Lecture 2. Basic Linux commands

understand various linux commands, understand absolute/relative path,
understand linux file tree, understand file access permission,
understand shell special symbols

Site for linux commands, system calls, and c library functions

http://linux.die.net/man/
section 1: linux command list
section 2: linux system call list
section 3: linux library function list

Or

http://www.oreillynet.com/linux/cmd/ for linux command list
http://en.wikipedia.org/wiki/List_of_C_functions for c standard library functions

command: utility programs. ls, cat, cp, ...
C library functions: system functions that you can use in your C program. Provided by the C compiler. printf(), scanf(), strlen(), ...
System calls: system functions that you can use in your C program. Provided by the Operating system. read(), write(), socket(), ...

1. Accessing lab server from PC
- download putty.exe from I-class or Interent
- run putty.exe
- login with
  user id: 12345 (your student ID)
  passwd: p12345 (p + your student ID)

For Mac, open the terminal window and enter "ssh userID@ip-addr". userID is your student ID (explained below) and ip-addr is either 165.246.38.151 or 165.246.38.152 (also explained below). For Chromebook, use "Secure Shell App" and enter "userID@ip-addr", where userID and ip-addr are as explained above.

2. upload/download a file from/to PC
- download psftp.exe from internet
- run psftp.exe
- login with the same user id and password
- move local location with "lcd", remote location with "cd"
  lcd . - show current directory in PC
  lcd .. - move to parent directory in PC
  lcd c: - move to C hard disk in PC
  cd . - show current directory in the lab server
  cd mydir - move to mydir in the lab server
- to download a file
  get myfile - will download "myfile" from the lab server into PC
- to upload a file
  put myf - will upload "myf" from PC to the lab server

3. Linux basics

1) Linux command classification

display information
   general: man
   process: ps, who, finger, top, last, history
   file:
      location: find, which, whereis, locate
      general: ls, file
      content: cat, more, od, xxd, cmp, diff, wc, head, tail, objdump
      search: grep
      others: echo
    other: tty, pwd, date, cal, df, du, uname, mount, hostname, whoami, env

manipulate information
   process: kill, gcc, make, ctrl-c, ctrl-z, &, time, gdb
   file:
      editor: vi, ed, sed
      file system: fsck
      directory: cd, mkdir, rmdir
      general: ln, mv, cp, rm, gzip, gunzip, tar, touch
      permission: chmod, umask, chown
   terminal/login: login, exit, su, passwd, stty, clear
   communication: write, mail, ftp, telnet, ifconfig
   shell: sh, csh, ksh

2) file tree
/   -- root directory
   bin   -- executable files
      ls, zip, cat, chown, df, du, env, ftp, grep, ...
   etc   -- system configuration files
      password(password file), hostname(the name of this server), …
   home   -- user home directories
      linuxer2(home for user linuxer2), park(home for user park), …
   usr   -- library files, header files
      lib(library files are here), include(header files are here), …

3) relative path, absolute path
If the path starts with '/' , it is an absolute path; otherwise it is a relative path.
   cd /home/linuxer1/12345   -- go to /home/linuxer1/12345
   cd 12345   -- go to directory 12345 in the current directory
                                 if the current location is /home/linuxer1,
                                    go to /home/linuxer1/12345
                                 if the current location is /bin
                                    go to /bin/12345
If the destination directory does not exist, the system issues an error.

4) special symbols
. : current directory
   cp f1 ./f2   -- copy f1 to f2 in the current directory
.. : parent directory
   cp f1 ../f2 -- copy f1 to f2 in the parent directory

```
>  : standard output redirection
   cat f1 > f3  -- display the content of f1 in f3 (same effect as "cp f1 f3")
|  : pipe. redirect the standard output of the first program into the standard input of
     the second program
   cat f1 | more
*  : match any file name
   ls b*   -- diplay all file/directory names that start with 'b'
```

4. Basic commands (http://linux.die.net/Linux-CLI/)

```
ls    : listing files and directories in the current directory
    ls                : list all files
    ls -l             : list all files in detail
        -rwxr-xr-x  1  linuxer1 linuxer1  14  Feb 26 2013  f1

    ls -al            : list all files including hidden files
    ls ex*            : list all files whose name start with "ex"

pwd : show the current directory (Present Working Directory)

cd  : change current directory
    cd /              : go to the "/" directory (the root directory)
    cd /dev           : go to /dev
    cd ..             : go to the parent directory
    cd .              : go to the current directory(no moving)
    cd                : go to the home directory (the directory you enter when logging)

man : shows the usage of commands/system calls/c-lib functions etc. space to move to the
next screen. 'q' to exit.
    man  ls      : shows the usage of "ls" command
    man  1 kill  : shows the usage of "kill" command (manual section 1)
    man  2 kill  : shows the usage of "kill" system call (manual section 2)
    man  kill    : same as "man 1 kill"
    man  3 printf : shows the usage of "printf" c library functions (manual section 3)
    man  printf : same as "man 3 printf" (printf appears at section 3)

ps : listing processes
    ps         : show the processes of the current user
       PID  TTY    TIME    CMD
      12009 tty1    00:00:00  bash
      ..............
      PID: process ID
      TTY: terminal id for this process
      TIME: time spent on this process
      CMD: executable file name for this process
    ps -ef            : show all processes of all users
    ps -ef | more     : pipeline the output of "ps -ef" to "more"
                      : "more" will show the result of "ps -ef" screen by screen

mkdir : make a directory
    mkdir  d1
```

```
rmdir : remove a directory

echo : echo
    echo korea          : echo korea
    echo korea > f1     : redirect the standard output file of “echo” to f1.
                          As a result “korea” will be written to file f1.

cp, rm, mv: copy, remove, change the name of a file
    cp f1 f2            : copy file f1 to file f2
    cp f1 f4
    rm f4               : remove file f4
    mv f2 f3            : change the name of file f2 to f3

grep : search a string
    grep -nr “ko” *  : find all files that contain string “ko”. -n means show the
                          line number that contains “ko”. -r means “do this recursively
                          searching down all sub-directories”. * menas “all files” in
                          the current directory.

cat  : show the contents of a file
    cat f1              : show the contents of f1
    cat f1 > f2         : redirect the standard output file of “cat” to f2.
                          As a result, the data in f1 will be copied to f2.
    cat > f3            : Read data from keyboard and send them to f3.
                          ^D will end the input.

more : show the contents of a file one screen at a time. <space> to move to the next
       screen. ‘q’ to stop.
    more f1
    ls -al | more   : send the output of “ls -al” as an input to “more”.
                      As a result we can see the output of “ls -al” one screen at a time.
    ps -ef | more   : send the output of “ps -ef” to “more”
    grep -nr "ko" * | more : display the result of "grep ..." screen by screen


less : similar to more

xxd  : show the contents of a file in hexadecimal numbers
    xxd f1

uname : show the operating system information

whereis : show the locations of the executable file of a command
   whereis  kill

which  : show the exact location of the executable file of a command
   which  kill

whoami : show my user name

find : find the location of a file
```

find / -name "stdio.h" -print : find the location of "stdio.h" starting from /

file : show the file type
file f1

locate : find the location of a file in the file database

5. Homework
5.1 Do following steps.
1) Log in to a Linux server. Find your login directory with "pwd". Find your user ID with "whoami".
$ pwd
$ whoami
2) Go up the file tree with "..", display the current location with "pwd", and find other students directory name with "ls".
$ cd ..
$ pwd
$ ls
3) Pick one of the student IDs (suppose it was 12345) and try to enter his/her directory with "cd".
$ cd 12345
What happens?
4) Use "ls -l" to see the file permission of the directory you tried to enter.
$ ls -l
.....
drwx------ 2 12345 12345 4096 Feb 27 11:47 12345
.....

d: this file is a directory
rwx: the owner can read/write/execute this file
--- : the group member has no read/write/execute permission
--- : other people has no read/write/execute permission
2 : there are two links to this file
12345 : the user ID of the owner of this file
12345 : the group ID of of the owner of this file
4096 : the byte size of this file
Feb 27 11:47 : the creation time or last modification time of this file
12345 : file name

Explain why you couldn't enter the directory you picked.
5) Find out your current location again with "pwd".
$ pwd
6) Go to the root directory with "cd /" and make sure you are really at the top directory with "pwd".
$ cd /
$ pwd
7) How many files you have in the root directory? Some of them are not directory files. Find them with "ls -l". Use "file" command to see more detailed info.
$ ls -l
$ file *
$ file lib

8) * is a wild card character meaning it will be replaced by all file names in the current directory.

```
$ file *
```

is same as

```
$ file app bin boot dev .... (all file names in the current directory)
$ file b*
```

is same as

```
$ file bin boot
```

9) If you want to go back to your login directory (suppose it was /home/sp1/12345), you can cd with absolute path, cd with relative path, or just "cd".

```
cd /home/sp1/12345   -- go to absolute path "/home/sp1/12345"
cd home/sp1/12345    -- go to "home" in the current directory and go to "sp1"
                        and then to "12345", that is go to relative directory
                        "home/sp1/12345"
cd                   -- go to the login directory of the current user
```

Use one of above methods to go back to your login directory (or home directory).

10) Confirm your current location with "pwd".

11) List all files in your directory with "ls" command.

12) Try "echo" command.

```
$ echo korea
korea
$ echo hello
hello
```

11) Try "echo" with ">" symbol. ">" is called "standard output redirection".

```
$ echo hello > f1
```

">" sends the result to "f1" so there is no output shown in the terminal but you will have f1 in the current directory.

12) Do "ls" to see you can find f1 in the current directory. Show its content with "cat".

```
$ ls
$ cat f1
```

13) Make a directory, d1, with "mkdir".

```
$ mkdir d1
```

14) Copy f1 into directory d1.

```
$ cp f1 d1   -- make a copy of f1 inside directory d1 under the same name.
$ cd d1
$ ls
$ cat f1
$ cd ..
$ cp f1 d1/f2  -- make a copy of f1 inside d1 under another name f2
$ cd d1
$ ls
```

5.2 Do followings and explain what happens and why.

```
$ cd
$ ls
$ ls -l
$ ls -al
$ cd /
$ cd bin
$ ls
$ ls bz*
```

```
$ cd
$ pwd
$ man ls
$ echo hello
$ echo hello > f4
$ ls
$ cp f4 f2
$ cat f4
$ cat f2
$ cat f2 > f3
$ ls -l f*
$ rm f2
$ ls
$ cat f4
$ xxd f4
$ mkdir d2
$ cp f4 d2
$ cd d2
$ pwd
$ ls
$ cd ..
$ grep -nr "he" *
$ ps
$ ps -ef
$ ps -ef | more
```

5.3 Run following commands and explain what happens.

chmod, clear, gzip, date, dd, df, diff, dmesg, du, env, exit, file, find, head, ifconfig, ln, mount, netstat, stat, tail, time, touch, tty, gunzip, whereis, which, whoami,