Getting Started with Linux

경희대학교 컴퓨터공학과

조 진 성

User Account

- Get your account
  - ✓ Ask the system administrator (super-user)
  - ✓ ID & Password
- Manual configuration
  - ✓ `$ vi /etc/passwd`
    - **user ID, password, UID, GID, real name, home directory, shell**
  - ✓ `$ mkdir`
  - ✓ `$ chown`
- By one command
  - ✓ `$ adduser`
  - ✓ `$ deluser`
- Cf) admintool in X-Windows

Password

- Passwords must be kept secret
  - ✓ Frequent change of passwords
  - ✓ Use of “non-guessable” passwords
  - ✓ Log all invalid access attempts

- Passwords are encrypted
  - ✓ /etc/passwd
  - ✓ /etc/shadow

- Change your password: passwd

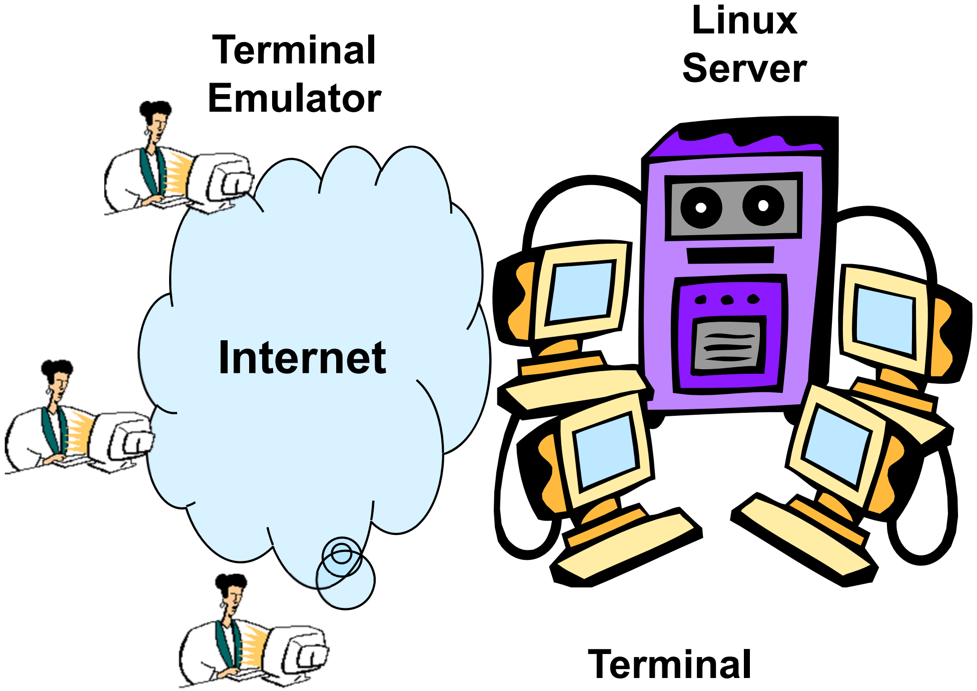
```
$ passwd
passwd:  Changing password for cjs
Enter login password:
New password:
Re-enter new password:
passwd (SYSTEM): passwd successfully changed for cjs
```

Login & Logout

- Login
  - ✓ Access to Linux/Unix system
  - ✓ Multiple login may be possible
  - ✓ Login procedure
    - When you see “Login: ” message, enter your ID
    - When you see “Password: ” enter your password
    - Then, a shell runs for you
- Logout
  - ✓ Exit from Linux system
  - ✓ Logout procedure
    - logout
    - exit
    - ^D (cf: set ignoreeof in csh)
  - ✓ If you do not process normal logout procedure?
    - Your shell process may remain in the system forever

Remote Access to Linux System

- Remote access protocol
  - ✓ telnet, rlogin, ssh
- Terminal emulator
  - ✓ PuTTY, TeraTerm, NetTerm, etc. in MS-Windows
  - ✓ X-shell
- Other remote access protocols (& commands)
  - ✓ ftp, rsh, rcp, scp, sftp



Shell

- Shell
  - ✓ Intermediate program between users and Linux
  - ✓ Command-line interpreter
  - ✓ Cf) “Command Prompt” in MS-Windows (cmd.exe)

- Several types of shells
  - ✓ Bourne shell (sh)
  - ✓ Korn shell (ksh)
  - ✓ C shell (csh, tcsh)
    - history (ex: !!, !g)
    - alias
    - job control (background job)
  - ✓ Bourne Again shell (bash)
    - default shell in Linux

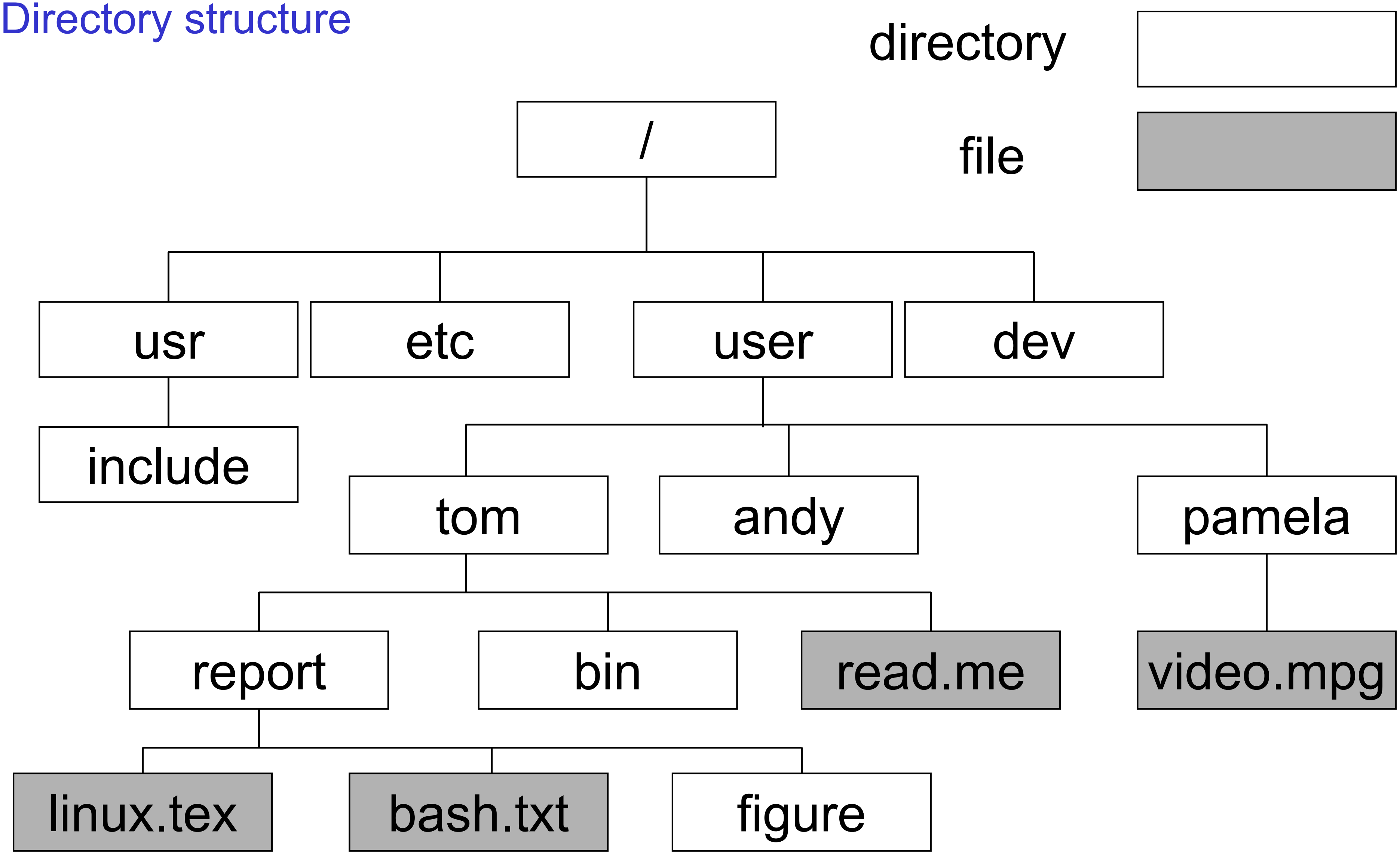
Shell (Cont'd)

- Meta characters
  - ✓ erase (^H): delete a character (backspace)
  - ✓ kill (^U): delete a line
  - ✓ werase (^W): delete an word
  - ✓ susp (^Z): stop foreground process
    - Resume as foreground process: **`$ fg`**
    - Resume as background process: **`$ bg`**
  - ✓ interrupt (^C): kill the foreground process without core dump (SIGINT signal)
  - ✓ quit (^\): kill the foreground process with core dump (SIGQUIT signal)
  - ✓ stop (^S|^Q): stop/resume terminal output
  - ✓ eof (^D): end of file(input)

- Shell programming
  - ✓ Shell dependent
  - ✓ Similar to *.bat in MS-DOS

Linux File System

- Directory structure



Linux File System (Cont'd)

- Path
  - ✓ Root directory /
  - ✓ Parent directory ..
  - ✓ Current directory . (or Working directory)
  - ✓ Home directory ~

- Absolute path
  - ✓ Path from root directory
  - ✓ /usr/local, /usr/bin, ~/doc/sample.doc

- Relative path
  - ✓ Path from current directory
  - ✓ ../src/sample.c, documents/report.txt

Linux File System (Cont'd)

- System directory

/	root directory
/usr	system directory (command, library, header files)
/usr/lib	system library
/usr/bin	user command
/usr/sbin	system administration command
/usr/include	C/C++ header file
/var	system log, miscellaneous information
/bin	single-user mode command
/sbin	single-user mode command
/usr/local	newly installed program(command, library, etc.)
/etc	system configuration
/dev	special device files
/home	User home directory
/tmp	directory for temporary files

Linux File System (Cont’d)

- User permission
  - ✓ Owner
  - ✓ Group
  - ✓ Other

- Access permission
  - ✓ Read
  - ✓ Write
  - ✓ Execute

Linux File System (Cont'd)

```
$ ls -l letter.0610 check_spell
    -rw-r--r--  1 alex pubs 3355 May 2 10:52 letter.0610
    -rwxr-xr-x  2 alex pubs  852 May 5 14:03 check_spell
```

✓ Type(first character) : *d for directory, - for file*

✓ Permission (the next nine characters)

✓ Number of links to the file

✓ Owner

✓ Group

✓ Size in bytes

✓ Modified date

✓ Name

Linux File System (Cont'd)

- Change the permission
  - ✓ By owner only
  - ✓ 'chmod' (change mode) command
  - ✓ Usage: **`chmod [a,u,g,o][+,-]{r,w,x}`** ***`filename`***
  - ✓ Example

```
$ chmod a+rw letter.0610
$ ls -l letter.0610
-rw-rw-rw- 1 alex pubs 3355 May 2 10:52 letter.0610

$ chmod o-rw check_spell
$ ls -l check_spell
-rwxr-x--- 2 alex pubs 852 May 5 14:03 check_spell

$ chmod 755 check_spell
$ ls -l check_spell
-rwxr-xr-x 2 alex pubs 852 May 5 14:03 check_spell
```

Basic Commands

- File system related commands

```
$ ls –al
$ mkdir mydir
$ cd mydir
$ rmdir mydir
$ pwd
$ cp header.h header2.h
$ cp –R mydir ../mydir2
$ rm header2.h
$ rm –rf mydir
$ mv print.c printf.c
$ ln –s /home/project/ myproject
$ ln –s ~kim/.bashrc .bashrc
$ mkfs
$ fsck
$ mount
$ du .
```

Basic Commands (Cont’d)

- File related commands

```
$ vi main.c
$ cat main.c
$ more main.c
$ ls –al | more
$ diff sort.c sort1.c > sort.diff
$ head main.c
$ tail main.c
$ file a.out
$ grep while main.c
$ find . –name ‘*.c’ –print
$ find . –name ‘*.c’ –exec grep SendPacket {} \;
$ wc main.c
$ od a.out
```

Basic Commands (Cont’d)

- Process related commands

```
$ ps
     PID TTY STAT  TIME COMMAND
   28475 pp1 S     0:00 -tcsh
   28478 pp1 S     0:00 -csh
   28486 pp1 T     0:00 vi
   28487 pp1 R     0:00 ps
$ ps –ef (or ax in Linux)
$ ps –ef | grep userID
$ kill –9 28486
```

- Miscellaneous

```
$ su – userID
$ date
$ who
$ clear
$ script
```

Basic Commands (Cont'd)

- File archive commands

```
$ tar cvf hw1.tar hw1.c hw1.h Makefile
$ tar tvf hw1.tar
$ tar xvf hw1.tar
$ tar cvf hw2.tar hw2
$ tar cvf hw3.tar *
$ compress hw1.tar
$ compress –d hw1.tar.Z
$ uncompress hw1.tar.Z
$ gzip hw2.tar
$ gzip –d hw2.tar.gz
```

Basic Commands (Cont’d)

- FTP (File Transfer Protocol)

```
$ ftp ceunix.khu.ac.kr
User(none):
Password:
ftp> ls (or dir)
ftp> cd hw1
ftp> lcd MyDocuments/Unix/Hw1
ftp> pwd
ftp> lcd .
ftp> binary (or ascii)
ftp> get hw1.tar
ftp> put hw2.c
ftp> mget *.c
ftp> mput *.h
ftp> prompt
ftp> quit (or bye)
$
```

Basic Commands (Cont’d)

- Manual pages: man
  - ✓ Section 1: shell commands
  - ✓ Section 2: system calls
  - ✓ Section 3: library function

```
$ man man
$ man gcc
$ man –s1 ls (AT&T style)
$ man 2 open (BSD style)
$ man 3 fopen
$ man strcpy
$ man –a exit
```

Basic Commands (Cont’d)

- Manual pages (Cont’d)

```
$ man mesg
```

```
MESG(1)             Linux Programmer's Manual             MESG(1)

NAME
       mesg - control write access to your terminal

SYNOPSIS
       mesg [y|n]

DESCRIPTION
       Mesg  controls the access to your terminal by others. It's
       typically used to allow/disallow others users to  write(1)
       to your terminal.

FLAGS
       y      Allow write access to your terminal.
       n      Disallow write access to your terminal.
       [none] Prints  out the current access state of your termi-
              nal.

SEE ALSO
       write(1),wall(1)

AUTHOR
       Miquel van Smoorenburg, miquels@drinkel.nl.mugnet.org

Linux 0.99               29 October 1993                        1
```

Summary

- How to use Linux
  - ✓ Terminal emulator
  - ✓ Login & Logout
  - ✓ Shell
  - ✓ Basic commands
    - File related commands
    - File system related commands
    - Process related commands
    - Miscellaneous commands