

Operating Systems		CIS3110	
Classes	MWF 11:30–12:20 in Richard 124		
Instructor	Wlodek Dobosiewicz (Reynolds 215), ext. 53216 dobo@cis.uoguelph.ca		
Office Hours	MTF 10:30am–11:20am		
Prerequisites	CIS2500 or CIS2650 (CIS2030 or ENGG2410 recommended).		
Textbook	Silberschatz, Galvin and Gagne <i>Operating Systems Concepts</i> (John Wiley) any edition, 8 th is the latest.		
Class notes	on Wiley website (Google "Silberschatz Galvin Gagne") also: www.cis.uoguelph.ca/~dobo/3110/		
Dishonesty	results in an "F" (not negotiable)		

Check the calendar for special accommodations and the definition of academic misconduct.

Calendar description

CIS*3110 Operating Systems W(3-1). [0.50]

Operating systems in theory and practice. Components in a system: scheduling and resource allocation; process management, multi-programming, multi-tasking; I/O control and file systems; mechanisms for client-server computing. Examples from contemporary operating systems.

Precondition

Students taking CIS3110 are not expected to have any deep background in operating systems other than as users. However, they are expected to be mature computer science students, implying that they:

- can design, implement and test moderately-sized programs (<1000 lines) in C, C++ or Java.
- are familiar with basic UNIX commands.

Postconditions

Students who complete successfully CIS3110 will be well informed about the essential aspects of operating systems internals as well as concurrency control and interprocess communication.

- Basic structure of a process: address space, registers, program counter and stack.
- Asynchronous activity: hardware and software interrupts, service routines.
- The implementation of multiprogramming: co-routines, the context switch, process status.
- Sharing of the processor (multiprogramming): Concurrent processes, the problem of shared variables, race conditions, critical sections and techniques for partial synchronization, such as mutual exclusion, semaphores and strict message passing.
- Performance implications of multiprogramming: processor scheduling, queuing.
- Management of main memory: MMU hardware, virtual memory, segmentation and paging algorithms, their performance, thrashing.
- Handling of secondary storage: block and byte I/O, internal buffer handling, file system structures.
- Operating System Kernel: Internal structure, system call interface.
- Client/server model for distributed computing, remote procedure calls.

Grading

Assignments and exams		
Programming assignment 1	Monday, January 26	10%
Programming assignment 2	Friday, February 13	10%
Programming assignment 3	Monday, March 30	20%
Mock Midterm	Wednesday, February 25	5% (in class)
Midterm	Monday, March 2	20% (in class)
Final	Tuesday, April 7 11:30am	35%

All the assignments and projects involve programming. Assignments are individual.

Reading materials

- Andrew Tanenbaum, *Modern Operating systems* (Prentice–Hall).
- Richard Stevens and Stephen Rago, *Advanced Programming in the UNIX Environment* (Addison–Wesley).
- Michael Beck *et al.*, *Linux Kernel Programming* (Addison–Wesley).
- Tutorial on network programming by [Beej](#)
- Tutorials in [LUPG](#).
- Tutorials about hardware and Windows in <http://computer.howstuffworks.com/>
- Many other books and web postings are readily available (wikipedia or google are good starting points for a search).

Syllabus revised January 7, 2009 by W. Dobosiewicz