COMP-310 / ECSE-427
Operating Systems (OS), Winter 22

McGill University, School of Computer Science

General Information

- **Instructor** Prof. Oana Balmau
- **When** Mondays, Wednesdays, Fridays 9:35am-10:25pm
- **Where** *https://mcgill.zoom.us/j/82520535065*
  (Live session link; check detailed syllabus for lab links)
- **Course Credits** 3
- **Course website** MyCourses
- **Ed link** *https://edstem.org/us/courses/16123/discussion/*

OS Team

Instructor: Prof. Oana Balmau oana.balmau@cs.mcgill.ca

TAs/CAs:

- Murray Kornelsen murray.kornelsen@mail.mcgill.ca
- Paul-Andre Henegar paul-andre.henegar@mail.mcgill.ca
- Loic Ho-Von loic.ho-von@mail.mcgill.ca
- Yong Zhang yong.zhang3@mail.mcgill.ca
- Changjun Zhou changjun.zhou2@mail.mcgill.ca
- Samuel Bergeron samuel.bergeron@mail.mcgill.ca
- Mushfique Rahman mohammad.rahman4@mcgill.ca

Graders:

- Jenny Cao yuanhui.cao@mail.mcgill.ca
- Jiaxuan Chen jiaxuan.chen2@mail.mcgill.ca

Email etiquette:

- *All **course-related discussion** should **be posted on Ed**, for everyone's benefit and fastest replies.*
- *For issues with **grading**, email the **responsible TA/CA/grader (not the instructor).***
- *For **personal and medical** issues, feel free to **send email to instructor**.*
- *Please **add the tag [COMP310/ECSE427]** in the beginning of your email subject, to make sure the email does not get lost in instructor's inbox.*
- *Please include your **name and McGill ID** in all email communication.*
- *Before sending email, please review the McGill email etiquette.*

Office Hours

Who?	When?	Where?
Oana Balmau	Mon 10:30am – 11:30am Weeks 2 – 15	Zoom: *https://mcgill.zoom.us/j/6796939400*
Murray Kornelsen	TBD	Zoom: TBD
Paul-Andre Henegar	TBD	Zoom: TBD
Loic Ho-Von	TBD	Zoom: TBD
Yong Zhang	TBD	Zoom: TBD

Samuel Bergeron	TBD	Zoom: TBD
Mushfique Rahman	TBD	Zoom: TBD

- *Office hours **start January 10** and **end April 11.** No office hours on public holidays and Reading week.*

Course Descriptions

This is an introductory course in Operating Systems (OS). In this course, students study the theoretical and practical aspects of modern OS. This includes the basic structure of an OS, as well as algorithms and techniques used to design and implement different components of an OS (e.g., processes, inter-process communication, scheduling, memory management, paging, and storage management).

Learning Objectives:

Primary learning objective:
- Explain the objectives and functions of modern Operating Systems (OS).

Secondary learning objectives:
- Analyze the tradeoffs inherent in OS design.
- Describe how computing resources are used by applications and managed by the OS.
- Contrast kernel and user mode in an OS.
- Describe how an OS is constructed of layers upon layers, based on separation of concerns, with well-defined interfaces.
- Describe the need for concurrency within the framework of an OS.
- Compare and contrast the common algorithms used scheduling of tasks in an OS.
- Explain memory hierarchy and cost-performance tradeoffs.
- Summarize the principles of virtual memory as applied to caching and paging.
- Compare and contrast different approaches to file organization.
- Summarize the use of journaling and how log-structured file systems enhance fault tolerance.

Prerequisites

- **ECSE-322** or **COMP-273**.
- Though it is not a formal prerequisite, students are encouraged to **review C programming basics.**

Evaluation Criteria

The class grade will be based on the following components.

- **Project,** consisting of 3 C programming assignments that build on each other
  Milestones:
  - Assignment 1: OS Shell — **10%** **Due: February 14**
  - Assignment 2: Scheduling — **20%** **Due: March 7**
  - Assignment 3: Memory management — **20%** **Due: April 4**
- **C Quiz** — **10%** **Date: February 25**
- **Written exam** — **40%** **Date: Winter 22 exam session.**

Note: If your written exam grade is higher than your C quiz grade, then the C quiz will not count, i.e., the written exam will count for 50%.

Late policy: No extensions. Please plan your work accordingly. **If an assignment is submitted past the deadline, it will receive a 0.** In rare exceptional situations, extensions may be granted on a case-by-case

basis if the delay is caused by a medical issue. If you are in this situation, please contact the instructor *before the deadline* by email. If you contact the instructor past the deadline, it is too late and you will receive 0 points.

The OS staff is committed to the principle of equal access. We encourage you to contact the instructor and the Office for Students with Disabilities to discuss your disability-related needs, including accommodations which you may need to succeed in this course.

Collaboration policy:

- The OS programming assignments can be done **individually, or in teams of 2.**
- If you commit to working in a team, **the team must be the same for all 3 assignments.**
- Please let the instructor know if you will be working in a team (via the Assignments tab in MyCourses), **no later than 7 days after the add/drop deadline.**
- If you work in a team, **both students receive the same grade for all assignments.**
- All the other quizzes/exams will be graded individually.

Deferred/Supplemental exams: Under rare circumstances (such as serious medical conditions, or probationary academic standing https://www.mcgill.ca/exams/dates/supdefer) it is possible to apply for a deferred or a supplemental exam. In case the request is approved, a deferred exam will count for 40% of the final grade. For a supplemental exam, the exam will count for 50% of the final grade. Note that the course has a heavy programming component that cannot be replaced by writing a supplemental exam.

Course Content

Detailed weekly schedule will be posted on MyCourses. Topics covered:

- What is an Operating System?
- User mode vs Kernel mode
- Process management
- Scheduling
- Synchronization primitives
- Multi-process structuring
- Inter-process communication
- Multithreading
- Virtual memory
- Demand paging
- File systems implementation
- Crash recovery in file systems
- Advanced topics: Virtual machines, OS security

Course Logistics

The course has 3 components: (1) Recorded lectures, (2) Live Sessions, and (3) Labs.

1. **Recorded Lectures:** Typically released on Mondays on MyCourses (but see detailed schedule). The lectures are designed to teach students the technical details necessary to understand key design aspects in operating systems and apply them in their own systems.

2. **Live Sessions:** Typically held on Wednesdays on Zoom (but see detailed schedule). These sessions are designed to provide an opportunity for students to practice their problem-solving skills. Each session is dedicated to solving exercises related to the Lecture of the week. Students need to come prepared i.e., by watching the week's recorded lecture to be able to productively participate in the class discussion and ask questions.

Live sessions will be recorded and will be posted to MyCourses the following day in case students cannot attend the zoom call. The instructor will lead the session.

3. **Lab sessions:** Typically held on Fridays on Zoom (but see detailed schedule). The goal of the lab sessions is two-fold: (1) Give a refresher on C programming and trickier topics that will come up as students work on the programing assignments, (2) Give an overview of each programming assignment and give students an opportunity to ask live questions about the projects (in addition to the office hours time and the Ed discussion board). The Zoom links for the lab sessions will be announced on MyCourses and Ed. **Lab sessions will be recorded and will be posted to MyCourses the following day in case students cannot attend the zoom call. Lab sessions are led by TAs/CAs.**

Reference Materials

The main course content is made of lectures, interactive sessions, and programming assignments. To complement the course content, students can use:
Operating Systems: Three Easy Pieces, by Remzi Arpaci-Dusseau and Andrea Arpaci-Dusseau.
The full e-book is available online for free: https://pages.cs.wisc.edu/~remzi/OSTEP/

Evaluation policies and 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 here and here for more information).

We encourage you to discuss the course materials with your peers, **but every assignment you turn in must be your own work.**

Cheating, plagiarism, and any form of dishonesty will be handled with maximum severity. The OS staff will use automated systems to detect possible cases of text or software plagiarism. Cases that warrant further investigation will be referred to the university disciplinary officers. If you are ever in doubt about whether an action on your part may constitute unacceptable collaboration, please ask the course staff before proceeding—*doing so afterward is too late*.

It is the responsibility of each student to carefully acknowledge all sources (e.g., papers, code, books, websites, individual communications) when submitting work.

Copying solutions obtained from the web (GitHub/Google/Chegg) is considered plagiarism.

In accord with McGill University's Charter of Students' Rights, students have the right to submit in English or in French any written work that is to be graded.

Note: *In the event of extraordinary circumstances beyond the University's control, the content and/or evaluation scheme in this course is subject to change.*