

# CMPUT 379 Wi14 - OPERATING SYSTEM CONCEPTS Combined LAB LEC Wi14

## Week-by-Week Summary

### Week 1: Jan. 6 - Jan. 10

- Check out Course Outline and Reference Materials. Be sure to read and participate in the Course Newsgroup. Also consider giving feedback and fun videos/comics on the Anonymous Video and Feedback Forum.
- Mon, Jan 6: Introductory Lecture
- Wed, Jan 8 and Fri, Jan 10: Lecture 2 - Chapter 2; Note that this loosely follows the topics in Chapter 2, but adds on other topics I think should be introduced now. This will be true of all the lectures. Sometimes uploaded slides will span multiple days; there will always be a slide showing the start of each lecture.
- We looked through the examples mem.c, seg1.c seg2.c and loopstomp.c - be sure to try them out in the lab yourself and play with them.
- We talked about system calls, linux "syscall" and the system call numbers in /usr/include/sys/syscalls.h and /usr/include/asm/unistd.h. Looking at Linux code and headers can show you a lot about how the code works and shows you examples of good programming.
- We looked through and ran syscalls.c, talking about /usr/bin/time and strace for profiling code and using a light debugger.
- **Read** Chapter 1 by Monday (Jan 13), Chapter 2 and 3 by Friday (Jan 17)

### Week 2: Jan. 13 - Jan. 17

- Mon, Jan 13: Lecture 4 - Chapter 3; Started talking about processes, and looked at task\_struct in linux/sched.h (which stores process info)
- Wed, Jan 15: Lecture 5 - Chapter 3; Ran sig1.c and sig2.c showing how to catch signals with signal handlers and how to use sigsetjmp/siglongjmp.
- Fri, Jan 17: Lecture 6 - Chapter 3; Ran fork1.c, fork2.c showing how to create processes with fork and dumbshell.c (a simple shell) showing how its useful to spawn children with fork() to run tasks. Ran zombie.c to show that parents that don't call wait() can cause processes to become zombies.
- Should have read Chapter 1, 2 and 3 so you can complete Exercise 1.

## Week 3: Jan. 20 - Jan. 24

- **Reminder:** Exercise 1 due Wednesday January 22 at 8:30 a.m.
- Mon, Jan 20: Lecture 7 - Chapter 3; Continued talking about IPC, focusing on message passing. Discussed UNIX signals as a rudimentary form of message passing and tried to think of if the message passing was direct/indirect, synchronous/asynchronous, etc.
- Wed, Jan 22: Lecture 8 - Chapter 4; Started talking about threads, why we care about multiprocessing and examples where threads can be more useful than processes. Looked at code example `pthread_hello.c`, `pthread_hello_arg2.c` and `pthread_join1.c`
- Fri, Jan 24: Lecture 9 - Chapter 4; finished discussing threads, particularly issues with dealing with threads, and the distinction between user-level and kernel-level threads.
- Should read Chapters 4 and 5 (at least by February 7, so can complete Exercise 2).

## Week 4: Jan. 27 - Jan. 31

- **Reminder:** Assignment 1 due Friday January 31 at 6:00 p.m.
- Mon, Jan 27: Lecture 10 - Chapter 5; Started talking about synchronization, introducing the idea of mutual exclusion, race conditions and critical sections. Talk a bit about initial solutions to synchronizing two processes.
- Wed, Jan 29: Lecture 11 - Chapter 5; Talked about atomic operations and spin locks for mutual exclusion. Introduced non-blocking semaphores for locking (binary semaphores) and synchronizing many resources/tasks (counting semaphores)
- Fri, Jan 31: Lecture 12 - Chapter 5; Went through several examples to apply mutual exclusion techniques. Introduced monitors, which enforce mutual exclusion, unlike spin locks or semaphores, which rely on the programmer to use them correctly.

## Week 5: Feb.3 - Feb. 7

- **Read** Chapters 4 and 5 by end of week; try to read Chapter 6 because we will start into scheduling this week.
- Mon, Feb 3: Lecture 13 - Tackling Big Projects; Talked about some skills that are useful to have for starting and completing projects; hopefully this will help you tackle Assignment 2, and better modularize your code. Look a little bit `server.c` and `client.c`.
- Wed, Feb 5: Lecture 13 - Tackling Big Projects and Lecture 14 - Chapter 6; Finished off discussion about tackling projects and some stuff to think about for your server assignment. Small review of Chapters 1-5, with pthread exercise to get you thinking about applying your knowledge. Introduced scheduling and some of the criteria we need to think about.
- Fri, Feb 7: Lecture 15 - Chapter 6; Mentioned a bit more about how you might write

- Thu, Feb 7: Lecture 15 - Chapter 6; mentioned a bit more about how you might write and test your server code for your assignment. Dove right into different scheduling algorithms.

## Week 6: Feb. 10 - Feb. 14

- **Reminder:** Exercise 2 due Wednesday February 12 at 8:30 a.m.
- Mon, Feb 10: Lecture 16 - Chapter 6; Talked a little bit about predicting CPU-bursts, priority scheduling (with binary heaps) and multilevel queues, finishing off the scheduling algorithms that we will go through in the textbook (feel free to read about realtime algorithms in Section 6.6 anyway).
- Wed, Feb 12: Lecture 17 - Chapter 6; Showed how some of these scheduling ideas are used in practice (in Linux's CFS scheduler). Talked about the hardness of multiprocessor scheduling (load balancing) and ways to tackle that.
- Fri, Feb 14: Ankush Roy will go through `select_server.c`, which will hopefully make it more manageable to tackle your second assignment.

## Week 7: Feb. 17 - Feb. 21

- **Reading Week, No classes**
- **Reminder:** make sure you have read Chapter 6 and remember that Exercise 3 due on Feb. 26
- **Reminder:** Assignment 2 is not trivial, so hopefully you've started

## Week 8: Feb. 24 - Feb. 28

- **Reminder:** Exercise 3 due Wednesday February 26 at 8:30 a.m.
- Mon, Feb 24: Lecture 18 - Chapter 7; Introduced deadlocks, the four necessary conditions for deadlock and resource allocation graphs as a useful tool for analyzing potential deadlock situations.
- Wed, Feb 26: Lecture 19 - Chapter 7; Continued into deadlock prevention using resource ordering to prevent circular waiting, with mild success trying to convince you that the proof that deadlocks cannot occur was both useful and interesting. Starting talking about deadlock avoidance with the notion of safe sequences and the Banker's algorithm to find safe sequences.
- Fri, Feb 28: Lecture 20 - Chapter 7; Finished off deadlocks with deadlock detection techniques.

## Week 9: Mar. 3 - Mar. 7

- **Reminder:** Assignment 2 due Friday March 7 at 6:00 p.m.
- The lecture time on March 3 and March 7 will be help sessions for the second assignment, in Lab CSC 2-19 and Lab CSC 1-59. Go to lab CSC 2-19 first, and if it is full,

then go to CSC 1-59.

- You will have an amazing guest lecturer, Professor Paul Lu! He will talk about memory management. He taught material from slides 1-16.

## Week 10: Mar. 10 - Mar. 14

- Mon, Mar 10: Lecture 21 - Chapter 8; Discussed swapping, fragmentation from static and dynamic allocation and, after a good historical basis of how to lay out memory, we finally got into paging, a typical modern approach. We didn't get to the proof about selecting page size, but we at least got to see Jean Chretien's support of proofs.
- Wed, Mar 12: Lecture 22 - Chapter 8; We talked about the implementation of page tables, particularly talking about hashed page tables, multi-level page tables, inverted page tables and finally, for hardware, the TLB.
- Fri, Mar 14: Dr. Paul Lu talked about Virtual Memory (Chapter 9) and amazed you with how great he is.

## Week 11: Mar. 17 - Mar. 21

- **Reminder:** Exercise 4 due Wednesday March 19 at 8:30 a.m.
- Mon, Mar 17: Lecture 23 - Chapter 9; we talked about page replacement strategies and criteria. We discussed least-recently-used (LRU) in depth, implemented with the clock algorithm. We also discussed thrashing and issues with page fault rates.
- Wed, Mar 19: Lecture 24 - Chapter 10; we talked about storage structure particularly for magnetic disk, allocation strategies and, of course, disk scheduling and the importance of seek distance as a criteria for scheduling disk.
- Fri, Mar 21: Lecture 25 - Testing and Kernel Modules; we talked about how to test scheduling environments, focusing on how to use models and data for simulation. I demoed a simple kernel module, so you could see how kernel code looks similar but with some important differences.

## Week 12: Mar. 24 - Mar. 28

- Mon, Mar 24 and Wed, Mar 28: Bob Beck will talk about the historical smashing the stack paper and exploit mitigation techniques. You have to read the paper before these two lectures, as Bob will assume you have this knowledge. A question from these lectures is on Exercise 5! And, besides, Bob is an expert in security and you will learn some really interesting stuff in this lecture.
- Friday, Mar 28: Jeeva Paudel, will lecture about interesting recent developments in parallel programming that make it easier to parallelize tasks. He is a very engaging speaker, and plus, he will make one question that will be on the final; for both of these reasons, you should give him your undivided attention!

## Week 13: Mar. 31 - Apr. 4

- **Reminder:** Exercise 5 due Wednesday April 2 at 8:30 a.m.
- Mon, Mar 31: Lecture 26 - Chapter 15; we talked about security, focusing on some of the terminology and categories of security breaches at a high-level.
- Wed, Apr 2: Lecture 27 - Chapter 15; we talked a bit more in-depth about network security breaches, such as the Morris worm (including source code).
- Fri, Apr 4: Lecture 27 - Chapter 15; we finished talking about security breaches and started into cryptography as a tool for security. I tried to use bitcoin as an example of a system that needs strong security, and that uses cryptographic hash functions in an interesting way. It is a good example of how to use cryptographic tools well, though this message may have been slightly lost due to the fact that it is a somewhat confusing/complicated example at first.

## Week 14: Apr. 7 - Apr. 9

- Mon, Apr 4: Lecture 28 - Chapter 15; finished talking about cryptographic tools and encryption. Started into how to review.
- Wed, Apr 6: Lecture 29 - Review; went through the Practice Final.
- **Reminder:** Assignment 3 due Wednesday April 9 at 6:00 p.m.

## Week 15: Apr. 16, 2014

- **Monday:** I will have office hours at 11-12, as usual. If you would like to ask questions/discuss the course discuss outside of this time, please email me to setup a time.
- **Reminder:** the final is at 9:00 a.m. in CSC B-10 (regular lecture location). It is 3 hours long and open-book (but no electronic devices).

Last modified: Sunday, 13 April 2014, 9:55 AM