Finding Files: find and locate

The find command line utility provides an extremely powerful and flexible method for locating files based on their properties, including name. It does not search the interior of files for patterns, etc.; that is more the province of **grep** and its variations.

The general form of a **find** command is:

```
$ find [location] [criteria] [actions]
```

where there are three classes of arguments, each of which may be omitted. If no location is given, the current directory **(.)** is assumed; if no criteria are given, all files are displayed; and, if no actions are given, only a listing of the names is given.

There are many logical expressions which can be used for criteria. For example, the command:

```
$ find /etc -name "*.conf"
```

will print out the names of all files in the **/etc** directory and its descendants, recursively, that end in **.conf**. To specify a simple action request:

```
$ find /etc -name "*.conf" -ls
```

will print out a long listing, not just the names.

A little more complicated example is the following:

```
$ find /tmp /etc -name "*.conf" -or -newer /tmp/.X0-lock -ls
```

will look in subdirectories under **/etc** and **/tmp** for files whose names end in **.conf**, or are newer than **/tmp/.Xo-lock** and print out a long listing.

You can perform actions with the **-exec** option, as in:

```
$ find . -name "*~" -exec rm {} ';'
```

where **{}** is a fill in for the files to be operated on, and **';'** indicates the end of the command. This can be unwieldy and one often pipes into the **xargs** program, as in:

```
$ find . -name "*~" | xargs rm
```

which accomplishes the same action. A third way to do the same action would be:

```
$ for names in $(find . -name "*~" ) ; do rm $names ; done
```

If a filename has a blank space in it (or some other special characters), some of the previous commands will fail.

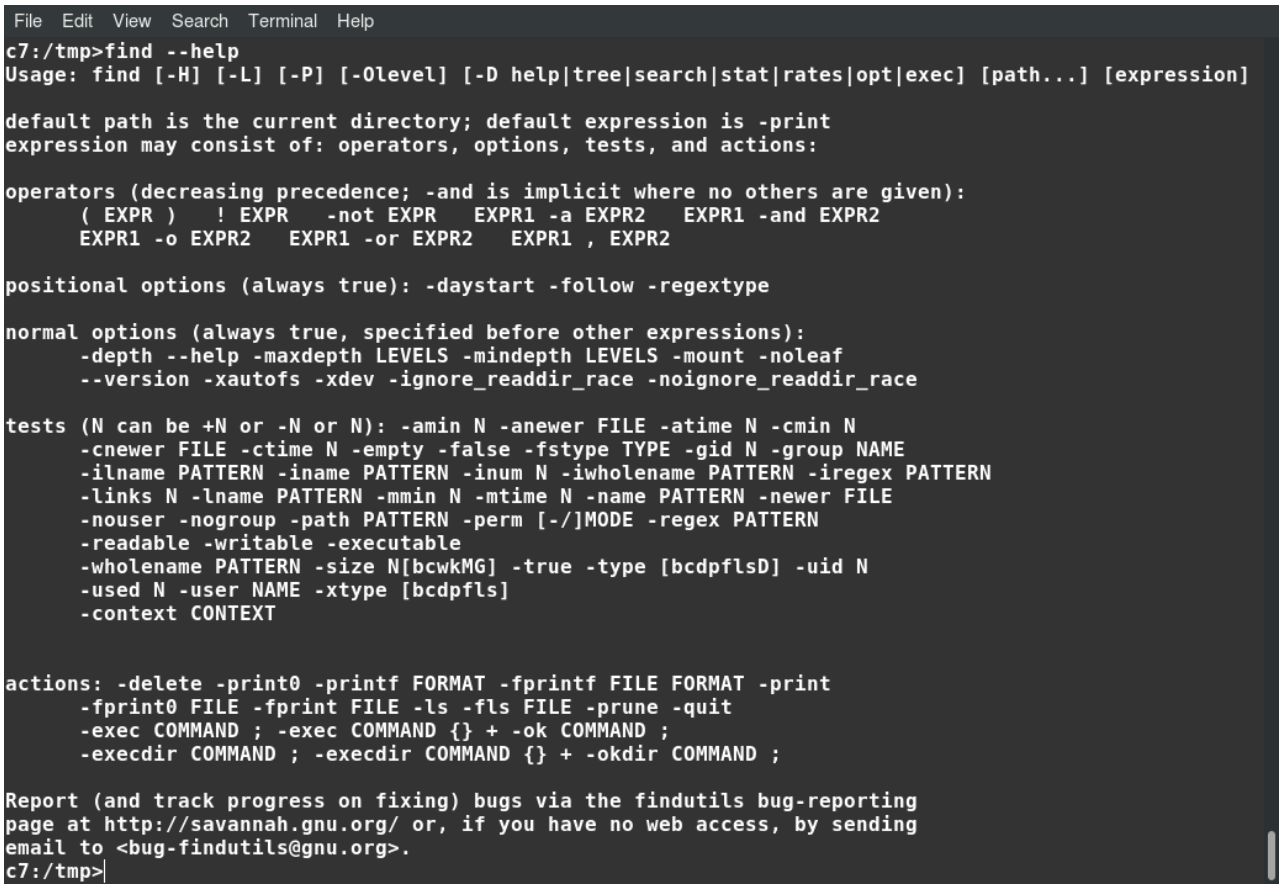
It is generally a disfavored practice to utilize such file names in UNIX-like operating systems, but it is not uncommon for such files to exist, either in files brought in from other systems, or from applications which are also used in other systems.

In such a case, the following variant will work just fine:

```
$ find . -name "*~" -print0 | xargs -0 rm
```

as will the command that uses **-exec rm {} ';'**.

There are many options to **find**, especially regarding selection of files to display. This can be done based on size, time of creation or access, type of file, owner, etc. A quick synopsis is provided by **find --help**:



Another method of locating files is provided by the **locate** command, which searches

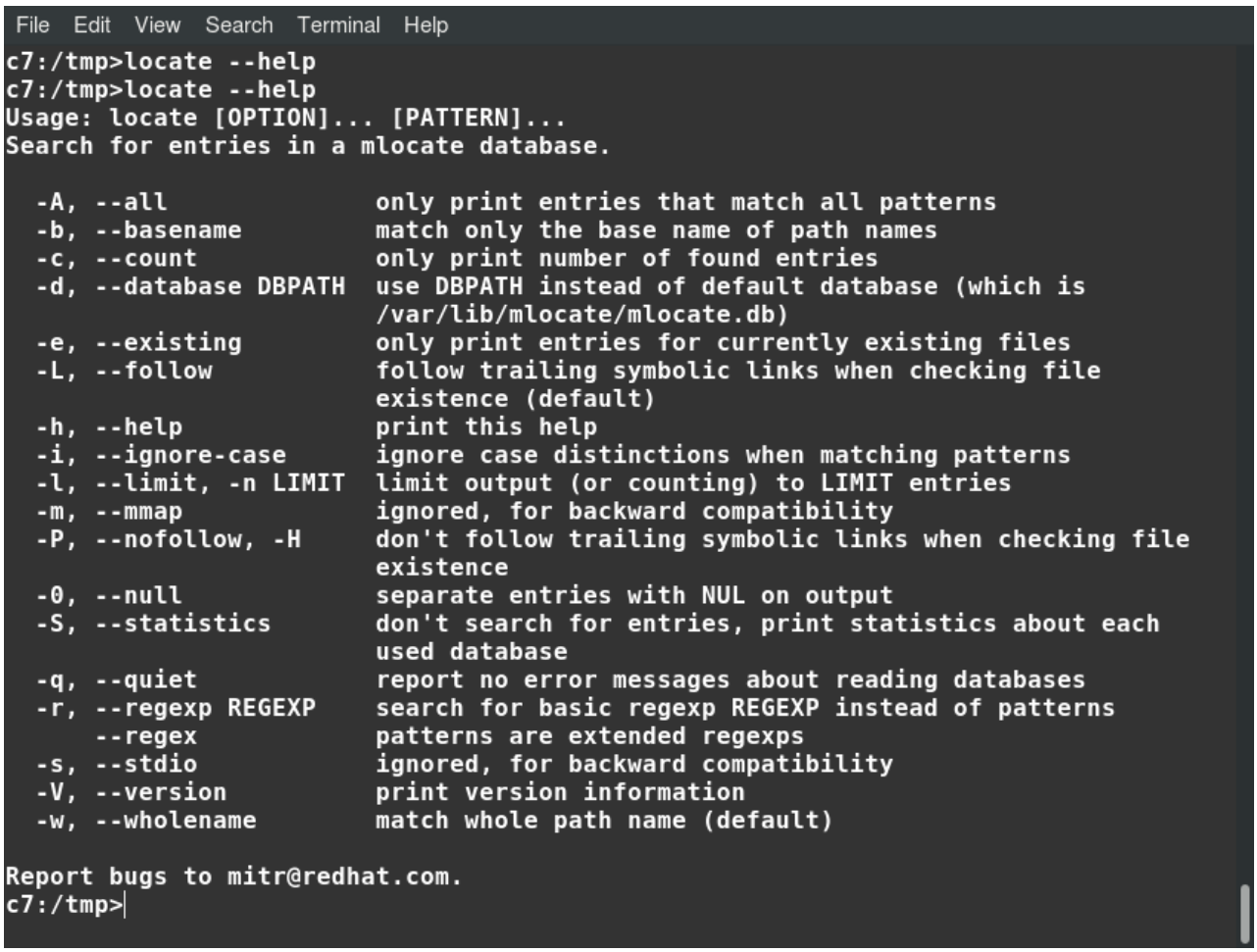
your entire filesystem (except for paths which have been excluded) and works off a database that is updated periodically with **updatedb**. Thus, it is very fast.

Thus, the command:

```
$ locate .conf
```

will find all files on your system that have **.conf** in them.

A quick synopsis is provided by **locate --help**:



locate will only find files that were already in existence the last time the database was updated. On most systems, this is done by a background **cron** job, usually daily. To force an update, you need to do:

```
$ sudo updatedb
```