CS335 – Operating Systems – Fall 2016
MWF 12:00–12:50 p.m., ENGR A219

See the course web page for more information and resources: http://www.cs.siu.edu/~cs335

Professor
Dr. Norman Carver, Faner 3121, phone 453-6048, email carver@cs.siu.edu.
Office hours: MWF 1:00–3:00pm (or by appointment).

Teaching Assistant
Thomas Ruble, email thomas.ruble@siu.edu
Office hours: TBA

Required Textbook

Catalog Course Description
An extended treatment of the components of operating systems including process management, concurrency, memory management, device management, file management, and security.

General Course Objectives
Few students ultimately are going to end up “hacking” OS kernels or even drivers, so focusing on students building simple/simulated kernel components seems of low value. By contrast, every student is going to end up working with OS’s and in particular developing software on and for OS’s. Thus, the primary goal of an OS course should be to teach students how OS’s function, with a focus on those aspects of OS’s that can enable students to work more effectively on computer systems and develop better software.

Detailed Course Objectives

• Familiarizing students with the features and components of modern operating systems.
• Improving students’ programming abilities through a detailed understanding of OS’s:
  – so they can be more efficient in developing and debugging software
  – so they understand how to develop more efficient and secure software
• Introducing students to system-level programming (i.e., syscall programming).
• Introducing students to concurrent programming (processes and threads).
• Introducing students to network and distributed programming.

Prerequisites
CS220 with grade of C or better or consent of the instructor.
Course Topic Outline (tentative)

1. Introduction to Operating Systems (chaps. 1, 2)
   - single tasking, multiprogramming, multitasking, multiprocessing
   - single user, multi-user;
   - batch, interactive, CLI, GUI
   - hardware (computer architecture): protected mode, multicore, SMP, etc.
   - libraries vs. system calls

2. Programming and the OS
   - executable formats, memory layouts
   - static vs. shared/dynamic libraries
   - compiling, linking/loading
   - process tracing, profiling, debuggers, core files, etc.
   - program efficiency and the OS
   - writing secure and robust programs

3. Process management (chaps. 2, 4, 6)
   - processes: creation, management, scheduling
   - threads: creation, management, scheduling, vs. processes, TLP

4. Concurrent computing (chaps. 5, 7)
   - process/thread synchronization
   - interprocess communication (IPC)
   - issues: race conditions, deadlock, mutex, critical sections, etc.

5. Memory management (chaps. 8, 9)
   - RAM and multitasking (partitions, pages, segments)
   - Virtual Memory: pages, paging/swapping, swap space, page replacement, etc.
   - kernel caching/buffering
   - 32 vs. 64 bit CPUs

6. Storage management (chaps. 10, 11, 12, 13)
   - logical filesystems, physical filesystems, RAID
   - device (e.g., disk) management, I/O systems

7. Security and protection (chaps. 14, 15)
   - users, permissions, privileged users, capabilities
   - memory protections
   - filesystem protections
   - Discretionary Access Controls (DAC) vs. Mandatory Access Controls (MAC)

8. Distributed and networked systems (chaps. 17)
   - distributed systems, coordination,
   - networked systems, TCP/IP, OS network stack, client-server architecture

9. OS virtualization (chaps. 16)