COMP1521 Week 2

Welcome!!!
Please sit down we shall start at 11:05am

Admin stuff

- Course website (https://cgi.cse.unsw.edu.au/~cs1521/23T1/)
- HELPLINES: course forum/course admins cs1521@cse.unsw.edu.au/me (z5362344@unsw.edu.au)
- Tutorial files (https://github.com/abbylxt/COMP1521-23T1-Tut)
- Email is the best form of contact to me (response rate is within 24hrs normally, max 48hrs)

Overview

- Intro to mips
- Worked examples

Intro to MIPS

mipsy and mipsy_web fun facts

mipsy

- Is a mipsy emulator which simulates the execution of a MIPS CPU and lets you run MIPS assembler on any computer (regardless of native architecture).
- developed at CSE by Zac Kologlu
- You can run mipsy on CSE with the command: `1521 mipsy`

mipsy_web (https://cs1521.web.cse.unsw.edu.au/mipsy/)

- A web-based version of mipsy that is still in very early stages
- Able to run a full MIPS simulator visually in any web browser

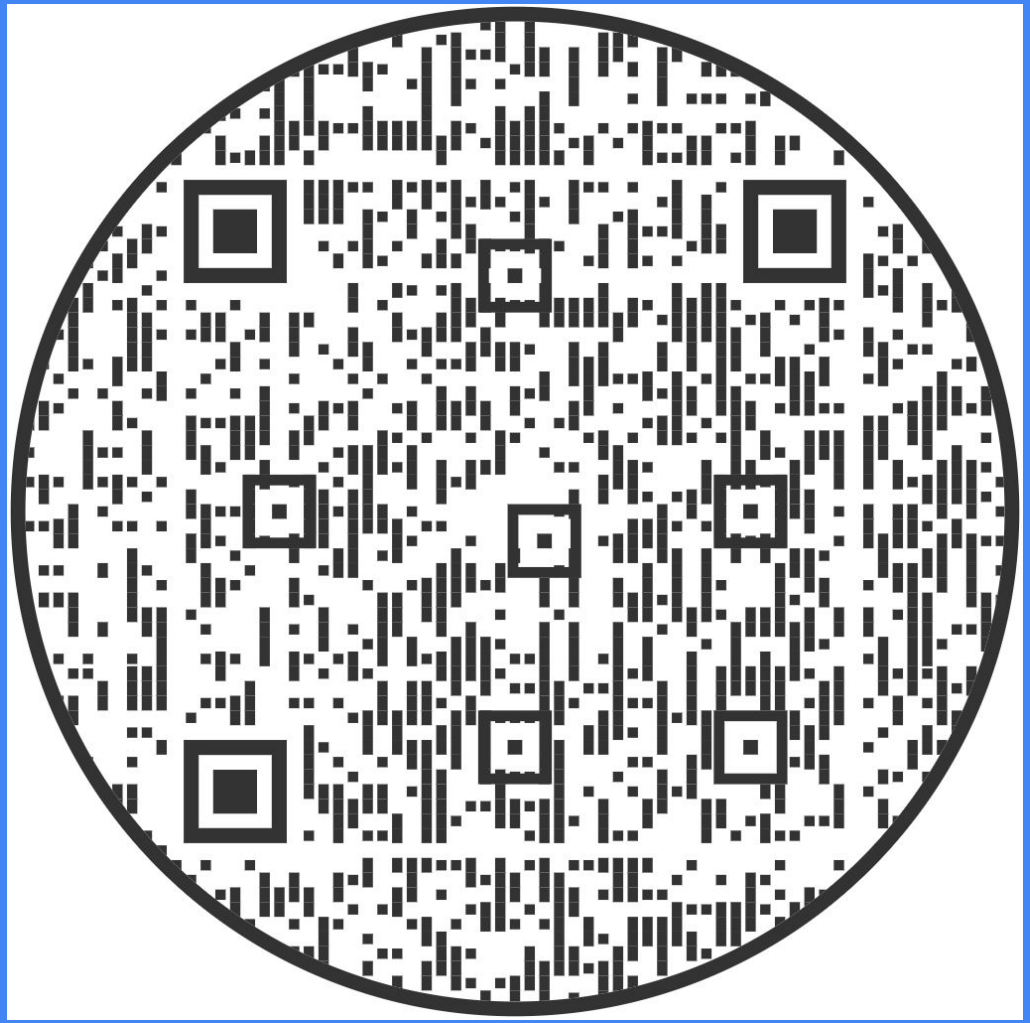
Tutorial Q2-9

Q2-3 Refer to course website

Q4-9 Refer to tutorial code

Clang commands

1. `clang -E x.c`
   Executes the C pre-processor, and writes modified C code to `stdout` containing the contents of all `#include`'d files and replacing all `#define`'d symbols.
2. `clang -S x.c`
   Produces a file `x.s` containing the assembly code generated by the compiler for the C code in `x.c`. Clearly, architecture dependent.
3. `clang -c x.c`
   Produces a file `x.o` containing relocatable machine code for the C code in `x.c`. Also architecture dependent. This is not a complete program, even if it has a `main()` function: it needs to be combined with the code for the library functions (by the linker ld).
4. `clang x.c`
   Produces an executable file called `a.out`, containing all of the machine code needed to run the code from `x.c` on the target machine architecture. The name `a.out` can be overridden by specifying a flag `-o filename`



Code: Pen