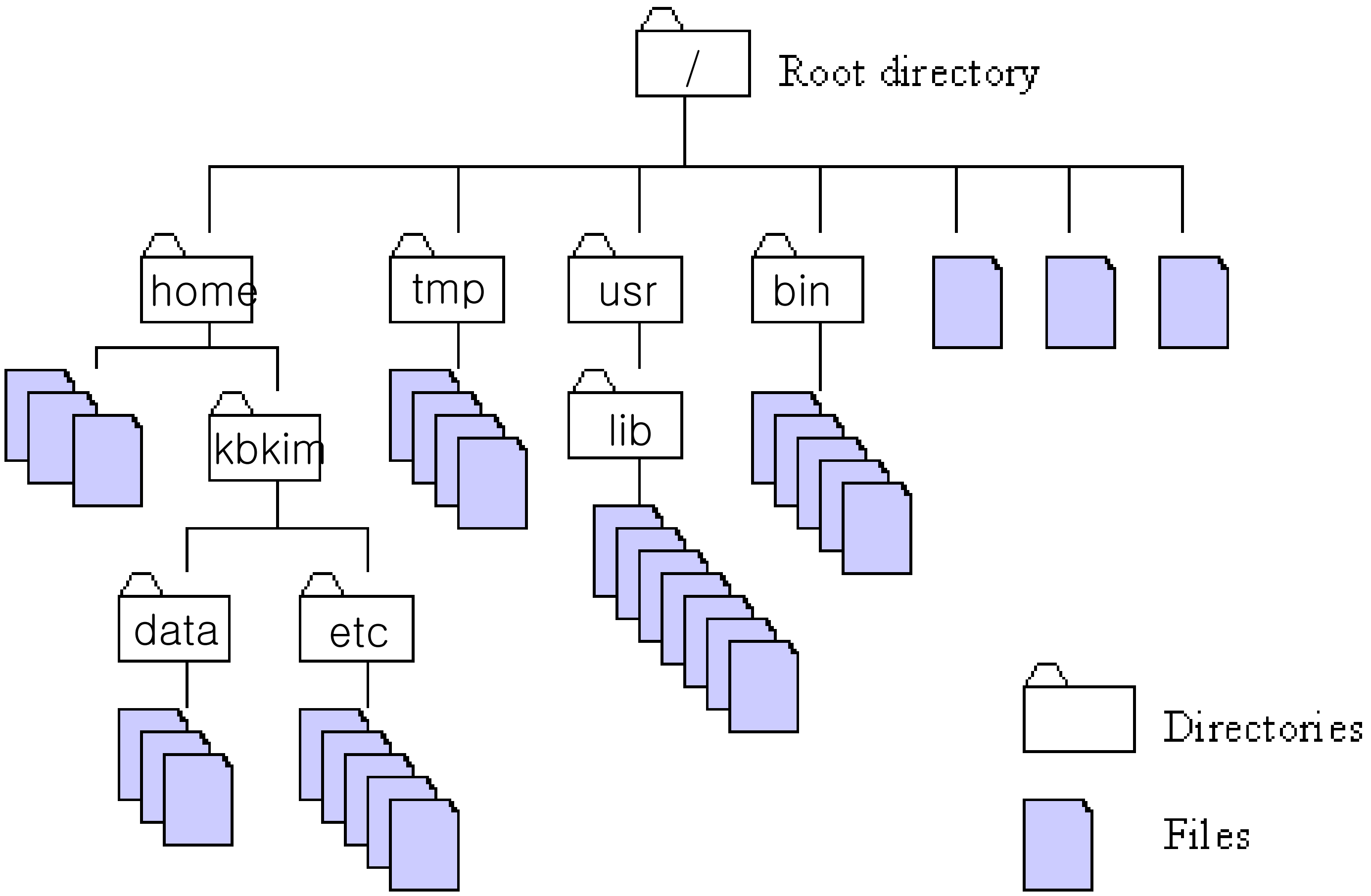
Files and Directories

Chonnam National University
School of Electronics and Computer Engineering

Kyungbaek Kim

Overview


/
Root directory
home
tmp
usr
bin
kbkim
lib
data
etc
Directories
Files

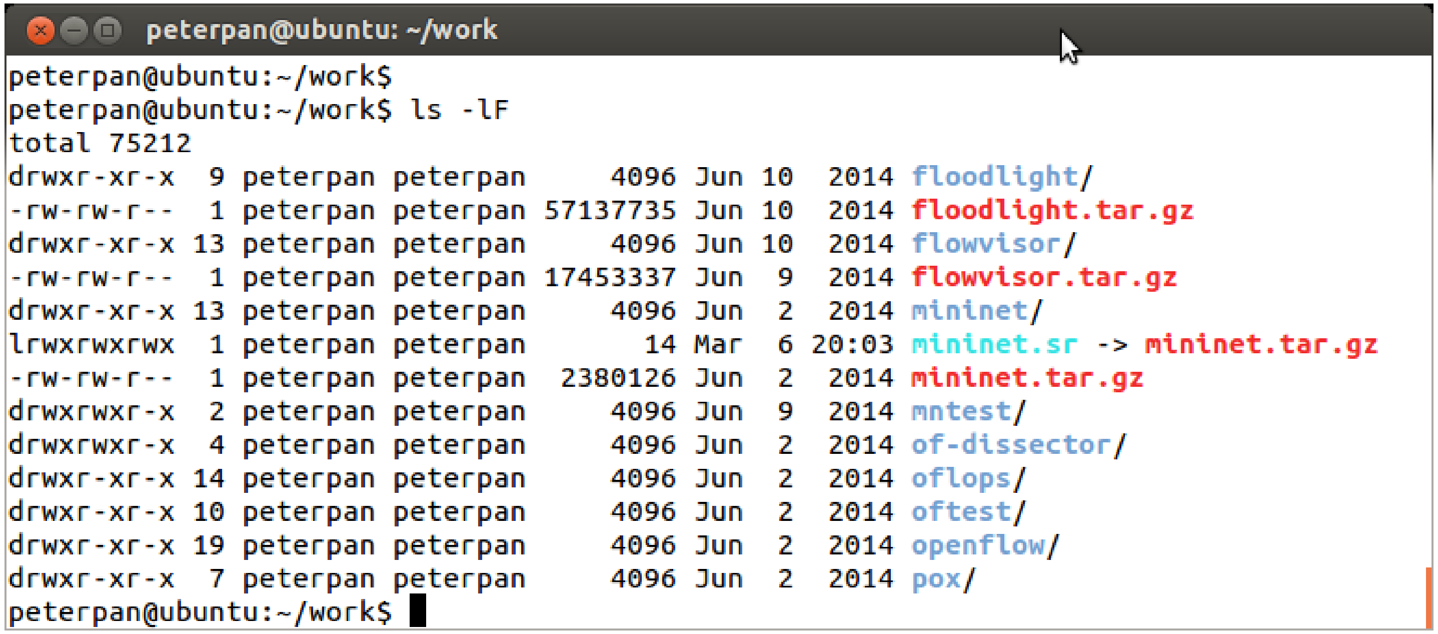
File

- A set of bytes to store data
- Each file has a filename
  - A label referring to a particular file
  - Permitted characters include letters, digits, hyphens (-), underscores (_), and dots (.)
  - *Case-sensitive*
  - File name starting with "." means it is a hidden file
- The "ls" command lists the names of files

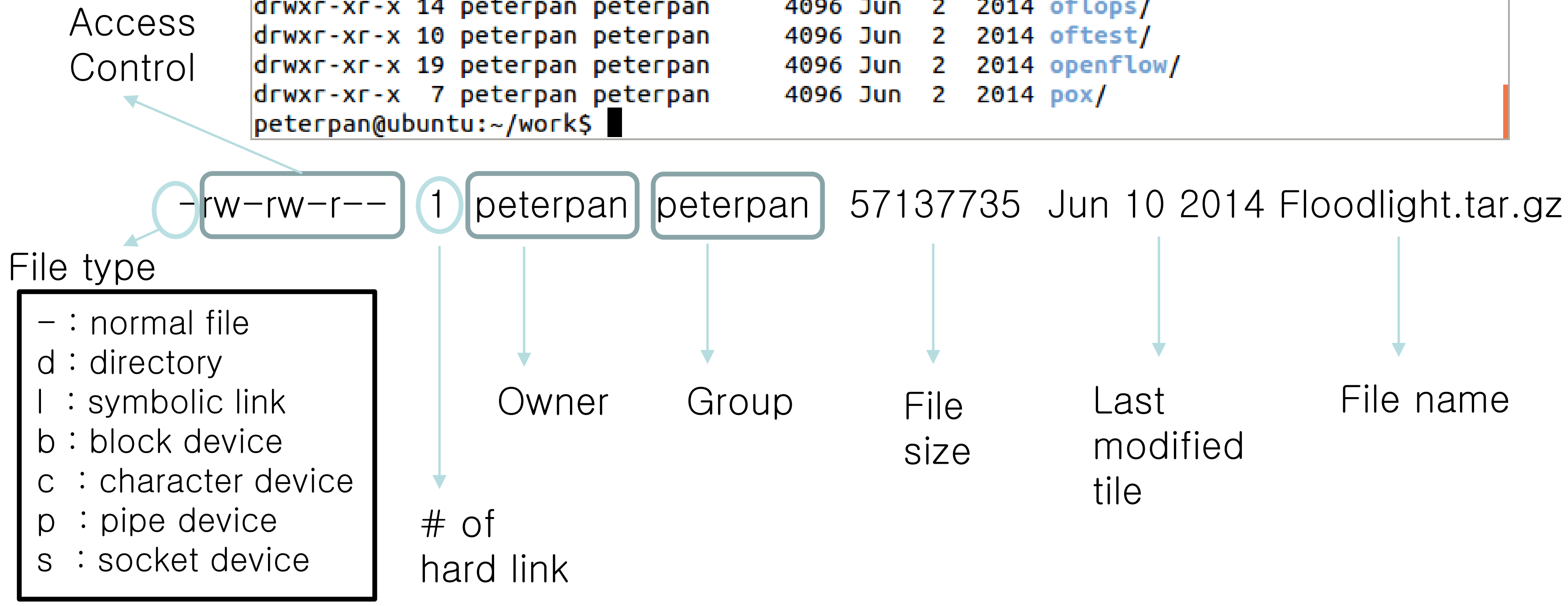
Listing Files with "ls"

- "ls" command
  - List the names of files
- Options
  - –a : *do not hide* entries starting with "."
  - –l : use a *long listing* format
  - –i : print *index number* of each file
  - –F : print file type (*: execution, /: directory, @: symbolic link)
  - –R : Recursively print directory contents
- e.g.) compare the results between "ls", "ls –a", "ls –al" and "ls –l"

Detail information of a file



```
peterpan@ubuntu:~/work$
peterpan@ubuntu:~/work$ ls -lF
total 75212
drwxr-xr-x  9 peterpan peterpan     4096 Jun 10  2014 floodlight/
-rw-rw-r--  1 peterpan peterpan 57137735 Jun 10  2014 floodlight.tar.gz
drwxr-xr-x 13 peterpan peterpan     4096 Jun 10  2014 flowvisor/
-rw-rw-r--  1 peterpan peterpan 17453337 Jun  9  2014 flowvisor.tar.gz
drwxr-xr-x 13 peterpan peterpan     4096 Jun  2  2014 mininet/
lrwxrwxrwx  1 peterpan peterpan       14 Mar  6 20:03 mininet.sr -> mininet.tar.gz
-rw-rw-r--  1 peterpan peterpan  2380126 Jun  2  2014 mininet.tar.gz
drwxrwxr-x  2 peterpan peterpan     4096 Jun  9  2014 mntest/
drwxrwxr-x  4 peterpan peterpan     4096 Jun  2  2014 of-dissector/
drwxr-xr-x 14 peterpan peterpan     4096 Jun  2  2014 oflops/
drwxr-xr-x 10 peterpan peterpan     4096 Jun  2  2014 oftest/
drwxr-xr-x 19 peterpan peterpan     4096 Jun  2  2014 openflow/
drwxr-xr-x  7 peterpan peterpan     4096 Jun  2  2014 pox/
peterpan@ubuntu:~/work$
```



-	normal file
d	directory
l	symbolic link
b	block device
c	character device
p	pipe device
s	socket device

Creating files with "cat"

- "cat" command

```
$ cat > shopping_list
cucumber
bread
yoghurts
```

- ">" sign means redirection of text types to the file "shopping_list"
- Press "**Ctrl+D**" after a line break to denote the end of the file
  - The next shell prompt is displayed
- "ls" demonstrates the existence of the new file

Simpler way with "touch"

- "touch" command
  - Simply create an empty file with a given filename.

```
$ touch a
$ ls –al
-rw-r--r-- 1 kbkim kbkim 0 2012-03-04 05:20 a
$
```

Displaying contents of a file with “cat”

```
$ cat shopping_list
cucumber
bread
Yoghurts
$
```

- The text in a file is displayed immediately
  - Starting on the line after the command
  - Before the next shell prompt
- What happens if the file size is too big?

“more” and “less”

- more command
  - Print the contents fit to a screen
  - Scroll down only
    - Enter key : one line scroll
    - Space key : next screen scroll
    - q key : quit more
- Less command
  - Similar to more, but support scroll up and down
    - j key : next line
    - k key : previous line
    - Space key, or “ctrl+f” : next screen scroll
    - “ctrl+b” : previous screen scroll

“head” and “tail”

- To check the first or last part of files
  - Print first or last 10 lines of each file
    - Adjust the number of lines with “-n” option
  - Great to check large sized files
    - Such as log files
- “tail –f file”
  - Continuously check the tail of a file
  - Good for real-time checking logfiles

```
$ tail /var/log/syslog
~~~~~
$ tail –n 30 /var/log/syslog
~~~~~
$
```

Deleting Files with “rm”

```
$ rm –i shopping_list
rm: remove regular file `shopping_list’? y
$ rm –f shopped_list
$
```

- Simply pass the name of the file to be deleted as an argument
  - “–i” option : interactively remove
  - “–f” option : forcefully remove
  - “–r” option : recursively remove
- The file and its contents are removed
  - There is no recycle bin!!
  - There is no “unrm” thing!!
- The “ls” command can be used to confirm the deletion

Copying and renaming files

```
$ cp cv.pdf new-cv.pdf
$ mv cv.pdf new-cv.pdf
$
```

- “cp” command
  - To copy the contents of a file into another file
- “mv” command
  - To rename (or move) a file
- For both commands, the existing name is specified as the first argument and the new name as the second
  - If a file with the name already exists, it is overwritten
  - For Interactive mode, use “-i” option

Filename Completion

- The shell can make typing filename easier
- Once an unambiguous prefix has been typed, pressing "Tab" button will automatically **type** the rest
  - $ rm sho
  - Pressing "Tab" button may turn it into this:

    $ rm shopping_list
- this also works with command names

Specifying Files with Wildcards

- Use the “*” wildcard to specify multiple filenames to a program
- The shell expands the wildcard, and passes the full list of files to the program
- Just using “*” on its own will expand to all the files in the current directory
  - Except the hidden ones
- Glob and Globbing
  - Glob: Names with wild cards
  - Globbing : Process of expanding them

```
$ ls –l *.txt
-rw-rw-r-- 1 fred users 108 Nov 16 13:06 report.txt
-rw-rw-r-- 1 fred users 345 Jan 18 08:56 notes.txt
```

File name with space

- If a filename contains spaces, or characters which are interpreted by the shell (such as *), put single quotes around them

```
$ rm ‘Beatles – Strawberry Fields.mp3’
$ cat ‘* important notes.txt *’
```

Hidden Files

The special "." and ".." directories do not show up when you do "ls" without any option.

- They are **hidden files**

- Files whose names start with "." are considered **hidden**
- To display use "**ls –a**" (all option)
- Hidden files are often used for configuration files

```
$ ls –a
.      ..      .bashrc      .profile      report.doc
$ ./a.out
```

Directories

- A directory is a collection of files and/or other directories
  - Directory hierarchy
    - subdirectories
  - In windows, it is called as “folder”
- The top level of the hierarchy is the “root directory”
  - The root directory is represented as “/”

Making and Deleting Directories

- “mkdir” command
  - Makes new and empty directories
  - –p : create new intermediate directories
    - e.g.) mkdir –p a/b/c ➔ mkdir a a/b a/b/c
- “rmdir” command
  - Deletes an *empty* directory
  - --ignore-fail-on-non-empty : ignore the error
  - –p : remove directory and its ancestors
    - e.g.) rmdir –p a/b/c ➔ rmdir a/b/c a/b a
  - CAREFUL there is no way to recover!!!

Copy/Move directories

- cp –r old_dir new_name_dir
  - Copy old directory to new directory
  - Including files and directories under old_dir
- cp –r old_dir existing_dir
  - Copy old directory under the existing directory
- cp file dir
  - Copy a file under the directory
- mv old_dir existing_dir
  - Move old directory under existing dir
- mv old_dir new_name_dir
  - Change the name of old_dir to new_name_dir
- mv file dir
  - Move a file under the directory

Current Directory

- Your shell has a current directory
  - The directory in which you are currently working
- Commands such as “ls” use the current directory if none is specified
- “pwd” (print working directory) command
  - See what your current directory
- “cd” command
  - Usage : cd [path]
  - Change the current directory
  - Without specifying a path to get back to **your home directory**

```
$ cd /mnt
$ pwd
/mnt
```

Special Dot Directories

- Every directory contains two special filenames which help making relative paths
- “..” directory : Parent Directory
- “.” directory : the directory it is in

```
$ pwd
/home/kbkim
$ cd ..
$ pwd
/home
```

```
$ cd ..
$ pwd
/
$ cd ./home
$ pwd
/home
```

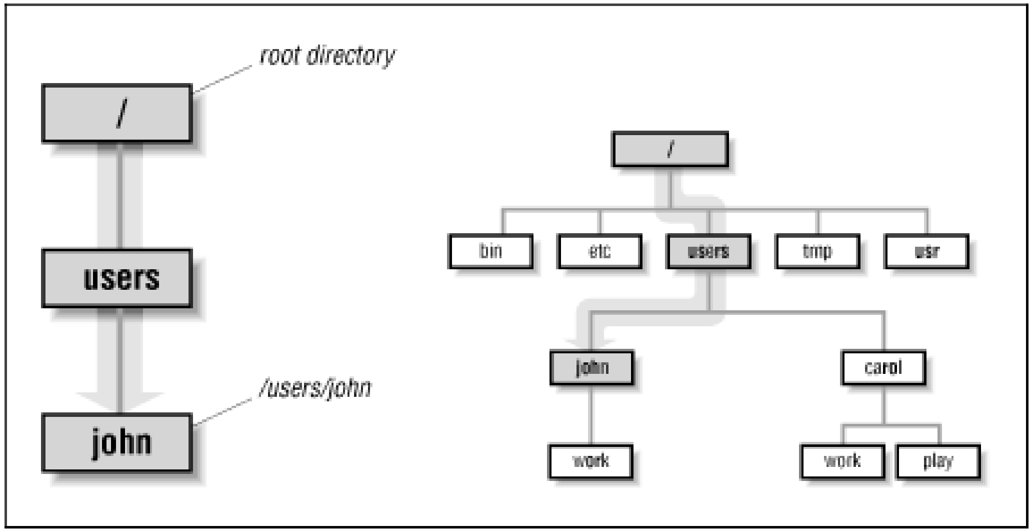
Using Dot Directories in Paths

- The ".." and "." directories can be used in paths just like any other directory name
- Usually used to go back several directories from the current directory
- "." most commonly used on the current directory

```
$ cd ../../../../far-away-directory/
$
```

Path

- Files and directories can be named by a path
  - A path shows programs how to find their way to a file or a directory
  - The root directory is referred to as “/” (slash)
    - Cf) “₩” **(backslash)** in MS Window
  - Other directories are referred to by name, and their names are separated by “/”
- If a path refers a directory, it can end in “/”



Absolute Path

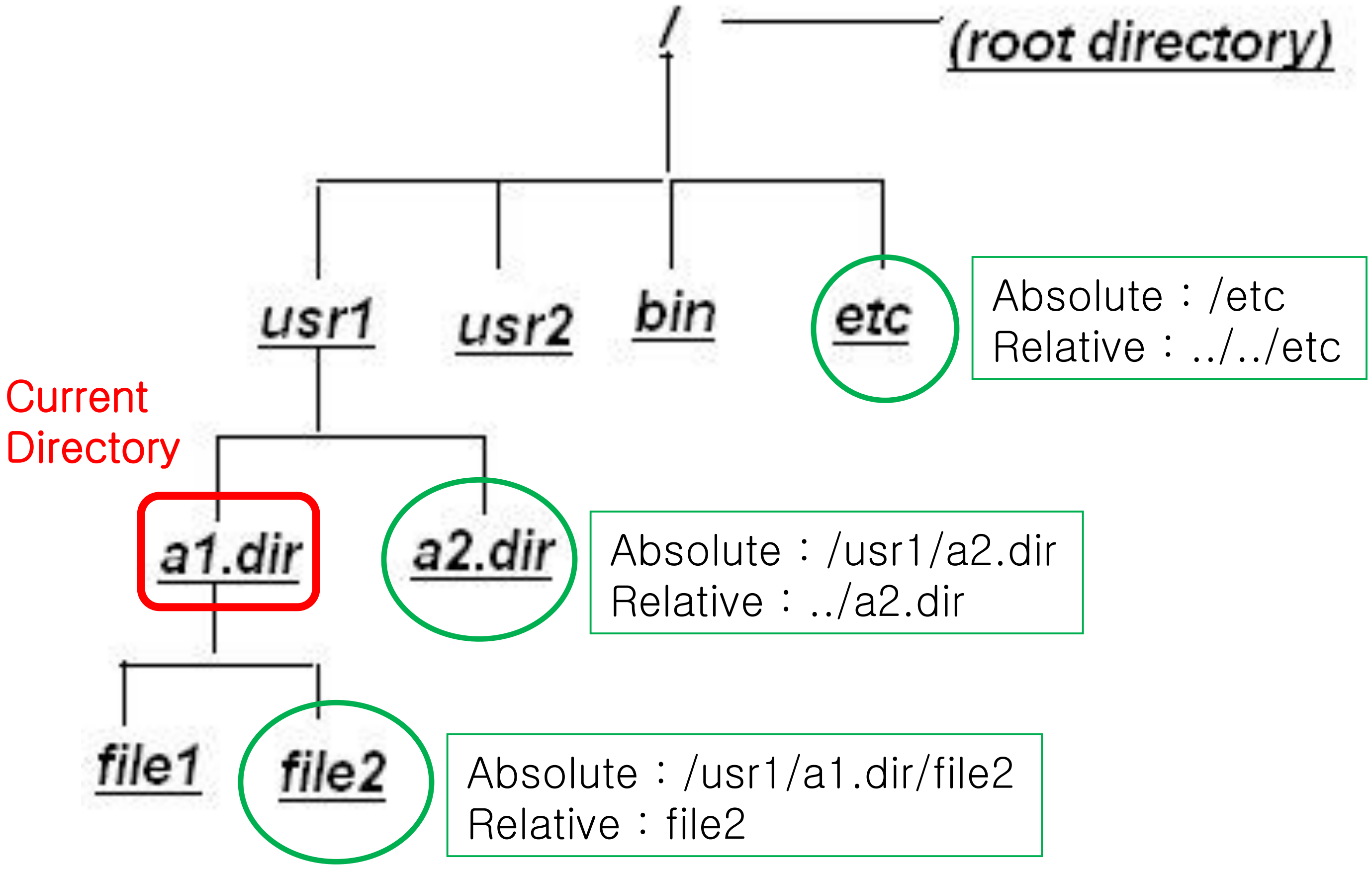
- An absolute path **starts at the root of the directory hierarchy**, and names directories under it
  - E.g) **/etc/hostname, /home/kbkim/**
  - Meaning the file called “hostname” in the directory “etc” in the root directory
- We can use “ls” to list files in a specific directory by specifying the absolute path

Relative Path

- A **path which does not start with "/"**
  - Paths do not have to start from the root directory
  - It is relative to some other directory, usually *current directory*
- Relative paths specify files inside directories in the same way as absolute ones

```
$ cd /usr/share/doc
$
$ cd /
$ cd usr
$ cd share/doc
```

Relative vs Absolute



Absolute : /etc
Relative : ../../etc

Absolute : /usr1/a2.dir
Relative : ../a2.dir

Absolute : /usr1/a1.dir/file2
Relative : file2

Paths to Home Directory

- The symbol “~” (tilde) is an abbreviation for your home directory
  - The “~” is expanded by the shell, so program only see the complete path
  - You can get the paths to other users’ home directory by concatenating “~” and a user account

```
(user kbkim)
$ cd /home/kbkim/documents/
$ cd ~/documents/
$ cat ~mglee/data.txt
```

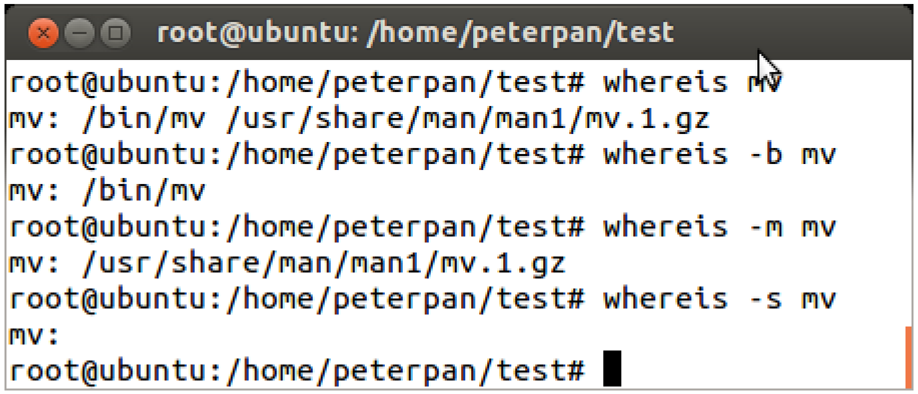
Looking for files in the system

- "locate" command
  - List files which contain the pattern you give in the filename
    - Lookup the databases prepared by "updatedb"
  - Pattern can contains globbing characters
  - Very useful for finding files when you do not know exactly where they are stored

```
$ locate mkdir
/usr/man/man1/mkdir.1.gz
/usr/man/man2/mkdir.2.gz
/bin/mkdir
…
```

Searching the directory of a command

- "which" command
  - Search command file from PATH environment
    - e.g.) which cp
- "whereis" command
  - Search command file from source/binary and manual sections.
    - –b: binary files, e.g.)
    - –m : manual files
    - –s : source files

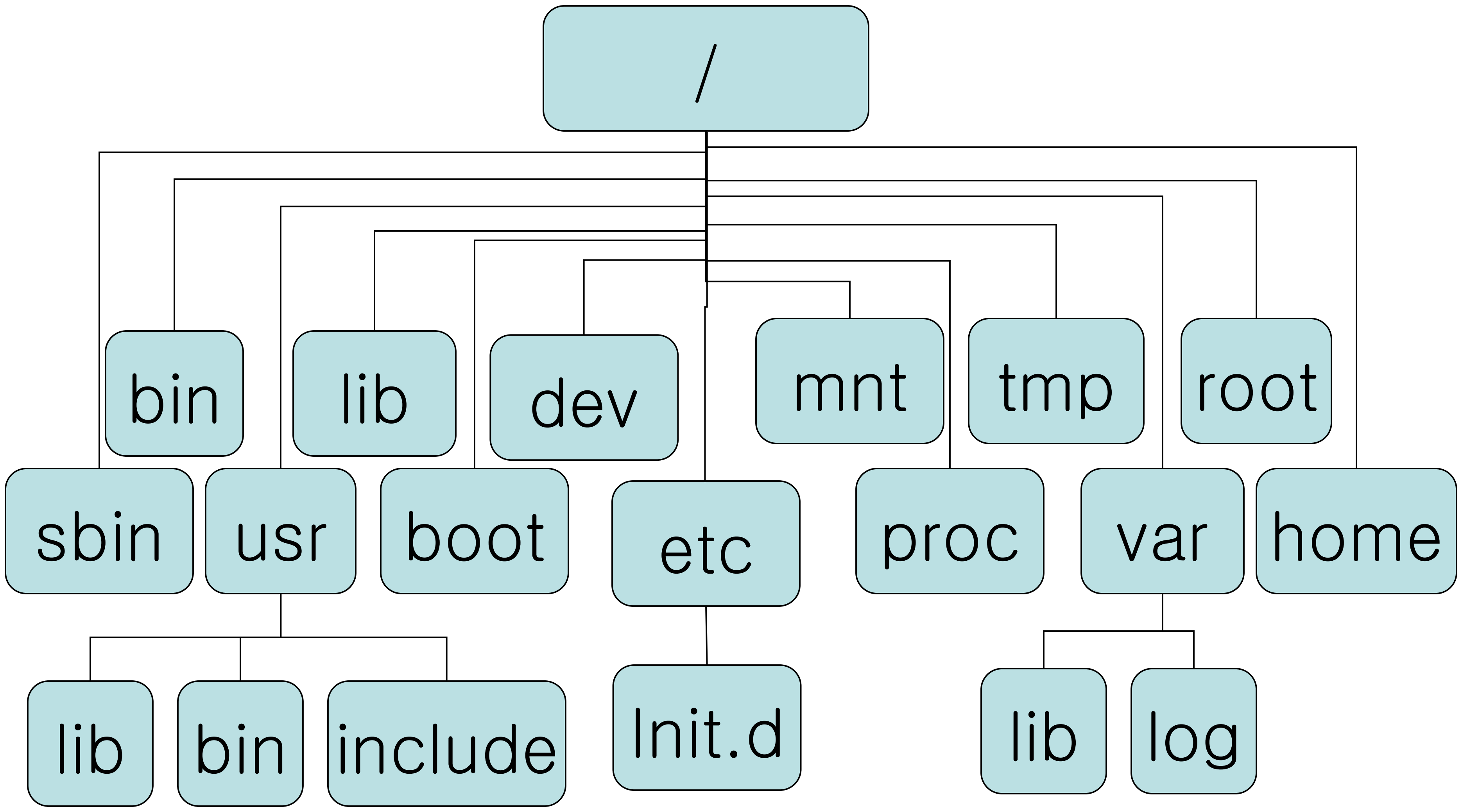


```
root@ubuntu:/home/peterpan/test# whereis mv
mv: /bin/mv /usr/share/man/man1/mv.1.gz
root@ubuntu:/home/peterpan/test# whereis -b mv
mv: /bin/mv
root@ubuntu:/home/peterpan/test# whereis -m mv
mv: /usr/share/man/man1/mv.1.gz
root@ubuntu:/home/peterpan/test# whereis -s mv
mv:
root@ubuntu:/home/peterpan/test#
```

find command

- Find files from given locations with given conditions
  - find [path condition] [operation]
    - –name filename : find by file name
    - –type filetype : find by file type
    - –user loginID : find by user
    - Operation
      - –print → print the absolute path of files (default op)
        - » e.g.) find . –name test
      - –ls → print files in detail
        - » e.g.) find . –name test –ls
      - –exec command {} ₩; → execute on the files
        - » e.g.) find . –name xyz –exec rm {} ₩;
      - –ok command {} ₩; → execute on the files interactively
        - » e.g.) find . –name xyz –ok rm {} ₩;

Linux Directories



Linux Directories (cont')

- /
  - Root (root partition)
- boot
  - Files used by the bootstrap loader, LILO. Kernel images are often kept here.
- bin
  - Utility Commands needed by normal users (cp, cat, ls, …)
- sbin
  - Like bin but commands are not intended for normal users. Commands run by LINUX (administrator permission)
- dev
  - Device files for devices such as disk drives, serial ports, keyboard, console etc.
- mnt
  - Mount points for temporary mounts by the system administrator.

Linux Directories (cont')

- etc
  - Configuration files specific to the machine.
  - Init.d subdirector : configurations of initialization
- var
  - Contains files that change for mail, news, printers log files, man pages, temp files
  - lib subdirectory
  - local subdirectory : var for /usrs/local programs
  - log subdirectory : log files
  - lock subdirectory : lock files
- tmp
  - Temporary files. Programs running after bootup should use /var/tmp.

Linux Directories (cont’)

- lib
  - Shared libraries needed by the programs on the root filesystem
- usr
  - Acronym of Unix System Resource
  - Contains all commands, libraries, man pages, games and static files for normal
  - Similar to “Program Files” in MS window
  - Bin subdirectory : executable files for normal users
  - Sbin subdirectory : executable files for administrator
  - Include subdirectory : header files
  - Lib subdirectory : libraries

Linux Directories (cont')

- proc
  - This filesystem is not on a disk.
  - Exists in the kernels imagination (virtual).
  - This directory Holds information about kernel parameters and system configuration.
- home
  - Contains the user's home directories
- root
  - The home directory for the root user

Compressing

- Sometimes you need to save the disk space
  - Compressing files and directories
- Tools to compress and decompress files and directories
  - gzip(gunzip)
  - bzip2(bunzip2)
  - zip(unzip)
  - tar

gzip and gunzip

- gzip compress the size of the given files using Lempel-Ziv coding.
- Compressing
  - gzip [filename/directory]
  - The file/directory is replaced by one with the extension .gz
- Decompressing
  - gzip –d [.gz file]
  - gunzip [.gz file]

bzip2 and bunzip2

- bzip2 compress the size of the given file using Burrows-Wheeler block sorting text compression algorithm and Huffman coding
- Compressing
  - bzip2 [filename/directory]
  - The file/directory is replaced by one with the extension .bz2
- Decompressing
  - bzip2 –d [.bz2 file]
  - bunzip2 [.bz2 file]

zip and unzip

- zip is a compression and file packaging utility for Unix/Linux
- Each file is stored in a single .zip file
- Compressing
  - zip [.zip-filename] [filename-to-compress]
- Decompressing
  - unzip [.zip file]

tar command

- GNU tar : archiving utility
- Files are archived in a single .tar file
- Compressing
  - tar –cvf [.tar file] [files]
- Listing
  - tar –tf [.tar file]
- Decompressing
  - tar –xvf [.tar file]

tar command with gzip/bzip2

- Tar supports both archive compressing through gzip and bzip2
  - –z : using gzip ➔ .tgz file
  - –j : using bzip2 ➔ .tbz2 file
- Compressing
  - tar –zcvf [.tgz–file] [files]
  - tar –jcvf [.tbz2–file] [files]
- Decompressing
  - tar –zxvf [.tgz–file]
  - tar –jxvf [.tbz2–file]