OS 1/28

Reagan Shirk

January 28, 2020

- Most of what we did today was working on the command line and getting used to the Ubuntu VM on GCP so I don't have a lot of notes

Generating an Executable File

- C file (.c) goes to a compiler
- The compiler translates from what we write to machine-specific code
- The object file (.o) is an intermediate machine-specific representation of just what is in a C file
- The Object file goes to a linker which brings together multiple files and produces the .exe file

Compiling Code Bases

- We want to:
  - Have a way to easily invoke the compiler
  - Only compile code that needs to be compiled

Make Files

- Managing the compiling/project management process
- Script of instructions on how to put programs in certain places

Our First C Program

```
#include <stdio.h>

int main(int c, char** argv)
{
    printf("Hello, world!\n");
}

gcc hello.c -o hello
```

Our First Makefile

```
# The top rule is exscuted by default:

all: hello

# Other rules are invoked as necessary
# Rule for creating the hello executable:

hello: hello.c
    gcc hello.c -o hello

clean:
    rm hello.o hello.exe
```