CSE320 System Fundamentals II

TONY MIONE

Topics

Command Options

Environment Variables

More Unix Commands

- Process status and control commands
- File ownership and permissions
- Locating files in a directory tree

Command History

Custom Commands

Acknowledgements

Some slides provided by Dr Yoon Seok Yang

Command Options

Unix options communicate specific customizations to a program or command

Styles

- '-' followed by a single character (usually from a-zA-Z0-9) [old style]
  - Multiple options can be squashed together (i.e.: '-a -e -f' can be specified '-aef')
  - Ordering of options usually doesn't matter unless the option takes an argument
- '--' followed by a string (i.e. --no-uid-translation) [newer style (past 20 year)]
  - Double dash is used to avoid confusion with multiple 'old style' options as shown above
- Both option types may take an 'argument'
  - Argument must be next on the command line (i.e. -o filename)
  - Sometimes '=' is used between the option and argument, other times, a space ' ' is used.

Environment Variables

Environment variables:

- store information about the user and process
- Help provide needed context to applications

Important variables:

- **HOME** – This is the user's home directory
- **PATH** – This contains a list of directories to be searched for commands and programs. Directories are ':' separated
- **USER** – This contains the name of the logged in user
- **PWD** – This holds the current working directory. The value can be changed but doing so will not affect the actual connected working directory.

Manipulating Environment Variables

printenv

- Prints the values of all environment variables

*echo **$VARNAME***

- Displays contents of VARNAME

*unset **VARNAME***

- Deletes the named environment variable

*export **VARNAME='string'***

- Sets VARNAME to the provided value

Process Status and Control

ps [options]

- Provides information about running processes

^C

- Stops running program
- Process is deleted
- Process resources (memory, devices) are freed

^Z

- Suspends running program
- Process and resources are still present

Process Status and Control

jobs

- This lists the processes in the immediate ‘process tree’
- jobs are shown as : [jobnum]<+/-> command
  - The job number is a small integer assigned by the OS.
  - + indicates it is the top backround job that will be pulled to foreground with ‘fg’
  - ‘command’ shows the entire command line originally used to start the job
- -l option adds process id to displayed information

*bg %**<jobnum>***

- Continues running program indicated by <jobnum> in the background
- Keyboard commands do not affect background process

*fg %**<jobnum>***

- Continues running program indicated by <jobnum> in the foreground

*kill **<pid>***

- Stops program running in process indicated by process id <pid>
- Deletes process
- Use -9 option if kill does not work

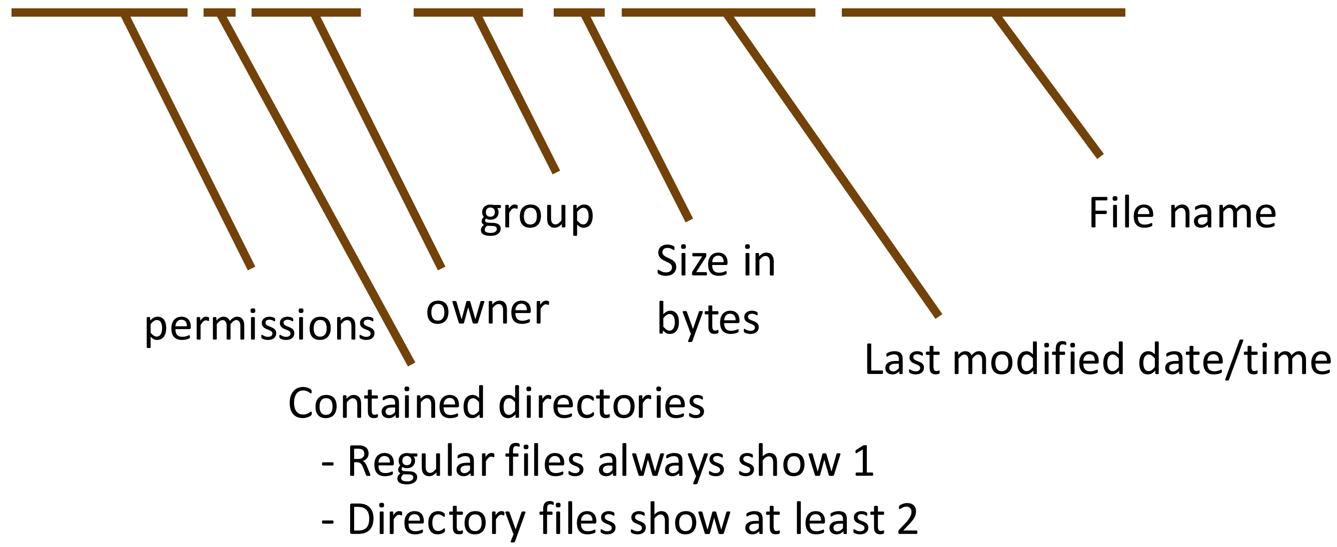
‘Long’ directory listing

ls –l

- Lists files in a directory with extra information (‘long’)

Interpreting ‘Long’ directory listing

```
 ls -l
total 12
-rwxrwxr-x 1 tmione tmione 4845 Jan 30 21:11 printSquares
-rw-rw-r--   1 tmione tmione  438 Jan 29 09:46 printSquares.c
```



permissions

Contained directories
- Regular files always show 1
- Directory files show at least 2

owner

group

Size in bytes

Last modified date/time

File name

File ownership

Files are owned by a user

Files also have an associated ‘group’

chown ***<username> <filename>***

- Changes the owner of <filename> to the user <username>
- -R – Recursively changes owner of all files in the entire directory tree

chgrp ***<groupname> <filename>***

- Changes the owning group of <filename> to the group <groupname>
- Some systems allow *chown* ***<user>.<group> <filename>***

File permissions

File permissions

- 9 characters
- Format : tuuugggooo
    - ‘t’ is type (‘-’=file, ‘d’=directory, ‘b’=block device, ‘c’=char device)
    - Each triplet represents read,write,execute (rwx) permissions for a user or subset of users
        - uuu – ‘owning user’
        - ggg – ‘group’ permissions
        - ooo – ‘other’ permissions
    - ‘-’ in a position means ‘not permitted’

Changing File Permissions

chmod – Change Mode – changes file permissions

- chmod <perms> <filename>
- <perms> can be given
  - As a combination of sets and permissions:
    - u+x – Add execution permission for the owning 'user'
    - o-rwx – Remove read, write, and execute from 'others'
  - As a 3 digit octal number where each octal digit is the permissions for a class of user:
    - 700 – Provide read, write, and execute for the owning user, no permissions for anyone else
    - 660 – Provide read and write for the owning user and group. No access is provided for others

File permissions examples

drwxr-xr-x

  Type: Directory file
  owner can read, write, and execute (search)
  group can read and execute (search)
  ‘other’ users can read and execute (search)

-rwxrw-rw-

  Type: regular file
  owner can read, write, and execute
  group can read and write (but not execute)
  ‘other’ users can read and write (but not execute)

Locating files

‘ls’ is not good at locating files in a large directory subtree

‘find’ searches a subtree for files with matching criteria

*find **<top_path>** [options] [expression]*

- <top_path> should be the top directory where the search will start
- [options] – Provide details to restrict scope of search (maxdepth, mindepth, etc)
- [expression] – These provide search parameters

Locating files

‘find’ options

- -maxdepth # - Do not descend past ‘#’ directory levels
- -mindepth # - Do not test for files above ‘#’ directory levels

‘find’ expression predicates

- -name ‘<filespec>’
  - <filespec> can contain wildcards (*, ?, [character class], etc.
  - Must be in ‘’! [single quotes]
- -iname ‘<filespec>’ – like -name but match is case insensitive
- -empty – Empty files (or directories)

Locating Files

‘find’ expression predicates

- -amin # - Files that were
  - # - accessed ‘#’ minutes ago
  - -# - accessed less than ‘#’ minutes ago
  - +# - accessed more than ‘#’ minutes ago
- -cmin # - Files that were changed ‘#’ minutes ago (see –amin)
- -readable, -executable, -writable
- -type ‘t’ – (d=directory, c=char special, b=block special, f=regular file, l=symbolic link, etc)
- -size # - Files that are (>,<,=) ‘#’ units large (Supports +/-)
  - #k - # * 1024
  - #M - # * 1024 * 1024
  - #G - # * 1024 * 1024 * 1024

Command History

Unix will save recent commands

Environment variables (place definitions in .bashrc)

- HISTSIZE – number of commands to save
- HISTFILESIZE – number of commands to save across sessions
- HISTFILE – File in which to save history

history – lists entire saved command history

'!' recalls commands or parts of commands

- !# - recalls command '#' from list
- !-# - Recalls '#'th previous command
- !! – recalls previous command
- !<string> - Recalls most recent command starting '<string>'

Command History

‘:’ after recall command specifies what part of the command to recall

- ^ - recall first argument
- $ - recall last argument
- # - recall ‘#’th argument
- n-m – recall ‘n’th through ‘m’th argument

Custom Commands

alias ***<commandname>=‘<command and arguments>’***

- Creates a command (<commandname>) that translates to the remaining text
- Can place these in .bashrc so they are present each time you start a shell
- Examples:
  - alias direct=‘ls –l | grep “^d”’
  - alias tmpfiles=‘ls –l *.tmp’
  - alias dgcc=‘gcc –O0 –ggdb’

Questions?