Computer System Organization [Spring 18, Mu]

Additional tools, R01 part 2

Logistics

- Stay current with Piazza
  - Ask questions there
  - Answer fellow classmates questions if you can
  - You are responsible for, at least, knowledge from instructor posts
- Lab 1 has been posted -- you would benefit to start right away
  - We will not go over part 1 in recitation
  - Next week I will speak on part 2, but you should have started by then!
- You have 5 grace days total for labs
  - used in half day increments

Using a graceday

- Inside of the lab directory
  - e.g. (this will not be the exact command): **cd cso-spring18-labs-USERNAME/clab**
  - **echo “0.5” > gracedays.md**
  - **git add gracedays.md**
  - **git commit gracedays.md -m “Using extension”**
  - **git push**
- We will automatically detect that a graceday was used as long as it is *pushed to github!!!*
- No grace days for recitations. Recitations due Wednesday 11:59pm

Makefile

- Simple program compiling
  - **gcc hello.c**
  - **gcc hello.c -o hello** # Name our output program
- Compiling foo
  - **gcc foo.c** # Error!
  - **gcc foo.c main.c** # Recompile everything
  - Compiling piece by piece
    - **gcc -c foo.c**
    - **gcc -c main.c**
    - **gcc foo.o main.o -o foo**
- What a pain! What if we have way more source files
  - What if we only changed one?

Makefile

- First, look for file called (in the order they are looked for)
    - GNUmakefile, makefile and Makefile
- Generally, just use Makefile
- To execute Makefile
    - **make**
    - **make *rule***
    - **make clean**
- make only recompiles files that *need* to be recompiled
- Allows you to compile with a single command

Makefile

```
foo : foo.o main.o
    gcc foo.o main.o -o foo

foo.o: foo.c
    gcc -c foo.c

main.o: main.c
    gcc -c main.c

clean :
    rm foo.o main.o foo
```

```
name: dep_1 dep_2
    command
    command

dep_1: file_1
    command

dep_2: dep_3
    command

dep_3: file_2
    command
```

Debugging: gdb

- An aside: GDB stands for **G**NU **Deb**ugger
  - GCC: **G**NU **C**ompiler **C**ollection
  - GNU: ???
- What ways can you debug a program?
- Any other debuggers?
  - Eclipse
- GDB is a command line debugger (no pretty GUI)

GDB

- Program must be compiled with **-g** flag (put this in your Makefile)
- **gdb foo**
- Some commands, there are others!

Command	**Description**
help *command*	provide help for the given command
list	list lines around line num, file, function, etc.
break *location*	create a breakpoint at location
run	start executing program
continue	continue after hitting breakpoint
step	move one source file line

What to turn in for this lab:

- Make sure you have a working hello.c
    - Should print something close to (with newline!): Hello, World
- Make sure foo.c now prints the correct sum of 1 to 10

New to this recitation

- Makefile
    - that allows **foo** to be run in GDB
    - Uses $^and $@ variables (and there should only be one rule to compile foo.c and main.c)
- Debugged and correctly implemented argv_product.c