CSC369: Operating Systems

Fall 2014

Andrew Petersen



- Administrivia
- Introduction to Operating Systems
- ...
- Computer Organization
- OS Structure
- What is a Process?

Administrivia

• Make sure to read the course information sheet.

- The discussion board is your best friend.
 - Announcements will be posted there.
 - It will be your best resource on assignments.

Course Objectives

- Understand the role of a modern OS.
- Become familiar with the major components of a modern OS and how they are implemented.

- How are we going to get there?
 - Discussion ("lectures" and tutorials) of material from online readings (textbook)
 - Practice (C assignments, pen-and-paper exercises)

Course Content

- The role of OS
- The system call interface
- Memory management
- CPU scheduling
- Processes vs. threads
- Concurrency (synchronization and deadlock)
- File and I/O systems

Homework is 50%

- Two kinds of homework: assignments and exercises
- The three assignments are each moderate-sized C coding assignments.
 - We are not using OS/161 this year; each assignment is "standalone".
 - Each should be completed individually.
- At each class meeting (lecture or tutorial), we will complete work in class or you will be asked to bring some work to class.
 - 3 points per day, for a total of 72 points. 60 is considered "full credit".

Why the Exercises?

- The goal is to make our discussions productive.
- The exercises are either:
 - Results of work in class
 - Preparation for class

 To get full credit, you'll need to be familiar with the assigned reading, which will help you get more our out of class.

Tests are 50%

- The midterm is 10% and covers system calls, scheduling, and some concurrency.
- The final exam is 40% and comprehensive.

- The tests will cover material very similar to our discussions and exercises in class.
- Assignments will also be represented.

Workload

- Karen and I are hoping this course is less work than previous offerings -- but has the same OS value.
- That being said, the course covers a lot of material.
 - Much of it is very abstract.
 - It will require practice to really understand.
- The material draws from many prerequisites.
 - UNIX tools, C programming (especially pointers)
 - Computer organization

Academic Dishonesty

- Please don't.
 - It's painful for everyone involved ...
 - It doesn't say good things about the perpetrator.
 - And it creates paperwork.
 - I hate paperwork.

- You may discuss concepts and tools with your classmates.
- Discuss assignment solutions with me or your TA.

Your Resources

- Lectures/tutorials: Please ask questions!
- Office hours: Please visit!
- Discussion board: Fast response (if everyone participates!)
- Email: Please reserve for administrative or private issues (24-48 hour response, usually)
- Anonymous email: For feedback



- Administrivia
 - What's coming up?
- Introduction to Operating Systems
- ..
- Computer Organization
- OS Structure

Coming Up

Today's activity is a mind map (we'll get to it shortly)

 Wednesday, we'll be discussing processesand system calls and will complete an activity in tutorial

• Next week: processes, address spaces

Today

- Introduction to Operating Systems
 - What is an OS?
- Computer Organization
- OS Structure
- What is a Process?

Setting the Stage

• What is an OS?

• What's its purpose?

• What impacts OS development?

What is an OS?

Applications

Operating System

Hardware

Goals of the OS

- Primary: convenience for the user
 - It should be easier to compute with the OS than without it.
- Secondary: efficient use of the system

- These goals are sometimes contradictory!
 - To be honest ... often contradictory

Other Views of the OS

- An OS is a resource allocator
 - Allows the proper use of resources (hardware, software, data) in the operation of the computer system
 - Provides an environment within which other programs can do useful work
- An OS is a control program
 - Controls the execution of user programs to prevent errors and improper use of the computer
 - Especially concerned with the operation and control of I/O devices

What Impacts OS Design?

- On one hand, its goals
 - Convenience
 - Efficiency and speed



Hardware

- On the other hand, practical constraints
 - What do applications need?
 - What does the hardware provide?



- Computer Organization
- OS Structure
- What is a process?
- Process Lifecycle

Computer System Structure

- CPU: Processing
- Memory system



What Components Do We Care About?

- Processor: efficient sharing of execution resources (scheduling, concurrency)
- Memory: improving execution performance and enabling multitasking (virtual memory)
- Disk (persistent storage): organizing data into a filesystem
- Network: distributed systems

Focusing on Memory...

- Main memory is the only storage the CPU can access (directly)
 - Ignore registers for the moment :)
- Viewed as a large array of bytes
 - The memory is byte-addressable

Storage Hierarchy

- Registers, main memory, and auxiliary memory form a memory hierarchy
- Caches can be installed to increase performance and hide access-time gaps



C's View of Memory

- A variable in C is a symbolic name for a data item (stored in memory)
 - The variable's type indicates its size and alignment
 - The variable's address is an index into memory
 - The variable's value is the data at that address
- A pointer is a variable whose value is an address
- An array is a contiguous chunk of memory that is interpreted as some number of variables
 - ... but it's really just a pointer

Memory Example

0xbfffeb08

	a	0xbfffeb07	'h' (0x68)
	1	b06	Unused
<pre>int main() {</pre>		b05	Unused
char $a = h';$		b04	Unused
<pre>int b = 0xdeadbeef;</pre>		b03	0xde
char $*c = \&a$		b02	0xad
int $*d = \&b$		b01	0xbe
<pre>printf("b=%d (0x%x)\n",</pre>	b	0xbfffeb00	0xef
		aff	Oxbf
		afe	Oxff
		afd	0xeb
	c=&a	0xbfffeafc	0x07
		afb	0xbf
		afa	Oxff
		af9	Oxeb
	d=&b	0xbfffeaf8	0x00

Memory Layout (Linux, x86)

