Introduction to Operating Systems
Fall 2015
M/W 11:30 – 13:20
Mark Morrissey

Office Hours

Mark Morrissey, markem@pdx.edu: M/W 1:30 – 2:30, FAB 120-13
Naomi Dickerson, nld2@pdx.edu: Tu 11:00 – 12:30, FAB Fishbowl

Note

The instructor wishes to thank Professor Jonathan Walpole for allowing the use of his materials in this course. While some modifications have been made to the lecture slides, if you happen to have a copy of Professor Walpole's slides these will suffice with some additional note taking.

Description

This course will introduce the core concepts of operating systems, such as processes and threads, scheduling, synchronization, memory management, file systems, input and output device management and security. The course will consist of assigned reading, weekly lectures, a midterm and final exam, and a sequence of programming assignments. The goal of the readings and lectures is to introduce the core concepts. The goal of the programming assignments is to give students some exposure to operating system code. Students are expected to read the assigned materials prior to each class, and to participate in in-class discussions.

Prerequisites

Students should have previous familiarity with programming in a high-level object-oriented language (such as C++ or Java); assembly language programming; CPU organization, instruction sets, registers; program development in the Unix environment (edit, compile, link, load, execute, makefile, using the shell); the Unix system call interface; basic data structures (lists, trees, graphs); object-oriented concepts (class, object, method. Junior standing (and all that implies) is a prerequisite for this course.

Textbook

The course will be based primarily on lecture slides. In addition, the following reference material will be used for the project assignments:

"**The BLITZ System**", by Harry Porter, approx. 200 pages. Do not print out all the documentation. You will waste your print quota. We will discuss in class those documents that you may wish to print out.

There is also a textbook associated with the course. The primary purpose of the textbook is to provide students with an alternative perspective on the materials covered, should it be needed. The textbook, which is also available freely online is:

Operating Systems: Three Easy Pieces
Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau, Arpaci-Dusseau Books.
Available at: http://pages.cs.wisc.edu/~remzi/OSTEP/

Note: you can use the **free** online version, or pay $10 for a complete .pdf with glossary, links, etc. Also you may choose the hardcopy options for $25 softcover or $37 hardcover. Any of these options include all of the material needed for the course, just choose the most convenient for your needs.

Grading

Your final grade will be calculated as follows:

project - 30%;
midterm exam - 35%;
final exam - 35%;

Students must achieve at least 35% of the available points on all graded work to pass this class. Each project counts as an individual graded item for this purpose.

On Academic Honesty

Students are expected to understand the idea of academic integrity and the ramifications of plagiarism, cheating, etc. Contact student services or your advisor if you are unclear on these concepts. Any work submitted by a student wherein substantial parts of the work are not those of the student or a misrepresentation of one student's work as that of another are examples of academic dishonesty. In cases of academic dishonesty a minimum penalty of the loss of all points for that grade point will occur. All incidents of academic dishonesty will be reported to the department head.

Extraordinary Circumstances:

If an extraordinary situation (for example severe illness) prevents you from working for a period of time, contact us as soon as possible to discuss your situation and arrange a special schedule. Scheduled work commitments do not constitute an extraordinary circumstance. Makeup exams will not be given except in cases of severe and documented medical or family emergencies. Please note that travel is not considered an emergency. If an emergency arises and you miss an exam, contact the instructor before the exam to arrange for a special circumstance.

Requests for Regrade:

Requests for regrading, plus the original graded material, must be submitted to the instructor in writing within one week of the time the graded assignment was made available. You must be specific in saying why you feel your answer deserves additional credit. A request for regrade will result in a reevaluation of the entire assignment and your total grade may increase or decrease as a result.

Students with Disabilities or Need for Special Accommodation:

Accommodations are collaborative efforts between students, faculty, and the Disability Resource Center. Students with accommodations approved through the DRC are responsible for contacting the faculty member in charge of the course prior to or during the first week of the term to discuss accommodations. Students who believe they are eligible for accommodations but who have not yet obtained approval through the DRC should contact the DRC immediately. The DRC may be reached at 503-725-4150 or visit the DRC web page: http://www.drc.pdx.edu/.

Projects

The programming assignments for this class are based on the **BLITZ** system. BLITZ is a collection of software, written by Harry Porter, designed to streamline the process of learning about, and experimenting with, operating system kernel code. BLITZ includes a complete operating system, assembler, linker, loader and debugger, together with software to emulate an underlying CPU and various devices. The emulated CPU and devices are representative of real-world systems, but without some of the low-level complexity that complicates the process of learning about the key underlying concepts. By using BLITZ students are able to study, in detail, the low-level operating system code that interacts with the hardware, as well as design, code and test their own modifications to the operating system.

Note that our class will have project submission requirements that are in addition to those listed on the BLITZ pages. This will be discussed in class.

The due dates for each of these projects are given in the class schedule below. No late work will be accepted. No exceptions.

Project 1: [Handout.pdf] [Directory Containing Files]
Project 2: [Handout.pdf] [Directory Containing Files]
Project 3: [Handout.pdf] [Directory Containing Files]
Project 4: [Handout.pdf] [Directory Containing Files]
Project 5: [Handout.pdf] [Directory Containing Files]

Our class will not be using this homework submission guidelines includes in the BLITZ project descriptions. We will use a different mechanism that we believe will be much easier for the students. The new method will be discussed in class, but is essentially doing a dropbox submission of entire files modified for the assignment. Note that the dropbox for project submissions will close at the start of class on the appropriate due date.

D2L

This course will use the online system d2l.pdx.edu. Students are expected to log on to D2L frequently. D2L is how the instructor and TAs will communicate with the entire class; through news items, discussion areas, etc. This is also where the grade book will be maintained and lecture slides distributed.

Syllabus

9-28	**Course Overview and Introduction to Operating Systems** Course outline. Overview of course project and expectations. Introduction to hardware support for operating systems: privileged mode execution, saving and restoring CPU state, traps and interrupts, timers, memory protection. Operating system techniques for protecting user and hardware resources. Overview of the key operating system abstractions and the use of system calls to manipulate them. Reading: *Intro: Introduction* Start Project 1 - Introduction to BLITZ
9-30	**The Process Concept** Complete the overview of the key operating system abstractions and the use of system calls to manipulate them. Program execution, the process concept, process-related state, the process table, saving and restoring process state, the role of the scheduler. Reading: *Virtualization: Processes; Process API; Direct Execution*
10-5	**Threads and Concurrency** Threads, process context switch vs thread switch, true concurrency vs pseudo concurrency, operating systems as concurrent programs, concurrency through multi-threading, concurrency through interrupt handling, concurrent access to shared memory, race conditions, mutual exclusion, and synchronization primitives based on atomic instructions. Reading: *Concurrency: Concurrency and Threads; Thread API* Project 1 due at start of class. Start Project 2: Threads & Synchronization
10-7	**Synchronization Primitives** Atomic instructions, locks, spinlocks, mutex semaphores, counting semaphores, and their use in solutions to Producer Consumer synchronization. Reading: *Concurrency: Locks; Locked Data Structures*
10-12	**Classic Synchronization Problems** Classic synchronization problems: Producer Consumer, Dining Philosophers, Readers and Writers, Sleeping Barber. Reading: *Concurrency: Locks; Locked Data Structures*
10-14	**Monitors and Message Passing** Monitors, condition variables, message passing, and their use in solutions to classic synchronization problems: Producer Consumer, Dining Philosophers, Readers and Writers, Sleeping Barber. Reading: *Concurrency: Condition Variables; Semaphores; Concurrency Bugs*

10-19	**Deadlock** Deadlock, livelock, deadlock detection, avoidance, and prevention. Reading: *Concurrency: Concurrency Bugs*
10-21	**Scheduling** Separation of policy from mechanism, scheduling mechanisms, preemptive vs non-preemptive scheduling, example scheduling policies, FIFO, round-robin, shortest job first, priority scheduling, Unix-style feedback scheduling, proportional share scheduling, lottery scheduling. Reading: *Virtualization: CPU Scheduling; Multi-level Feedback; Lottery Scheduling; Multi-CPU Scheduling* Project 2 due at start of class. Start Project 3: Synchronization Problems
10-26	**Memory Management** Memory addresses and binding, static and dynamic addresses translation, address translation using base and limit registers, memory management algorithms using linked lists and bitmaps, external and internal fragmentation, paged virtual memory. Reading: *Virtualization: Address Spaces; Memory API; Address Translation; Segmentation; Free Space Management; Intro to Paging;*
10-28	**Midterm Exam** In class, closed-book exam based on material covered so far.
11-2	**Virtual Memory 1** Physical address spaces, virtual address spaces, page table design, single-level and multi-level page tables, hardware support for dynamic address translation using a TLB, hardware and software managed TLB refill. Reading: *Virtualization: Address Spaces; Memory API; Address Translation; Segmentation; Free Space Management; Intro to Paging; Translation Lookaside Buffers*
11-4	**Virtual Memory 2** Inverted page tables, the memory hierarchy, TLB miss faults, segmentation faults, protection faults, page faults, hardware support for memory protection, segmentation. Reading: *Virtualization: Address Spaces; Memory API; Address Translation; Segmentation; Free Space Management; Intro to Paging; Translation Lookaside Buffers; Advanced Page Tables* Project 3 due at start of class. Start Project 4: Kernel Resource Managers
11-9	**Virtual Memory 3** Implementation issues, page sharing, copy-on-write, page fault handling, segmentation,

	segmentation with paging. Reading: *Virtualization: Address Spaces; Memory API; Address Translation; Segmentation; Free Space Management; Intro to Paging; Translation Lookaside Buffers*
11-11	**Holiday**
11-16	**Paging Algorithms** Demand paging, swapping, placement and replacement algorithms, memory hierarchy revisited, overview of cache architecture, performance modeling for memory management systems. Reading: *Virtualization: Swapping Mechanisms; Swapping Policies*
11-18	**Input/Output** Devices, memory mapped devices, DMA, device drivers, interrupt handling, scheduled vs non-scheduled I/O processing, block vs character devices. Reading: *Persistence: I/O Devices* Project 4 due at start of class. Start Project 5: User Level Processes
11-23	**Secondary Storage Management** Disks, sectors, tracks, blocks, disk head scheduling algorithms, the file abstraction, directories, links. Reading: *Persistence: Hard Disk Drives; RAID*
11-25	**File Systems 1** File system architecture, file system data structures and system calls. Reading: *Persistence: Files and Directories; File System Implementation;*
11-30	**File Systems 2** File system architecture and design criteria. Reading: *Persistence: Fast File System; FSCK and Journaling; Log Structured File System* Project 5 due at start of class.
12-2	**Security** Protection domains and mechanisms, access control lists, capabilities, user authentication, encryption, common internal and external attacks. Reading: *Persistence: Data Integrity and Protection*
12-10	**Final Exam** (tentative)