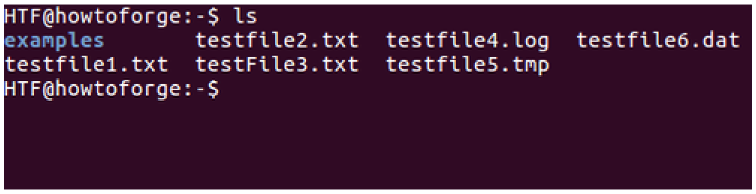
14 Practical Examples of Linux Find Command for Beginners

Find is one of the most frequently used Linux commands, and it offers a plethora of features in the form of command line options. In this tutorial, which is aimed at beginners, we will discuss the basic usage of the command as well as some of the useful command line options it offers.

NOTE: Unless otherwise specified, we will be using the following files for all our find command-related examples in this tutorial.

On this page



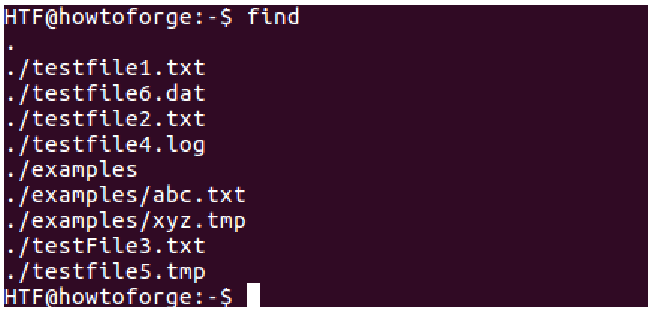
```
HTF@howtoforge:-$ ls
examples       testfile2.txt  testfile4.log  testfile6.dat
testfile1.txt  testFile3.txt  testfile5.tmp
HTF@howtoforge:-$
```

1. How to list all files in current directory and its subdirectories

The find command lets you quickly list all the files in current directory and its subdirectories. For this, all you have to do is to run the command without any arguments or options.

```
find
```

Here is the output in our case:



```
HTF@howtoforge:-$ find
.
./testfile1.txt
./testfile6.dat
./testfile2.txt
./testfile4.log
./examples
./examples/abc.txt
./examples/xyz.tmp
./testFile3.txt
./testfile5.tmp
HTF@howtoforge:-$
```

Of course, you'll have to provide the complete path if the directory whose contents you want to list is not your current directory.

2. How to search file by name

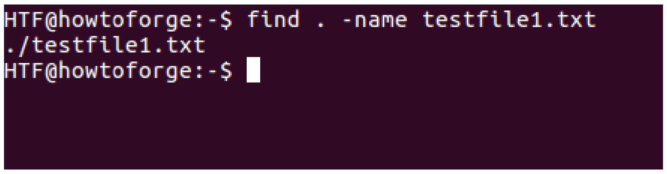
If you want, you can use the find command to search for a specific file by its name. The -name command line option lets you do this. Here's the syntax:

```
find [dir-path] -name [filename]
```

For example, the following command will search the current directory for a file named 'testfile1.txt.'

```
find . -name testfile1.txt
```

Here is the output



```
HTF@howtoforge:-$ find . -name testfile1.txt
./testfile1.txt
HTF@howtoforge:-$
```

Similarly, you can search for the file in another directory. If the directory is a subdirectory of your present working directory, then you don't have to do anything as the find command automatically searches in all sub-directories.

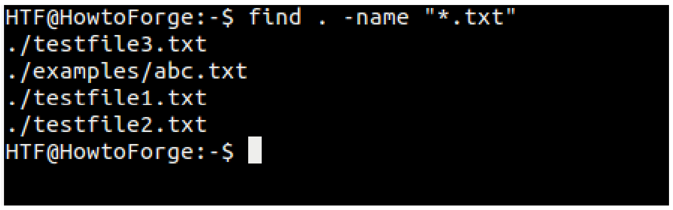
But if it's a separate directory altogether, then you'll have to provide the complete path to it. For example, the following command will search for the file in user's home directory:

```
find /home -name testfile1.txt
```

3. How to search for files of particular type

The find command also lets you search for same type of files in a directory (and its subdirectories). For example, the following command will search for all .txt files in your present working directory.

```
find . -name "*.txt"
```



```
HTF@HowtoForge:-$ find . -name "*.txt"
./testfile3.txt
./examples/abc.txt
./testfile1.txt
./testfile2.txt
HTF@HowtoForge:-$
```

In case you aren't aware, * is a wildcard character. For more information on wild card characters, head here.

4. How to perform case insensitive search

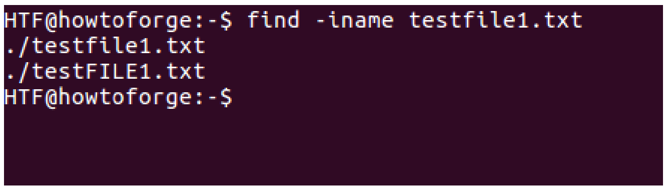
By default, the find command performs a case sensitive search (it treats upper and lower case characters as different). But if you want, you can force the command to carry out case-insensitive search. This you can do by using the -iname command line option.

```
find -iname [filename]
```

For example,

```
find -iname testfile1.txt
```

Here is the output



```
HTF@howtoforge:-$ find -iname testfile1.txt
./testfile1.txt
./testFILE1.txt
HTF@howtoforge:-$
```

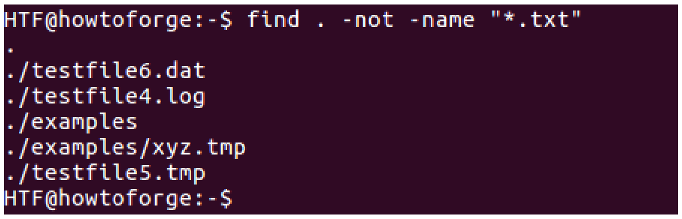
5. How to only display names that don’t match search pattern

If you want, you can even ask the find command to print the file names that do not match the search pattern (also known as invert search). You can access this feature using the ! or -not operator.

For example

```
find . -not -name “*.txt”
```

Here is the output



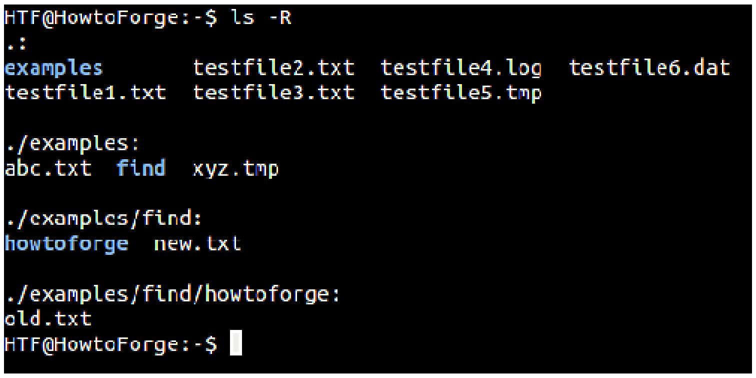
```
HTF@howtoforge:-$ find . -not -name "*.txt"
.
./testfile6.dat
./testfile4.log
./examples
./examples/xyz.tmp
./testfile5.tmp
HTF@howtoforge:-$
```

So as you can see, all files with extension other than .txt are produced in the output.

6. How to limit search to directory level(s)

The find command also allows you to limit search to a particular directory depth. One command line option that lets you do this is -maxdepth.

For example, consider the following directory structure:



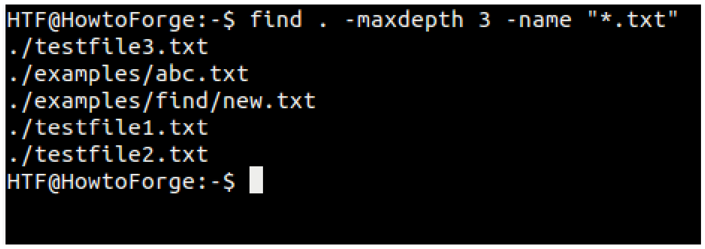
```
HTF@HowtoForge:~$ ls -R
.:
examples       testfile2.txt  testfile4.log  testfile6.dat
testfile1.txt  testfile3.txt  testfile5.tmp

./examples:
abc.txt  find  xyz.tmp

./examples/find:
howtoforge  new.txt

./examples/find/howtoforge:
old.txt
HTF@HowtoForge:~$
```

Now suppose, if you want find to search only up till the 'find' subdirectory (meaning it should ignore the 'howtoforge' sub-directory), then you can use the following command:

```
find . -maxdepth 3 -name "*.txt"
```

Here '-maxdepth 3' forces 'find' to go inside and search only three levels, with the first level being your top level (or the current working) directory.

Here's the output of the command:

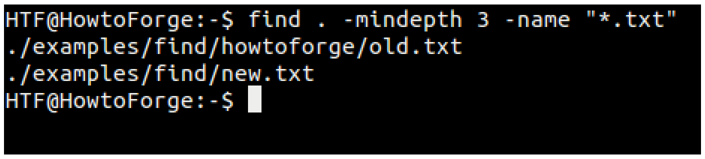


```
HTF@HowtoForge:~$ find . -maxdepth 3 -name "*.txt"
./testfile3.txt
./examples/abc.txt
./examples/find/new.txt
./testfile1.txt
./testfile2.txt
HTF@HowtoForge:~$
```

Like maxdepth, there is another option called mindepth (usage: '-mindepth [N]'). When used, this option forces the find command to go 'N' level down before starting the search operation

For example,

```
find . -mindepth 3 -name "*.txt"
```



```
HTF@HowtoForge:~$ find . -mindepth 3 -name "*.txt"
./examples/find/howtoforge/old.txt
./examples/find/new.txt
HTF@HowtoForge:~$
```

So, only directories 'find' and below are searched.

On similar lines, if you want to search a .txt file in subdirectories that fall between level 2 and 4, then you can use the following command.

```
find -mindepth 2 -maxdepth 4 -name "*.txt"
```

7. Display all the empty files

If you want, you can use the find command to display all the empty files in a particular directory (and its subdirectories). This can be done using the -empty option.

For example, to display all the empty files in your current working directory, use the following command:

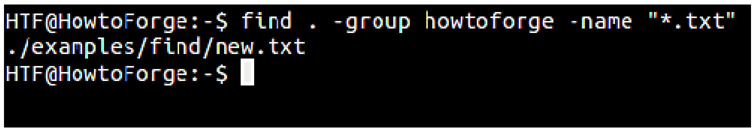
```
$ find . -empty
```

8. How to search files belonging to particular group

The find command also lets you search for files belonging to a particular group - the -group option lets you do this. For example, the following command will list all the files – present in current working directory and its subdirectories - that belong to the 'howtoforge' group.

```
find . -group howtoforge -name "*.txt"
```

Here's the output of the above command in my case:



```
HTF@HowtoForge:~$ find . -group howtoforge -name "*.txt"
./examples/find/new.txt
HTF@HowtoForge:~$ 
```

9. How to search files owned by a particular user

The find command also lets you search for files based on ownership - the -user option lets you do this. For example, the following command will display all .txt files (present in the current directory) owned by user 'himanshu':

```
find . -user himanshu -name "*.txt"
```

10. How to search recently modified files

The find command allows you to search recently-modified files. This can be done using the -mmin option. The option requires you to pass a number which will be treated as number of minutes.

So for example, if you want to search for .txt files (in your current directory) whose data was modified 1 minute ago, you can use the following command:

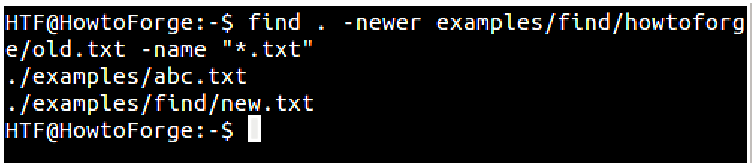
```
find . -mmin 1 -name "*.txt"
```

11. How to search for files that were modified more recently than a file

Yes, the find command even lets you search for files that were modified more recently that a particular file. This feature can be accessed using the -newer option that requires a file name (against which you want to compare) to be passed to it.

Here's an example:

```
find . -newer ./examples/find/howtoforge/old.txt -name "*.txt"
```



```
HTF@HowtoForge:~$ find . -newer examples/find/howtoforg
e/old.txt -name "*.txt"
./examples/abc.txt
./examples/find/new.txt
HTF@HowtoForge:~$ 
```

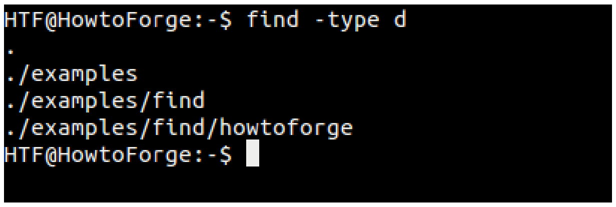
12. How to display only directory names in output

There's also a command line option that enables the find command to only display directory names in the output. The option in question is -type and you need to pass d as a value to it.

For example:

```
find -type d
```

Here's the output of the above command in our case:



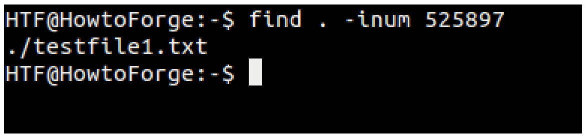
```
HTF@HowtoForge:~$ find -type d
.
./examples
./examples/find
./examples/find/howtoforge
HTF@HowtoForge:~$
```

Aside from d, there are several other letters that can be passed as value to -type option. To know about them, head to the find command's man page.

13. How to search files based on their inode numbers

You can also pass an inode number to the find command and ask it to find the corresponding file name (if any). This feature can be accessed through the -inum option which requires an inode number as a value.

Following is an example:



```
HTF@HowtoForge:~$ find . -inum 525897
./testfile1.txt
HTF@HowtoForge:~$
```

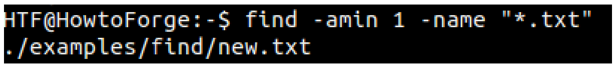
Tip: You can find a file's inode number using the 'ls -li' command.

14. How to search files based on their last access time

The find command also lets you search for files based on when they were last accessed – you can ask the tool to display file(s) that were last accessed 'n' minutes ago. This feature can be accessed using the -amin command line option.

For example, the following command searches for .txt files in present directory that were accessed 1 min ago:

```
find -amin 1 -name "*.txt"
```



```
HTF@HowtoForge:~$ find -amin 1 -name "*.txt"
./examples/find/new.txt
```

Conclusion

The examples mentioned in this tutorial should give you a good idea about how the find command works on a basic level, as well as about some of the useful command line options it offers. Do try these examples on your system, and also go through the tool's man page.

view as pdf | print

Share this page: Tweet Follow @howtoforgecom 27.8K followers Recommend 37 G+1 7

Suggested articles

2 Comment(s)

Add comment

Name *

Email *

p

I'm not a robot
reCAPTCHA
Privacy - Terms

Submit comment

Comments

From: Gustavo Takeshi **at:** *2017-03-22 16:34:14* Reply

Hello all,Very good content!
I think it could be interesting to add the following command,It is useful when you know the text but don't know the file you stored it.
grep -iR "text you want to find inside file"
Regards, Gustavo.

From: MaT **at:** *2017-03-25 09:11:22* Reply

This grep example is nice, but the article itself is about the program called find. So the logic behind your comment is like: "Yes, you wrote great article about how to prepare sushi, but look, this recipe for fish and chips is also great and you did not include it in your article..." :-) Ok, maybe next time they will publish another article about grep (maybe they already did) and thent I can imagine that there could be a link at the and of the article: "And sometimes when the find program is not sufficient for your needs, you can try to use another program called grep (link there)."

Tutorials **14 Practical Examples of Linux Find Command for Beginners**

Sign up now!

Tutorial Info

Author:	Himanshu Arora
Published:	Mar 22, 2017
Tags:	centos, debian, linux, opensuse, shell, ubuntu

Share This Page

Tweet Follow 27.8K followers

Recommend 37

G+1 7