CS 4400 – Computer Systems

Instructor: Matthew Flatt
TAs: Christian Roy
     Graham Zuber
     Guy Watson
     Qixiang Chao
     Simon Redman
CS 4400 Organization

- Video lectures
- Before-class quiz on videos
- Recitation-style class
- Lab sessions
- Homework assignments
Registering for CS 4400 Fall 2017

• There’s a waiting list

• Preference given to students who need the course now to satisfy requirements
  e.g., CE student to take ECE 5780 in the Spring

• Waiting list or permission code: e-mail instructor
  --- please include details in request
Course Information

https://www.eng.utah.edu/~cs4400/

- Prerequisite: CS 3810
- Recommended: CS 3505
Why CS 4400?

Explore layers of abstraction — especially the lower ones, but above hardware

... 
Java Virtual Machine 
C 
Operating System 
Memory Hierarchy 
Instruction Set Architecture 
Hardware
Why CS 4400?

Explore layers of abstraction — especially the lower ones, but above hardware

...  
Java Virtual Machine  
C  
Operating System  
Memory Hierarchy  
Instruction Set Architecture  
Hardware
Why CS 4400?

Explore layers of abstraction — especially the lower ones, but above hardware

...  
Java Virtual Machine  
C  
Operating System  
Memory Hierarchy  
Instruction Set Architecture  
Hardware
Course Skills

**Unix** both technically and culturally

- Processes, file descriptors, sockets
- Shells, gcc, gdb

**C** as a “portable assembly language”

- Exposed data representations
- Unsafe
- Manual memory management
Course Skills

**Unix** both technically and culturally
- Processes, file descriptors, sockets
- Shells, gcc, gdb

**C** as a “portable assembly language”
- Exposed data representations
- Unsafe
- Manual memory management

**ANSI C = C89 = C90**
default gcc on CADE machines
Course Skills

Unix both technically and culturally

- Processes, file descriptors
- Shells, gcc, gdb

C as a “portable assembly language”

- Exposed data representations
- Unsafe
- Manual memory management

ANSI C = C89 = C90
default gcc on CADE machines

Seriously!
Course Skills

**Unix** both technically and culturally

- Processes, file descriptors, etc.
- Shells, *gcc*, *gdb*

**C** as a “portable assembly language”

- Exposed data representations
- Unsafe
- Manual memory management

**ANSI C** = C89 = C90

default *gcc* on CADE

Seriously!

We’ll count C99/C11 homework as wrong
Course Skills

**Unix** both technically and culturally

- Processes, file descriptors, sockets
- Shells, `gcc`, `gdb`

**C** as a “portable assembly language”

- Exposed data representations
- Unsafe
- Manual memory management

**x86-64** but transferrable to, e.g., ARM
Course Concepts

Representing data, especially numbers

Instruction sets

Optimization

Linking

Processes and signals

Memory allocation

Networking APIs

Concurrency
Useful Outcomes of CS 4400

You will be a more effective programmer

• detecting and fixing bugs more efficiently

• understanding and tuning program performance

You will be comfortable using the terminal and command line

You will have a firm foundation for specialized systems classes and real-word software development
Course Structure: Homework Assignments

match

bomb (disassembly)

performance

linking

shell

malloc

server

2 weeks each, sometimes student-specific
Course Structure: Videos, Classes, and Lab Sessions

Before Monday & Wednesday:

• video lectures posted
• quiz on video due 1 hour before class

Monday & Wednesday:

• class meets for extended examples

Thursday:

• lab session in CADE (WEB L224)
Command-Line Arguments
Running Programs at a Command Line

$ /bin/cat one.txt two.txt
Running Programs at a Command Line

$ /bin/cat one.txt two.txt

prompt  program  arguments
Running Programs at a Command Line

A command line is itself a program known as a **shell**

The default shell is **/bin/bash**

```
$ /bin/echo a  b
```
Running Programs at a Command Line

A command line is itself a program known as a **shell**

The default shell is `/bin/bash`

$ /bin/echo a b

argument argument
Running Programs at a Command Line

A command line is itself a program known as a **shell**

The default shell is `/bin/bash`

```
$ /bin/echo "a  b"
```

argument
Shell Quoting

Both

""

and

,

are quotes in bash, but with different rules

More information:

man bash