

How to Find a File in Linux Using the Command Line

The 'find' command offers powerful options to hone your search

by [Juergen Haas](#)

157

Updated August 13, 2019

Despite the popularity of window managers that offer graphical user interfaces, the best way to search for files in Linux still requires a shell. The **find** command, with its myriad options and switches, offers the most powerful and precise features to surface what you're looking for.

Note: All modern Linux distributions support **find** from the shell. Access the shell (sometimes called the *terminal window*) by clicking the relevant icon or pressing **Ctrl+Alt+T**.

The 'find' Command

The command used to search for files is called **find**.

The basic syntax of the **find** command is:

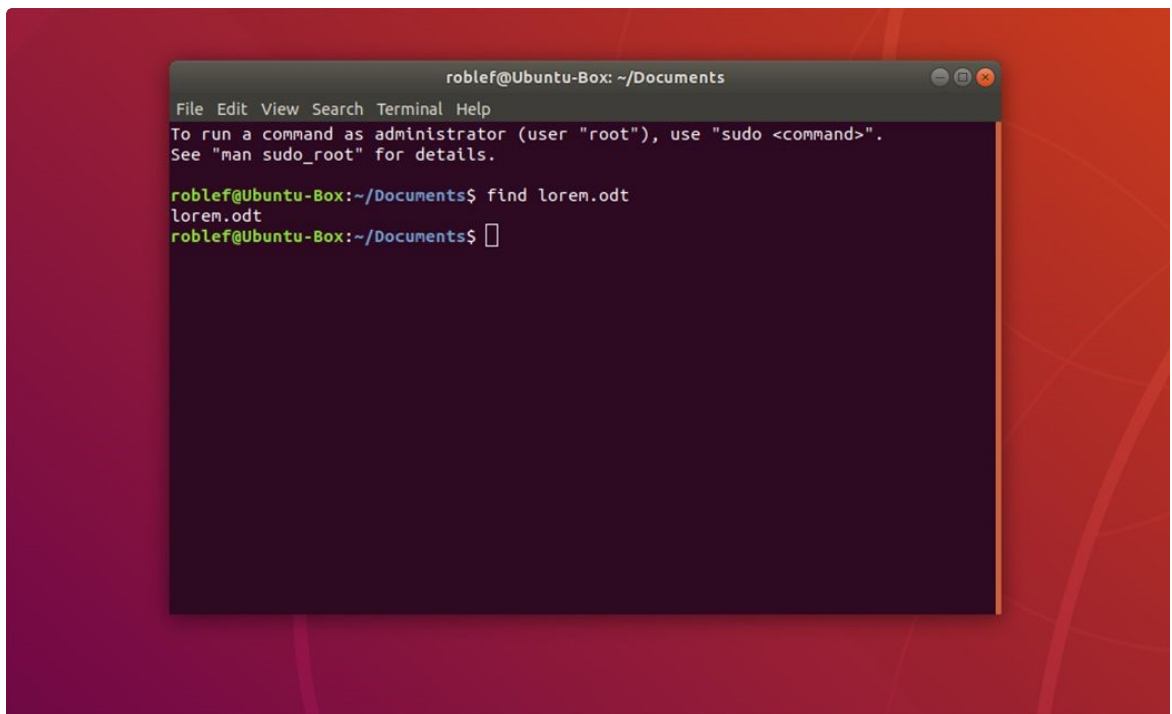
```
find filename
```

The currently active path marks the search location, by default. To start searching the whole drive you would type the following:

```
find / filename
```

If, however, you want to start searching for the folder you are currently in then you can use the following syntax:

```
find . filename
```



When you search by name across the whole drive, use the following syntax:

```
find / -name filename
```

The first part of the find command is the **find** command itself.

The second part is where to start searching from.

The next part is an expression which determines what to find.

Finally the last part is the name of the thing to find.

Search Location Shortcuts

The first argument after **find** is the location you wish to search. Although you may specify a specific directory, most people use a [metacharacter](#) to serve as a substitute. The three metacharacters that work with this command include:

Period: specifies the current and all nested folders

Forward Slash: specifies the entire filesystem

Tilde: specifies the active user's home directory

Tip: Searching the entire filesystem is likely to generate a lot of access-denied errors. Run the command with elevated privileges (e.g., by using **sudo**), if you need to search in places your standard account normally cannot access.

Expressions

The most common expression you will use is **-name**. The **-name** expression lets you search for the name of a file or folder.

There are, however, other expressions you can use:

- amin n**: The file was last accessed n minutes ago
- anewer**: The file was last accessed more recently than it was modified
- atime n**: The file was last accessed more n days ago
- cmin n**: The file was last changed n minutes ago
- cnewer**: The file was last changed more recently than the file was modified
- ctime n**: The file was last changed more than n days ago
- empty**: The file is empty
- executable**: The file is executable
- false**: Always false
- fstype type**: The file is on the specified file system
- gid n**: The file belongs to group with the ID n
- group groupname**: The file belongs to the named group
- ilname pattern**: Search for a symbolic link but ignore case
- iname pattern**: Search for a file but ignore case
- inum n**: Search for a file with the specified node
- ipath path**: Search for a path but ignore case
- iregex expression**: Search for a expression but ignore case
- links n**: Search for a file with the specified number of links
- lname name**: Search for a symbolic link
- mmin n**: File's data was last modified n minutes ago
- mtime n**: File's data was last modified n days ago
- name name**: Search for a file with the specified name
- newer name**: Search for a file edited more recently than the file given
- nogroup**: Search for a file with no group id
- nouser**: Search for a file with no user attached to it
- path path**: Search for a path

- readable:** Find files which are readable
- regex pattern:** Search for files matching a regular expression
- type type:** Search for a particular type
- uid uid:** Files numeric user id is the same as uid
- user name:** File is owned by user specified
- writable:** Search for files that can be written to

Example Usage of the Find Command

How to Find Files Accessed More Than a Certain Number of Days Ago

To find all the files within your home folder accessed more than 100 days ago:

```
find ~ -atime 100
```

How to Find Empty Files and Folders

To find all the empty files and folders in your system:

```
find / -empty
```

How to Find All of the Executable Files

To find all of the executable files on your computer:

```
find / -exec
```

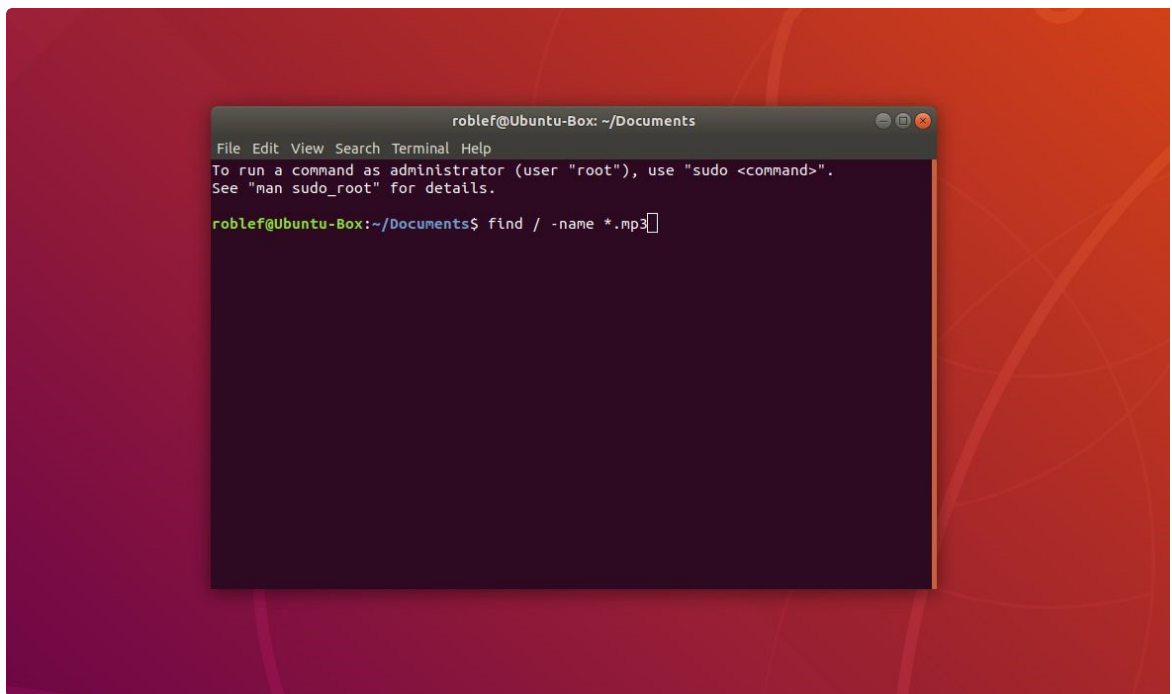
How to Find All of the Readable Files

To find all of the files that are readable:

```
find / -read
```

Patterns

When you search for a file you can use a pattern. For example, search for all files with the extension [mp3](#):



```
find / -name *.mp3
```

How to Send Output from the Find Command to a File

The main problem with the find command is that it can sometimes return too many results to look at in one go. Pipe the output to the [tail command](#) or you can output the lines to a file as follows:

```
find / -name *.mp3 -fprint nameoffiletoprintto
```

How to Find and Execute a Command Against a File

To search for and edit a file at the same time:

```
find / -name filename -exec nano '{}' \;
```

The above command searches for a file called filename and then runs the nano editor for the file that it finds.