Introduction to UNIX

EECS 2031

13 November 2017

Introduction

- UNIX is an operating system (OS).
- Our goals:
  - Learn how to use UNIX OS.
  - Use UNIX tools for developing programs/ software, specifically shell programming.

Processes

- Each running program on a UNIX system is called a process.
- Processes are identified by a number (process id or PID).
- Each process has a unique PID.
- There are usually several processes running concurrently in a UNIX system.

3

ps command

```
indigo 167 % ps a                                # a = list all processes
  PID TTY      STAT   TIME COMMAND
 1950 pts/22   Ss     0:00 -csh
 1981 pts/22   S+     0:00 ssh video
 2047 pts/23   Ss     0:00 -csh
 2067 pts/23   S+     0:01 ssh voice
 2097 pts/25   Ss+    0:00 -csh
 2249 pts/25   S      0:00 /bin/sh /cs/local/bin/moodle
 2264 pts/25   Sl   199:47 /cse/local/pkg/firefox/firefox
 3162 pts/11   Ss     0:00 -csh
 3304 pts/11   S+     0:46 ssh -X music
 6963 pts/13   Ss+    0:00 -tcsh
 8567 tty1     Ss+    0:00 /sbin/mingetty /dev/tty1
 8568 tty2     Ss+    0:00 /sbin/mingetty /dev/tty2
 8570 tty3     Ss+    0:00 /sbin/mingetty /dev/tty3
 8572 tty4     Ss+    0:00 /sbin/mingetty /dev/tty4
```

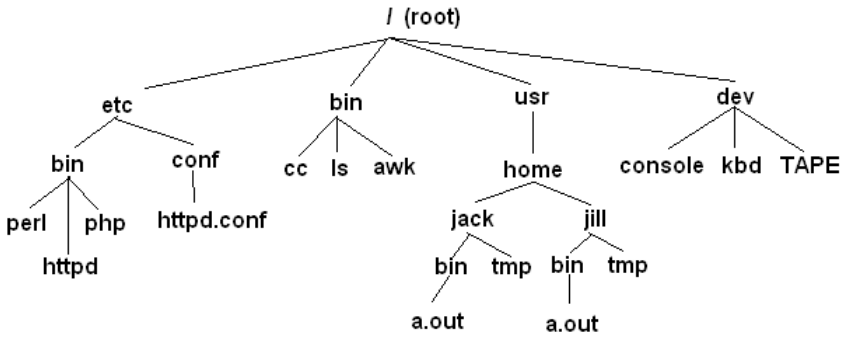
4

The File System

- Directory structure
- Current working directory
- Path names
- Special notations

5

Directory Structure



6

Current Working Directory

- Every process has a current working directory.
- In a shell, the command **ls** shows the contents of the current working directory.
- **pwd** shows the current working directory.
- **cd** changes the current working directory to another.

7

Path Names

- A path name is a reference to something in the file system.
- A path name specifies the set of directories you have to pass through to find a file.
- Directory names are separated by '/' in UNIX.
- Path names beginning with '/' are absolute path names.
- Path names that do not begin with '/' are relative path names (start search in current working directory).

8

Special Characters

- **.** means the current directory
- **..** means the parent directory
  - **cd ..**
  - **cd ../Notes**
- **~** means the home directory
  - **cat ~/lab3.c**
- To go directly to your home directory, type
  - **cd**

9

Frequently Used Terminal Keystrokes

- Interrupt the current process: Ctrl-C
- End of file: Ctrl-D
- Read input (stdin) from a file
  - a.out < input_file
- Redirect output (stdout) to a file
  - ls > all_files.txt # overwrites all_files.txt
- Append stdout to a file
  - ls >> all_files.txt # append new text to file

Wildcards (File Name Substitution)

- Goal: referring to several files in one go.
- ? match single character
  - ls ~/C2031/lab5.???
  - lab5.doc lab5.pdf lab5.out
- * match any number of characters
  - ls ~/C2031/lab5.*
- […] match any character in the list enclosed by [ ]
  - ls ~/C2031/lab[567].c
  - lab5.c lab6.c lab7.c
- We can combine different wildcards.
  - ls [e]*.c
  - enum.c ex1.c ex2.c

11

File Manipulation Commands

ls, cp, mv, rm

touch

pwd, mkdir, rmdir

cd

chmod, chown, chgrp

find

12

find command

Search from the current directory:

```
% find . -name "ex1.c"
./Ptr2Ptr/ex1.c
./ex1.c

% find . -name "e*.c"
./Midterm/err.c
./ex2.c
./Ptr2Ptr/ex2.c
./Ptr2Ptr/ex1.c
./enum.c
./ex1.c
```

Search from the home directory:

```
% find ~ -name "ex1.c"
/cs/home/utn/Temp_2031/Misc/ex1.c
/cs/home/utn/Demo_2031/Ptr2Ptr/ex1.c
/cs/home/utn/Demo_2031/ex1.c
```

Search from the specified directory:

```
% find ./Test1/Archive/ -name "*.c"
./Test1/Archive/convertMain.c
```

13

Commonly Used Commands

- Get on-line help with **man**

  `man chgrp`

- Some commonly used commands

  `cat, more`
  `who`
  `grep`
  `echo`
  `date`
  `sort`
  `wc`
  `ps, kill`
  `history`

14

cat, more, tail

```
% cat phone_book
Yvonne 416-987-6543
Amy 416-123-4567
William 905-888-1234
John 647-999-4321
Annie 905-555-9876
```

```
% more phone_book
```

Similar to cat, except that the file is displayed one screen at a time.

```
% tail myfile.txt
```

Display the last 10 lines

```
% tail -5 myfile.txt
```

Display the last 5 lines

```
% tail -1 myfile.txt
```

Display the last line

15

echo

- When one or more strings are provided as arguments, echo by default repeats those strings on the screen.

```
% echo This is a test.
This is a test.
```

- It is not necessary to surround the strings with quotes, as it does not affect what is written on the screen.
- If quotes (either single or double) are used, the quotes are not repeated on the screen.

```
% echo 'This is'"a test."
This is a test.
```

- To display single/double quotes, use `\'` or `\"`

16

echo (cont.)

```
% echo a \t b
a t b
% echo 'a \t b'
a        b
% echo "a \t b"
a        b
```


UNIX Filter grep

```
% grep 'char' *.c
arr.c:  char s[] = "2031";
char.c:   char c;
char.c:   c = getchar();

% grep '1302ESC' cse*/lab3.c

% grep -i 'ChaR' *.c
arr.c:  char s[] = "2031";
char.c:   char c;
char.c:   c = getchar();
```

```
% grep -v 'char' *.c
```

Search for lines that do not contain pattern “char”.

```
% grep 'bea[nm]' *.txt
```

Search for lines that contain either bean or beam.

```
% grep '[0-9][0-9][0-9]' *.c
```

Search for lines that contain a sequence of 3 (or more) digits.

grep (cont.)

```
% grep –n 'char' *.c
```
Also display the line numbers.

```
% grep '[3]' *.c
% grep '3' *.c
```
Search for lines that contain digit 3.

```
% grep '\[3\]' *.c
```
Search for lines that contain string [3].

```
% grep '\[' *.c
```

19

wc

```
% wc enum.c
 14  37 220 enum.c

% wc [e]*.c
 14  37 220 enum.c
 17  28 233 ex1.c
 21  46 300 ex2.c
 52 111 753 total
```

```
% wc -c enum.c
220 enum.c

% wc -w enum.c
37 enum.c

% wc -l enum.c
14 enum.c
```

20

sort

```
% cat phone_book
Yvonne 416-987-6543
Amy 416-123-4567
William 905-888-1234
John 647-999-4321
Annie 905-555-9876

% sort phone_book
Amy 416-123-4567
Annie 905-555-9876
John 647-999-4321
William 905-888-1234
Yvonne 416-987-6543
```

Try these options:

sort –r
reverse normal order

sort –n
numeric order

sort –nr
reverse numeric order

sort –f
case insensitive

sort –k 2
sort on column 2

21

cmp, diff

```
% cat phone_book
Yvonne 416-987-6543
Amy 416-123-4567
William 905-888-1234
John 647-999-4321
Annie 905-555-9876

% cat phone_book2
Yvonne 416-987-6543
Amy 416-111-1111
William 905-888-1234
John 647-999-9999
Annie 905-555-9876
```

```
% cmp phone_book phone_book2
phone_book phone_book2
differ: char 30, line 2

% diff phone_book
phone_book2
2c2
< Amy 416-123-4567
---
> Amy 416-111-1111
4c4
< John 647-999-4321
---
> John 647-999-9999
```

22

who

```
% who
ossama   pts/13       Nov  7 00:22 (ip-198-96-36-11.dynamic.yorku.ca)
hoda      pts/21       Nov  4 16:49 (gomez.cs.yorku.ca)
gordon   pts/24       Nov  5 10:40 (bas2-toronto08-1096793138.dsl.bell.ca)
minas    pts/29       Nov  2 14:09 (monster.cs.yorku.ca)
jas       pts/37       Oct 18 12:36 (brayden.cs.yorku.ca)
utn      pts/93       Nov  7 12:21 (bas2-toronto44-1177753778.dsl.bell.ca)
```

- User name
- Terminal associated with the process
- Time when they logged in

23

kill

```
% ps a
  PID TTY          TIME CMD
 2117 pts/24   00:00:00 pine
 2597 pts/79   00:00:00 ssh
 5134 pts/67   00:00:34 alpine
 7921 pts/62   00:00:01 emacs
13963 pts/24   00:00:00 sleep
13976 pts/43   00:00:00 sleep
13977 pts/93   00:00:00 ps
15190 pts/90   00:00:00 vim
24160 pts/44   00:00:01 xterm
. . .
```

```
% kill 7921
```

24

history

```
% history 10
   323  12:45   ls
   324  12:47   cd Demo_2031/
   325  12:48   ls
   326  12:48   m ex1.c
   327  12:49   who
   328  12:50   history 10
   329  12:52   ls -a
   330  12:56   ls Stack/
   331  12:57   ls
   332  12:57   history 10
```

25

Pipes

- Pipe: a way to connect the output of one program to the input of another program without any temporary file.
- Pipeline: connection of two or more programs through pipes.
- Examples:

```
ls   | wc –w          # count number of files
who | sort           # sort user list
who | wc –l          # count number of users
who | grep ‘utn’     # look for user ‘utn’
ps a | grep 'emacs’ # look for process emacs
```

26

NEVER-DO List in UNIX

- Never switch off the power on a UNIX computer.
  - You could interrupt the system while it is writing to the disk drive and destroy your disk.
  - Other users might be using the system.
- Avoid using `*` with `rm` such as `rm *, rm *.c`
- Do not name an important program core.
  - When a program crashes, UNIX dumps the entire kernel image to a file called `core`.
  - Many scripts go around deleting these `core` files.
- Do not name an executable file `test`.
  - There is a Unix command called `test`.

27

Next time …

- UNIX Basics 2
- Writing Shell Scripts

28