CSE 351

Introduction & Course Tools
INTRODUCTION

- I graduated from UW in December with a degree in Computer Engineering
- Currently am a 5th-year Masters student
- This is my 4th time as a TA for 351
- I run marathons, race triathlons, etc
WHY TAKE 351?

• Aside from it being a CSE requirement…
• The labs are fun
• You learn how computers work!
• Introduction to the C language, as well as x86_64 assembly
COURSE TOOLS

- Text editor
- GNU Compiler Collection (GCC)
- GNU Project Debugger (GDB)
- You can find all of these installed on the CSE Home VM
This is a personal preference
Try several, choose the one you like
Command-line
  - Nano
  - Vim
  - Emacs
Graphical
  - Gedit
  - Emacs
GCC

• This is a command-line utility that compiles your C files
• To create an executable program in C, there are two phases:
  • Compiling
  • Linking
• Compile: `gcc -Wall -std=gnu99 -c main.c`
  • This produces an object file called `main.o`
• Link: `gcc main.o -o test`
  • This produces an executable program called `test`
For this class, you will only be writing simple programs, so you can easily combine the compiling & linking phases

Compile & Link: `gcc -Wall -std=gnu99 main.c -o test`

This accomplishes the same thing as before in just one command
#include <stdio.h>

int main(int argc, char *argv[]) {
    printf("Hello World!\n");
}

HELLO WORLD
TRY IT ON YOUR OWN

• If you have a laptop with you, download the following file: http://courses.cs.washington.edu/courses/cse351/14sp/sections/1/HelloWorld.c
• Navigate to the directory where it is located, compile it, and run it
TRY IT ON YOUR OWN

• Navigating to the directory:
  • The command `cd` can help

• Looking inside each directory:
  • Run the `ls` command

• Compiling the program:
  • `gcc HelloWorld.c -o hello`

• Running the program:
  • `./hello`
ABOUT `PRINTF()`

- Used for printing to the console
- You can’t just concatenate strings with variables like you can in Java
- Insert placeholders to print out variables
  - The placeholder depends on the type of the variable
  - "%d", signed int
  - "%u", unsigned int
  - "%f", float
  - "%s", string
  - "%x", hexadecimal int
  - "%p", pointer
PRINTF() EXAMPLES

- `printf("I am %d years old", 20)`
  - Prints “I am 20 years old”
- `printf("My name is %s", "Matt")`
  - Prints “My name is Matt”
- `printf("%d in hex is %x", 2827, 2827)`
  - Prints “2827 in hex is 0xb0b”
ANOTHER EXAMPLE

• Download the file http://courses.cs.washington.edu/courses/cse351/14sp/sections/1/MoreComplicated.c

• Again, navigate to the file, compile it, and run it

• This example prints out the result of some bitwise and logical operations
• When you don't know how to use a particular shell command, you have several options
• One option is this site: http://google.com
• Another option is using the man command:
  • man 3 printf
  • This will give a detailed description of printf()