дате изменения](#)

[Лимиты](#)

[Действия над найденными объектами](#)

[Запуск по расписанию в CRON](#)

Общий синтаксис

```
find <где искать> <опции>
```

<где искать> — путь к корневому каталогу, откуда начинать поиск. Например, `find /home/user` — искать в соответствующем каталоге. Для текущего каталога нужно использовать точку «`.`».

<опции> — набор правил, по которым выполнять поиск.



опцию *maxdepth*.

Описание опций

Опция	Описание
-name	Поиск по имени.
-iname	Регистронезависимый поиск по имени.
-type	Тип объекта поиска. Возможные варианты: <ul style="list-style-type: none">■ f – файл;■ d – каталог;■ l – ссылка;■ p – pipe;■ s – сокет.
-size	Размер объекта. Задается в блоках по 512 байт или просто в байтах (с символом «с»).
-mtime	Время изменения файла. Указывается в днях.
-mmin	Время изменения в минутах.
-atime	Время последнего обращения к объекту в днях.
-amin	Время последнего обращения в минутах.
-ctime	Последнее изменение владельца или прав на объект в днях.
-cmin	Последнее изменение владельца или прав в минутах.
-user	Поиск по владельцу.
-group	По группе.
-perm	С определенными правами доступа.



-depth	с самого глубоко вложенного каталога.
-maxdepth	Максимальная глубина поиска по каталогам. -maxdepth 0 – поиск только в текущем каталоге. По умолчанию, поиск рекурсивный.
-prune	Исключение перечисленных каталогов.
-mount	Не переходить в другие файловые системы.
-regex	По имени с регулярным выражением.
-regextype <тип>	Тип регулярного выражения.
-L или -follow	Показывает содержимое символьных ссылок (симлинк).
-empty	Искать пустые каталоги.
-delete	Удалить найденное.
-ls	Вывод как <code>ls -dgils</code>
-print	Показать найденное.
-print0	Путь к найденным объектам.
-exec <команда> { } \;	Выполнить команду над найденным.
-ok	Выдать запрос перед выполнением -exec.

Также доступны логические операторы:

Оператор	Описание
-a	Логическое И. Объединяем несколько критериев поиска.
-o	Логическое ИЛИ. Позволяем команде <code>find</code> выполнить поиск на основе одного из критериев поиска.
-not или	Логическое НЕ. Инвертирует критерий



Полный набор актуальных опций можно получить командой **man find**.

Примеры использования find

Поиск файла по имени

1. Простой поиск по имени:

```
find / -name "file.txt"
```

** в данном примере будет выполнен поиск файла с именем **file.txt** по всей файловой системе, начинающейся с корня **/**.*

2. Поиск файла по части имени:

```
find / -name "*.tmp"
```

** данной командой будет выполнен поиск всех папок или файлов в корневой директории **/**, заканчивающихся на **.tmp***

3. Несколько условий.

а) Логическое И. Например, файлы, которые начинаются на **sess_** и заканчиваются на **cd**:

```
find . -name "sess_*" -a -name "*cd"
```

б) Логическое ИЛИ. Например, файлы, которые начинаются на **sess_** или заканчиваются на **cd**:

```
find . -name "sess_*" -o -name "*cd"
```

в) Более компактный вид имеют регулярные выражения, например:



```
find . -regex '.*\/(sess_.*\|.*cd\).'
```

** где в первом поиске применяется выражение, аналогичное примеру а), а во втором – б).*

4. Найти все файлы, кроме .log:

```
find . ! -name "*.log"
```

** в данном примере мы воспользовались логическим оператором !.*

Поиск по дате

1. Поиск файлов, которые менялись определенное количество дней назад:

```
find . -type f -mtime +60
```

** данная команда найдет файлы, которые менялись более **60** дней назад.*

Или в промужутке:

```
find . -mmin -20 -mmin +10 -type f
```

** найти все файлы, которые менялись более **10** минут, но не более **20**-и.*

2. Поиск файлов с помощью **newer**. Данная опция доступна с версии 4.3.3 (посмотреть можно командой **find --version**).

а) дате изменения:

```
find . -type f -newermt "2019-11-02 00:00"
```

** покажет все файлы, которые менялись, начиная с **02.11.2019 00:00**.*



** найдет все файлы, которые менялись в промежутке между 31.10.2019 и 01.11.2019 (включительно).*

б) дате обращения:

```
find . -type f -newerat 2019-10-08
```

** все файлы, к которым обращались с 08.10.2019.*

```
find . -type f -newerat 2019-10-01 ! -  
newerat 2019-11-01
```

** все файлы, к которым обращались в октябре.*

в) дате создания:

```
find . -type f -newerct 2019-09-07
```

** все файлы, созданные с 07 сентября 2019 года.*

```
find . -type f -newerct 2019-09-07 ! -newerct  
"2019-09-09 07:50:00"
```

** файлы, созданные с 07.09.2019 00:00:00 по 09.09.2019 07:50*

По типу

Искать в текущей директории и всех ее подпапках только файлы:

```
find . -type f
```

** **f** – искать только файлы.*

Поиск по правам доступа

1. Ищем все справками на чтение и запись:



2. Находим файлы, доступ к которым имеет только владелец:

```
find / -perm 0600
```

Поиск файла по содержимому

```
find / -type f -exec grep -i -H "content" {}  
\;
```

** в данном примере выполнен рекурсивный поиск всех файлов в директории / и выведен список тех, в которых содержится строка **content**.*

С сортировкой по дате модификации

```
find /data -type f -printf '%TY-%Tm-%Td %TT  
%p\n' | sort -r
```

** команда найдет все файлы в каталоге **/data**, добавит к имени дату модификации и отсортирует данные по имени. В итоге получаем, что файлы будут идти в порядке их изменения.*

Лимит на количество выводимых результатов

Самый распространенный пример – вывести один файл, который последний раз был модифицирован. Берем пример с сортировкой и добавляем следующее:

```
find /data -type f -printf '%TY-%Tm-%Td %TT  
%p\n' | sort -r | head -n 1
```

Поиск с действием (exec)



```
find . -name "sess_*" -type f -print -exec rm
{} \;
```

* **-print** использовать не обязательно, но он покажет все, что будет удаляться, поэтому данную опцию удобно использовать, когда команда выполняется вручную.

2. Переименовать найденные файлы:

```
find . -name "sess_*" -type f -exec mv
{} new_name \;
```

или:

```
find . -name "sess_*" -type f | xargs -I '{}'
mv {} new_name
```

3. Переместить найденные файлы:

```
find . -name "sess_*" -type f -exec mv {}
/new/path/ \;
```

* в данном примере мы переместим все найденные файлы в каталог **/new/path/**.

4. Вывести на экран количество найденных файлов и папок, которые заканчиваются на **.tmp**:

```
find . -name "*.tmp" | wc -l
```

5. Изменить права.

Рассмотрим пример, как найти все папки и задать им определенные права:

```
find /home/user/* -type d -exec chmod 2700 {}
\;
```




права **2700**.

Или так можно изменить права для всех файлов:

```
find ./ -type f -exec chmod 0644 {} \;
```

** в данном примере мы ищем все файлы (**type f**) в текущей директории **./** и ставим для них права **0644**.*

6. Передать найденные файлы конвейеру (pipe):

```
find /etc -name '*.conf' -follow -type f -exec  
cat {} \; | grep 'test'
```

** в данном примере мы использовали **find** для поиска строки **test** в файлах, которые находятся в каталоге **/etc**, и название которых заканчивается на **.conf**. Для этого мы передали список найденных файлов команде **grep**, которая уже и выполнила поиск по содержимому данных файлов.*

7. Произвести замену в файлах с помощью команды **sed**:

```
find /opt/project -type f -exec sed -i -e  
"s/test/production/g" {} \;
```

** находим все файлы в каталоге **/opt/project** и меняем их содержимое с **test** на **production**.*

Чистка по расписанию

Команду **find** удобно использовать для автоматического удаления устаревших файлов.

Открываем на редактирование задания **crontab**:

```
crontab -e
```

[Обо мне](#)[Услуги](#)[Настройка серверов](#)[Контакты](#)[Донаты](#)[Помощь online](#)

© Дмитрий Моск +7(906)250-48-69

```
0 0 * * * /bin/find /tmp -mtime +14 -exec rm  
{ } \;
```

** в данном примере мы удаляем все файлы и папки из каталога **/tmp**, которые старше **14** дней. Задание запускается каждый день в **00:00**.*

** полный путь к исполняемому файлу **find** смотрим командой **which find** – в разных UNIX системах он может располагаться в разных местах.*

Linux

UNIX



Была ли полезна вам эта инструкция?

Нравится

3

[Обо мне](#)

[Услуги](#)

[Настройка серверов](#)

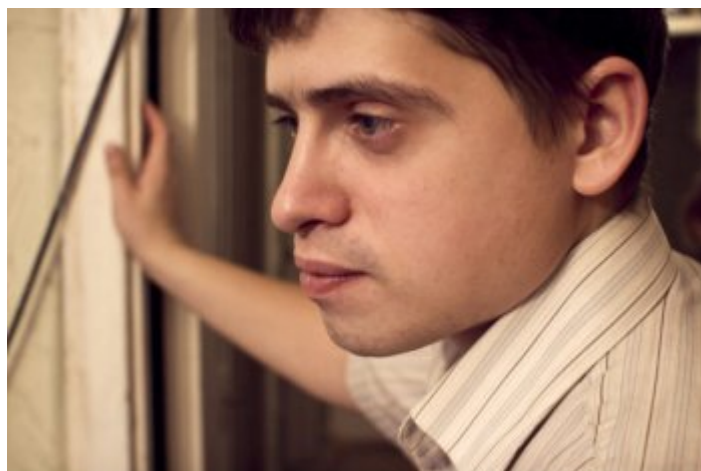
[Контакты](#)

[Донаты](#)

[Помощь online](#)



© Дмитрий Моск +7(906)250-48-69



*Дмитрий Моск – IT-специалист.
Настройка серверов, услуги DevOps.*

Мини-инструкции

[Инструкция по восстановлению пароля Windows](#)

[Как настроить ротацию логов во FreeBSD](#)

[Как добавить фотографию в Exchange](#)

[Описание команды find в Linux](#)

[Как открыть службы Windows](#)

[Установка и базовая настройка Ansible на Linux CentOS или Ubuntu](#)

[Как настроить автозавершение команд в Linux](#)

[Весь список ...](#)

Нужна помощь? Пишите:

Что хотите узнать...



Контактная эл. почта

Получить инструкцию

Реклама



[Настройка серверов](#)