Week 01 2

deeota (Demetres) Kostas-Heliokinde

September 14, 2023

Contents

1 Announcements

1.1 Lab 1 is tomorrow

- Go to quercus to check room assignments
  - they are flexible
    - try to use them so that there is space
  - if the TAs ask you to move
    - *please, kindly move*

1.2 Assignment 1 is up!

- Have a read over it
- Start thinking about
  - what are you missing for this?
  - what do you still need to learn?
- Due Sept 27th at 4pm

2 Compilation in C

- Starting the same as Tuesday

2.1 Basic usage of `gcc`

- Open a terminal
  - (which starts a shell program)
- Then, gcc is a built-in program (like `ls` or `cd`)
  - so we just type `gcc`
- It will need arguments
  - using `$ gcc source_code.c`
    - this will compile the `source_code.c`
      - and produce an executable program!

2.2 Additional arguments

- In this course
- We will keep some additional arguments
- We will add `-Wall`
  - to tell us *all possible warnings*
- We will use `-std=gnu99`
  - to use the C99 standard

- We will also use `-o <prog_name>`
  - to specify a specific program name
    - (replace `<prog_name>` as needed)

2.3 So how would we compile?

- If we had a single source file
  - called `source_code.c`
- and wanted to call the resulting program
  - `super-python`
- We would type `gcc -Wall -std=gnu99 -o super-python source_code.c`

2.4 The '-g' argument

- not required yet
- but there's nothing wrong with adding it
- it will allow you to
  - **debug your program with gdb**
    - the debugger that goes with gcc

3 Permissions

- every file in *NIX has metadata
- part of this metadata relates to
  - users (who created and responsible for files)
  - and permissions
    - who can read, write and execute files

3.1 ls -l and reading permissions

- recall the `ls` program
  - in particular, the longer output
  - when we passed the argument `-l`

3.2 10 characters to describe permissions

- 1 leading character to describe file type
  - `-` just means a regular file
  - `d` means it is a directory file
  - `l` means it is a link!
  - `c` means it is a "character device"
    - these are hardware-oriented (devices)
  - `b` means block device
    - hardware in blocks - a.k.a (HD and SS) drives

3.3 10 characters to describe permissions

- the remaining 9 characters
  - use '-', 'r', 'w' and 'x'
  - for read, write and execute permissions
- for three groups
  - the file's owner/creator
  - the group the file belongs to
  - anybody else

3.4 the `chmod` command

- a way to change the permissions
  - as long as you have permission
- you'll often find yourself saying
  - `chmod u+x <script-file.sh>`
- by default, regular files are not executable
  - u+x means, add (+) executable (x) permission
    - to the owner (u)
- What might this mean?
  - `chmod o-x <program-name>`
- See `man chmod` or `chown` for more