CS 321 - Operating Systems

Meets MWF 10:30-11:30 AM Room 106 Chapman Building University of Alaska Fairbanks	CS F321-F01 (#32845) 3.0 Credits, Spring 2005 Prerequisite: CS 301 (Assembly)	Instructor: Dr. O. Lawlor ffosl@uaf.edu, 474-7678 Office: 210C Chapman, 2-4 MWF

	Textbook: Operating System Concepts, 7th Edition by Silberschatz, Galvin, & Gagne; 2005 Wiley & Sons	ADA Compliance: Will work with Office of Disabilities Services (203 WHIT, 474-7043) to provide reasonable accomodation to students with disabilities.	Course Website (& links to Blackboard): http://lawlor.cs.uaf.edu/2005/cs321/ UNIX Machines: on nanook.uaf.edu, in Chapman lab, or Linux CDs available

Course Goals and Requirements

By the end of the course, you will be able to design system-level libraries for a variety of tasks; be familiar with the general abilities and interfaces provided by common operating systems; and understand in a deep way the implementation of modern processor execution, memory, and storage. To understand this, you will need to have experience writing programs in some standard systems programming language (C or C++), with at least some idea of how your code relates to assembly language and how it runs on the real machine.

Grading

As shown in the example questions, your work will be evaluated on correctness, rationale, and insight, not on successful regurgitation of random trivia. Grades for each assignment and test may be curved upward, by scaling to a distribution with a median of at least 80%. Each homework and the midterm will then be clamped to the range [0%,100%]. Your grade is then computed based on four categories of work:

1. **HW:** Homeworks and machine problems, to be distributed through the semester. The curved and clamped score for neatly typeset homeworks or well-packaged and commented machine problems will be scaled up by 10%.
2. **PROJ:** Two short individual or group projects. I'll provide a list of possible topics, let you pick groups and a topic, and provide a staged series of requirements, including (1) project description, (2) informal project design, (3) code, and (4) a short presentation.
3. **MT:** Midterm Exam covering the time section, tentatively held Wednesday, March 9 at the usual class time.
4. **FINAL:** Final Exam covering the space section, to be held Monday, May 9 at 10:15AM.

The final score is then calculated as:

TOTAL = 20% **HW** + 30% **PROJ** + 25% **MT** + 25% **FINAL**

Assignments will be due at the beginning of class. Late work will not be accepted under any circumstances. Exams must be taken as scheduled, except in extreme circumstances. Academic dishonesty (including plagarism or cheating) is unacceptable and will be handled according to University board regulations.

Course Outline and Schedule (Tentative)

First section: Time Management

- Event-driven programming [1 lecture]
  - DOS-style polling loop
  - Mac/X (or other GUI) event loop
  - Win32 wndProc
- Processes (Ch. 3) [1 lecture]
  - Semantics: multiprogramming
  - Creation: UNIX fork, Win32
- Hardware Implementation (Ch. 3.1) [1 week]
  - Resources: Stack, registers, heap
  - Protected (privileged, supervisor, "ring 0") mode kernel & security

Calendar	**Monday**	**Wednesday**	**Friday**
January			21 First day of class
Events, Processes, Hardware	24	26 HW0 Due	28 Add Deadline
Signals, Interrupts	31		

  - System calls, timers, and other hardware interfaces
- Signals & interrupts (Ch. 4.4.3 & 13.2.2) [1 week + 1 homework]
  - Hardware interrupts
  - UNIX/Win32 signals/handlers
  - Interrupt safety (reentrancy)
- Threads (Ch. 4) [1 week]
  - Kernel-level: pthreads, win32
  - User-level: coroutines
- Concurrent Interaction (Ch. 6 & 7) [2 weeks + 1 homework]
  - Motivation: Race conditions
  - Locks (pthread lock, win32 mutex), semaphores (win32)
  - Deadlock prevention
  - Not covered: deadlock detection & response
- CPU Scheduling (Ch. 5) [1 week]
  - Starvation, poor utilization
  - Prioritization
  - Priority Inversion
  - Job scheduling: Shortest-Job-First

Second Section: Space Management

- Memory allocation (Ch. 8.3.2) [1 week + 1 homework]
  - Memory heirarchy & cost-capacity-speed tradeoff
  - Low-level memory allocation: sbrk
  - Mid-level memory allocators
- Virtual memory: uses (Ch. 9) [1 week]
  - DLL/text page sharing, copy-on-write
  - Memory-mapped files: UNIX mmap, mprotect, SYSV IPC; Win32 MapViewOfFile (Ch. 9.7)
  - Software distributed shared memory
- Virtual memory: implementation (Ch. 8, 9.4) [1 week]
  - Page table and TLB (presence, permissions, and dirty bits)
  - Demand paging & page replacement strategies
- Filesystem (Ch. 10 & 11) [1 week]
  - Layouts: File Allocation Table (FAT), inode, b-tree
  - Caching, fragmentation, corruption during crash
- Accounting and security [2 weeks]
  - Terminology: Tampering and authentication, secrecy and encryption
  - Common security holes: buffer overflow, unquoted inputs, excess priviledge

February		2 PROJECT1 Topic Due	4 Drop Deadline
Threads	7	9 HW1 Due	11
Concurrency 1	14	16 PROJECT1 Design Due	18
Concurrency 2	21	23	25
CPU Scheduling	28		
March		2 HW2 Due	4
Review and Midterm	7	9 MIDTERM	11
Spring Break	14 (BREAK)	16 (BREAK)	18 (BREAK)
Memory Allocation	21 Last day to Withdraw	23 PROJECT1 Code Due	25
VM Usage	28	30 PROJECT2 Topic Due	
April			1
VM Implementation	4	6 HW3 Due	8
File System	11	13	15
Security	18	20 PROJECT2 Design Due	22
Accounting	25	27 HW4 Due	29 (BREAK) Springfest
May			
Semester Project Demos	2 PROJECT Demos	4 PROJECT Demos, PROJECT2 Code Due	6 Review for final exam
Finals Week	9 FINAL at 10:15AM		