Utilities

September 27, 2016

Useful ideas

- Listing files and display text and binary files
- Copy, move, and remove files
- Search, sort, print, compare files
- Using pipes
- Compression and archiving
- Your fellow users
- Communicating with other users

Special characters

- The following lot tend to have meaning to the shell, so be careful when trying to use them:

```
& ; | * ? ’ " ‘ [ ] ( ) $ < > { } # / \ ! ~
```

Quoting special characters

- You can quote special characters in three ways
  - Use the backslash
  - Use single quotes
  - Sometimes use double quotes, but some of the previous lot may still be interpreted

```
$ echo '$PATH'
$PATH
$ echo "$PATH"
/usr/sbin:/usr/bin:/sbin:/bin
$ echo \$PATH
$PATH
```

Very useful utilities

- ls : list directories and files
- rm : remove files and directories
- cat : "catenate" a file
- less, more, pg : page through files
- hostname : display what the kernel thinks of as a "hostname"

ls

- ls has a ton of options; some of the most useful are
  - -b : show non-graphical characters in a useful fashion
  - -a : show all directory entries, including those beginning with a period
  - -d : named directories are not descended into
  - -l : show long information
  - -i : show inode number

ls

- ls options
  - -h : “human” readable; this option has become quite common among utilities
  - -t : order by modification time
  - -r : reverse the ordering
  - -1 : only file per line

cat

- cat has very few options
  - -n : number all output lines
  - -v : show graphical characters

rm

- rm and its most useful options
  - -i : prompt before removal
  - -r : remove recursively
  - -f : don't ask questions and don't make comments (okay, the man page actually phrases this as "ignore nonexistent files and arguments, never prompt")

less (another pager along the lines of more and pg)

- less - while less does indeed have some command line arguments, its most useful keyboard interactions are while it is displaying text
  - SPACE - go forward one page
  - b - go back one page
  - d - go forward half a page
  - u - go back half a page

less, continued

- less
  - CTRL-L : repaint the screen
  - g : go to a particular line
  - /pattern : search for a pattern (use '*' at the beginning of the pattern to search multiple files)
  - :n : go to the next file

Working with files

- Common commands for working with files
  - cp
  - mv
  - lpr
  - grep
  - head
  - tail
  - sort
  - uniq
  - diff
  - file

cp

- cp has several very useful options
  - -a : “archive” mode; attempts to preserve as many attributes as possible in the copy
  - -i : prompt before overwriting
  - -l : just create a new hard link rather copying (rsnapshot uses this to cleverly make its snapshots)
  - -r : make a recursive copy

mv

- mv does not have a lot of very useful options, but -i is an exception:
  - -i : inquire before overwrite

lpr

- lpr has two particularly useful options
  - -P PRINTER : specify a printer
  - -# COUNT : print COUNT copies of a file

a2ps

- ap2s has two very useful options
  - -o FILENAME : send output to a file rather than the default printer
  - -P NAME : send output to a printer NAME rather than the default printer

grep/egrep/fgrep

- grep/egrep/fgrep is a very powerful and useful program; the version fgrep that is quite useful for fixed strings
  - -i : ignore case
  - -v : invert match
  - -c : count matching lines
  - -l : show files with matching lines
  - -h : suppress prefixing of files names
  - -H : print the file name for each match
  - -C LINES : print LINES of context for each match

head

- head prints initial lines
  - -n LINES : print LINES of the file

tail

- tail prints final lines
  - -f : “follow” a file as it changes
  - -n LINES : print LINES of the file

sort

- sort lets you sort a file
  - -u : print only unique lines
  - -b : ignore initial blank lines
  - -n : numerical sort
  - -r : reverse a sort
  - -k : specify a key
  - -t : field separator

uniq

- uniqueness
  - -i : ignore case
  - -u : print only unique lines

diff

- file and directory differences
  - -r : recursive diff

file

- no really useful options

pipes

- use the system call pipe(2), a distinguishing characteristic of Unix

Four more utilities

- echo
- date
- script
- unix2dos

echo

- echo is usually both a binary and a built-in, and what usually distinguishes the two is the -n option
  - -n : do not send an end-of-line

date

- date lets you display the date with a great deal of freedom
  - –iso-8601
  - specification formats

```
$ date --iso-8601
2016-01-28
$ date '+%A'
Tuesday
```

date

- date also lets you change the system date

script

- script lets you record your session (but with lots of control characters. . .)

dos2unix, unix2dos

- Unix and Microsoft Windows use different conventions for end of line. One easy way to convert between the two is with dos2unix and unix2dos. I have never needed any options when doing this.

Compression and archiving

- xz
- unxz
- xzcat
- bzip2
- bzcat
- bunzip2
- gzip
- zcat
- gunzip
- tar
- cpio

xz

- Up and coming format; really nifty how you can just cat two xz files together!

xzcat, unxz

- xzcat : Catenate a .xz file
- unxz : uncompress a .xz file

bzip2

- Pretty much the standard these days. Excellent compression, with plenty of options.
  - -1, -2, -3, . . . : choose compression level (from least to most compressed; also least to most memory consumed during run)
  - –fast : synonym for -1 (uses less memory during run)
  - –best : synonym for -9 (uses more memory during run)

bunzip2 and bzcat

- Reverse bzip2; bzcat by default goes to stdout

gzip, zcat, gunzip

- Older standard, still pretty common

tar

- Originally "tape archive." Allows you to package files together into an archive. Newer versions automatically recognize compression.
- The historical option syntax (no dashes) is a bit tricky to use.
  - c : create a tar file
  - x : extract from a tar file
  - t : print the contents of a tar file
  - p : preserve
  - -C : change directory for this process

User and system information

- top and htop
- w, who, last
- finger
- uptime
- df
- free
- write
- mesg

top and htop

- Show pretty comprehensive information about what is going on
- htop is not as common

w, who, last

- w and who let you see who else is on the system
- last lets you see login sessions, both current and older

```
$ w
 10:52:34 up 40 min,  3 users,  load average:
USER     TTY      FROM              LOGIN@
langley  tty8     :0                10:12
$ who
langley  tty8         2015-09-08 10:12 (:0)
```

uptime

- uptime gives you a very brief description of the system

```
$ uptime
 10:53:20 up 40 min,  3 users,  load average:
```

free

- shows memory utilization for the entire system.
- use the -h option, it's the flexible and readable

```
$ free -h
             total       used       free
Mem:           15G       3.3G        12G
-/+ buffers/cache:       1.8G        13G
Swap:          59G         0B        59G
```

write

- write to another user

mesg

- allows or prevents write messages
- y for allow, n for don't allow