Operating Systems
ECSE 427/COMP 310 Winter 2018

Instructor:

Dr. Rola Harmouche
Department of Electrical and Computer Engineering
Office: McConnell Engineering Building, Room 637
Email: rola.harmouche@mcgill.ca
Office Hours: Tuesdays 1pm-2pm and Thursdays 10am-11am. Dr. Harmouche is also available outside of office hours. Please contact to arrange a time. Check mycourses for any changes to office hours.

Lectures:

- Tuesday, Thursday and Friday, 8:35AM-9:25am Macdonald Harrington Building G-10, Jan 08, 2018 - Apr 16, 2018.

Class tests:
Class test 1: Friday February 16, 8:35AM-9:25am, rooms TBD.
Class test 2: Friday March 23rd , 8:35AM-9:25am, rooms TBD.

Tutorials:

- Wednesdays, 2:35 PM - 3:25 PM, Macdonald Engineering Building 279 (Engineering students)

- Thursdays 4:30 PM or 5:30 PM (Computer science students – check mycourses for confirmation)

Tutorials start on the week of January 8, 2018

FAQ hour:

- TBA (check mycourses): The TAs will be dedicating 1 hour a week to answer student assignment / programming questions and issues in a group setting.

Teaching Assistants:

Information on teaching assistants is available on my courses.

Credits: 3. Course Hours: (3,1,5) (Lectures, Labs and tutorials, outside work).

Prerequisites: ECSE 322 or COMP 273.

Required Textbook: A. Tenenbaum and H. Bos, Modern Operating Systems, 4th Edition, Pearson, 2015. (Available at the McGill Bookstore)

Web page: Information for the course will be posted on the course myCourses web page. The myCourses page for the course should be the first place you look to get information about the course. The use of email and discussion board for communication with the instructors or the teaching assistants is encouraged. You must send email from myCourses or from your McGill Email address.

Course Description and Learning Outcomes:

Description: This is an introductory course in computer operating systems. In this course we will study the theoretical and practical concepts behind modern operating systems. In particular, we will study the basic structure of an operating system, its components, design strategies, algorithms and schemes used to design and implement different components of an operating system. Major components to be studied include: processes, inter-process communication, scheduling, memory management, virtual memory, storage management, network management, and security.

Primary learning outcome: To get a clear understanding of the major principles/ algorithms that underlie an operating system and how they interplay within it.

Secondary learning outcomes: After taking this course, you should be able to: (i) identify the core functions of operating systems and how they are architected to support these functions, (ii) explain the algorithms and principles on which the core functions are built on, (iii) explain the major performance issues with regard to each core function, and (iv) discuss the operating system features required for particular target applications.

Course Content:

CONCEPT MAP
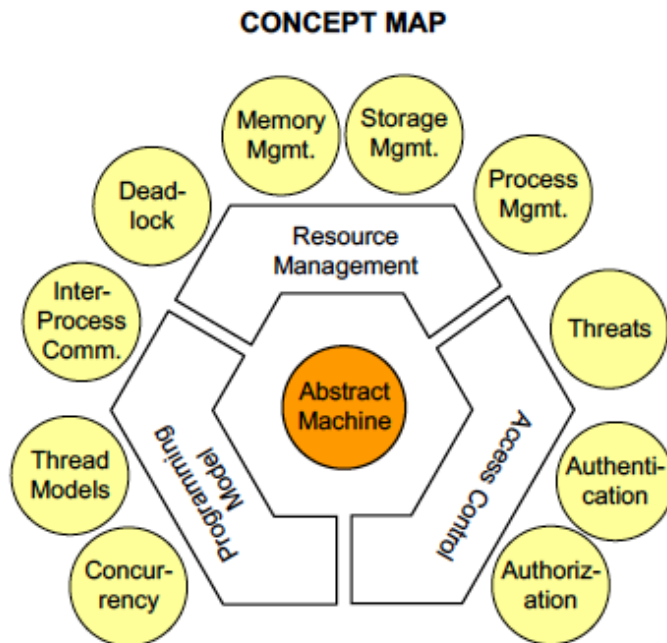
Course Schedule:

The table below shows a **tentative course schedule.** I will make a "best effort" to stick to the proposed schedule. Except for the Midterm, dates might change.

Week	Topics	Reading Material	Comments
1	OS Introduction: OS tour; simple OS; beyond simple OS	1.5, 1.7	1.8 highly recommended
2	Process Concept: processes, introduction to process scheduling, IPC, Threads – multithread programming, issues	1.6, 2.1, 2.2	1_{st} Mini Assignment out
3	Synchronization: Critical section problem, Peterson's solution, synchronization hardware, mutex, semaphores, monitors, well-known problems in synchronization	2.3, 2.5	
4	Deadlocks: models, characterization, prevention, avoidance, detection, recovery	Chap. 6	2_{nd} Mini Assignment out
5	File Systems: file concept, file system structure, implementation, allocation methods, free-space management, efficiency & performance, access methods, directories, disk structure, disk scheduling, etc	4.1-4.5	
6	Advanced File Systems: Log-structured file systems, fault tolerance, examples of modern file systems, SSD issues	Same as week 5	Class test 1
7	CPU Scheduling: basic concepts, scheduling criteria, algorithms, thread scheduling, real-time scheduling, lottery scheduling, stride scheduling, performance of scheduling algorithms	2.4, 10.3.4	
8	Main Memory: swapping, memory allocation, segmentation, paging	3.1-3.7	Programming Assignment out
9	Virtual Memory: demand paging, copyon-write, page replacement, allocation of frames, thrashing	Same as week 8	
10	Virtualization and Clouds: Virtualization requirements, hypervisors, memory virtualization, I/O virtualization, clouds	7.1-7.11	Class test 2
11	Security: goals of protection, domain of protection, access matrix, implementation,	9.1-9.4	

	access control, capability-based systems		
12	Security: security problem, program threats, system & network threats, user authentication	9.7-9.10	

Instructional Method:
The course will consist of three hours of instructor led classes per week together with a maximum of one hour of tutorial per week taken by the TAs. The class time will be devoted to the presentation and development of new concepts and the application of these concepts to examples and problems, while the tutorials will discuss solutions to the programming assignments/projects and written assignments. The primary focus of the tutorials is to provide sufficient "how-to" knowledge through the discussion of the assignments to help in the development of the programming project series.
Students are strongly encouraged to use the My Courses discussion groups to talk about the programming and written assignments. These discussion groups will be monitored by the TAs and by the instructor for providing the necessary answers. For **answering questions pertaining to lecture and assignment material, priority will be given to the My Courses discussion board over emails.**

Assignment information:
Programming assignments must be done using the C programming language.
Assignments will be graded on Ubuntu 16.04. Please follow all submission instructions. Failure to do so will result in deducted marks. You are expected to make sure that your programs run properly using the gcc compiler on the following McGill SSH servers:
- mimi.cs.mcgill.ca
- linux.cs.mcgill.ca
- ubuntu.ca.mcgill.ca
In particular, assignment will be graded on mimi.cs.mcgill.ca. Finally, you are expected to debug your code using the tools available. The first two tutorials will focus on helping you with C basics, setting up your environment, and using the necessary tools for completing and debugging your assignments.

Evaluation Procedures:

Activity	Weight (in percent)
Mini Programming Assignment 1:	8%
Mini Programming Assignment 2:	9%
Programming Assignment:	13%
Class test 1 (during class time)	10%
Class test 2 (during class time)	10%
Final (comprehensive)	50%
Total	100%

Double Grading Policy: This course has a significant portion of the grade allocated for

the programming component. You are expected to submit only your work in these assignments. You can receive advice or tips from others (instructor, teaching assistants, or peers), but the final submission should be yours. You are expected to know all the design decisions in the program and explain all aspects of the program handed in as part of the assignment. To test this condition, we will randomly select some students and ask them to explain their programming assignments. The eventual marks for an assignment will be the minimum of the two marks. For example, if 85 is the marks obtained in the first (normal) evaluation of the programming assignment and 50 is the marks obtained in the second evaluation, then effectively you have 50. We will also be using tools to check for plagiarism.

Late Assignment Policy: There will be two deadlines for each assignment: proper deadline and cut-off date. After the proper deadline, there will be a penalty of 10% for each day the assignment is late until the cut-off date. After the cut-off date, the assignment cannot be handed in. No individual requests for extensions will be granted unless they are for medical reasons.

The deadlines will be set for 11:55 pm or 11:59pm. Please observe the time and date very carefully. It is your responsibility to make sure that the assignment is properly submitted via MyCourses.

Regrading Policy: If you find your assignments or exams are not marked according to the marking scheme, you are encouraged to consult me or the TAs. When you resubmit your assignment or exam for regarding, we reserve the right to regrade the full exam or assignment without restricting the attention to the disputed portion.

Academic Integrity: McGill University values academic integrity. Therefore, all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see www.mcgill.ca/students/srr/honest/ for more information). (approved by Senate on 29 January 2003)

In accord with McGill University's Charter of Students' Rights, students in this course have the right to submit in English or in French any written work that is to be graded." (approved by Senate on 21 January 2009 - see also the section in this document on Assignments and evaluation.)