

Chapter 2

Unix Utilities for non-programmers

Graham Glass and King Ables,
UNIX for Programmers and Users,
Third Edition, Pearson Prentice Hall, 2003.

Original Notes by Raj Sunderraman
Converted to presentation and updated by
Michael Weeks

8/20/2008

cs3320

1

Getting Started

- Obtain an account -OR- install Linux
- Logging in
 - locally: enter username and password
 - remotely: use the *ssh/telnet* utility to sign on to
- Upon logging on you will see a prompt (usually \$ or %) which is displayed by a special kind of program called a SHELL program.
- To set your password, use *passwd*

8/20/2008

cs3320

2

Shells

- Popular shells:
 - Bourne Shell
 - Korn Shell
 - C Shell
 - Bash (Bourne Again SHell)
- All have common core functionality; some differences.
- Each shell has its own programming language. (Shell programming).

8/20/2008

cs3320

3

Running Utilities

- To run a utility, simply type the name of the utility after the prompt.
- Some utilities: *date*, *man*, *clear*, *stty*, *passwd*
 - Utility: *date* [yymmddhhmm[.ss]]
 - Utility: *clear*
 - Utility: *man* [-s section] word
 - *man* -k keyword
- To logout, enter CTRL-D (or *exit*)

8/20/2008

cs3320

4

Special Characters

- list meta characters with *stty -a* command
- ^ means control
- ^C means hold down control key (CTRL) and press the C key

8/20/2008

cs3320

5

Special Characters

- erase: CTRL-H erase character before cursor
- werase: CTRL-W erase word before cursor
- kill: CTRL-U erase entire line
- rprnt: CTRL-R reprint line
- To test these:
 - type some input like "one two three four" at the terminal prompt
 - move cursor around
 - try the above characters

8/20/2008

cs3320

6

Special Characters

- `intr: CTRL-C` interrupt running program
- `susp: CTRL-Z` suspend running program
- `stop: CTRL-S/CTRL-Q` stop printing to screen
- `eof: CTRL-D` give program end of file
- To test these:
 - Try a command like `find *`
 - For eof, try `cat > testfile` then `CTRL-D` on new line

8/20/2008

cs3320

7

Some common UNIX utilities

- `pwd`
- `cat` (like concatenate), `more`, `page`, `head`, `tail`
- `ls`, `cd`
- `mv`, `cp`, `rm`
- `mkdir`, `rmdir`
- `file` (determine file type), `wc` (word count), `lp`
- `vi`, `pico`, `emacs`
- `groups`, `chgrp`, `chmod`

8/20/2008

cs3320

8

Pathnames

- `/home/mweeks/x.c`
- `/export/home/mweeks`
- `/Users/mweeks`
- `/usr/oracle/m01/app/oracle/product/8.0.4/bin/sqlpl`
`us`
- `.` refers to current directory (example: `./a.out`)
- `..` refers to parent directory (example: `../p2.cpp`)

8/20/2008

cs3320

9

Example of `cat`

`$ cat > letter`

Hi Mom,

Please send me money!

David

`^D`

`$ ls -l letter`

```
-rw-r--r--  1 qcheng  other    38 May 28 11:20  
letter
```

8/20/2008

cs3320

10

List Files (`ls`)

- `$ ls -algFsdR <file-spec>`
- options
 - `a`: hidden files
 - `l`: long listing
 - `g`: group
 - `F`: put character after file name indicating executable*, link@, directory/, socket=
 - `s`: num. disk blocks
 - `d`: dir details not contents
 - `R`: recursive listing

8/20/2008

cs3320

11

Show File Contents (`cat`, `more`)

- `$ cat <file-spec-list>`
 - list contents of file(s) on screen without pause
- `$ more <file-spec-list>`
 - same as `cat`; pauses after each screen -more-
 - space bar takes you to next screen;
 - q quit
 - enter shows next line
 - h help key for more commands

8/20/2008

cs3320

12

Show File Contents (*page*, *head*, *tail*)

- `$ page <file-spec-list>`
 - same as *more*; clears screen before each page
 - quicker
- `$ head -n <file-spec>`
 - display *n* lines from front of file
- `$ tail -n <file-spec>`
 - display *n* lines from end of file

8/20/2008

cs3320

13

Working with Directories

- `$ mkdir <dname>`
 - make a new directory
- `$ cd <dir>`
 - built-in to shell
 - changes shell to a different directory
- `$ pwd`
 - print working directory

8/20/2008

cs3320

14

Rename Files (*mv*)

- rename files (simply change labels in the file hierarchy)
- `$ mv -i <old> <new>`
 - *i* for inquire
 - (prompt if *<new>* already exists)
- `$ mv -i <file-spec> <dir>`
- `$ mv -i <dir> <dir>`

8/20/2008

cs3320

15

Copy Files (*cp*)

- `$ cp -i <old> <new>`
 - *i* for inquire
 - (prompt if *<new>* already exists)
- `$ cp -i <file-spec> <dir>`
- `$ cp -ir <file-spec> <dir>`
 - *r* for recursive

8/20/2008

cs3320

16

Removing Files (*rm*, *rmdir*)

- remove (delete) file
- `$ rm -fir <file-spec>`
 - *f*: force; inhibits all prompts/messages
 - *i*: inquire
 - *r*: recursive
- Be careful! This is a permanent deletion!
- remove (delete) directory
- `$ rmdir <dir>`

8/20/2008

cs3320

17

Line Printing Commands

- Print file *<file-spec>* to printer *dest* :
 - `$ lp -d dest -n copies <file-spec>`
 - also *lpr*
- Get status of jobs at printer *dest*
 - `$ lpstat dest`
- Stop a print job
 - Need job number *<request-id>* from above
 - `$ cancel <request-id>`

8/20/2008

cs3320

18

Word Count

- `word count`
- `$ wc -clw <file-spec>`
 - `c` counts the bytes
 - `l` counts the newline characters
 - `w` prints the word counts

8/20/2008

cs3320

19

File Attributes

- `-rwx----- 1 cscqxcx cscqxcx 3661 Jan 9 23:12 RingBufferActOutHandler.java`
 - `-rwx-----` : file type (first char), permissions
 - file type (- regular, d dir, b buffered file disk drive, c unbuffered file terminal, l link, p pipe, s socket)
 - `1` : hard link count
 - `cscqxcx` : file owner
 - `cscqxcx` : file's group
 - `3661` : file size in bytes
 - `Jan 9 23:12` : file modification date
 - `RingBufferActOutHandler.java` : file name

8/20/2008

cs3320

20

File Type

- `$ file <fname>`
- ascertains the type of file (ascii/binary etc.)
- example:
 - `[cscqxcx@techie BattlefieldOnFramework]$ file BFComputation.c`
`file BFComputation.c`
 - `BFComputation.c: ASCII C program text, with CRLF line terminators`

8/20/2008

cs3320

21

File Type

- example:
 - `[cscqxcx@techie testbed]$ file *`
 - `BattleFieldOnFramework:` `directory`
 - `BattleFieldOnFramework.09.24.tar:` `POSIX tar archive`
 - `RingBufferActOutHandler.java:` `ASCII C program text`
 - `Samples:` `directory`
 - `try:` `ASCII text`

8/20/2008

cs3320

22

Process Attributes

- When a process executes, it has
 - a real user ID
 - an effective user ID
 - a real group ID
 - an effective group ID

8/20/2008

cs3320

23

Process Attributes

- When you log in, your login shell process has its
 - real/effective user id set to your user id
 - real/effective group id set to your group id

8/20/2008

cs3320

24

Process' File Permissions

- When a process runs, the file permissions apply as follows:
 - If process' effective user id = owner of file; User Privilege applies
 - If process' effective user id is not owner of file, but its effective group id = file's group id; Group Privilege applies
 - Otherwise Other Privilege applies
- Real user id is used for accounting purposes only

8/20/2008

cs3320

25

Process Access Rights

- Process Access Rights
 - depends on who executes the process,
 - NOT who owns the executable
- Sometimes undesirable
- Example: game of rogue
 - file of best scores needs to be modified ONLY by rogue program process when player ends game; NOT by players' shells.

8/20/2008

cs3320

26

Process Access Rights

- Special feature: 2 special file permissions
 - set user ID (s instead of x)
 - set group ID
- When an executable with set user ID is executed, the effective user ID of the process becomes that of the executable, NOT of the players shell
- Allows process to modify best scores file

8/20/2008

cs3320

27

Groups

- list the groups a particular user belongs to
 - `$ groups <userid>`
- change the group a file belongs to
 - `$ chgrp -R <group-name> <file-spec>`
 - *R stands for recursive*
- create subshell with effective group id = group
 - `$ newgrp <group-name>`

8/20/2008

cs3320

28

Change File Permissions

- change file mode; u,g,o,a r,w,x,s
 - user, group, other, all
 - read, write, execute, (set)
 - plus (+) adds permissions
 - minus (-) subtracts
- Also allows octal numbers

8/20/2008

cs3320

29

Change File Permissions

- Examples
 - `$ chmod -R XXX <file-spec>`
 - `$ chmod -R g+w <file-spec>`
 - `$ chmod -R u-rw <file-spec>`
 - `$ chmod -R u+w,g-r <file-spec>`

8/20/2008

cs3320

30

Change File Permissions

- Example with octal numbers
 - mweeks@carmaux:~\$ ls -l example.txt
 - -rw-r--r-- 1 mweeks mweeks 51 2007-08-23 16:06 example.txt
 - mweeks@carmaux:~\$ chmod 765 example.txt
 - mweeks@carmaux:~\$ ls -l example.txt
 - -rwxrw-r-x 1 mweeks mweeks 51 2007-08-23 16:06 example.txt
- Notice:
 - 7 = 111 (binary) r "on", w "on", x "on"
 - 6 = 110 (binary) r "on", w "on", x "off"
 - 5 = 101 (binary) r "on", w "off", x "on"

8/20/2008

cs3320

31

Change File Owner

- change owner of file
 - \$ *chown* <new-owner> <file-spec>
- only super user can do this

8/20/2008

cs3320

32

The vi Editor

- visual text editor
- Very common on Unix systems
- Created by Bill Joy in 1976
- Many versions exist, including vim
 - vi improved

8/20/2008

cs3320

33

The vi Editor

- Command mode vs Text Entry Mode
 - To enter text entry mode use the following commands:
 - i,I,a,A,o,O,R
 - to get back to command mode use ESC
- line ranges: 1,\$ 1,. ,,\$,.-2 5,12 ,.+12

8/20/2008

cs3320

34

The vi Editor

- Cursor Movement:
 - up arrow k
 - down arrow j
 - left arrow h
 - right arrow l
 - beginning of line O
 - first non-whitespace char in line ^
 - end of line \$

8/20/2008

cs3320

35

The vi Editor

- Cursor Movement:
 - next word w
 - previous word (back) b
- Goto line number *nn*
 - :nn
- Go to last line
 - :\$
- Arithmetic expression
 - 1,\$
 - 1,.
 - ,,\$
 - ,.-2

8/20/2008

cs3320

36

The vi Editor

- CTRL-D (down half screen)
- CTRL-U (up half screen)
- CTRL-F (forward one screen)
- CTRL-B (back one screen)

8/20/2008

cs3320

37

The vi Editor

- Deleting Text:
 - delete character under cursor `x`
 - delete word `dw`
 - delete line `dd`
(also `D`)
 - delete many lines `:<range>d`
- Replacing text:
 - replace char under cursor `r`
 - replace word under cursor `cw`
 - replace entire line `cc`
 - substitute next 4 characters `4s`

8/20/2008

cs3320

38

The vi Editor

- Pasting text:
 - `:<range>y` (yank lines)
 - `5yy` (yank 5 lines)
 - `p`
 - `P`
 - `:nnp` (after line `nn`)
 - `pu`

8/20/2008

cs3320

39

The vi Editor

- Searching:
 - search forward `/sss/`
 - search backward `?sss?`
- Searching/Replacing:
 - `:<range>s/sss/ttt/`
 - `:<range>s/sss/ttt/g`

8/20/2008

cs3320

40

The vi Editor

- Saving/Loading:
 - write to file `:w <fname>`
 - write again `:w`
 - write `:<range>w`
`<fname>`
 - write and quit `:wq`
 - edit new file `:e <fname>`
 - read (insert) a file `:r <fname>`

8/20/2008

cs3320

41

The vi Editor

- Misc
 - re-draw screen `CTRL-L`
 - execute a shell command `!:comand`
- Quitting:
 - quit `:q`
 - write and quit `:wq`
 - quit without writing `:q!`
 - exit and write if changed `:x`

8/20/2008

cs3320

42

Review

- Shells and Special Characters
- Common Utilities
 - Working with Directories
 - Removing Files
- File Attributes/Types
- Process Attributes
- File permissions, groups, and owner
- The vi editor