Advanced Shell Commands

Jeremy Sanders

October 2011

1. a2ps - Convert text files (or programs) to PostScript for printing. By default two columns of text are put onto a single side. mpage can put more than one page of PostScript into an output PostScript file.

   ```
   a2ps prog.f | lp -d ewps     # send output to ewps
   a2ps -1 prog.f | lp -d lp2   # one column
   ```

2. apropos - Print Unix commands which have a certain word in their summary in the man pages.

   ```
   > apropos fortran | more
   g77                 (1)  - GNU project Fortran Compiler (v0.5.24)
   ```

3. at - Run a command at a particular time

   ```
   > at 19:20 01.02.03
   mycommand
   [CTRL+D]
   ```

   Example runs mycommand at 7:20pm on 1st February 2003. Other time formats possible - see man at. Use atq to show future jobs and atrm to delete jobs.

4. awk - Run an awk script on a file or the input.

   ```
   # print first column of test.txt
   awk '{print $1}' test.txt
   # sort test.txt, then print out 1st col
   sort test.txt | awk '{print $1}'
   # print column 1 and sum of cols 2 and 3
   awk '{print $1, $2+$3}' test.txt
   # print sum of column 1
   awk 'BEGIN {sum=0} {sum=sum+$1} END {print sum}' test.txt
   ```

5. bzip2 - compress or decompress bz2 compressed data. This usually compresses data better than gzip, but is much slower.

   ```
   > bzip2  foo.tar           # compress tar file
   > bunzip2 foo.tar.bz2      # decompress tar.bz2 file
   > gtar cjf foo.tar.bz2 foo # compress directory to bz2 tar file
   ```

6. chmod - Modify the attributes of a file or directory. This lets you decide whether others can look at the file or whether a program is an executable.

```
chmod og-r file  # others and group can’t read file
chmod og+r file  # others and group can read file
chmod +x file    # allow file to be an executable
chmod -w file    # prevent easy writing to file
chmod -R og-r dir  # modify all files in dir not to be readable
                   # by group or others
```

7. `cmp` - Are two files the same? diff is probably more useful for text files

   ```
   > cmp home.ps home.pdf
   home.ps home.pdf differ: byte 2, line 1
   ```

8. `convert` - Convert between different image formats (ImageMagick)

   ```
   > convert file.jpg file.png
   ```

9. `cron` - Run a command at specified time intervals (e.g. daily, monthly, biannually). Complex to explain - see `man cron`.

10. `diff` - Intelligently look for differences in text files. If there are no differences, nothing is printed out, otherwise lines which only occur in the first are shown with “<” and lines which only occur in the second are shown with “>”. The number before tells you on which line the difference was. diff can be used to create “patch files” - patches which can be applied to an existing file to convert it to a new version of it.

    ```
    > diff paper1.tex paper2.tex
    1c1
    < \documentclass[a4paper,12pt]{article}
    ---
    > \documentclass[a4paper]{article}

    # unified diff format is more robust
    > diff -u paper-old.tex paper.tex
    ...
    -  extract files from that archive.
    +  extract files from that archive (same as gtar on Linux).
    ...
    ```

11. `ds9` - FITS image viewer (very useful).

12. `du` - Shows how much space is used by current and lower directories.

    ```
    > du -k   # show space used in kilobytes
    12      ./testdir
    20      ./frog
    352     .
    > du -h   # show space used in a human-readable form
    ```

13. `egrep` - Extended grep with more support for regular expressions. e.g. look for frog or toad in lines in files.

    ```
    egrep "(frog|toad)" file1 file2...
    ```

14. `find` - Find files according to name and other attributes

```
> find                # get a list of files in this directory and
                      # subdirectories
> find | wc -l        # count all files
> find -name "*.txt"  # get a list of all files called *.txt
> find /home/foo -name "*.gz"   # look in specific directory
> find -newer foo.txt  # find files modified more recently than foo.txt
> find -type d        # find all directories
> find -type f        # find all regular files
> find -size +2M      # find files at least 2MB in size
```

15. `finger` - Examine a user (or users) on this or another system.

```
> finger
Login     Name             Tty      Idle  Login Time   Office
jss       Jeremy Sanders   pts/0      17  Sep 14 10:48 (:0)
freddy    Freddy Starr     pts/1          Sep 14 14:24 (:0)

> finger -l jss
Login: jss                              Name: Jeremy Sanders
Directory: /home/jss                    Shell: /bin/bash
On since Sat Sep 14 10:48 (BST) on pts/0 from :0
   17 minutes 59 seconds idle
On since Sat Sep 14 14:24 (BST) on pts/1 from :0
Mail last read Wed Aug 14 22:26 2002 (BST)
No Plan.

> finger -l jss@cass32.ast.cam.ac.uk
...
```

16. `ftp` - Transfer files using the ftp protocol over the internet. This is an old method of transferring files. You shouldn’t use this now (see sftp or scp instead), unless you are getting files using “anonymous ftp”, where you don’t need a password or username (lftp is easier to use in this case).

```
> ftp xalph3.ast.cam.ac.uk
Connected to xalph3.ast.cam.ac.uk (131.111.68.53).
220 xalph6.ast.cam.ac.uk FTP server ready.
Name (xalph3.ast.cam.ac.uk:jss): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:  <<< type email address here
230-
230-
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /pub/jss
250 CWD command successful.
ftp> bin   <<< this makes sure files are transferred correctly
ftp> get myfile.dat
...
```

17. `fv` - FITS viewer. Useful tool to manipulate and view FITS files. Needs XANADU environment set (see Users’ Guide).

18. `gdb` - GNU debugger - a command line debugging tool. Various frontends exist to make debugging easier (e.g. ddd, kdbg).

19. `gimp` - Very good conversion, drawing and image manipulation program. Even reads FITS files! See `http://www.gimp.org/`

20. `head` - Print lines from the start of a file or the input (see also tail).

    ```
    grep fred file.txt | head -100  # print first 100 lines with fred
    head -200 file.txt | more    # more first 200 lines from file.txt
    ```

21. `jpeg2ps` - Convert a jpeg file to an eps file (very good for astro-ph papers, as it makes very small eps files).

22. `lftp` - An easy command-line interface to downloading files with ftp or http.

23. `lftpget` - Download a URL, including web or ftp.

    ```
    > lftpget http:///www-xray.ast.cam.ac.uk/title.jpg # download title.jpg
    > lftpget ftp://ftp.ast.cam.ac.uk/pub/README       # download README
    ```

24. `ln` - Make a hard link or symbolic link to a file.

    A hard link to a file is like another directory entry for a file (you can see these in `ls -l`). The file is deleted when all the directory entries are deleted. Hard links only work within a single file system.

    A symlink is a filename which points to an existing filename. Symlinks are generally more useful unless you know what you are doing.

    ```
    > ln -s myfile symlinkname     # make a symlink with name symlinkname
                                   # point to file myfile
    > ln -s /data/disk1/xx mydir   # make a symlink (mydir) to a directory
    ```

25. `make` - Interpret a Makefile. Makefiles are often provided with programs in order to build them from their source. Makefiles contain rules for making targets according to dependencies. Often you can do things like:

    ```
    > make build
    > make install
    ```

26. `mpage` - A program to take a PostScript file and modify it so that more than one side is put onto a single side. To send a shrunk `paper.ps` to `ewps`:

    ```
    mpage -2 -Pewps paper.ps
    ```

27. `ncftp` - An easy interface to ftp with command-line completion (Solaris only by default). `lftp` is a probably better program.

    ```
    ncftp ftp.some.edu             # do anonymous ftp to ftp.some.edu
    ```

28. `nedit` - Fairly simple graphical text editor.

29. `paste` - Add files together as columns. Useful for merging data.

    ```
    paste file1 file2 > file3
    ```

30. `ping` - See whether a computer is responding on the network. This allows you to check network connections.

    ```
    > ping www.stsci.edu
    www.stsci.edu is alive
    ```

31. `pstoedit` - Make an editable xfig file from a Postscript file. Download from `http://www.pstoedit.net/`

32. `ps2epsi` - Take a single page Postscript file and convert it to an embeddable form (does not always work).

33. `ps2pdf` - Convert a Postscript file to PDF.

```
ps2pdf document.ps                  # make document.pdf
ps2pdf -sPAPERSIZE=a4 document.ps   # use a4 paper output
```

34. `rsync` - Command to copy a set of directories and files easily (ignoring files which already exist in the destination) to other directories or over the internet. It is *extremely* useful for synchronising files on different computers.

```
rsync -a /home/user/fred/ /data/fred/jim/
 # copy contents of /home/user/fred into /data/fred/jim
 # NOTE: slashes are important here after the directories: CARE
rsync --delete -a /home/user/fred/ /data/fred/jim/
 # same as above, but delete files in destination which aren’t
 # in sender: EVEN MORE CARE!
rsync -za -e ssh cass10:/home/user/fred/ /data/fred/jim/
 # copy files from cass10 using ssh to current computer
```

35. `scp` - Copy a file (securely) to/from another computer over the network or internet.

```
# copy file.txt from /home/fred on cass12 to this directory
scp cass12.ast.cam.ac.uk:/home/fred/file.txt .
(type in username and password to do this)

# Recursively (copy subdirectories too) files from /home/jim/dir
# on capc41 (log in as jim) into directory dir54 here
scp -R jim@capc41.ast.cam.ac.uk:/home/jim/dir dir54/

# copy boo.dat here onto /home/username on cass12
scp boo.dat cass12:
```

36. `screen` - A program which allows you to disconnect from terminal sessions and connect back to them later. This is useful to check up on running programs from home or so on, or run programs while you are logged out. See the useful Unix tips guide for more information.

37. `sed` - The stream editor - can replace phrases or characters (or regular expressions).

```
sed "s/hello/hi/g" file1 > file2  # replace hello with hi globally
sed "s/cat[s]*/sheep/g" file # replace cat or cats with sheep
sed "s/\bjames\b/jim/g" file # match james on word boundaries
sed "d/+-/-+/" file1 > file2  # swap + and - signs in file
```

38. `sftp` - Securely transfer files to or from another computer using an ftp-like interface.

```
> sftp username@cass12.ast.cam.ac.uk
[type in password here]
> cd dir1
> get fred.txt    << get the file fred.txt from ˜username/dir1
> put secrets.dat << upload the file secrets.dat there
> bye             << logout
```

39. `sha1sum` - Generate a checksum for a file. If the checksums for two files are the same they are very likely to have the same contents. Useful for ensuring a file is the same on different systems. Often used to make sure the downloaded file is not corrupt or hacked. md5sum is also used (has some security flaws).

```
> sha1sum programming.tex
b43a0697c5c67e808b817e82829f70937889b98d  programming.tex
```

40. `sleep` - Do nothing for specified time (in seconds by default)

41. `sort` - Sort a file. Unfortunately the Solaris version can't sort numbers which have exponentials in them (e.g. 2e10).

```
sort data.txt > out.txt   # sort alphabetically, write out.txt
sort -n data.txt > out.txt    # sort numerically (gen. format)
sort -n -k 2 data.txt > out.txt      # sort num. by 2nd column
sort -r -n data.txt > out.txt # num reverse sort on 1st column
grep 'hello' data.txt | sort | more       # show sorted output
```

42. `spell` - Spell check a file.

43. `strace` - what system calls is a program using? - advanced users only.

44. `tail` - Print lines from the end of a file or the input (see also tail).

```
grep fred file.txt | tail -100  # print first 100 lines with fred
tail -200 file.txt   # print last 200 lines from file.txt
```

45. `tee` - Take piped input, send to a file, and pass it on to the next program in the pipe.

```
sort filename | tee out.txt | more
```

46. `telnet` - Insecure way to log into another computer. Use ssh instead.

47. `time` - Tells you how long a command took to execute

```
> time sleep 2
0.010u 0.000s 0:02.01 0.4%      0+0k 0+0io 156pf+0w
^^^^^^ ^^^^^^ ^^^^^^^ ^^^^
USER   SYS    REAL    CPU%
```

User tells you how long it took in the program on the CPU (0 here, as sleep doesn't do much except wait), in the Kernel (SYS), and real apparent time (REAL).

48. `uniq` - Take a sorted file, and remove identical lines

```
sort test.txt | uniq      # remove identical lines after sorting
sort test.txt | uniq -c   # count identical lines after sorting
```

49. `units` - Do unit conversions (google can also do this)

```
> units
2438 units, 71 prefixes, 32 nonlinear units

You have: 1 kpc
You want: cm
        * 3.0856776e+21
        / 3.2407793e-22
```

50. `vi` - Editor with cult following and strange keys. To get out of it type `:q`
51. `w` - The w command shows you who is on the system and what command they are executing

    ```
    > w
      2:44pm  up  4:38,  2 users,  load average: 0.00, 0.02, 0.06
    USER     TTY      FROM           LOGIN@   IDLE   JCPU   PCPU  WHAT
    jss      pts/0    :0            10:48am 19:23   0.05s  0.05s  bash
    ```

52. `wait` - Wait until background processes have finished in the current shell (useful for scripts)
53. `wc` - Counts the characters, words and lines in a file

    ```
    > wc glossary.tex unix_basic.tex
        284    2612   16201 glossary.tex
        153    1234    7519 unix_basic.tex
        437    3846   23720 total
    > grep fred filename | wc
    ```

54. `which` - Show which program the shell will run if you type the command

    ```
    > which ls
    /usr/bin/ls
    ```

55. `xargs` - Run a command on a set of files input from standard input

    ```
    find -name "*.txt" | xargs rm   # delete all files called *.txt
                                    # does not work if spaces in names
    find -name "*.txt" -print0 | xargs -0 rm       # works with spaces
    ```

56. `xmgrace` - Grace - a plotting package with a GUI. Have a go if you hate gnuplot/sm.
57. `zip / unzip` - Create or unzip a .zip file. Useful for transferring data to and from Windows.

    ```
    > zip file.zip dirname/   # create zip file containing directory
    > unzip -l file.zip       # list contents of zip file
    > unzip file.zip          # unzip zip file in current directory
    ```