

Finding Files

If you ever need to locate a file or directory you can use the `find` command. It can be used to find files by name, size, permissions, owner, modification time, and more.

`find [path...] [expression]` - Recursively finds files in path that match expression. If no arguments are supplied it find all files in the current directory.

```
$ find
.
./.profile
./.bash_history
./PerformanceReviews
./PerformanceReviews/Fred
./PerformanceReviews/current
./PerformanceReviews/current/tps-report-violations.log
./PerformanceReviews/John
./sales.data
...
```

Here are some useful ways in which to use the `find` command.

`find . -name pattern` - Displays files whose name matches pattern. This is case sensitive.

`find . -iname pattern` - Same as `-name`, but ignores case.

`find . -ls` - Performs an ls on each of the found files or directories.

`find . -mtime num_days` - Finds files that are num_days old.

`find . -size num` - Finds file that are of size num.

`find . -newer file` - Finds files that are newer than file.

`find . -exec command {} \;` - Run command against all the files that are found.

Let's look at some examples. Let's say you are looking for a file or directory named "apache." You think it is in /opt somewhere and are not quite sure if it is "Apache" or "apache." You could provide `find` with the path of /opt, use `-iname` to ignore case, and look for "apache."

```
$ find /opt -iname apache
/opt/web/Apache
```

To find all the files in /usr/local that end in "conf", you can use this command.

```
$ find /usr/local -name *.conf
/usr/local/etc/dhcpd.conf
/usr/local/etc/httpd.conf
```

If you are looking for files that are more than 10 days old, but less than 13 days old in the current directory you can use this command.

```
$ find . -mtime +10 -mtime -13
./.profile
./PerformanceReviews
./PerformanceReviews/John
./tpsreports
```

```
./tpsreports/coversheet.doc
```

Find files that start with an "s" and perform an `ls` on them.

```
$ find . -name "s*" -ls
52  11 -rw-r--r--  1 bob users 10400 Sep 27 08:52 ./sales.data
48   1 -rw-r--r--  1 bob users   35 Sep 27 08:47 ./tpsreports/sr.txt
53 112 -rw-r--r--  1 bob sales  2566 Sep 27 08:54 ./sales-lecture.mp3
```

The `-size` argument to `find` takes a number followed by a letter that represents the unit of space. Valid options are:

c for bytes

k for Kilobytes (units of 1024 bytes)

M for Megabytes (units of 1048576 bytes)

G for Gigabytes (units of 1073741824 bytes)

Here is an example of how to find files that are larger than 300 megabytes.

```
$ find . -size +300M
./PerformanceReviews/current/tps-report-violations.log
```

Here is how to find directories that are newer than a given file. In this case you are looking for directories that are newer than "b.txt."

```
$ find . -type d -newer b.txt
.
./PerformanceReviews
./PerformanceReviews/current
./tpsreports
```

On some occasions you may want to run a command against a list of files. You can use the `find` command with the `-exec` option to do this sort of thing. Use a pair of braces (`{}`) to act as a placeholder for the current file being processed. The command is terminated with the semicolon (`;`) character. You need to either escape or quote the semicolon like this `';'` or like this `\;` . If you want to run the command `file FILE_NAME` on every file in the current directory you would use the following command.

```
$ find . -exec file {} \;
.: directory
./profile: ASCII text
./bash_history: ASCII text
./PerformanceReviews: directory
./PerformanceReviews/Fred: directory
./PerformanceReviews/current: directory
./PerformanceReviews/current/tps-report-violations.log: ASCII text
./PerformanceReviews/John: empty
./sales.data: data
```

As you can see `find` is a really powerful tool and it has even more features than you have seen so far. Take a look at the man page or refer to the links at this end of this chapter.

Locate - A fast find

Every time you run the `find` command it evaluates each file and returns the appropriate response. This can be a slow process at times. For instance, if you are looking for a file on the system and cannot narrow its location down to a subdirectory you would run `find / -name something`. That command looks at each and every file on the system. If you know the file's name or at least part of its name and just want to know where it resides, the `locate` command is the right tool for that job.

`locate pattern` - List files that match pattern.

Once a day all the files on the system are indexed by a process called `updatedb`. When you run `locate` it is simply querying the index or database created by `updatedb` and not looking at each file on the system. This is really, really fast. The down side is that the data is not in real time. If you are trying to find a file you created just a few minutes ago, chances are it is not yet indexed and `locate` will not find it. Also, `locate` can potentially return a file that matches your search, but the file may have removed from the system since the index was last updated. On some servers `locate` is not installed or enabled, so your only choice may be to use `find`.

Here is what it looks like when `locate` is disabled.

```
$ locate bob
locate: /var/locatedb: No such file or directory
```

If it is enabled you will get a quick response to your queries. Notice that you do not need to know the entire file name, just a portion works.

```
$ locate tpsrep
/home/bob/tpsreports
/home/bob/tpsreports/coversheet.doc
/home/bob/tpsreports/sales-report.txt
```

Deep Dive

- [Find](#) - Ubuntu documentation on the find command.
- [Locate](#) - An article on the locate command.
- [The /etc/passwd file](#) - An article on the `/etc/passwd` file.

<http://www.LinuxTrainingAcademy.com>