**CS241 Lecture 01, August 23, 2015**

> echo $USER

Ed Karrels (filling in for Lawrence\_Angrave)

> cat "CS241 Learning Objectives.txt".   
You will be able to ...

Interact with OS in C via system calls

Understand how OS allocate, deallocates and accesses memory

Understand how virtual memory works

Create, use, manipulate processes and threads

Understand how OS schedules processes and threads

Communicate and synchronize between threads and processes

Determine when deadlock and race conditions may occur and how to avoid them

Manipulate filesystem structures (inodes etc.)

Communicate across networks

> grep "The People" CS241.txt

> man –S 2 " The Experience CS241"

Not your regular course. This is a UIUC-and-by-Angrave course.

A byte of CS241 every day is good for you.

Sat,Sun,Tue: Piazza.Programming Assignment. Write the book. Play.

Class: Lecture MWF. Thursday Section. Weekly Exam Quiz. Other stuff.

> Grades

90:A-, 80:B- 70:C-

0x1e% Quizzes. 0x1e% Final Exam. 0x1a% Programming. 0xe% HW&Section

> Why do we need an O/S ?

- Protect from errant programs

- Hide device diffs (HDD, SSD, USB, CD)

- Hide system diffs

- Shared resources (printer, CPU time)

POSIX? Standardized Unix (Portable Operating System Interface)

> Process memory

Environment

Program Arguments

Stack

Heap

Unitialized vars

Initialized vars

Code

+ Dynamically linked library functions + Guard pages + Multiple threads.

> Fun stuff:

Low level! UIUC programmers don't just program in python/js, they could *write* python/js

Powerful! Create things that others will use. Make programs that others can only dream of.

> Trickier stuff:

Know your tools. C Programming / System programming is brutal if you don't know the details.

Concurrency (muli-threading, multi-process)

Synchronization

Signals

Critical Section

Race Conditions

Deadlock

Analysis of Reader-Writer, Dining Philosphers, Producer Consumer

Demos: [We may do these next lecture]

printf. vs write

C Strings & pointers

null. sizeof. strcpy. strcmp

casting

malloc & free

Your turn:

https://courses.engr.illinois.edu/cs241/

Navigate to the github wiki -

https://github.com/angrave/SystemProgramming/wiki/C-Programming---Common-Gotchas

**TODO**

1) Read “C Programming, Part 1: Introduction” from the wiki

<https://github.com/angrave/SystemProgramming/wiki>

2) Create 5 C programs that illustrate 5+ mistakes add them to your subversion area before your section. Print them out bring them to next lecture

Hello world in C++:

#include <iostream>

int main() {

std::cout << "Hello world!" << std::endl;

}

And in C:

#include <stdio.h>

int main() {

printf("Hello world!\n");

return 0; // echo $?

}

Printf - %d, %s

Strings – null terminated

strcpy, strlen, strcat, strchr

Pointers – size of object, special-case void\*

Print a string, one char at a time

$ bash

$ if ./equal $a def; then echo Yes; else echo No; fi

STDIN\_FILENO = 0

STDOUT\_FILENO = 1

STDERR\_FILENO = 2

Send stdout and stderr to different files:

ls > foo 2> bar

Pipe stderr and stdout:

ls 2>&1 | nextcmd