LINUX 作業系統實務
07. File System

2020 TKU

Sherry Yin

File types

- Ordinary file – regular file, contains only data as stream of characters
- Directory file – a folder containing the names of other files and subdirectories as well as a number associated with each name
- Device file – a device or peripheral.

Regular file

- Text file: contains only printable characters.
  - Lines: terminated with the linefeed (LF), also known as newline
    - cat –e: display the newline character
    - od: makes all characters visible
- Binary file: contains both printable and nonprintable characters within ASCII range (0 to 255).

Directory file

- Contains details of the files and subdirectories under it.
  - The filename
  - A unique identification number (the inode number)

Device file

- Contains nothing at all
- Its attributes are not stored in itself, but somewhere else.

File name

- Up to 255 characters
- No ‘/’ or NULL character (ASCII value 0)
- Avoid using unprintable characters or “$ ~ ? * &”
- Try to use only:
  - Alphabetic and numerals
  - The period (.), hyphen (-) and underscore (_).
- Never use a – at the beginning of a filename.

File system hierarchy

- Root directory (/) is different from the user-id root.
- When access a file in the current directory, the first / should be dropped.
- Thus, cat /progs/foo.c is different from cat progs/foo.c

Unix file system – first group

- /bin and /usr/bin always in PATH variable, where all the commonly used commands (binaries, hence the name bin)
- /sbin and /usr/sbin usually only system admin can execute
- /etc configuration files of the system, /etc/passwd
- /dev contains all the device files
- /lib and /usr/lib library files in binary form
- /usr/include standard header files used by C programs. # include <stdio.h>
- /usr/share/man stores the man pages. Sub directories man1, man2…

Unix file system – second group

- /tmp users are allowed to create temporary files
- /var the variable part of the file system, contains all the print jobs and outgoing/incoming mails.
- /home users are housed here. /home/romeo for the user romeo.

HOME

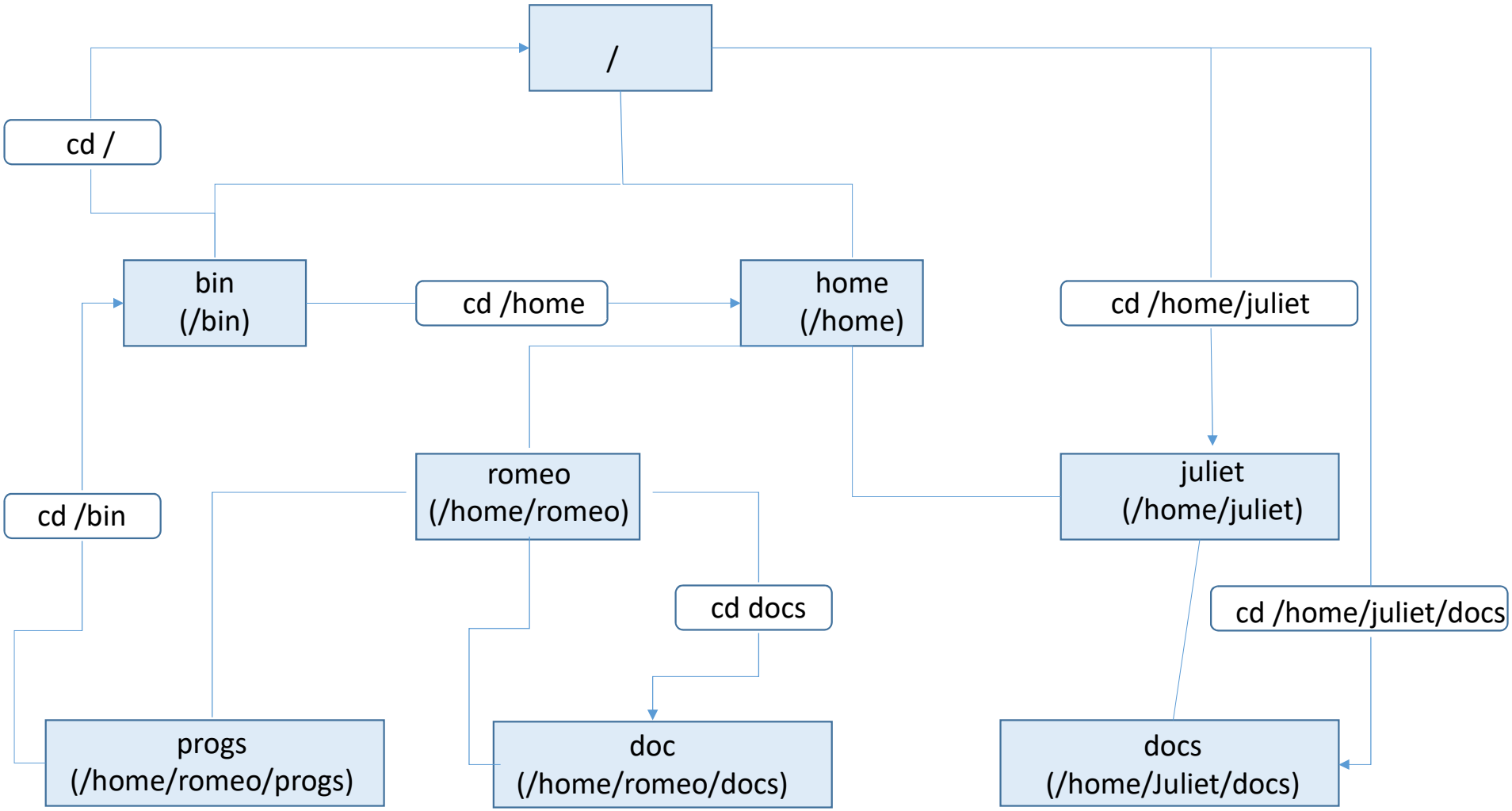
$ echo $HOME

/home/romeo

- This is the user's home directory.
- It is set in /etc/passwd at the time of opening a user account.
- ~ is used to refer to the home directory.
  - ~/ refers to one's own home directory
  - ~Juliet refers to the home directory of Juliet
  - Use $HOME/ or ~/ rather than the absolute path in order to move scripts easier.

pwd and cd: navigating the file system

- pwd: displays the absolute pathname of the current directory.
- cd: **c**hange **d**irectory
  - cd: change to your home directory
  - cd progs: change subdirectory to progs under the current directory.
  -

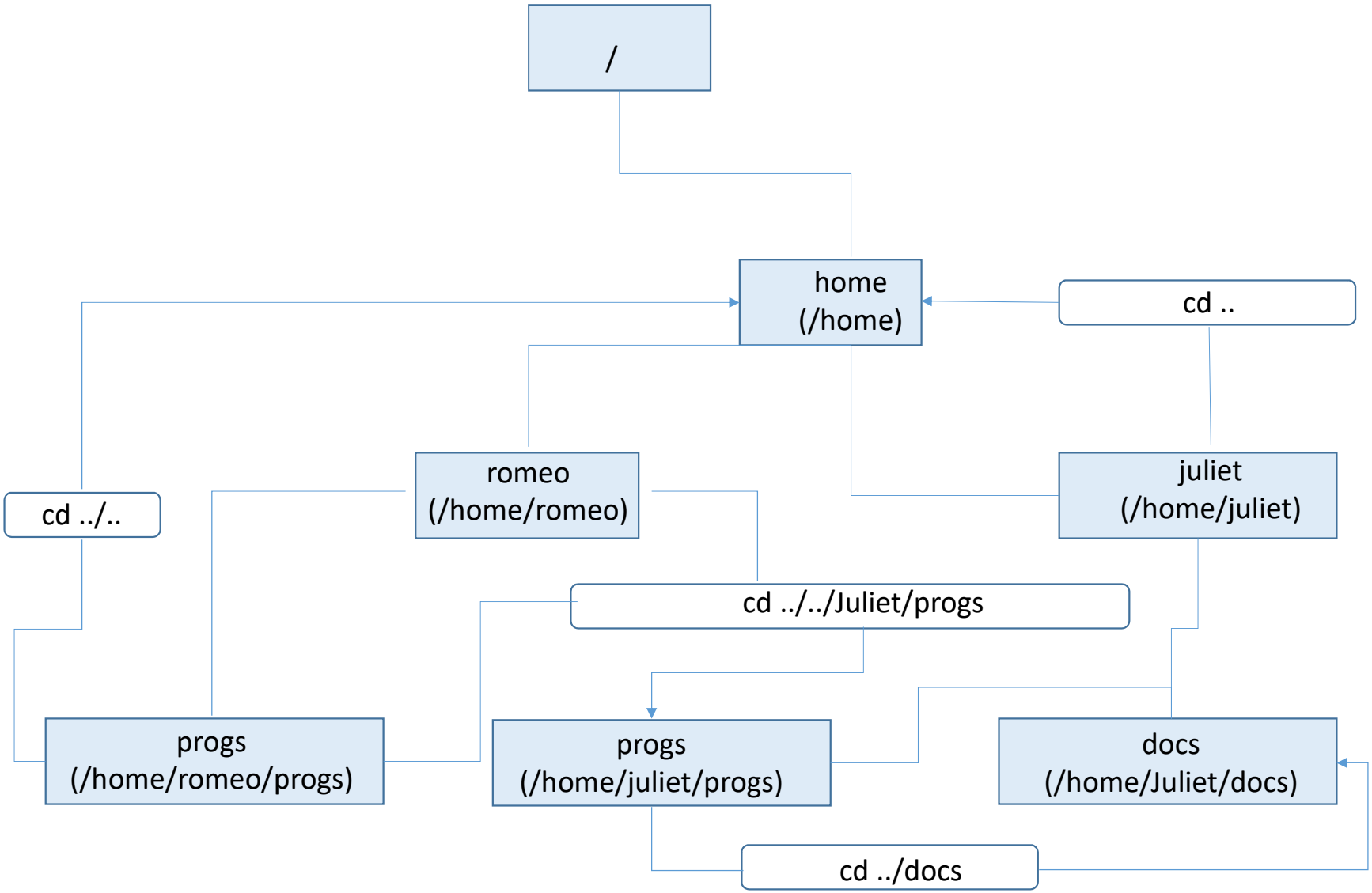

/
cd /
bin
(/bin)
cd /home
home
(/home)
cd /home/juliet
cd /bin
romeo
(/home/romeo)
juliet
(/home/juliet)
cd docs
cd /home/juliet/docs
progs
(/home/romeo/progs)
doc
(/home/romeo/docs)
docs
(/home/Juliet/docs)

. and ..

- . (a single dot): the current directory
- .. (two dots): the parent directory
- Pathnames that begin with either of these two simbols above are relative pathnames.
- cd progs = cd ./progs
- cat != ./cat
- Make use of type, which or whereis to check whether your program has a duplicate somewhere else.
- cd ../../.. (the .. on the right of the / is the parent of the ..on the left.

Use of the ..

- cp /home/Juliet/addressbook.sam . (copy file to the current firectory)
- cp addressbook.sam .. (copy file from the current directory to the parent directory.


/
home
(/home)
cd ..
cd ../..
romeo
(/home/romeo)
juliet
(/home/juliet)
cd ../../Juliet/progs
progs
(/home/romeo/progs)
progs
(/home/juliet/progs)
docs
(/home/Juliet/docs)
cd ../docs

mkdir: making directories

- mkdir patch: creates a directory named patch
- mkdir patch patch2 patch3: creates three directories
- mkdir progs progs/include progs/lib: creates a directory tree
  - Note: the sequence here is **important**
- Possible error (mkdir: Failed…; Permission denied) causes :
  - Directory/file with the same name already existe
  - Permission denied (e.g. in /bin, /etc, /home/otherUser)
  - No space left on the file system

rmdir: removing directories

- rmdir patch: deletes a directory named patch (empty directories)
- rmdir patch patch2 patch3: deletes three directories (empty directories)
- rmdir progs/lib progs/include progs: deletes a directory tree
  - Note: the sequence here is **important (reverse from the mkdir)**
- Two things must be fulfilled:
  - The directory is empty
  - Your current directory is above the target directory (so rmdir . will not work)

Behind the scene – mkdir, rmdir

Filename	Inode Number
.	386444
..	417585
foo	499770

mkdir progs →

Filename	Inode Number
.	386444
..	417585
foo	499770
progs	162112

rmdir progs →

Filename	Inode Number
.	386444
..	417585
foo	499770

ls: listing files

- ls: lists file names
- ls calendar /bin/perl (check whether these two exist)
  calendar (displays the name of the file)
  /bin/perl: No such file or directory
- When ls a directory, it displays its contents. Unless you add the -d option.
- alias ls='ls --color=tty'

ls command options

https://www.rapidtables.com/code/linux/ls.html#options

option	description
`ls -a`	list all files including hidden file starting with '.'
`ls --color`	colored list [=always/never/auto]
`ls -d`	list directories - with ' */'
`ls -F`	add one char of */=>@| to enteries
`ls -i`	list file's inode index number
`ls -l`	list with long format - show permissions
`ls -la`	list long format including hidden files
`ls -lh`	list long format with readable file size
`ls -ls`	list with long format with file size
`ls -r`	list in reverse order
`ls -R`	list recursively directory tree
`ls -s`	list file size
`ls -S`	sort by file size
`ls -t`	sort by time & date
`ls -X`	sort by extension name

cp: copying files

- It creates the destination file if it does not exist
- It also overwrites any existing files (so better use ls to check before cp)
- cp fork.c progs/fork.c.bak (copied to .bak under progs)
- cp fork.c progs (creates file with same name under progs)
- cp file1 file2 file3 dir (the last entry must be an existing directory, cp will not create directories)
- cp file* dir (copy all the files name starting with ‘file’)

ls command options

https://www.rapidtables.com/code/linux/cp.html

option	description
`cp -a`	archive files
`cp -f`	force copy by removing the destination file if needed
`cp -i`	interactive - ask before overwrite
`cp -l`	link files instead of copy
`cp -L`	follow symbolic links
`cp -n`	no file overwrite
`cp -R`	recursive copy (including hidden files)
`cp -u`	update - copy when source is newer than dest
`cp -v`	verbose - print informative messages

mv: renaming files

- mv fork.txt fork.c (creates or overwrites destination)
- mv dir1 dir2 (rename a directory)
- mv *.avi *.xvid (**not working**)
- rename *.avi *.xvid (**only works in Linux, not Unix**)

option	description
`mv -f`	force move by overwriting destination file without prompt
`mv -i`	interactive prompt before overwrite
`mv -u`	update - move when source is newer than destination
`mv -v`	verbose - print source and destination files
`man mv`	help manual

rm: deleting files

- rm chap01 chap02 chap03 (**be careful** with rm chap*)
- rm * (does not include filenames beginning with a dot, so it leaves all the hidden files undeleted.)
- rm –rf * (deletes everything in the current directory and below, **extremely dangerous**)
- **Do NOT** run rm –rf * in the / directory, it wipes out the entire system!
- To avoid mistakes, run: alias rm='rm -i'

cat: displaying and concatenating files

```
$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
…
```

cat cont.

- cat foo.c foo1.c foo2.c > foo4.c (save these files' contents in foo4.c)
- For executable files, cat will only show junk.
- For large files, use **more** to view it instead of **cat**

  more /etc/inetd.conf (press q to exit)

  (at the bottom, you will see the filename and percentage of the file that has been viewed.

  f or the spacebar: one page forward

  b: one page back

  10f: scrolling forward by 10 pages

  .: the dot command repeats the last command you used)

more vs less

- Both are like vi, j for one line up, k for one line down
- less can search for a pattern in reverse direction
  - ?ftp (searches backwards)
- less cannot repeat the last command by using . (dot)

wc: counting line, words and characters

$ wc filename

```
        3       20      103    filename
```

(3 lines, 20 words and 103 characters)

- A character can be a space, tab or newline
- wc -l filename (count lines)
- -w: count words
- -c: count characters
- wc works on data stream as well

lp: printing a file

$ lp file1.txt

request id is xxxx (1 file)

$ lp –dlaser file1.txt (printer name is laser)

$ lp –n3 –m file1.txt (prints 3 copies and mails user a message after the file has been printed)

- cancel laser: cancels current job on printer laser
- Cancel xxxx: cancels job with request-id xxxx

od: viewing nonprintable characters

$ od –bc /bin/cat | more

pipe the octal value of executable /bin/cat, and pipe the output to **more**

each line displays 16 bytes of data in octab

Output:

0000000 177 105 114 116…

177 E L F …

(All C executables have the same first four characters)

dos2unix, unix2dos, and Tofrodos: converting between DOS and UNIX

- Windows files (DOS files): end of line is CR (\r) and LF (\n)
- UNIX files: only LF
- Thus, conversion of the DOS file to UNIX is just a simple matter of removing the \r.
- Never perform this conversion on a binary file.

tar (tape archiver): the archival program

- -c: creates an archive
- -x: extracts files from archive
- -t: displays files in archive
- -f: specify the name of the archive
- -v: display the progress

$ tar –cvf archive.tar libc.html User_guide.ps

$ tar –xvf archive.tar (extract the two files)

$ tar –tvf archive.tar (displays the files in the tar file)

gzip: the compression program

$ gzip libc.html (it removes the file libc.html and creates libc.html.gz)

- -l: displays the amount of compression achieved
  $ gzip -l libc.html.gz

$ gunzip libc.html.gz (restore the original file)

$ gzip -d libc.html.gz (restore the original file)

$ gzcat, gzmore (or zcat and zmore) to view compressed plain text files.

- gzip archive.tar (archived and compressed, creates archive.tar.gz or archive.tgz)
  - To restore it, use:
    - gunzip archive.tar.gz
    - tar -xvf archive.tar

zip: the compression and archival program

- It combine the compressing function of gzip with the archival function or tar.

$ zip archive.zip libc.html User_Guide.ps

$ cd ; zip –r sumit_home.zip . (recursive compression the home directory)

$ unzip archive.zip

$ unzip -v archive.zip (view the zipped files)

Test

1. Can the files note and Note coexist in the same directory?
2. Switch to the root directory with cd, and then run cd .. followed by pwd. What will happen?
3. What will cat foo foo foo display? (foo is a text file for example)