



481.29

Рейтинг

FirstVDS

Виртуальные и выделенные серверы в ДЦ в Москве

[Подписаться](#)

1shaman

15 дек 2021 в 15:00

15 супер полезных примеров команды find в Linux

🕒 9 мин 👁 59K

Блог компании FirstVDS, Настройка Linux*, Системное администрирование*

[Перевод](#)

Автор оригинала: Abhishek Prakash

Команда FIND в Linux Полезные примеры



Практические примеры, которые приведены в этой статье, помогут вам освоить очень эффективную и крайне полезную команду find.

Она используется для поиска файлов и папок через командную строку Linux.

Команда `find` – одна из самых мощных и широко применимых команд. При этом она крайне объёмная и насчитывает более 50 опций, в которых легко запутаться, особенно в сочетании с командами `exec` или `xargs`.

Если вы сисадмин или разработчик, избежать команды `find` при работе с командной строкой не получится. Так что давайте научимся её не бояться и пользоваться её возможностями в полной мере.

Для этого разберём самые распространённые случаи практического применения команды `find`. Но для начала покажу вам синтаксис и принцип работы с командой.

Команда `find` в Linux

Общий синтаксис команды `find` выглядит так:

```
find [directory to search] [options] [expression]
```

Всё, что в квадратных скобках, указывать необязательно. А значит, выполнить команду `find` можно вообще без опций и параметров. Она выдаст список всех файлов и папок в текущем расположении. Мало полезного, да?

Так что давайте взглянем на параметры подробнее:

- `directory to search` (папка поиска) – это расположение, с которого вы хотите начать поиск. Поиск по умолчанию рекурсивный и начинается с текущего расположения.
- `options` (опции) содержит указание типа поиска: по имени, типу файла, времени изменения и так далее – тут может быть более 50 вариантов.
- `expression` (выражение) содержит поисковый запрос. Если вы ищете файл по имени, параметр `expression` должен содержать имя файла. Если ищете файлы с именем, соответствующим заданному шаблону, поисковое выражение – это шаблон.

Приведу простой пример:

```
find . -type f -name myfile
```

Такая команда выполнит поиск файла (именно файла, не папки) с именем `myfile` в

текущей папке и подпапках. Опция `-type f` сужает поиск до файлов. Точка (`.`) указывает на текущую папку.

Рассмотрим несколько примеров применения команды `find`.

Поиск файлов и папок по имени

Так выполняется поиск файлов и папок по имени:

```
find . -name SEARCH_NAME
```

Поскольку тип объекта не указан, команда выполняет поиск и файлов, и папок.

Пример ниже – поиск файлов и папок с именем «`mystuff`»:

```
abhishek@LHB:~/Examples$ find -name mystuff
./new/mystuff
./mystuff
```

Поиск только файлов или только папок

Если нужно искать только файлы, на помощь придёт опция `type -f`:

```
find . -type f -name SEARCH_NAME
```

Тип и имя можно указывать в любом порядке. Возьмём пример выше и ограничим круг поиска файлами:

```
abhishek@LHB:~/Examples$ find -type f -name mystuff
./mystuff
```

Если нужно найти папку, укажите тип `type -d`:

```
find . -type d -name SEARCH_NAME
```

Вот пример нашего поиска уже по папкам:

```
abhishek@LHB:~/Examples$ find -type d -name mystuff
./new/mystuff
```

Поиск без учёта регистра

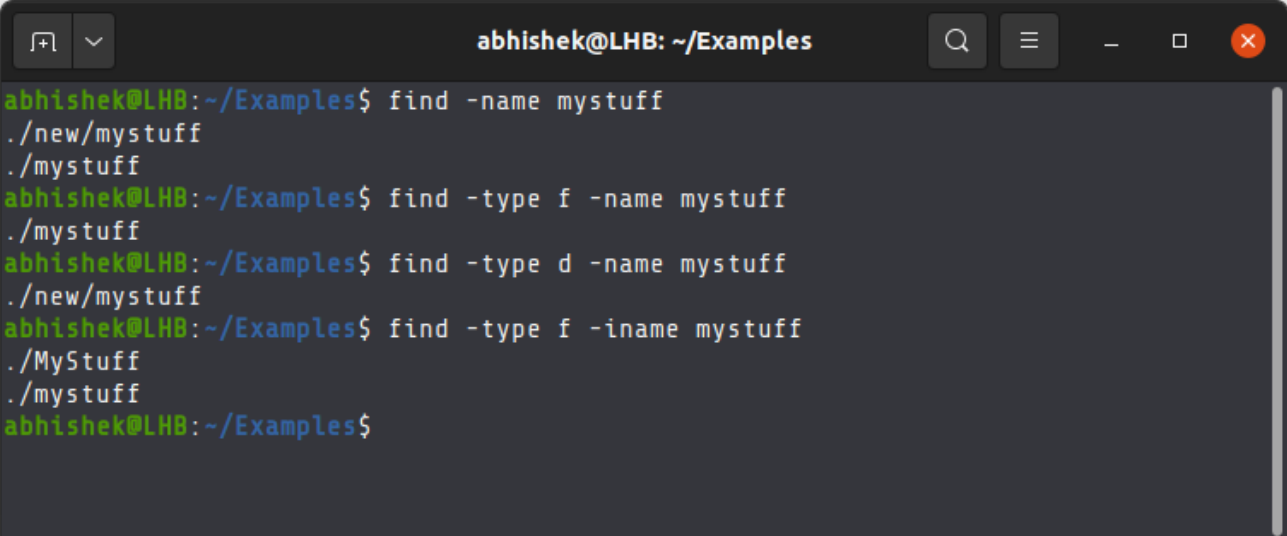
Команда `find` по умолчанию учитывает регистр. Чтобы выполнить поиск по имени файла без учёта регистра, надо ввести опцию `-iname` вместо `-name`.

```
find . -type f -iname SEARCH_NAME
```

С поиском по папкам (`type -d`) это тоже работает.

```
abhishek@LHB:~/Examples$ find -iname mystuff
./new/mystuff
./MyStuff
./mystuff
```

Скриншот последних трёх примеров:

A screenshot of a terminal window with a dark background. The window title is 'abhishek@LHB: ~/Examples'. It shows three sequential 'find' commands and their outputs. The first command 'find -name mystuff' returns './new/mystuff' and './mystuff'. The second command 'find -type f -name mystuff' returns './mystuff'. The third command 'find -type d -name mystuff' returns './new/mystuff'. The fourth command 'find -type f -iname mystuff' returns './MyStuff' and './mystuff'. The prompt is now 'abhishek@LHB:~/Examples\$'.

Поиск файлов по расширению (важно)

Одно из самых популярных применений команды `find` – поиск файлов определённого типа, то есть по заданному расширению.

Скажем, вы хотите найти все файлы C++ в текущих папках. Файлы C++ имеют расширение `.cpp`, и вот как их можно найти:

```
find . -type f -name "*.cpp"
```

С такими опциями команда `find` найдёт только файлы (`-type f`) с именами, оканчивающимися на `.cpp`.

```
abhishek@LNB:~$ find . -type f -name "*.cpp"
./file.cpp
./.cargo/registry/src/github.com-1ecc6299db9ec823/libz-sys-1.1.3/src/zlib/contrik
./.cargo/registry/src/github.com-1ecc6299db9ec823/libz-sys-1.1.3/src/zlib/contrik
./.cargo/registry/src/github.com-1ecc6299db9ec823/libz-sys-1.1.3/src/zlib/contrik
```

При работе с командой `find` всегда заключайте поисковое выражение в двойные кавычки.

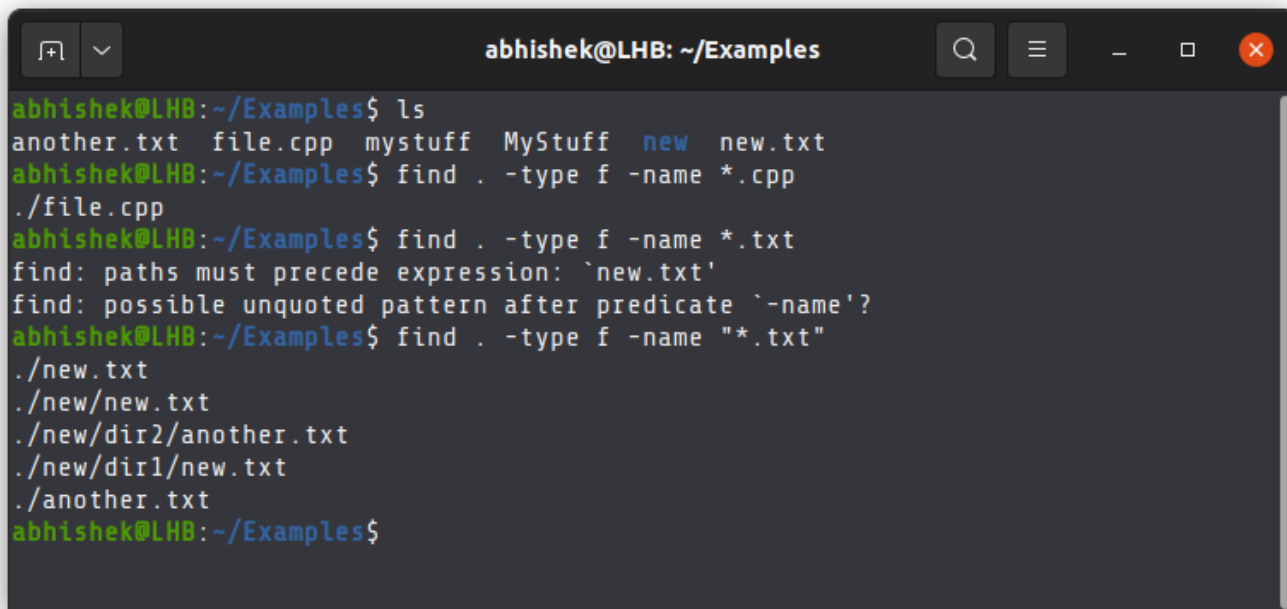
С чем связана рекомендация заключать поисковый запрос в двойные или одинарные кавычки? Дело в том, что без кавычек оболочка будет работать с символом `*` как с джокером и выполнит подстановку.

Вот что будет, если ввести запрос без кавычек:

```
find . -type f -name *.cpp
```

Оболочка распознает подстановочный знак `*` и заменит его всеми файлами в текущей папке, чьи имена заканчиваются на `.cpp`.

Это сработает, если такой файл всего один, но если их несколько, оболочка пожалуется на некорректный синтаксис.



```
abhishek@LHB: ~/Examples
abhishek@LHB:~/Examples$ ls
another.txt  file.cpp  mystuff  MyStuff  new  new.txt
abhishek@LHB:~/Examples$ find . -type f -name *.cpp
./file.cpp
abhishek@LHB:~/Examples$ find . -type f -name *.txt
find: paths must precede expression: `new.txt'
find: possible unquoted pattern after predicate `-name'?
abhishek@LHB:~/Examples$ find . -type f -name "*.txt"
./new.txt
./new/new.txt
./new/dir2/another.txt
./new/dir1/new.txt
./another.txt
abhishek@LHB:~/Examples$
```

В нашем случае файл .cpp всего один, и после подстановки команда выглядит так: `find . -type f -name file.cpp`. Она работает, поскольку `file.cpp` – корректный поисковый запрос.

А вот файлов .txt в той же папке два, и когда команда расширяется до `find . -type f -name another.txt new.txt`, выводится предупреждение, потому что поисковых запросов больше одного.

Именно поэтому сам поисковый запрос всегда следует заключать в двойные кавычки.

Поиск нескольких файлов с несколькими расширениями (или условием)

Команда, рассмотренная выше, нужна для поиска файлов по расширению. А что если нужно найти файлы с несколькими разными расширениями?

Вместо того чтобы прогонять команду `find` несколько раз, введите её один раз с опцией `-o`, которая работает как логическое условие «или»:

```
find . -type f -name "*.cpp" -o -name "*.txt"
```

Например:

```
abhishek@LHB:~/Examples$ find . -type f -name "*.txt" -o -name "*.cpp"
./new.txt
./file.cpp
```

```
./new/new.txt  
./new/dir2/another.txt  
./new/dir1/new.txt  
./another.txt
```

Поиск файлов в заданной папке

Все приведённые примеры иллюстрируют поиск в текущей папке, потому что команда включает в себя точку (.).

Чтобы выполнить поиск в заданной папке, не покидая текущего расположения, можно заменить точку абсолютным или относительным путём к нужной папке.

```
abhishek@LHB:~/Examples$ find ./new -name mystuff  
./new/mystuff
```

Поиск файлов в нескольких папках

Если нужные вам файлы могут находиться в нескольких папках, можно выполнить поиск во всех этих расположениях за один раз. Просто укажите все пути к папкам при введении команды `find`:

```
find ./location1 /second/location -type f -name "pattern"
```

Поиск пустых файлов и папок

Опция `-empty` позволяет использовать команду `find` для поиска пустых файлов и папок.

Найти таковые в текущей папке можно следующим образом:

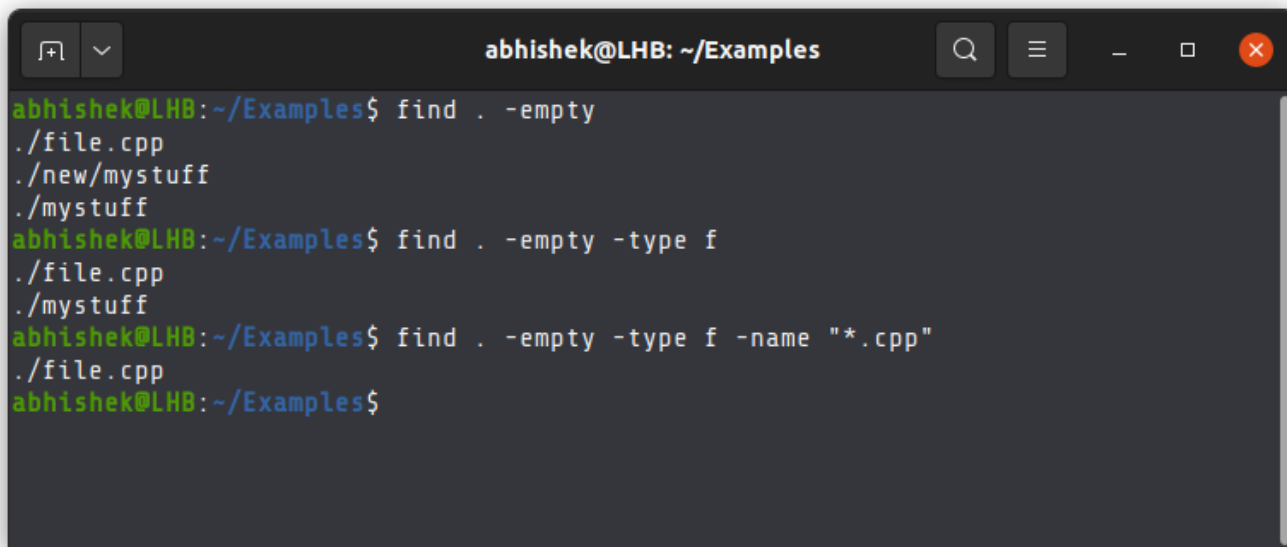
```
find . -empty
```

Можно указать тип объектов, чтобы искать только файлы или только папки:

```
find . -empty -type f
```

Кроме того, можно в таком режиме искать файлы по имени:

```
find . -empty -type f -name "*.cpp"
```

A screenshot of a terminal window titled 'abhishek@LHB: ~/Examples'. The terminal shows three sequential commands and their outputs. The first command is 'find . -empty', which outputs './file.cpp', './new/mystuff', and './mystuff'. The second command is 'find . -empty -type f', which outputs './file.cpp' and './mystuff'. The third command is 'find . -empty -type f -name "*.cpp"', which outputs './file.cpp'. The prompt 'abhishek@LHB:~/Examples\$' is visible at the end of each line.

```
abhishek@LHB:~/Examples$ find . -empty
./file.cpp
./new/mystuff
./mystuff
abhishek@LHB:~/Examples$ find . -empty -type f
./file.cpp
./mystuff
abhishek@LHB:~/Examples$ find . -empty -type f -name "*.cpp"
./file.cpp
abhishek@LHB:~/Examples$
```

Поиск крупных и мелких файлов (поиск по размеру файла)

Команда `find` поможет найти крупные или мелкие файлы, если выполнить поиск по размеру. Но это работает только для файлов, не для папок.

Используется опция `-size` с аргументом `+N` для файлов размером более `N` и `-N` для файлов размером менее `N`.

А вот как можно найти файлы точного заданного размера (50 КБ):

```
find . -size 50k
```

Так выполняется поиск файлов размером более 1 ГБ в текущей папке:

```
find . -size +1G
```

А так — файлов, не превышающих 20 байт:


```
find . -size -20c
```

Для поиска файлов размером более 100 МБ, но менее 2ГБ, введите:

```
find . -size +100M -size -2G
```

Поиск по размеру тоже можно сочетать с поиском по имени файла. Таким образом, найти в корневом каталоге все файлы размером более 500 МБ с именем, оканчивающимся на .log, можно так:

```
find / -size +500M -name "*.log"
```

Для справки:

- c – байты
- k – килобайты

 +34 245 17

Поиск недавно изменённых файлов (поиск по времени изменения или создания)

Вы ведь знакомы с параметрами mtime, atime и ctime?

- Mtime – время последнего изменения файла
- Ctime – время создания файла
- Atime – время последнего доступа к файлу

Вы не раз столкнётесь с ситуацией, когда нужен список всех недавно изменённых файлов. В таких случаях на помощь приходит поиск по времени изменения.

Найти все файлы, претерпевшие изменения за последние трое суток (3*24ч), можно так:

```
find . -type f -mtime -3
```

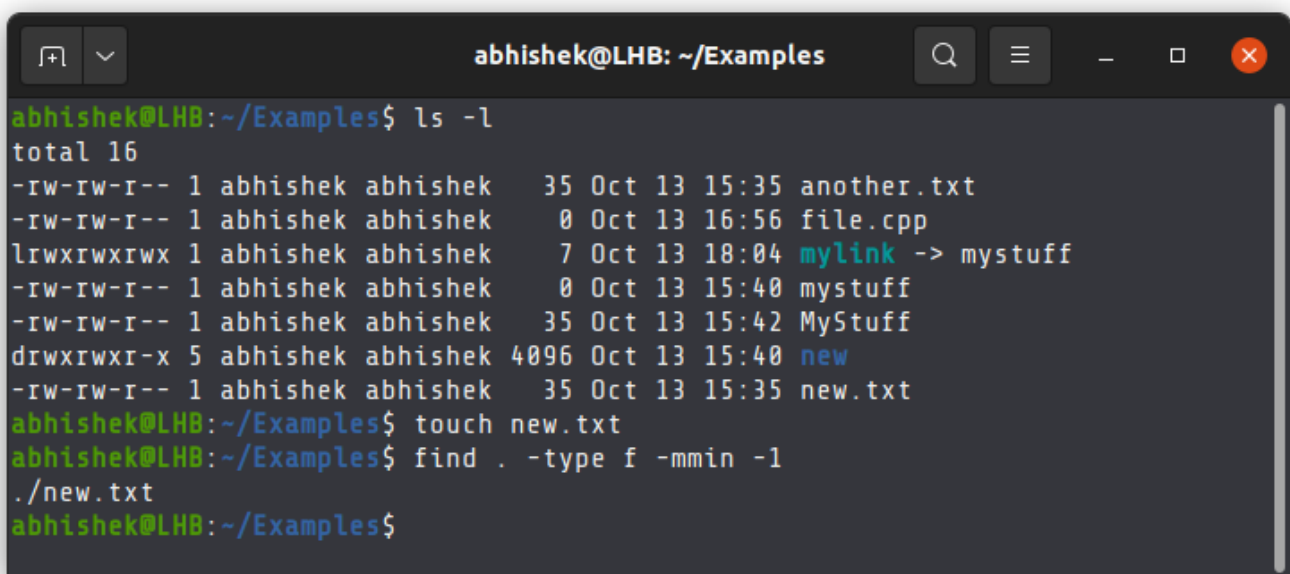
А все файлы, созданные пять и более дней назад, ищутся так:

```
find . -type f -ctime +5
```

Понимаю, что 24 часа – большой срок. Что если нужно выявить файлы, изменённые всего пару минут назад? Для этого предусмотрены опции `mmin`, `amin` и `cmin`.

Так выглядит команда поиска всех файлов, изменённых за последние пять минут:

```
find . -type f -mmin -5
```



```
abhishek@LHB: ~/Examples
abhishek@LHB:~/Examples$ ls -l
total 16
-rw-rw-r-- 1 abhishek abhishek 35 Oct 13 15:35 another.txt
-rw-rw-r-- 1 abhishek abhishek  0 Oct 13 16:56 file.cpp
lrwxrwxrwx 1 abhishek abhishek  7 Oct 13 18:04 mylink -> mystuff
-rw-rw-r-- 1 abhishek abhishek  0 Oct 13 15:40 mystuff
-rw-rw-r-- 1 abhishek abhishek 35 Oct 13 15:42 MyStuff
drwxrwxr-x 5 abhishek abhishek 4096 Oct 13 15:40 new
-rw-rw-r-- 1 abhishek abhishek 35 Oct 13 15:35 new.txt
abhishek@LHB:~/Examples$ touch new.txt
abhishek@LHB:~/Examples$ find . -type f -mmin -1
./new.txt
abhishek@LHB:~/Examples$
```

Можно не только указать имя файла, но и ограничить временной промежуток с двух сторон. Команда ниже выполнит поиск всех файлов `.java`, изменённых не ранее 30 и не позднее 20 минут назад.

```
find . -type f -mmin +20 -mmin -30 -name "*.java"
```

Поиск файлов с определёнными настройками доступа

Надеюсь, вы имеете представление о разрешениях файлов в Linux.

Команда `find` позволяет выполнить поиск файлов по разрешению и режиму доступа.

```
find -perm mode
```

Поиск в текущей папке, к примеру, все файлы с режимом доступа `777`:

```
find . -perm 777
```

А так можно найти все файлы с правами на чтение и запись для всех типов пользователей (только точное совпадение; файлы с правами на выполнение для всех не отобразятся):

```
find . -perm a=r+w
```

Поиск файлов по владельцу

Можно также найти файлы, принадлежащие определённому пользователю.

Вот как обнаружить в текущей папке все файлы пользователя Джона:

```
find . -type f -user John
```

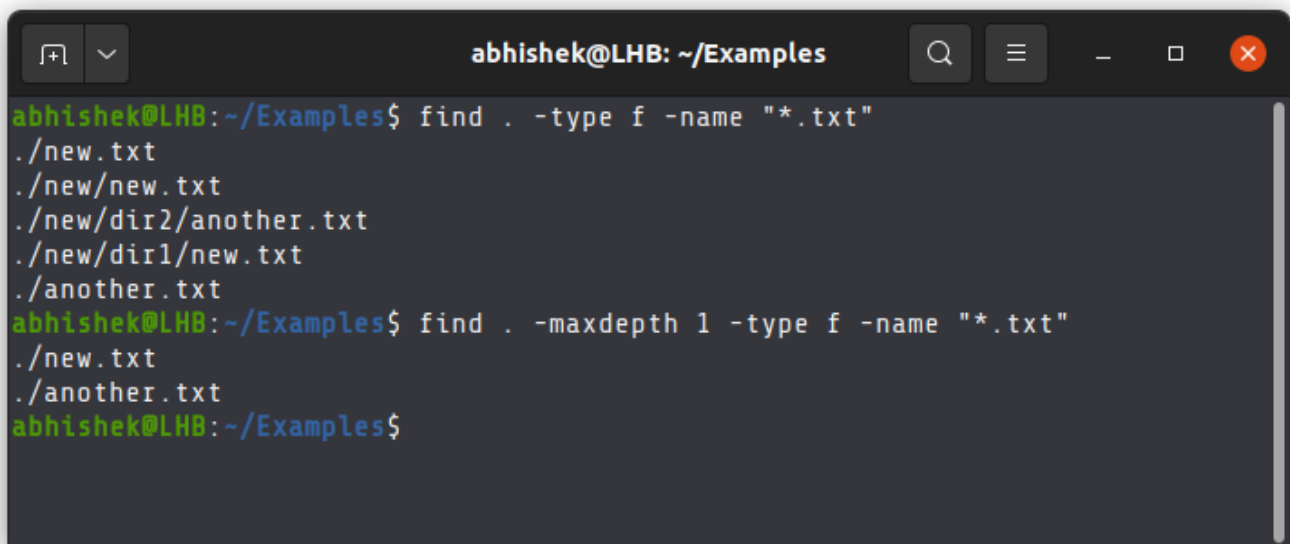
Эта опция сочетается с другими, будь то размер или время и имя файла:

```
find . -type f -user John -name "*.cpp"
```

Отключение рекурсивного поиска для поиска только в текущей папке

По умолчанию команда `find` выполняет поиск во всех подпапках текущего расположения. Если это не требуется, можно ограничить глубину поиска значением «1». Так вы ограничитесь поиском в текущей папке, не залезая в подпапки.

```
find . -maxdepth 1 -type f -name "*.txt"
```



```
abhishek@LHB: ~/Examples
abhishek@LHB:~/Examples$ find . -type f -name "*.txt"
./new.txt
./new/new.txt
./new/dir2/another.txt
./new/dir1/new.txt
./another.txt
abhishek@LHB:~/Examples$ find . -maxdepth 1 -type f -name "*.txt"
./new.txt
./another.txt
abhishek@LHB:~/Examples$
```

Исключение папки из поиска

Если нет необходимости производить поиск в той или иной папке, можно исключить её с помощью опций `path`, `prune` и логического «или».

```
find . -path "./directory_exclude/*" -prune -o -name SEARCH_NAME
```

Будьте внимательны: путь к папке должен оканчиваться на `*`, затем идёт `-prune` и только потом `-o`.

Попросту говоря, при поиске с опцией `prune` папка, указанная с помощью `path`, игнорируется. `Prune` всегда сопровождается флагом `-o` (логическое «или»), чтобы папки, которые не были исключены, просматривались на наличие искомого объекта.

Дальнейшая работа с результатами команды `find`: `exec` и `xargs`

Итак, мы изучили различные способы поиска файлов по заданным параметрам. Это хорошо. А теперь следующий шаг: рассмотрим, какие действия можно выполнять с результатами команды `find`.

Например, как найти файлы с именем, соответствующим определённому шаблону, и переименовать их за одно действие? Или выявить и удалить пустые файлы?

Вам уже известно, что в Linux можно использовать перенаправление ввода-вывода, чтобы объединить результаты одной команды с вводом другой. Но с результатами команды `find`

это не работает – по крайней мере, не напрямую.

Чтобы выполнить действия над результатом команды `find`, есть два варианта:

- Применить `exec`
- Применить `xargs`

Использование `find` и `exec`

Допустим, вам нужен подробный список (`ls -l`) файлов, найденных командой `find`. Вот как его получить:

```
find . -type f -name "*.txt" -exec ls -l {} +
```

Результат будет таким:

```
abhishek@LNB:~/Examples$ find . -type f -name "*.txt" -exec ls -l {} +
-rw-rw-r-- 1 abhishek abhishek 39 Oct 13 19:30 ./another.txt
-rw-rw-r-- 1 abhishek abhishek 35 Oct 13 15:36 ./new/dir1/new.txt
-rw-rw-r-- 1 abhishek abhishek 35 Oct 13 15:36 ./new/dir2/another.txt
-rw-rw-r-- 1 abhishek abhishek 35 Oct 13 18:51 ./new/mystuff/new.txt
-rwxrwxrwx 1 abhishek abhishek 35 Oct 13 15:37 ./new/new.txt
-rw-rw-r-- 1 abhishek abhishek 35 Oct 13 18:16 ./new.txt
```

Многие забывают ввести `{}` + в конце команды `exec`. Но это необходимо – как и пробел между скобками `{}` и плюсом +.

Фигурные скобки ссылаются на результат выполнения команды `find`. Их содержимое может иметь следующий вид: {файл 1, файл 2, файл 3}. Символ + используется как конец команды `exec`.

Есть ещё один вариант оформления `exec`:

```
find . -type f -name "*.txt" -exec ls -l {} \;
```

В данном случае плюс заменён на точку с запятой. Дополнительная косая черта означает, что точка с запятой не является специальным символом.

Преимущество сочетания `{}` + заключается в меньшем количестве команд (`ls -l file1 file2 file3`), тогда как комбинация `{}` `\;` запустит цепочку `ls -l file1` , `ls -l file2` и так далее.

Однако сочетание `{}` `\;` даёт возможность использовать `{}` несколько раз в одном и том же выражении `exec`. Так, приведённая ниже команда переименует все обнаруженные файлы с расширением `.old`.

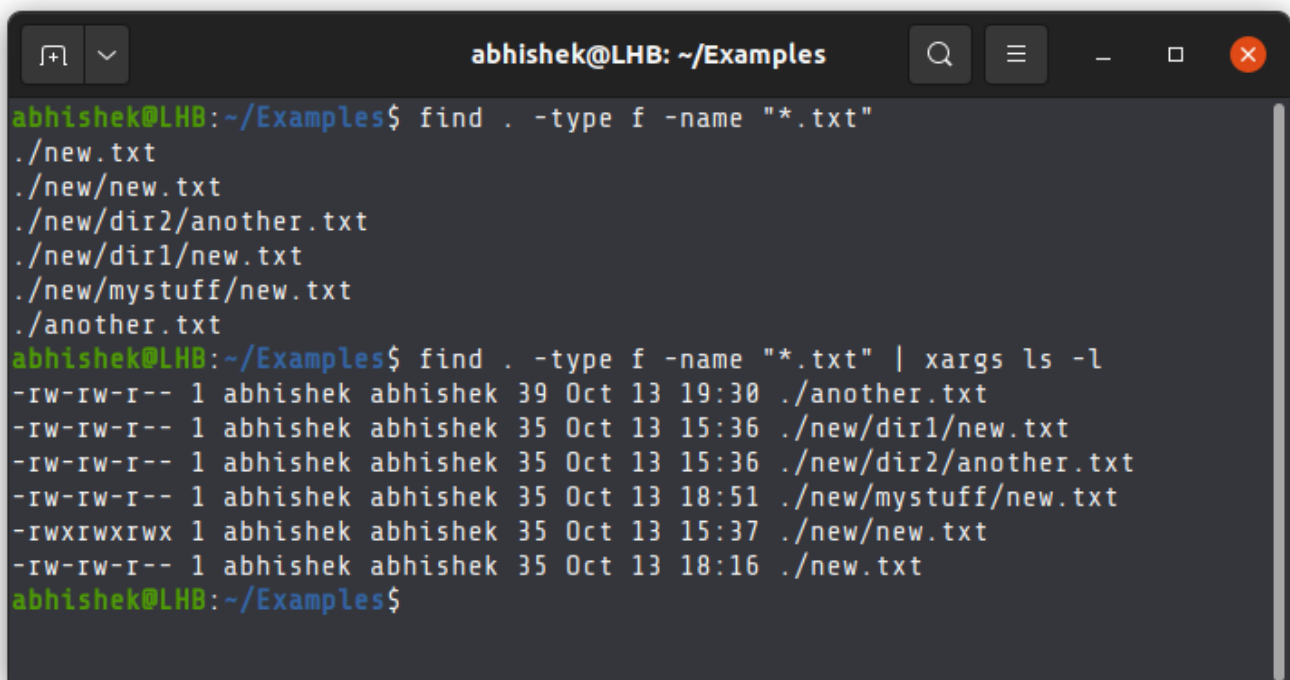
```
find . -type f -name "*.txt" -exec mv {} {}.old \;
```

Использование команды `xargs`

Многие пользователи Linux сталкиваются с необходимостью перенаправления ввода-вывода довольно часто. Но команда `exec` с цепочкой символов `{}` + кажется им слишком сложной.

И тут на помощь приходит `xargs`. Нужно просто перенаправить вывод команды `find` в команду `xargs` через конвейер.

```
find . -type f -name "*.txt" | xargs ls -l
```



The screenshot shows a terminal window titled "abhishek@LHB: ~/Examples". The user runs the command `find . -type f -name "*.txt"`, which lists several text files. Then, the user runs `find . -type f -name "*.txt" | xargs ls -l`, which displays the same files with their permissions, ownership, and timestamps.

```
abhishek@LHB: ~/Examples
abhishek@LHB:~/Examples$ find . -type f -name "*.txt"
./new.txt
./new/new.txt
./new/dir2/another.txt
./new/dir1/new.txt
./new/mystuff/new.txt
./another.txt
abhishek@LHB:~/Examples$ find . -type f -name "*.txt" | xargs ls -l
-rw-rw-r-- 1 abhishek abhishek 39 Oct 13 19:30 ./another.txt
-rw-rw-r-- 1 abhishek abhishek 35 Oct 13 15:36 ./new/dir1/new.txt
-rw-rw-r-- 1 abhishek abhishek 35 Oct 13 15:36 ./new/dir2/another.txt
-rw-rw-r-- 1 abhishek abhishek 35 Oct 13 18:51 ./new/mystuff/new.txt
-rwxrwxrwx 1 abhishek abhishek 35 Oct 13 15:37 ./new/new.txt
-rw-rw-r-- 1 abhishek abhishek 35 Oct 13 18:16 ./new.txt
abhishek@LHB:~/Examples$
```

Синтаксис куда проще, верно? К тому же команда `xargs` тоже весьма эффективна. Подробнее о ней – в статье по [ссылке](#).

Сочетание команд `find` и `grep`

Теперь вы умеете совмещать команду `find` с `xargs` и `exes`, и пора перейти на следующий уровень – объединить `find` и `grep`.

Для сисадминов и разработчиков комбинация команд `find` и `grep` – одна из самых распространённых и вместе с тем самых полезных.

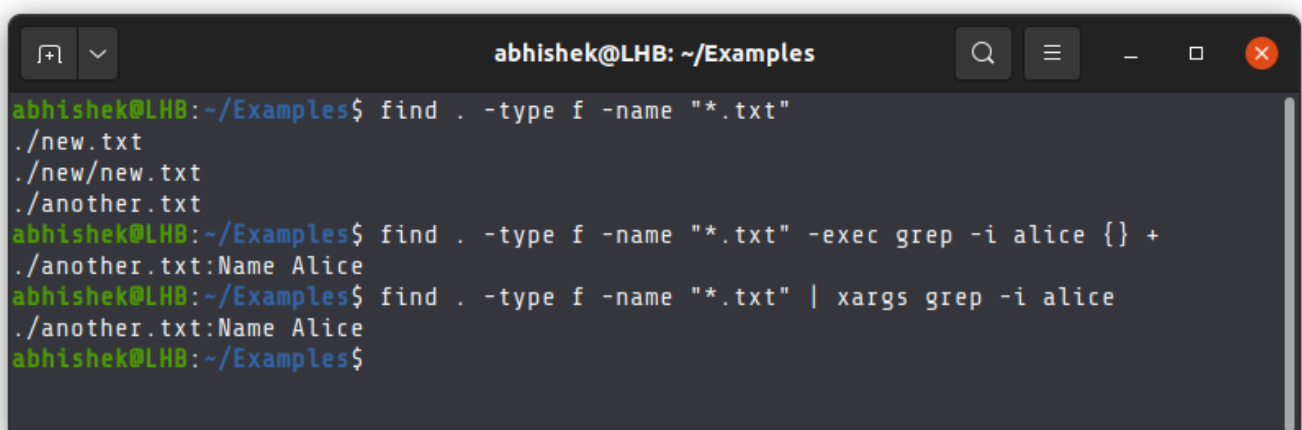
Команда `find` находит файлы с именем, соответствующим шаблону, а затем команда `grep` выполняет поиск по их содержимому.

Например, вам нужно найти все файлы `.txt`, в которых есть имя «Alice». Объединить команды `find` и `grep` можно так:

```
find . -type f -name "*.txt" -exec grep -i alice {} +
```

А можно с помощью `xargs`:

```
find . -type f -name "*.txt" | xargs grep -i alice
```

A screenshot of a terminal window with a dark background. The window title is 'abhishek@LHB: ~/Examples'. The terminal shows three commands and their outputs. The first command is 'find . -type f -name "*.txt"', which lists several files. The second command is 'find . -type f -name "*.txt" -exec grep -i alice {} +', which shows the output for one file. The third command is 'find . -type f -name "*.txt" | xargs grep -i alice', which also shows the output for one file.

```
abhishek@LHB: ~/Examples
abhishek@LHB:~/Examples$ find . -type f -name "*.txt"
./new.txt
./new/new.txt
./another.txt
abhishek@LHB:~/Examples$ find . -type f -name "*.txt" -exec grep -i alice {} +
./another.txt:Name Alice
abhishek@LHB:~/Examples$ find . -type f -name "*.txt" | xargs grep -i alice
./another.txt:Name Alice
abhishek@LHB:~/Examples$
```

Конечно, пример элементарный, но если команда `grep` вам знакома, можете использовать её на своё усмотрение.

И это далеко не все возможности команды `find`...

Перечислить все опции и примеры использования команды `find` практически невозможно. Её возможностям нет границ, но если вы освоите её принципы, она окажется очень кстати во многих ситуациях. Решающий фактор – как сочетается логика действия разных опций и команд.

Надеюсь, моя подборка примеров использования `find` была для вас полезна. Если у вас есть вопросы или предложения, как сделать эту статью лучше, добро пожаловать в комментарии.

НЛО прилетело и оставило здесь промокоды для читателей нашего блога:

- 15% на все тарифы VDS (кроме тарифа Прогрев) – **HABRFIRSTVDS**.
- 20% на выделенные серверы AMD Ryzen и Intel Core – **HABRFIRSTDEDIC**.

Доступно до 31 декабря 2021 г.

Теги: FirstVDS, FirstDEDIC, команды, линукс

Хэбы: Блог компании FirstVDS, Настройка Linux, Системное администрирование



FirstVDS

Виртуальные и выделенные серверы в ДЦ в Москве

Подписаться

Сайт Сайт Сайт ВКонтакте Telegram



93

Карма

-1

Рейтинг

@1shaman

Системный администратор

Подписаться



Комментарии 17

Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ

**Stolyarr**

7 часов назад

Как я случайно сэкономил компании полмиллиона долларов

7 мин

13K

Перевод

+47

22

26

**materiatura**

20 часов назад

Закон больших чисел и закон больших граблей

Простой

10 мин

5.2K

+37

43

25

**SergeyNovak**

9 часов назад

Загружаем OpenIPC по сети на камеру без удаления заводской прошивки

Средний

13 мин

2.5K

Тutorial

+36

32

0

**rumin_v_rabote**

9 часов назад

Особенности национального бизнеса на маршрутках

Простой

6 мин

3.6K

Интервью

+30

19

26

**mvideo**

9 часов назад

Press F. Что из мира IT мы потеряли в 2023 году

🕒 12 мин 👁 10K

📌 +24

🔖 16

💬 11

 **DrArgentum**

10 часов назад

Эльфы и пингвины: что такое ELF и как он работает в Linux?

🟢 Простой 🕒 37 мин 👁 4.5K

Тutorial

📌 +22

🔖 73

💬 13



ru_vds

5 часов назад

Даже в космосе пылища: какой мусор кружит в вакууме

🕒 4 мин 👁 2.3K

📌 +21

🔖 4

💬 10



Xenia_Gavrilova

7 часов назад

Как сделать консистентный UX для 40+ продуктов. Уроки, которые я извлекла из перезапуска дизайн-системы

💧 Средний 🕒 17 мин 👁 758

Ретроспектива

📌 +13

🔖 14

💬 1



alexeyinkin

7 часов назад

Как главный хакатон в России стёр 2000 наград, и как предотвратить это снова


🟢 Простой 🕒 9 мин 👁 1.8K

Мнение

📌 +12

🔖 4


💬 13





AnotherAnkor

11 часов назад

Добро пожаловать в нагрузочное тестирование! Вы поседаете через полгода


 Простой


 11 мин


 2.4K

Обзор

Хабр IT-гид

 +9

 16

 0

Показать еще

Ваш аккаунт

- Профиль
- Трекер
- Диалоги
- Настройки
- ППА

Разделы

- Статьи
- Новости
- Хабы
- Компании
- Авторы
- Песочница

Информация

- Устройство сайта
- Для авторов
- Для компаний
- Документы
- Соглашение
- Конфиденциальность

Услуги

- Корпоративный блог
- Медийная реклама
- Нативные проекты
- Образовательные программы
- Стартапам



Настройка языка

Техническая поддержка

ИНФОРМАЦИЯ

Сайт	firstvds.ru
Дата регистрации	13 сентября 2012
Дата основания	6 декабря 2002
Численность	51–100 человек
Местоположение	Россия
Представитель	FirstJohn

ССЫЛКИ

FirstVDS – VDS хостинг от 174 рублей.
firstvds.ru

Выделенные серверы за 5 минут. FirstDEDIC

firstdedic.ru

CLO.ru – облако с почасовой оплатой

clo.ru

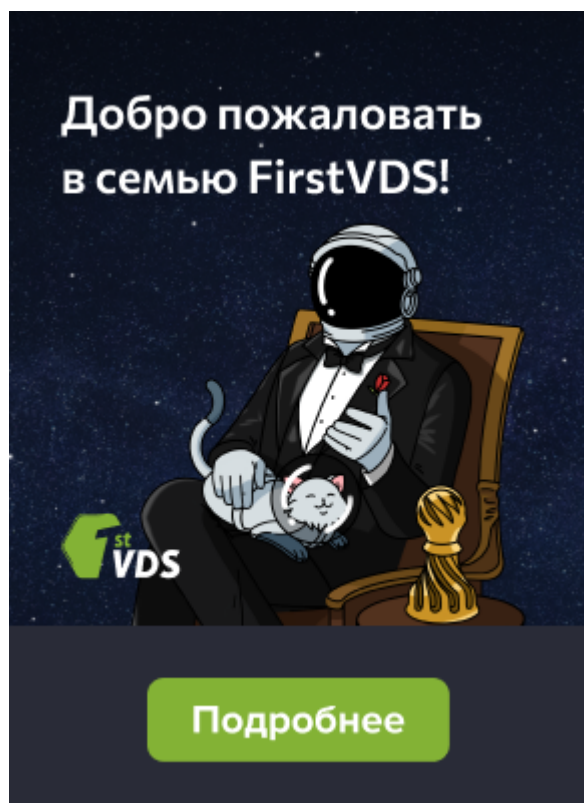
В КОНТАКТЕ

Сайт vk.com заблокиро

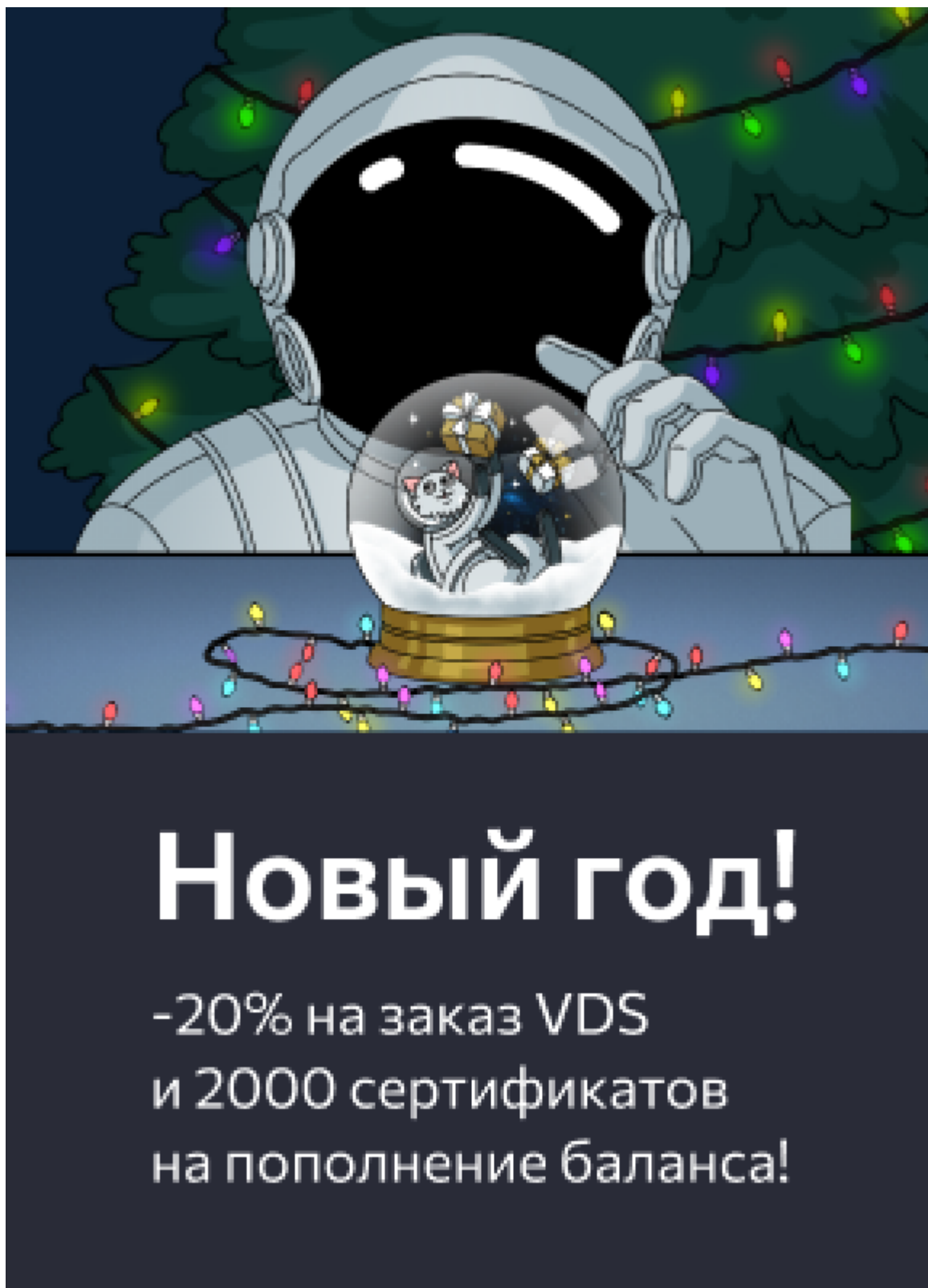
Сайт **vk.com** не позволяет устан

Подробнее

ВИДЖЕТ



ВИДЖЕТ



БЛОГ НА ХАБРЕ

3 янв в 15:16

Beyond Good & Evil: История недооцененного шедевра Ubisoft

 2К  3

29 дек 2023 в 15:10

Горные лыжи, нобелевский лауреат по литературе и прослушка

 1.8K  14

28 дек 2023 в 15:11

Танцуйте дальше: Почему Dance Dance Revolution остается чемпионом среди ритм-игр

 1.5K  4

27 дек 2023 в 18:13

Приземление МКС: проблема планетарного масштаба

 21K  284

26 дек 2023 в 15:11

Простая проблема приводит к слишком большим числам для нашей Вселенной

 25K  13