Editing and Compiling C Programs

Situation: Suppose that you want to download, edit, and run the code `functions.c` (provided in Workshop Week 2).

Here, 3 options are described:

- use lab PCs,
- use your laptop offline, and
- use your laptop online to edit and run codes that reside in your personal folder on the university's computer system. Your uni's folder has the name `H:` or starting with `H:`, and will be referred to as `H:` in this document.

Note: This document is short. If unclear, you should seek help from your classmates and tutors.

Option 1: using lab PC (in labs only!)

- Suppose you already downloaded `functions.c` from LMS, and it is now under Download folder
- Open that `functions.c` in `jEdit` and use Save As to save to `H:`
- Now change `functions.c` as required and Save
- Run `MinGW`, and in the `minGW` terminal run:

Command	Notes
`cd H:`	use your uni's folder as current directory (CDIR)
`ls`	display content of CDIR, you should see functions.c
`gcc –o functions functions.c`	compile, produce functions.exe
`./functions.exe`	run functions.exe
	then, back to jEdit to change/save and then compile, run again.

- Note: In this way, `functions.c` will be kept permanently on your uni's folder `H:`. If you didn't save to `H:`, the file will be automatically deletted.

Option 2: using your own laptop

Aim: be able to edit and run C programs off-line on your own laptop/desktop.

If you can do that already, skip 3 pages, to page 6. Otherwise:

- Make sure to have an editor (such as `jEdit`) installed. You can use any editor such as `Atom`, `emacs`, `jEdit`, `Visual Studio Code`, …
- For Windows: make sure to install `minGW` (or, skip this step if you already use some other similar tool such as `Cygwin`). When installing `minGW`, remember to also mark `msys: open_ssh bin` for installation. Note: if you installed `minGW` already, run `minGW installation manager` to add `msys: open_ssh bin` . This will allow you to use `scp` and `ssh` (see next pages).

Case 2a: If you use a Windows laptop

- Suppose you already downloaded `functions.c` from LMS, and put it your working directory (say. C:\comp20007)
- Open that `functions.c` in `jEdit`
- Run `MinGW`. In this `minGW` terminal run:

Command	Notes
`cd C:` `cd comp20007`	use your C:\comp20007 as current directory (CDIR)
`ls`	display content of CDIR, you should see `functions.c`
`gcc –o functions functions.c`	compile, producing `functions.exe`
`./functions.exe`	run `functions.exe`
	then, back to `jEdit` to change/save and then compile, run again.

Note: In this way, `functions.c` will be kept in your laptop. If something wrong happens to your laptop (oh, I didn't mean that, just if ☺) you will lose `functions.c` and all of your other precious works.

Case 2b: If you use a MacBook

- Suppose you already downloaded `functions.c` from LMS, and opened it in `jEdit`
- Then open a Terminal (ie. run app `Terminal`) and run

Command	Notes

`cd`
 	
`mkdir comp20007`
`cd comp20007`
	use home directory (/Users/your_name) as current directory (CDIR)
	make a new directory
	change CDIR to comp20007
	now, on `jEdit`, use Save As to save in comp20007

`ls`	display content of CDIR, you should see functions.c
`gcc –o functions functions.c`	compile, producing executable file `functions`
`./functions`	run `functions`
	then, back to `jEdit` to change/save and then compile, run again.

Note: In this way, `functions.c` will be kept on laptop. If something wrong happens to your laptop (oh, I didn't mean that, just if ☺) you will lose functions.c and all of your other precious works.

Common for both Case 2a and 2b: backup your files using your uni's H: folder

- You can copy your file to your uni's folder and have a good backup. For that you need to run command `scp` on your Terminal windows

```
scp functions.c your_uni_login_name@dimefox.eng.unimelb.edu.au:
```

(note: there is a colon : at the end). The file will be copied to your uni's H: driver. You can also copy the whole directory comp20007 using:

```
cd ..
scp –r comp20007 your_uni_login_name@dimefox.eng.unimelb.edu.au:
```

You will have a copy of your valuable works in a uni's server! And of course, if you use a lab PC you can edit and run these files directly.

Note: `dimefox` is a university server, after `scp`, you can login into that server to edit/compile/run and debug your codes. You can also use `nutmeg` instead of `dimefox`.

Option 3: use your laptop to edit and run programs in uni's `H:` directly

- At first, note that you can copy files from your laptop to your uni's folder `H:` as described in the previous page.

Tools: 3a. connect to dimefox/nutmeg, why?

Aim: use your laptop/desktop to access the CIS's servers `dimefox` (or `nutmeg`) and also access your uni's folder `H:`

Note:

- Your uni's folder `H:` is a good place to keep your files. You can access the folder from anywhere (with an Internet connection, of course).
- After accessing `dimefox`, you can edit and run C programs, submit assignments, and use various useful tools for this course such as `make`, `valgrind`, `gdb`. Some of these might be not available in your laptop.

Tools: 3a. connect to dimefox/nutmeg, how?

- if you are not on uni's ground, make sure to run `VPN` first
- open Terminal (such as `minGW` terminal, or Mac's `Terminal`)
- on the `Terminal` run:

```
ssh login_name@dimefox.eng.unimelb.edu.au
```

then, your terminal will work with dimefox.

Try some unix commands such as:

- `ls` (list the content of current directory)
- `cd` (change current directory),
- `cp` (copy files, run `man cp` for details)
- `mkdir` (make new directory).

At home: (use Google to) learn some basic Unix commands if needed .

Sample session on dimefox:

```
cd
ls
echo Hello Linux > hello.txt
ls
cp hello.txt hello_1.txt
more hello.txt
ls
man cp
emacs helloworld.c
```

Tools: 3b. Remote editing programs in H:

- *Aim*: Avoid time-consuming filecopy from your laptop to `H:`, be able to edit your uni's files remotely from anywhere.
- Tools: in `dimefox` terminal, using `emacs`/`nano`/`vim` etc. While vim seems inconvenient, `emacs` and `nano` could be mastered quickly.
- quick use of `emacs` (try `emacs hello.c` on `dimefox`'s terminal)
  - relatively simple and easy: just remember Ctrl-x Ctrl-s for "Save", Ctrl-x Ctrl-c for "Quit".
  - Cut/Copy/Delete a chunk of code inside the emacs window: Ctrl-Space to mark the start, arrows to expand, Esc-w to copy, Ctrl-w to cut, Ctrl-y to paste the chunk
  - Undo: Ctrl-x Ctrl-u
  - ESC a few times to exit from some mode/command if having troubles
  - how to use `emacs` more effectively and professionally: https://www.digitalocean.com/community/tutorials/how-to-use-the-emacs-editor-in-linux

FAQ

- `minGW`: I got "command not found" when trying `scp` or `ssh`. Why?
- You just missed it when installing `minGW`. Now you need:
  - run `minGW Installation Manager`
  - When the full dialog box opens with a list of software components, scroll down to find, then right-click "msys-openssh-bin" and click "Mark for Installation".
  - Click "Installation" (top left corner), then "Apply Changes", then "Apply".
  - Wait until installer prompts you with a message indicating completion.
- Go back to your `minGW` window, now you can use `scp` and `ssh`