LINUX FOUNDATION

TRAINING

Finding Files on the Filesystem

A Linux Foundation Training Publication
www.training.linuxfoundation.org

Overview

In Linux there is a saying “everything is a file”, which could be more accurately stated “everything is accessible in the filesystem”. The Linux command line comes equipped with a way of recursively searching the filesystem for files and directories based on specified criteria in the find command.

Key Ideas

find: The find command accepts search criteria and returns results matching it. Generally takes the form: “find /path/to/search/directory criteria”

wildcard: Wildcards are used to represent a set of possible values. For example, the “*” wildcard means “any text”. For example, “*.txt” means all text files. “*dude*” means “any text containing the word dude”.

recursion: By default, the find command searches recursively, meaning it searches the specified directory, and any directory within it, to an infinite depth of directories.

change time (ctime): The last time changes were made to a file’s metadata (owner, permissions, etc)

access time (atime): The last time a file was accessed, read from, or written to.

modified time (mtime): The last time the contents of a file were changed.

Example Scenario

The following series of examples shows different scenarios where you can use the find command.

Now Do It

1. List all files in the current directory.

2. Find all directories called “tmp”.

3. Find all filenames containing the text “ssh” in any case, and do not include directories in the results.

4. Find all filenames containing the text “ssh” in any case, and do not include directories in the results, or any .gz files, or files that contain “sshd” in their names.

5. Find filenames containing the text “ssh” in any case, that are either .gz files or contain “sshd” and do not include directories in the results. In other words, the

files that were excluded in the previous search.

6. Find all files not owned by the root user.
7. Find all files that have changed in the last two days
8. Find all files that are larger than 1M but smaller that 3M.

If you remember nothing else...

By default, arguments are combined additively: results match all of the criteria. You can specify alternative matching (OR) using the -o argument.

Answer Key

1. In this case, “.” means the current directory, and since no criteria is specified, all contents of the current directory are displayed.

```
# find .
```

2. In this case, you’ve specified that you want to start your search in the root root directory (“/”) which means include the whole filesystem. The “-type d” argument specifies you are only interested in directories, and the “-name” argument specifies that they should be called “tmp”.

```
# find / -type d -name “tmp”
```

3. In this case, you’ve used the “-type f” argument to specify that you’re looking for files, and the “-iname” argument to specify that the results should be case insensitive and the “*ssh*” specifies that any file with ssh in its name, regardless of where in the filename, should be returned.

```
# find /  -type f -iname *ssh*
```

4. This is the same search as before, but this time there are additional criteria added. The “!” is the mathematical representation of “not”. Note that find has combined these criteria additively, so that only results that match all of the criteria are returned.

```
   # find /  -type f -iname *ssh* ! -name “*.gz” ! -name “sshd*”
```

5. By default, find combines arguments additively. The “-o” argument specifies “or”: the argument following it should considered as an alternatives to the argument before it. In this case, you are saying return the files with names including the characters ssh and .gz or sshd.

```
   # find /  -type f -iname *ssh*  -name “*.gz” -o -name “sshd*
```

6. If there are no other users on the system, this command will not return any results.

```
# find / ! -user root -type f
```

7. The “ctime -2” argument specifies within the last two days. If instead “2” was given, without the “-”, the command would return files that had changed 2 days ago. If a “+” was given instead of the “-”, files that had changed before two days ago would be returned. Similar searches can be constructed using mtime and atime. To use minutes instead of days as the denomination, use “min” instead of “time” (i.e. cmin)

```
   # find / -ctime -2
```

8. In this case, the “+” means “greater than” and the “-” means “less than”.

9. # find / -size +1M -size -3M

LINUX FOUNDATION

Get Certified!

Get more information at http://training.linuxfoundation.org/certification/lfcs