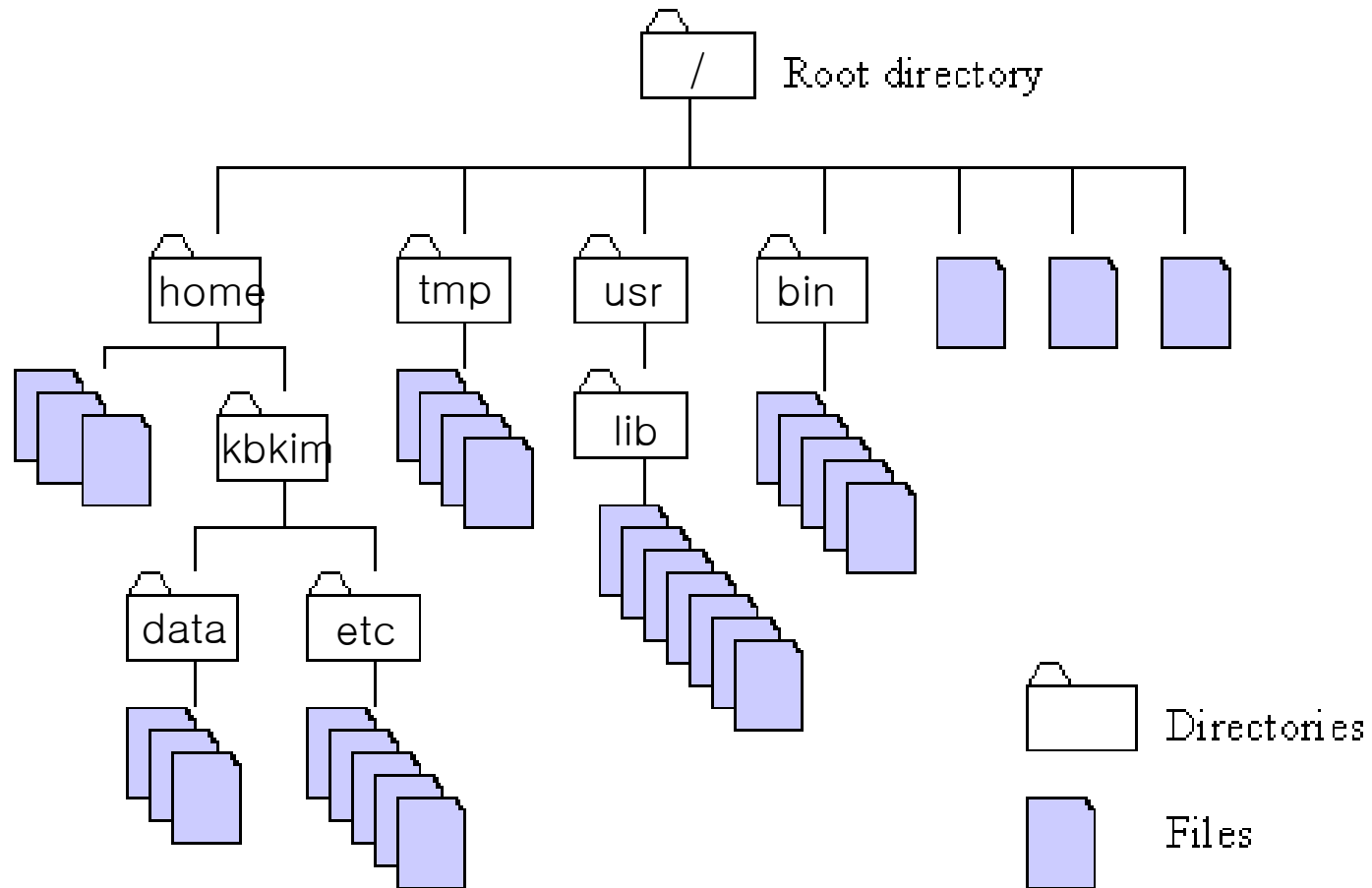


Files and Directories

Chonnam National University
School of Electronics and Computer
Engineering

Kyungbaek Kim

Overview



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*
- The “ls” command lists the names of files

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
- e.g.) compare the results between “ls”, “ls -a”, “ls -al” and “ls -l”

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 Files' contents 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?

“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 /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 old-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

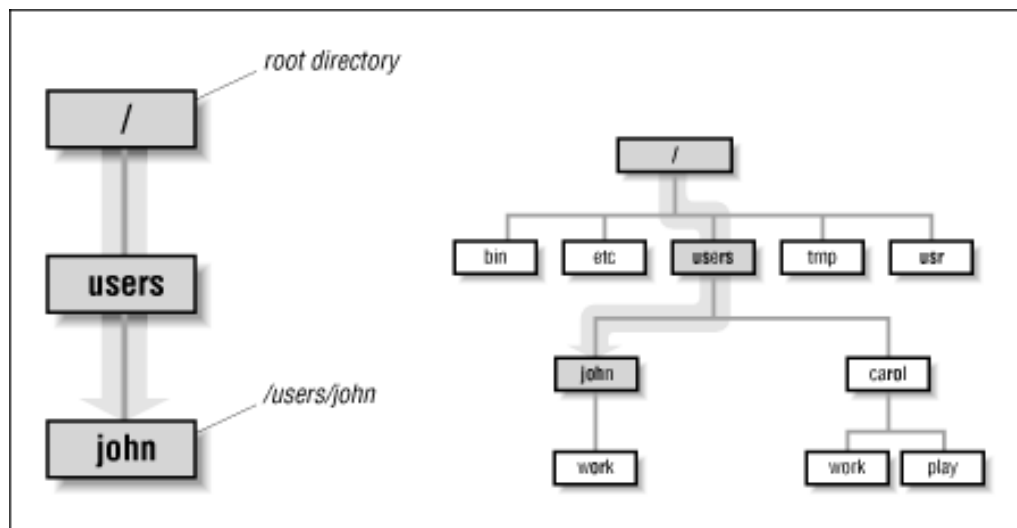
```
$ rm 'Beatles - Strawberry Fields.mp3'  
$ cat '* important notes.txt *'
```

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”

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 “/”
 - Cf) “\” (backslash) in MS Window
 - Other directories are referred to by name, and their names are separated by slash (“/”)
- 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 **the 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
```

Current Directory

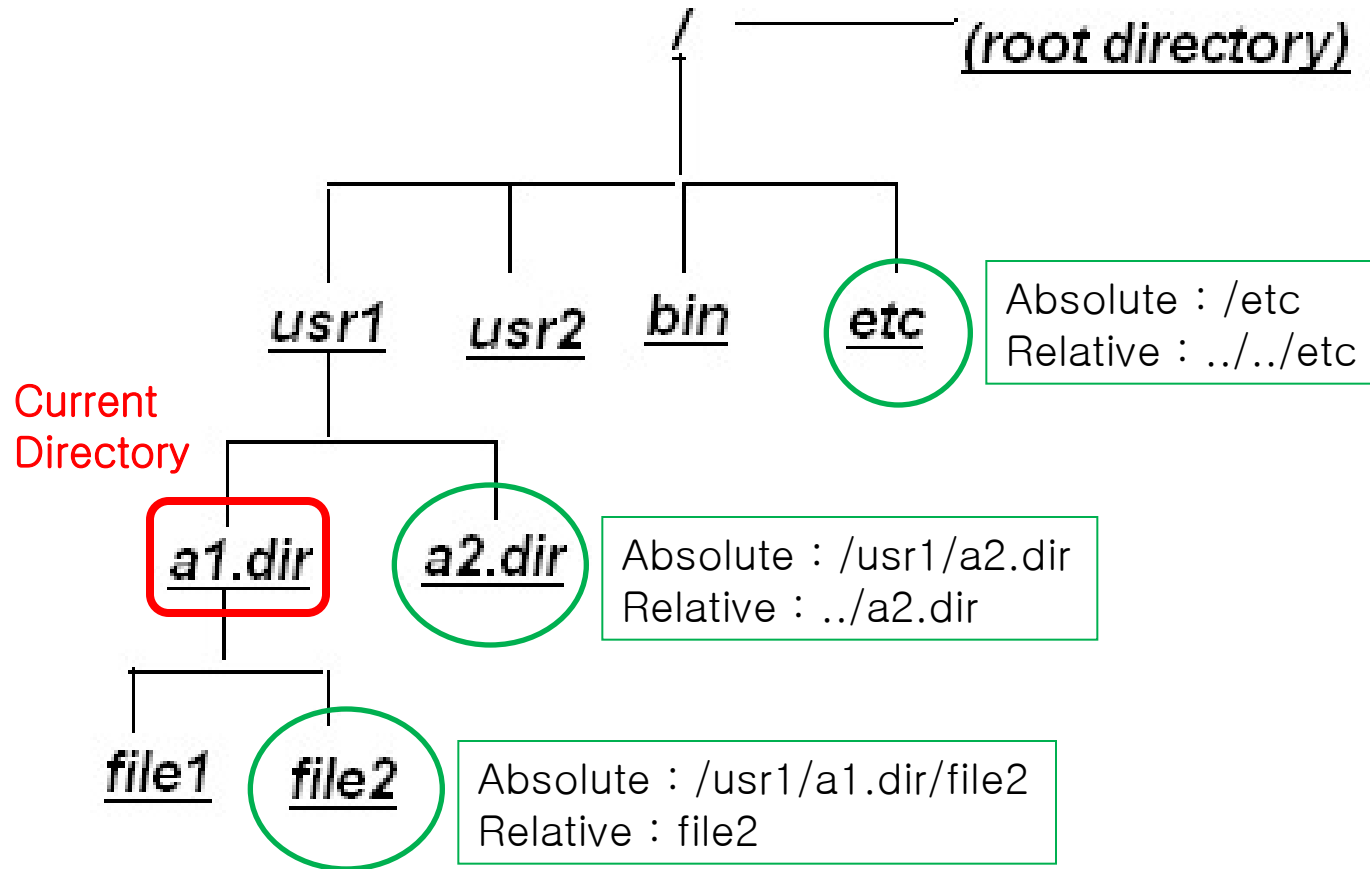
- Your shell has a current directory
 - The directory in which you are currently working
- Commands such as “ls” use the current directory if none is specified
- “pwd” (print working directory) command
 - See what your current directory
- “cd” command
 - Change the current directory
 - Without specifying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

```
$ cd /mnt/cdrom
```

```
$ pwd
```

```
/mnt/cdrom
```

Relative vs Absolute



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/  
$
```

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
```

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
```

Making and Deleting Directories

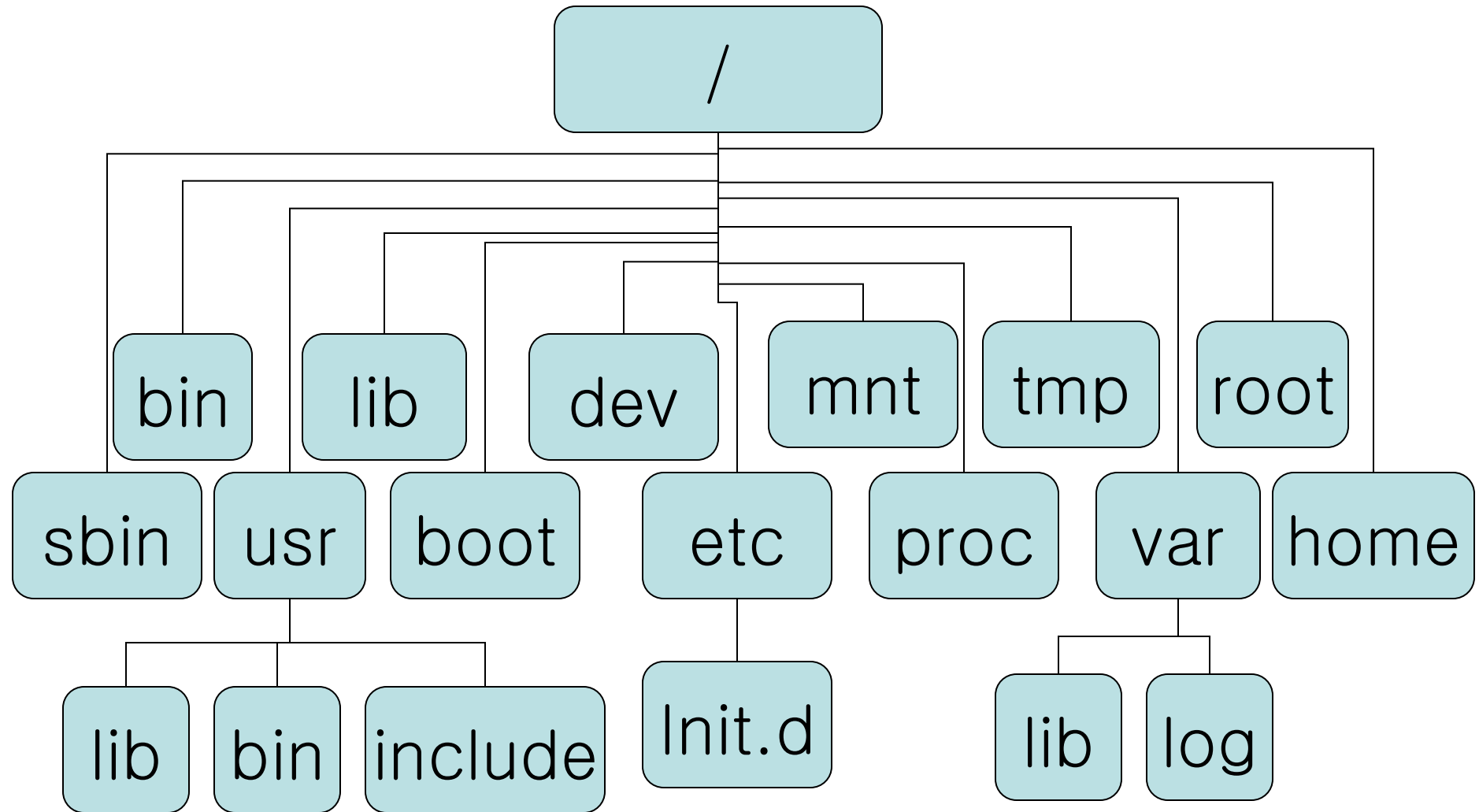
- “mkdir” command
 - Makes new and empty directories
- “rmdir” command
 - Deletes an *empty* directory
 - “-r” (recursive) option to delete the directories and all the files and directories they contains
 - “-i” option : interactive mode
 - “-f” option : forcefully delete
 - **CAREFUL there is no way to recover!!!**

Looking for files in the system

- “locate” command
 - List files which contain the pattern you give in the **filename**
 - 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
...
```

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 /usr/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
 - 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
- Each file is stored in a single .tar file
- Compressing
 - tar -cvf [.tar file] [files]
- 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]`