Carleton University
Department of Systems and Computer Engineering
SYSC 4001B: Operating Systems, Winter 2020
Course Outline

Instructor Information and Office hours

Instructor: Thomas Kunz
tkunz@sce.carleton.ca
Office Hours: Tuesdays 2-3 pm, CB 5202.

TA Information and Office hours

Posted on the course website, http://kunz-pc.sce.carleton.ca/sysc4001/

Calendar Description

Introduction to operating system principles. Processes and threads. CPU scheduling. Managing concurrency: mutual exclusion and synchronization, deadlock and starvation. Managing memory and input/output. Concurrent programming, including interprocess communication in distributed systems. See also the description at http://calendar.carleton.ca/undergrad/courses/SYSC/

Prerequisites

Precludes additional credit for SYSC 3001 and COMP 3000.
Prerequisite(s): SYSC 2006 with a minimum grade of C-.

Students who have not satisfied the prerequisites for this course must either withdraw from the course or obtain a prerequisite waiver by visiting the Engineering Undergraduate Academic Support Office.

Assumed Knowledge

Upon entry into this course, students are expected to have basic programming skills (in C).

Course Objectives

Operating systems exploit the hardware resources of one or more processors to provide a set of services to system users. The operating system also manages secondary memory and I/O devices on behalf of its users. There is a tremendous range and variety of computer systems for which operating systems are being designed: from embedded devices (e.g., the on-board computers for the space shuttle or a luxury sedan) and smartphones over PDAs and Laptops to PCs, workstations, and mainframes, to supercomputers. The variety is not just in the capacity and speed of machines, but in applications and system support requirements. Also, operating system research has been characterized by a rapid rate of change, and there is no indication that this will let up. Just think about such topics as the Java Virtual Machine, Android, or iOS.

In spite of this variety and pace of change, certain fundamental concepts apply consistently throughout. The intent of this course is to provide a thorough discussion of the fundamentals of operating system design, and to relate these to contemporary design issues and current directions in the development of operating systems. The objective is to provide you with a solid understanding of the key mechanisms of modern operating systems, the types of design tradeoffs and decisions involved in OS design, and the context within which the operating system functions (hardware, other system programs, application programs, interactive users).

Learning Outcomes

Students will learn the following:

1. Understand the concepts of (and differences between) processes and threads
2. Write concurrent programs using IPC, semaphores, and other operating system services
3. Identify cases of deadlock, livestock and starvation
4. Know different techniques for Memory management
5. Know different techniques for CPU scheduling, including uniprocessor scheduling, multiprocessor scheduling, and real-time scheduling
6. Explain how I/O devices challenge an Operating System and what general purpose features are provided by an OS to manage the collection of vastly different I/O devices
7. Explore different disk scheduling policies
8. Understand how file systems work and are organized
9. Can conduct comparative studies on different management algorithms for memory, CPU, file systems, I/O

Graduate Attributes (GA's)

The Canadian Engineering Accreditation Board requires graduates of engineering programs to possess 12 attributes at the time of graduation. Activities related to the learning outcomes listed above are measured throughout the course and are part of the department's continual improvement process. Graduate attribute measurements will not be taken into consideration in determining a student's grade in the course. For more information, please visit: https://engineerscanada.ca/.

Graduate Attribute	Learning Outcome (s)
1.4 Programming and algorithms	2, 4, 5
1.5 Computer systems	1,3
3.3 Experimental procedure	9
3.5 Interpretation of data (synthesis) and discussion	9
4.1 Clear design goals	2, 6, 7, 8
4.5 Design implementation / task(s) definition	9
5.3 Tools for design, experimentation, simulation, visualization, and analysis	2, 9

Textbooks (or other resources)

A number of operating systems textbooks exist, covering most or all of these topics. However, the terminology and approaches differ slightly, and to do justice, we need to use a relatively recent textbook. Therefore, the following textbook is mandatory (that is, I expect students to have access to one):

- William Stallings, Operating Systems: Internals and Design Principles, 9th edition, Pearson 2018, ISBN-9780134670959.

We will not cover all chapters in this textbook, but will focus on Chapters 2-12. The 4th edition of the textbook won the 2002 Texty Award for the best Computer Science and Engineering textbook, awarded by the Text and Academic Authors Association, Inc. Its particular strength is the in-depth discussion of the various topics by making explicit reference to the implementation of the basic concepts in popular operating systems: Windows 8, Linux, and Android. Additional information is available from the course homepage, http://kunz-pc.sce.carleton.ca/sysc4001/. Finally, I will use cuLearn for managing course interactions, grades, as well as assignment and lab submissions.

For the programming assignments/labs in this course, you may also want to consult the following book (the PDF version is posted on cuLearn). Note that the book does not help with the course lectures, but is a very useful reference working with/programming in C and Linux.

- Neil Matthew; Richard Stones, Beginning Linux Programming, 4th Edition, Wrox 2007, Print ISBN-13: 978-0-470-14762-7, Web ISBN-10: 0-470-14762-8

Evaluation and Grading Scheme

There will be five assignments, collectively worth 20% of the final mark, lab assignments worth 10%, a midterm exam worth 20% and a final exam worth 50%. To pass the course (in addition to obtaining an appropriate overall mark), the final exam must be passed (i.e., obtain at least 50%). The final examination is for evaluation purposes only and will not be returned to students. You will be able to make arrangements with the instructor or with the department office to see your marked final examination after the final grades have been made available.

Breakdown of course requirements

The midterm exam will be Tuesday, February 25 in the evening, from 6 pm to 7:30 pm. Due to space limitations, we cannot write it during the regular class time. The midterm will be written in Health Science 1301.

The final exam will be scheduled during the university exam period, April 2020. Students who miss the final exam may be granted permission to write a deferred examination (see the Undergraduate Calendar for regulations on deferred exams). These students have additional months to study and a less crowded examination schedule compared to their colleagues who write the final exam in April. As such, it is only fair to expect substantially better performance from these students on the deferred examination than on the final exam. Note that the above

formulation leaves it up to the instructor whether the deferred examination will be harder or the marking scheme will be more rigorous.

There will be a number of assignments and labs. While the lectures will focus on general principles and algorithms, the labs and assignments will deal with some of the issues in the context of a real operating system, Linux. Assignment due dates will be clearly stated on the assignment handouts. Late assignments will not normally be accepted, and will receive a mark of 0; however, students who cannot submit an assignment by the due date for valid medical or compassionate reasons should contact the instructor immediately and prior to the due date to arrange for appropriate accommodations (e.g., an extension of the due date). Arrangements must be made in a timely manner, otherwise the assignment will be considered late and not accepted. See also the discussion on deferred term work below.

Students are encouraged to discuss design issues when working on assignments and labs; however, you are expected to write your own programs. There is a fine line between cooperating with your colleagues (discussing problems and ideas) and copying program code (plagiarism). Not only is plagiarism an instructional offense (see the current Undergraduate Calendar, Academic Regulations of the University, Section 10), but doing the assigned work by yourself is by far the best way to prepare for the exams. To facilitate discussion of assignment-related issues, *cuLearn* maintains a discussion topic for each assignment that will be monitored by the TAs and myself.

Both exams will be **open textbook**. Only proper hardcopies of the official course textbook will be accepted, no alternative textbooks, photocopies, ebooks, etc. However, I will allow older versions of the textbook.

Due Dates: Right now, I plan to schedule the assignments as indicated in the table below. No late deadlines will be given. If you foresee any problem with an assignment deadline, please come and talk to me **early**, in particular **before** the deadline is passed.

	Handed Out	**Due**	**Weight**
Assignment 1	January 16	January 29, noon	2 %
Assignment 2	January 30	February 12, noon	4 %
Assignment 3	February 13	March 4, noon	4 %
Assignment 4	March 5	March 18, noon	5 %
Assignment 5	March 19	April 3, noon	5 %
Lab Assignments	See below		10%
Midterm Exam	February 25	6 pm to 7:30 pm	20 %
Final Exam	April Exam Period		50 %

The lab assignments should not take all three hours of scheduled lab time, allowing you time to also work on the programming assignments. That is particularly true for weeks without lab assignments (for example, the week of the midterm or the last week in the term). The TA will be available during the scheduled lab times to help with both the lab assignments as well as the programming assignments. The list of lab assignments and weeks are as follows:

	Topic
Lab 1	Linux Basics
Lab 2	Linux Basics and Processes
Lab 3	Processes and Signals
Lab 4	Semaphores
Lab 5	Posix Threads
Lab 6	Shared Memory
Lab 7	Message Queues
Lab 8	Linux Scheduler
Lab 9	Low-Level I/O
Lab 10	Socket Programming

There will be a total of 10 lab assignments, which you are expected to work on during the assignment lab periods. These labs will exercise various aspects of the Linux OS. Each lab assignment will ask you to submit something using *cuLearn* before the end of the lab. At the end of the term, we will check your submissions (to ensure you did complete the lab), and assign marks based on how complete your set of lab assignments is. You can earn 1 mark per lab, for a total of 10 marks.

Tentative Week-by-Week breakdown

Weeks 1-2 – Introduction and Operating System Overview
Week 3 – Process Description and Control
Week 4 – Threads
Week 5 – Concurrency: Mutual Exclusion and Synchronization
Week 6 – Concurrency: Deadlock and Starvation
Week 7 – Memory Management
Week 8 – Virtual Memory
Week 9 – Uniprocessor Scheduling
Week 10 – Multiprocessor and Real-Time Scheduling
Week 11 – I/O Management and Disk Scheduling
Weeks 12-13 – File Management

General Regulations

Attendance: Students are expected to attend all lectures and lab periods. The University requires students to have a conflict-free timetable. For more information, see the current *Undergraduate Calendar, Academic Regulations of the University, Section 2.1.3, Course Selection and Registration and Section 2.1.7, Deregistration.*

Health and Safety: Every student should have a copy of our Health and Safety Manual. A PDF copy of this manual is available online: http://sce.carleton.ca/courses/health-and-safety.pdf

Deferred Term Work : Students who claim illness, injury or other extraordinary circumstances beyond their control as a reason for missed term work are held responsible for immediately

informing the instructor concerned and for making alternate arrangements with the instructor and in all cases this must occur no later than three (3.0) working days after the term work was due. The alternate arrangement must be made before the last day of classes in the term as published in the academic schedule. For more information, see the current *Undergraduate Calendar, Academic Regulations of the University, Section 4.4, Deferred Term Work.*

Appeal of Grades : The processes for dealing with questions or concerns regarding grades assigned during the term and final grades is described in the *Undergraduate Calendar, Academic Regulations of the University, Section 3.3.4, Informal Appeal of Grade and Section 3.3.5 Formal Appeal of Grade.*

Academic Integrity: Students should be aware of their obligations with regards to academic integrity. Please review the information about academic integrity at: https://carleton.ca/registrar/academic-integrity/. This site also contains a link to the complete Academic Integrity Policy that was approved by the University's Senate.

Plagiarism: Plagiarism (copying and handing in for credit someone else's work) is a serious instructional offense that will not be tolerated.

Academic Accommodation: You may need special arrangements to meet your academic obligations during the term. You can visit the Equity Services website to view the policies and to obtain more detailed information on academic accommodation at http://www.carleton.ca/equity/ For an accommodation request, the processes are as follows:

- **Pregnancy or Religious obligation**: Please contact your instructor with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details see https://carleton.ca/equity/wp-content/uploads/Student-Guide-to-Academic-Accommodation.pdf
- **Academic Accommodations for Students with Disabilities**: The Paul Menton Centre for Students with Disabilities (PMC) provides services to students with Learning Disabilities (LD), psychiatric/mental health disabilities, Attention Deficit Hyperactivity Disorder (ADHD), Autism Spectrum Disorders (ASD), chronic medical conditions, and impairments in mobility, hearing, and vision. If you have a disability requiring academic accommodations in this course, please contact PMC at 613-520-6608 or pmc@carleton.ca for a formal evaluation. If you are already registered with the PMC, contact your PMC coordinator to send me your ***Letter of Accommodation*** at the beginning of the term, and no later than two weeks before the first in-class scheduled test or exam requiring accommodation (*if applicable*). **Requests made within two weeks will be reviewed on a case-by-case basis**. After requesting accommodation from PMC, meet with me to ensure accommodation arrangements are made. Please consult the PMC website (www.carleton.ca/pmc) for the deadline to request accommodations for the formally-scheduled exam (*if applicable*).
- **Survivors of Sexual Violence:** As a community, Carleton University is committed to maintaining a positive learning, working and living environment where sexual violence will not be tolerated, and where survivors are supported through academic

accommodations as per Carleton's Sexual Violence Policy. For more information about the services available at the university and to obtain information about sexual violence and/or support, visit: https://carleton.ca/sexual-violence-support/.

- **Accommodation for Student Activities:** Carleton University recognizes the substantial benefits, both to the individual student and for the university, that result from a student participating in activities beyond the classroom experience. Reasonable accommodation must be provided to students who compete or perform at the national or international level. Please contact your instructor with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details, see https://carleton.ca/senate/wp-content/uploads/Accommodation-for-Student-Activities-1.pdf