**Find Files in Linux Using the Command Line**

This credit will be applied to any valid services used during your first 60 days.

Top of Form

Sign Up

Bottom of Form



When you have to find a file in Linux, it’s sometimes not as easy as finding a file in another operating system. This is especially true if you are running Linux without a graphical user interface and need to rely on the command line. This article covers the basics of how to find a file in Linux using the CLI. The find command in Linux is used to find a file (or files) by recursively filtering objects in the file system based on a simple conditional mechanism. You can use the find command to search for a file or directory on your file system. By using the -exec flag (find -exec), matches, which can be files, directories, symbolic links, system devices, etc., can be found and immediately [processed within the same command](https://www.linode.com/docs/guides/find-files-in-linux-using-the-command-line/#how-to-find-and-process-a-file-in-linux).

**Find a File in Linux by Name or Extension**

Use find from the command line to locate a specific file by name or extension. The following example searches for \*.err files in the /home/username/ directory and all sub-directories:

find /home/username/ -name "\*.err"

**Using Common find Commands and Syntax to Find a File in Linux**

find expressions take the following form:

find options starting/path expression

* The options attribute will control the find process’s behavior and optimization method.
* The starting/path attribute will define the top-level directory where find begins filtering.
* The expression attribute controls the tests that search the directory hierarchy to produce output.

Consider the following example command:

find -O3 -L /var/www/ -name "\*.html"

This command enables the maximum optimization level (-O3) and allows find to follow symbolic links (-L). find searches the entire directory tree beneath /var/www/ for files that end with .html.

**Options and Optimization for find**

The default configuration for find will ignore symbolic links (shortcut files). If you want find to follow and return symbolic links, you can add the -L option to the command, as shown in the example above.

find optimizes its filtering strategy to increase performance. Three user-selectable optimization levels are specified as -O1, -O2, and -O3. The -O1 optimization is the default and forces find to filter based on filename before running all other tests.

Optimization at the -O2 level prioritizes file name filters, as in -O1, and then runs all file-type filtering before proceeding with other more resource-intensive conditions. Level -O3 optimization allows find to perform the most severe optimization and reorders all tests based on their relative expense and the likelihood of their success.

| **Command** | **Description** |
| --- | --- |
| -O1 | (Default) filter based on file name first. |
| -O2 | File name first, then file type. |
| -O3 | Allow find to automatically re-order the search based on efficient use of resources and likelihood of success. |
| -maxdepth X | Search current directory as well as all sub-directories X levels deep. |
| -iname | Search without regard for text case. |
| -not | Return only results that do not match the test case. |
| -type f | Search for files. |
| -type d | Search for directories. |

**Find a File in Linux by Modification Time**

The find command contains the ability to filter a directory hierarchy based on when the file was last modified:

find / -name "\*conf" -mtime -7

find /home/exampleuser/ -name "\*conf" -mtime -3

The first command returns a list of all files in the entire file system that end with the characters conf and modified in the last seven days. The second command filters exampleuser user’s home directory for files with names that end with the characters conf and modified in the previous three days.

**Use grep to Find a File in Linux Based on Content**

The find command can only filter the directory hierarchy based on a file’s name and metadata. If you need to search based on the file’s content, use a tool like [grep](https://www.linode.com/docs/tools-reference/search-and-filter-text-with-grep). Consider the following example:

find . -type f -exec grep "example" '{}' \; -print

This searches every object in the current directory hierarchy (.) that is a file (-type f) and then runs the command grep "example" for every file that satisfies the conditions. The files that match are printed on the screen (-print). The curly braces ({}) are a placeholder for the find match results. The {} are enclosed in single quotes (') to avoid handing grep a malformed file name. The -exec command is terminated with a semicolon (;), which should be escaped (\;) to avoid interpretation by the shell.

**How to Find and Process a File in Linux**

The -exec option runs commands against every object that matches the find expression. Consider the following example:

find . -name "rc.conf" -exec chmod o+r '{}' \;

This filters every object in the current hierarchy (.) for files named rc.conf and runs the chmod o+r command to modify the find results’ file permissions.

The commands run with the -exec are executed in the find process’s root directory. Use -execdir to perform the specified command in the directory where the match resides. This may alleviate security concerns and produce a more desirable performance for some operations.

The -exec or -execdir options run without further prompts. If you prefer to be prompted before action is taken, replace -exec with -ok or -execdir with -okdir.

**How to Find and Delete a File in Linux**

**Caution**

Be very careful using this.

To delete the files that end up matching your search, you can add -delete at the end of the expression. Do this only when you are positive the results will only match the files you wish to delete.

In the following example, find locates all files in the hierarchy starting at the current directory and fully recursing into the directory tree. In this example, find will delete all files that end with the characters .err:

find . -name "\*.bak" -delete

**More Information**

You may wish to consult the following resources for additional information on this topic. While these are provided in the hope that they will be useful, please note that we cannot vouch for the accuracy or timeliness of externally hosted materials.