Locate and Find

COMMAND

COLIN MURPHY

There are many branches in a UNIX-type file system, so it will come as no surprise to you when you realise that you have forgotten on which twiglet you have left a now desperately sought after file.

Even if you have an infallible memory you may still have to deal with an unfamiliar landscape, different distributions have different file layouts, which will only frustrate you when you are trying to find a system file – the XF86Config file is a good example of this. Thankfully, there are a number of tools and utilities that can help you find the files you need.

Most Linux distributions will come with the *locate* utility, which will give you the location of any files that have the text you are looking for in their filenames. So the command:

```
colin@localhost colin]$ locate XF86Config
```

will give me

```
/etc/X11/XF86Config
/etc/X11/XF86Config.old
/etc/X11/XF86Config.test
/usr/X11R6/lib/X11/XF86Config-4.eg
/usr/X11R6/man/man5/XF86Config.5x.bz2
```

showing me all the files that have the text 'XF86Config' in the filename, some of which I really should get around to deleting the next time I do some tidying up! If you have some idea of the filename then *locate* will help you track down its location.

locate relies upon its own database, which, to be of any use, needs to be updated regularly. If you are in the habit of leaving your machine on overnight then this usually will happen automatically when the nightly *crond* jobs are run. If your machine is never on late enough for these jobs to run then you will need to run the *updatedb* command yourself at some point – maybe as you log out, something which could be automated by adding the command to your *~/.bash_logout* file in your home directory. The advantage of having *locate* use a database means queries will be answered very quickly, the disadvantage is the database will be out of date, especially if you have been creating or copying a lot of files in one session, even this can be solved by running *updatedb* from the command line, leaving you with time to go get a coffee.

find is another command line tool that will help you find files, but this time you can search for files

based on other criteria like file size or by the date the file was last modified. This is a feature-rich and powerful command made obvious because the manpage is so big. The basic thing to remember when calling 'find' is that you need to provide the search criteria as well as the search pattern, so the command:

```
[root@localhost /]$ find /var/log/ -name↵
'*.log'
```

will give me

```
/var/log/security.log
/var/log/auth.log
/var.log/user.log
```

amongst many other files. Breaking this down – we will only look in the /var/log directory and any directories below it. We are basing our search criteria on the names of files only and that we only want to know about filenames that end in ".log".

There are many search criteria other than *-name*, all of which are listed in the manpages, here are just some:

- *-mmin -n* allows you to look for files that were modified no less that n minutes ago – useful if you wanted to see which log files had just been written to. +n would allow you to look for files that are older than n minutes.

- *-size +n* will look for files that are bigger than n in 512 byte sized blocks. Put a "c" after the value to search for byte counts or a "k" to search for kilobyte counts,

- *-user name* will allow you to look for files that only belong to the named user.

The search criteria can by combined to make for a more powerful search, so

```
[root@localhost /]$ find -mmin -30 -user colin
```

will only tell me about files that have been modified in the last half an hour and only belong to the user colin. ∎

Crond

Crond is the batch daemon which starts other processes at predetermined times, which are described in the control file etc/crontab.

Bash logout

#/.bash_logout is one of the bash shell command files, this one being run when you exit from a bash shell session, usually when you are shutting the machine down to turn it off.