

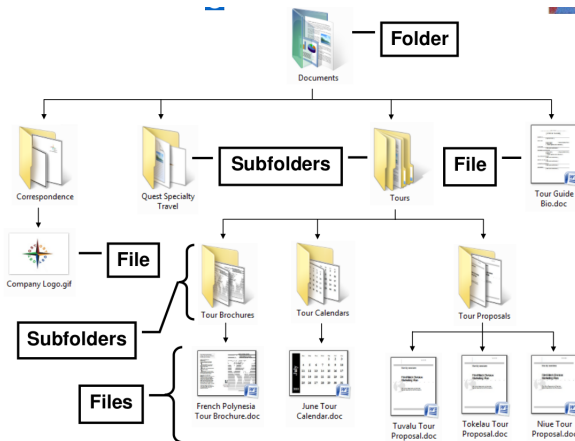
Introduction to UNIX-like systems

Intro. to Computing

`https://mandarm.github.io/computing.html`

File system structure

Files are organised in a hierarchical structure of folders, sub-folders, and files.



Courtesy: <https://www.slideshare.net/okmomwalking/windows-7-unit-b-ppt>

File system structure: terminology

- Folders $\equiv$ *directories*
- Top of the hierarchy: *root directory* (/)
- Default starting location: *home directory* (~)
- Location of a file or directory: specified by *path*
- Current location in terminal or file browser: *current working directory*
- Paths: *absolute* or *relative*
 - absolute path: from root (starts with /)
Example: `/usr/bin/python3`, `/tmp`, `/home/student`
 - relative path: from current working directory (does not start with /)
Example: `Desktop/wsl`

Note the difference between (forward) slash (/, used in Unix-like systems) and backslash (\, used in Windows-like systems) !

Navigating the file system

- **cd** : change directory

Examples:

cd ← go to home directory

cd /home/student (your home directory, same as above)

cd /mnt/c/Users/

- **pwd** : print current working directory

Navigating the file system: special directory names

- `~` : home directory

Example: `cd ~/Desktop`

- `.` (dot) : current working directory

Example: `./program1`

- `..` (dot dot) : parent directory (one level up)

Example: `cd ..`, `cd ../assignment2`, `cd ../..`

- `-` (dash) : go back to the immediately previous location

Example:

`cd /home/student`

`cd /mnt/c/Users`

`cd CSSC`

use your own username instead of CSSC

`cd -`

check location using `pwd`

`cd -`

check location again using `pwd`

Listing files

- `ls` : view list of files in current directory
- `ls <path>` : view list of files in specified path
- `ls -l` : view detailed list of files
- `ls -lt` : view detailed list of files sorted by modification time
- `ls -ltr` : view detailed list of files sorted by modification time *in reverse order*

Managing files from the terminal

Listing files

- `ls` : view list of files in current directory
- `ls <path>` : view list of files in specified path
- `ls -l` : view detailed list of files
- `ls -lt` : view detailed list of files sorted by modification time
- `ls -ltr` : view detailed list of files sorted by modification time *in reverse order*

Example:

```
$ /bin/ls -l
total 68
drwx----- 2 mandar mandar 4096 Jul 19 00:45 assignments
drwx----- 2 mandar mandar 4096 Jul 22 2016 exams
-rw-r--r-- 1 mandar mandar 13521 Jul 19 00:41 index.html
drwx----- 2 mandar mandar 4096 Jul 19 00:45 lectures
```

Managing files from the terminal

■ `cp` : copy a file

```
cp program1.c program2.c
```

```
cp -i source-file target-file
```

```
cp -i source-file target-directory
```

Managing files from the terminal

■ **cp** : copy a file

```
cp program1.c program2.c
```

```
cp -i source-file target-file
```

```
cp -i source-file target-directory
```

■ **mv** : rename (move) a file

```
mv program1.c program2.c
```

```
mv -i source-file target-file
```

```
mv -i source-file target-directory
```

Managing files from the terminal

■ **cp** : copy a file

```
cp program1.c program2.c
cp -i source-file target-file
cp -i source-file target-directory
```

■ **mv** : rename (move) a file

```
mv program1.c program2.c
mv -i source-file target-file
mv -i source-file target-directory
```

-i ≡ interactive
(asks for confirmation)

■ **rm** : remove (delete) a file

```
rm program1.c
rm -i file1 file2.c *.bak
rm -r some-directory (remove directory and everything inside it)
rm -rf some-directory (remove directory and everything inside it;
dangerous!)
```

Creating / deleting directories

- **mkdir** : create a directory

Examples:

```
mkdir clab
```

```
mkdir clab/assignment1
```

Create directories as appropriate.

OR

```
mkdir -p clab/assignment1 clab/assignment2
```

- **rmdir** : remove an (empty) directory

Example: `rmdir assignment2`, `rmdir clab/programs`

Convenience features

Tab completion: type a part of a command / file name, and hit tab to complete or get a list of suggestions.

Aliases

- **What are they?** → user-defined easy-to-remember names for commands (+ arguments)

Examples:

```
alias c="clear -x"
```

```
alias cp="cp -i"
```

```
alias l="ls -l"
```

```
alias ls="ls -F"
```

```
alias lt="ls -lt"
```

```
alias ltr="ls -ltr"
```

```
alias mv="mv -i"
```

```
alias rm="rm -i"
```

- **Where to define them?** → in `~/.bash_aliases`
- **How does one create / edit files?** → `notepad.exe bash_aliases`

(“simple and stupid” method that works only on WSL)

Viewers / pagers

- Useful for quickly viewing a file (not editing)
- Use `less` or `bat`

Example: `less ~/.bash_aliases`

- space: move forward one page
- backspace or b: move backward one page
- q : exit the pager
- / : search for a string in the file
- run `man less` for more information

Input/output redirection

Terminology

- “standard input” or *stdin* $\equiv$ input device, i.e., keyboard
- “standard output” or *stdout* $\equiv$ output device, i.e., terminal

Redirection

- Taking input from a **file**: `python3 prog1.py < input.txt`
- Printing stdout to a **file**: `python3 ./prog1 > output.txt`
- Taking stdin / printing stdout from / to a **program**:
`python3 prog1.py | less` vertical bar / pipe character (|)
- Input/output from/to file / program may be combined
`python prog1.py < input.txt | diff - desired-output.txt > differences.txt`

Searching / installing packages

1. Update list of available packages: `sudo apt update`
(you only need to do this occasionally, or if things don't work as expected)
2. Search for the package name if necessary. Examples:
`apt-cache search less`
`apt-cache search gnome-terminal`
3. Install the package: `sudo apt install <package name>`

Installing useful applications

- **Terminal:** `sudo apt install gnome-terminal -y`
- **Simple text editor:** (for quick edits) `sudo apt install gedit -y`
OR `sudo apt install gnome-text-editor -y`
- **Python**
 - check python version by running `python3 --version`
 - if you are not running Python 3.14
 - `sudo apt install software-properties-common`
 - `sudo add-apt-repository ppa:deadsnakes/ppa`
 - `sudo apt update`
 - `sudo apt install python3.14 python3.14-venv`

Find out more about these on your own.

- `cmp`, `diff` (comparing two files)
- `grep` (searching for patterns)
- `awk`, `sed` (programming)
- `wc` (counting characters, words, lines)
- `sort`
- `head`, `tail` (first few / last few lines)
- `ps`, `top`, `kill` (checking what programs are running)
- `find` (finding files or directories)

Useful references / cheat-sheets

http://cli.learncodethehardway.org/bash_cheat_sheet.pdf

<https://ubuntudanmark.dk/filer/fwunixref.pdf>

<http://www.ucs.cam.ac.uk/docs/leaflets/u5>

<http://mally.stanford.edu/~sr/compuGng/basic-unix.html>

<http://www.math.utah.edu/lab/unix/unix-commands.html>