

Operating Systems (Fall 2026) Course Outline

Contents

[Course Information](#)

[Course Calendar Description](#)

[Learning Outcomes](#)

[Topics](#)

[Grading](#)

[Communication, Lectures, and Assignments](#)

[Tutorials](#)

[Collaboration](#)

[Course Notes/Multimedia](#)

[Required Textbooks](#)

[Course Software](#)

[Undergraduate Academic Advisor](#)

[SCS Computer Laboratory](#)

[Extenuating Circumstances](#)

[Academic Accommodations and Regulations](#)

[Academic Accommodation](#)

[Generative AI Usage](#)

[Academic Integrity](#)

[Student Rights & Responsibilities](#)

Course Information

- **Course Number:** COMP 3000
- **Term:** Fall 2026
- **Title:** Operating Systems
- **Institution:** Carleton University, School of Computer Science

- **Instructor:** Anil Somayaji (<https://people.scs.carleton.ca/~soma>) (he/him, anilsomayaji at cunet.carleton.ca): by appointment in 5137 HP and online
- **Teaching Assistants:**
 - TAs will be announced in the first class
- **Lectures:**
 - A: Tue. & Thu. 14:35-15:55
 - B: Wed. & Fri. 8:35-9:55
- **Tutorials:**
 - A1: Wed. 13:05-14:25
 - A2: Wed. 19:35-20:55
 - A3: Mon. 16:05-17:25
 - B1: Mon. 14:35-15:55
 - B2: Tue. 18:05-19:25
 - B3: Thu. 16:05-17:25
- **Course Website:** https://homeostasis.scs.carleton.ca/wiki/index.php/Operating_Systems_%28Fall_2026%29

For information about Carleton's academic year, including registration and withdrawal dates, see [Carleton's The Academic Year \(https://calendar.carleton.ca/academicyear/\)](https://calendar.carleton.ca/academicyear/).

Course Calendar Description

Operating system implementation course stressing fundamental issues in design and how they relate to modern computer architectures. Assignments involve the modification and extension of a multitasking operating system.

Includes: Experiential Learning Activity

Precludes additional credit for SYSC 4001.

Prerequisites: COMP 2401 with a minimum grade of C- and COMP 2402.

Learning Outcomes

By the end of this course, students should:

- be able to write C code that uses low-level Linux services and should be able to implement simple Linux kernel extensions (modules),
- have a strong conceptual model of how an operating system works that allows them to determine the relative role of application and operating system code when debugging software, and

- understand the basic use and architecture of virtual machine-based cloud architectures.

Note that in order to achieve these objectives students should come into this course with a strong background in C programming and general application development.

Topics

Below is an approximate list of topics the course will cover. Because the concepts of operating systems are highly interconnected, many topics will be covered multiple times at increasing levels of detail. A more detailed list along with associated learning materials will be developed through the term on the [main course webpage](#).

- Processes
- Computer architecture
- Filesystems
- Concurrency
- CPU Scheduling
- Virtual Memory
- Inter-Process Communication
- Networking
- High performance I/O
- Virtual Machines
- Containers
- Security
- Cloud Infrastructure

Grading

The marking scheme for this course is:

- 5% for lecture participation.
- 20% for tutorial participation.
- 20% for the assignments, if exam average is not too low (see below).
- 20% for the Midterm Exam, Oct. 15th & 16th during class.
- 35% for the Final Exam (during the final exam period).

Note the following:

- The midterm and final exams are closed book: no notes, no collaboration, no online resources, no AI. Exams will be written in an exam book based on printed exams, except where accommodations require other forms.
- The lecture and tutorial participation are fundamentally graded on a credit/no credit

basis. In other words, if you show up and try you should be able to get full credit. Depending upon class performance the requirements for what is required to get full credit may change over the course of the semester; however, the expectation is that everyone who tries should be able to get 100% of these marks.

- If your assignment average is more than 25% higher than your weighted exam average (midterm & final), assignment grades are dropped and the weight assigned to the exams. For example, if you get 100% on assignments but only 70% for the exams (weighted average), the assignments would no longer count towards your final grade. (Missed exams count as a zero for grade calculation purposes.)
- Late assignments are only accepted until solutions are posted, unless extenuating circumstances apply and have been discussed with the course instructor. You have to get your tutorial marked by a TA within 8 days of the tutorial being posted, again unless extenuating circumstances apply.

I also calculate grades using alternative marking schemes at the end of the semester, assigning the highest grade for each student from any of the marking schemes. Thus your final grade may be higher than might be suggested by strict following of the above scheme.

Communication, Lectures, and Assignments

The course website page listed above is the canonical source of information on this course. Please refer to it for updates. When significant changes are made to this document it will be either announced in lecture and/or posted in the course discussion forum.

Online course announcements and discussions will be on Microsoft Teams (<https://teams.microsoft.com>). While you may discuss assignments there, do not post outright answers to them (unless the solutions have been posted). You may discuss tutorials freely, as they are graded on a participation basis. Note that solutions are never posted for tutorials.

I am best contacted via Teams direct message. I will also respond to email, preferably to my CUNET address.

Assignment and exam submissions will be through Brightspace (<https://brightspace.carleton.ca>). Grades will also be posted there.

All lectures will be conducted in person except when announced otherwise, in which case they will take place over Zoom. Note that the Wednesday and Friday lectures will be repeat performances of the Tuesday and Thursday lectures, except for the first week of class when they are reversed.

Lecture recordings will be posted to the course webpage, generally within 48 hours of lecture. One or both of the lectures may be posted, depending on the week.

Lecture participation is based on interactions during lecture, such as questions asked and participation in online polls. You do not need to participate in every lecture to get full marks for this; the grade is based on the quality and quantity of your interactions. Note that grades will be calculated so that lecture participation can only improve your grade.

Tutorials

Tutorials are graded based on participation and effort, not correctness. They are marked out of four points. The expectation is that if you make a reasonable attempt at all parts of the tutorial, you will get 4/4 points. Thus for most students tutorials are an easy way to improve your overall grade. However, if you don't do the tutorials, you can significantly harm your grade for the course.

To get credit for tutorials, you must demonstrate to your assigned TA that you have made a reasonable attempt at the questions. This demonstration will typically be through a brief conversation where you show your TA your work, either in person or virtually. Normally this conversation should happen during your assigned tutorial time but may also take place during TA office hours. If this conversation is not feasible, at your TA's discretion you may submit written answers.

The scheduled tutorial time is a designated time when you can work with TAs and other students together, in person, on that week's tutorial. TAs will generally also be available online during these times as well, so if you aren't on campus you can still get help and get credit. You can only get credit for a tutorial within 8 days of the tutorial being made available. (Tutorials are posted generally on Sundays, so you have until the end of the following Monday to get credit.) Partial attempts may be given partial credit; however, you may improve your tutorial mark as long as it is still being accepted. Exceptions will be made only if there are extenuating circumstances.

Plan to take a 2-5 hours each week to do the assigned tutorial. The Teams tutorial channel is a forum for getting help with tutorials, potentially from other students. While you may collaborate with others and use outside resources (including AI), when asked you should indicate who helped you and the sources you used. When submitting written answers, collaborators and sources should be clearly listed.

Your work should be your own. Please don't represent the work of others as your own, because if you do, you are subject to being reported to the Dean for plagiarism.

Collaboration

Collaboration on all work is allowed except for the midterm and final. Collaboration, however, should be clearly acknowledged. See below for further details.

For assignments, while you may get help from others and even collaboratively solve technical problems, the code and answers submitted should all be your own work. For example, you *may not* divide an assignment into parts, give a part to another student or anyone else to solve, and then submit that work as your own. You have to have participated in the creation of every part of your submitted work. An easy way to make sure this happens is never share files regarding coursework or copy and paste answers into email. Instead, meet together to work on an assignment and then separate to write up your solutions.

AIs should be treated the same as human collaborators, in that you may use them, but the answers you submit should be your own work, whether it is code or text. Note that artificially boosting your

assignment grades may not be helpful if there is too much of a discrepancy between them and your exam grades (see Grading above).

Similarity between submitted assignments and projects that has not been appropriately documented will be treated as plagiarism - the same as copying on a midterm or a final - and will be submitted to the Dean for disciplinary action.

All exams are expected to be completed individually, and without electronic aid; any communication or access of electronic devices (e.g. cellphones) during exams will be considered cheating. Use of Smart Glasses (or other similar assistive technologies) before or during a test is strictly prohibited and is considered an academic integrity violation. Even if you have prescription glasses with assistive technologies, you will not be allowed to use them during the test, so please make sure that you have acceptable prescription glasses. The proctor has the right to ask to inspect your glasses before or during the test. In addition, unless authorized with PMC accommodations, you are NOT allowed to use headphones nor ear buds during a test. Also, during a test, if you choose to wear a baseball cap, sunglasses or anything else that would prevent a proctor from monitoring where your eyes are focused during the test, you will likely be asked by a proctor to move to a different seat. Lastly, you must submit your exam before leaving the classroom or it won't be graded.

Course Notes/Multimedia

Video from lectures will be available via the lecture pages on the main course website generally within two days after lectures are delivered. These same pages will also contain code and notes given in class.

Do not rely upon the lectures and notes to cover all material related to this class. Mastery of the tutorial material is essential for doing well on the assignments and exams. Textbook and other outside readings should be used as supplements to help you understand concepts covered in lecture and tutorial.

Required Textbooks

The course will be using the textbook Operating System Concepts, 10th Edition by Silberschatz, Galvin, & Gagne (<https://www.wiley.com/en-us/operating-system-concepts-10th-edition-p-9781119320913>). You should plan to read the assigned sections BEFORE class, at least briefly, so you are familiar with the context of the lecture; then, read the sections in more detail after lecture.

You may also wish to consult the free online textbook Operating Systems: Three Easy Pieces (<http://pages.cs.wisc.edu/~remzi/OSTEP/>). I used this textbook in previous runs of this class.

This course focuses much more on reading code rather than writing code. Thus, John Aycock's book, Reading and Modifying Code (<http://pages.cpsc.ucalgary.ca/~aycock/reading-and-modifying-code.pdf>), is worth reading to better understand how reading code differs from writing code.

Course Software

In this course we will be working with the Ubuntu (<http://www.ubuntu.com/>) Linux distribution. You may use other Linux distributions in the tutorials to complete the assigned work; there will be differences, however, in some aspects (such as installing software), particularly if you use a distribution not based on Ubuntu or Debian.

Undergraduate Academic Advisor

The Undergraduate Advisors for the School of Computer Science are available in Room 5302HP; or by email at (scs.ug.advisor at cunet.carleton.ca). The undergraduate advisors can assist with information about prerequisites and preclusions, course substitutions/equivalencies, understanding your academic audit and the remaining requirements for graduation. The undergraduate advisors will also refer students to appropriate resources such as the Science Student Success Centre, Learning Support Services and Writing Tutorial Services.

SCS Computer Laboratory

Students taking a COMP course can access the SCS computer labs. The lab schedule and location can be found at: <https://carleton.ca/scs/tech-support/computer-laboratories/>. All SCS computer lab and technical support information can be found at: <https://carleton.ca/scs/tech-support/>. Technical support staff may be contacted in-person or virtually, see this page for details: <https://carleton.ca/scs/tech-support/contact-it-support/>.

Extenuating Circumstances

While I expect you to take this course seriously, I do not expect it to be the most important thing in your life. Physical and mental health issues, family responsibilities, natural and personal disasters, and other factors may come up over the semester. I want you to succeed in this course, and to that end I will do what I reasonably can to help you succeed. If your request can be accommodated with not unreasonable effort by me or the TAs, and the accommodation is one that is reasonable to give to anyone in your circumstances, I will do what I can.

Having said this, the main thing I have to give is extra time, but there is only so much time in a semester. Sometimes you'll need to get extra time by trying again in another semester, and that is okay. Having said that, I will always respect you enough to not pressure you to make a specific choice; you must decide what is best for your circumstances.

Remember that sometimes we all need help. The Carleton Wellness Website (<https://wellness.carleton.ca/>) is a good resource if you aren't sure what to do next.

Academic Accommodations and Regulations

Academic Accommodation

Carleton is committed to providing academic accessibility for all individuals. You may need special arrangements to meet your academic obligations during the term. The accommodation request

processes are outlined on the Academic Accommodations website (<https://students.carleton.ca/course-outline/>).

Generative AI Usage

As our understanding of the uses of AI and its relationship to student work and academic Integrity continue to evolve, students are required to discuss their use of AI in any circumstance not described here with the course instructor to ensure it supports the learning goals for the course.

Academic Integrity

Students are expected to uphold the values of academic integrity, which include fairness, honesty, trust, and responsibility. Examples of actions that compromise these values include but are not limited to plagiarism, accessing unauthorized sites for assignments or tests, unauthorized collaboration on assignments or exams, and using artificial intelligence tools such as ChatGPT when your assessment instructions say it is not permitted.

Misconduct in scholarly activity will not be tolerated and will result in consequences as outlined in Carleton University's Academic Integrity Policy (<https://carleton.ca/secretariat/wp-content/uploads/Academic-Integrity-Policy-2021.pdf>). A list of standard sanctions in the Faculty of Science can be found [here](https://science.carleton.ca/academic-integrity/) (<https://science.carleton.ca/academic-integrity/>).

Additional details about this process can be found on the [Faculty of Science Academic Integrity website](https://science.carleton.ca/academic-integrity/) (<https://science.carleton.ca/academic-integrity/>).

Students are expected to familiarize themselves with and abide by [Carleton University's Academic Integrity Policy](https://carleton.ca/secretariat/wp-content/uploads/Academic-Integrity-Policy-2021.pdf) (<https://carleton.ca/secretariat/wp-content/uploads/Academic-Integrity-Policy-2021.pdf>). Claiming ignorance of or confusion about the academic integrity standards as described in the Policy does not excuse a student from responsibility for violations of those standards.

Student Rights & Responsibilities

Students are expected to act responsibly and engage respectfully with other students and members of the Carleton and the broader community. See the [7 Rights and Responsibilities Policy](https://carleton.ca/studentaffairs/student-rights-and-responsibilities/#sect1.1) (<https://carleton.ca/studentaffairs/student-rights-and-responsibilities/#sect1.1>) for details regarding the expectations of non-academic behaviour of students. Those who participate with another student in the commission of an infraction of this Policy will also be held liable for their actions.

Retrieved from "[https://homeostasis.scs.carleton.ca/wiki/index.php?title=Operating_Systems_\(Fall_2026\)_Course_Outline&oldid=25084](https://homeostasis.scs.carleton.ca/wiki/index.php?title=Operating_Systems_(Fall_2026)_Course_Outline&oldid=25084)"

This page was last edited on 24 August 2026, at 22:01.

Content is available under GNU Free Documentation License 1.2 unless otherwise noted.