CSE 351 Section 1

Introduction & Course Tools

Introduction

Contact Info

Matt Dorsett
madman2@cs.washington.edu

Office Hours: Th 2:30-3:20 in CSE 002

About me

- Senior, Computer Engineering
    - Minor in Math & Physics
- Brand-new TA
- Love (most) sports
    - Still don't understand cricket
- Ironman triathlete by day, student by night...
- Come to my office hours!
    - I have office hours the day before HW is due

Why CSE 351 is awesome

- Understand computers
- Learn a bit of C
- Work on cool labs!

Student Introductions

- Name
- Year
- Majors/Minors
- Hobbies

Course Tools

Basic course tools

- Text editor
- GNU Compiler Collection (GCC)
- GNU Project Debugger (GDB)
- (all of these are in the CSE home VM!)

The CSE Home VM

- Why?
  - Like having a lab computer at home
  - Helps us troubleshoot code instead of fixing configuration problems

Information link:

http://www.cs.washington.edu/lab/labVMs/homeVMs.shtml

Text editors

- There is no ”best” text editor

- Command-line editors:
  - Nano (simple, but limited)
  - Vim

- Graphical editors:
  - Gedit
  - Emacs

GCC

- Command line utility
- Compiles C and C++ programs

- What is compiling?
  - Convert code so the processor can execute it

- How to compile?
  - Simple command is:

```
gcc <your code file> -o <program name>
```

Hello World in C

The code:

```
#include <stdio.h>

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

Let's talk about each part

Your turn!

- Find a person with a laptop and gcc
- Using a browser or wget, download the file:

  http://students.washington.edu/dutchsct/CSE351/HelloWorld.c

  ```
  wget http://students.washington.edu/dutchsct/CSE351/HelloWorld.c
  ```
- Open a terminal window
- Navigate to the source file

Your turn!

- Now run:

```
gcc HelloWorld.c -o hello
```

- Check that it's there:

```
ls
```

- Now run the program:

```
./hello
```

About printf()

- Takes a format string with placeholders for numbers, strings, etc.

- Common placeholders:
    - ”%d”, signed int
    - ”%u”, unsigned int
    - ”%f”, float
    - ”%s”, string
    - ”%x”, hexadecimal int
    - ”%p”, pointer address

printf() examples

```
printf(”I am %d years old”, 20)
```

Output: `I am 20 years old`

```
printf(”My name is %s”, ”Matt”)
```

Output: `My name is Matt`

```
printf(”%d in hex is %x”, 2827, 2827)
```

Output: `2827 in hex is 0xb0b`

A more complicated example

- Now, download the file:

  http://students.washington.edu/dutchsct/CSE351/MoreComplicated.c

  ```
  wget http://students.washington.edu/dutchsct/CSE351/MoreComplicated.c
  ```

- Compile the file

  ```
  gcc MoreComplicated.c -o MoreComplicated
  ```

- Run the file

  ```
  ./MoreComplicated
  ```

- Open this in a text editor, find examples of printf and function calls in C.

Linux shell commands

- See what directory you're in

```
pwd
```

- See what's in the directory

```
ls
```

- Go into a directory

```
cd <directory name>
cd .. (takes you up a directory)
```

chmod

- If "permission denied" error when trying to run
  `./MyProgramName`

- You may need to change the executable's permissions

- To give yourself R/W/X permissions, run
  `chmod 744 [filename]`

Linux man pages

Say you need to know more about printf()

You can certainly use Google

Another option is

```
man 3 <function>
```

ex: `man 3 printf`

Brings up the man page for the C version of the function you provide

For more general Linux help, use man without the 3