

CSC369H1 – Operating Systems

Course syllabus for the 2024 winter semester

Coordinator Mario Badr
Office BA4270
Email address csc369-2024-01@cs.toronto.edu
Website <https://q.utoronto.ca/courses/337408>

Contents

1	Course details	2
1.1	Communication	2
1.2	Lectures	3
1.3	Tutorials	3
1.4	Textbook	3
1.5	Assessment	4
1.5.1	Exercises	4
1.5.2	Assignments	5
1.5.3	Midterm Examination	5
1.5.4	Final Examination	5
2	Course policies	5
2.1	Academic integrity	6
2.2	Grading Errors	6
2.3	Accommodations	6
2.3.1	Grace credits	7
2.3.2	Tests and exams	7
2.3.3	Disabilities	7
2.3.4	Emergencies	8
2.3.5	Religious observances	8
2.4	Copyright	8
2.4.1	Lecture recordings	9
2.4.2	Your course work	9

1 Course details

Welcome to Operating Systems (CSC369H1)! The course introduces the major principles of operating systems, with a focus on systems programming in C. For a high-level overview of the topics we cover, see Table 1. In many ways, the material is “core CS”, and past students have indicated that the assignments and concepts have helped them in interviews.

Table 1: A tentative schedule of topics covered during lectures.

<i>Week</i>	<i>Monday</i>	<i>Wednesday</i>
W01	Introduction	Processes
W02	System calls	Scheduling
W03	Multi-level feedback	Lottery scheduling
W04	Managing memory	Address translation
W05	Paging	Page table design
W06	Replacement policies	Complete VM systems
W07	Threads	POSIX threads
W08	Locks	Condition variables
W09	Semaphores	Concurrency bugs
W10	File systems	The Very Simple File System
W11	File system implementations	File system reliability
W12	Solid state drives	Conclusion

The course offers you with opportunities to learn synchronously and in-person through lectures and tutorials. All lectures and tutorials start at 10 minutes past the hour of the time specified on ACORN. The location of lectures is on ACORN, but the location of your tutorials will be announced on Quercus. The rest of this section gives more information about lectures, tutorials, and assessment. To learn more about our course policies, see Section 2.

1.1 Communication

All course announcements are posted on Quercus. You are responsible for reading all announcements made by the teaching team in a timely manner. We highly recommend enabling notifications for Quercus announcements.

If you have a personal question (i.e., the answer is only useful to you), please use csc369-2024-01@cs.toronto.edu and ensure that you share your UTORid in the body of your message. We recommend reviewing our course policies (Section 2) before sending an email.

If you have a question related to course content, please use Piazza. As a courtesy to others (and the teaching team), please search to see if your question has already been posted. This is especially true closer to deadlines, where you may find many earlier Piazza answers helpful to you.

Throughout the term, the instructor and teaching assistants host in-person office hours. The time and location of office hours will vary from week to week. Details will be posted on Quercus.

1.2 Lectures

Lectures provide the conceptual underpinnings (and, sometimes, the historical context) for the programming exercises and assignments. You should prepare for, and reinforce concepts from, lectures by reading the textbook (Section 1.4). Your attendance in lectures is recommended, but not mandatory. The first lecture is on Monday, January 8th, 2024.

If the lecture hall supports it, lecture recordings will be made available on Quercus. This is not a guarantee, and you should not rely on recordings in case they are not available (e.g., due to technical issues). You may access these recordings through the “OCCS Student App”; you can find a link to this app on the course website. Please see Section 2.4.1 for our policy on lecture recordings.

1.3 Tutorials

Tutorials reinforce material from lectures. Your attendance in tutorials is recommended, but not mandatory. Because tutorials are active in nature (i.e., you are working on a problem), recordings may not be available (and would likely not be very useful). The first tutorial is on Friday, January 12th, 2024 at the place specified on ACORN. The location of subsequent tutorials will be posted on Quercus. Please note that there is no tutorial on Friday, March 29th, 2024 (the university is closed for Good Friday).

1.4 Textbook

A lot of your learning will happen outside of lectures and tutorials. The course textbook, which can be accessed online for free, will serve as an excellent resource:

Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. *Operating Systems: Three Easy Pieces*. Arpaci-Dusseau Books. November, 2023 (Version 1.10)

You may find that following a traditional textbook is useful (but this is not required). For a solid treatment of the theoretical concepts and historical context, we recommend:

Andrew, S. Tanenbaum, and Bos Herbert. Modern operating systems (fourth edition). Pearson Education, 2015.

If you would prefer a greater emphasis on implementation details, we recommend:

Stallings, William. Operating systems internals and design principles (ninth edition). Pearson Education, 2017.

1.5 Assessment

You are assessed both on your understanding of operating systems concepts and their implementation. Assessments include: regular programming exercises, assignments, a midterm, and a final exam. The marking scheme is summarised in Table 2.

Table 2: The overall marking scheme.

<i>Item</i>	<i>Count</i>	<i>Total Weight</i>
Exercises	10	10%
Assignments	3	30%
Midterm Examination	1	15%
Final Examination	1	45%

For your exercises and assignments to be graded, they must meet the minimum standards of a professional computer scientist. This means that your submissions *must* compile cleanly, without errors or warnings, on the teach.cs lab machines. All submissions are done through MarkUs, and *submissions that are missing files or do not compile may receive a grade of 0*.

1.5.1 Exercises

There are a total of 10 programming exercises that you must complete individually. Programming exercises are small relative to assignments, but cover key concepts in operating systems, C programming, or both. These exercises are designed to help complement your understanding of the concepts and C programming skills you need in larger assignments.

A programming exercise is typically released on a Friday at 10:00 AM and is due the following Friday at 10:00 AM. See Section 2.3.1 for our policy on late submissions. There is no programming exercise released in Week 6 or Week 12. You access, and submit, all programming exercises using the MarkUs platform.

1.5.2 Assignments

There are a total of 3 assignments that you must complete individually. Assignments may involve the programming of some subsystem of an operating system in a sandboxed environment. Or you may be asked to solve a problem (or set of problems) related to an operating system concept.

An assignment is typically released on a Tuesday at 4:00 PM and due on a Tuesday at 4:00 PM. See Section 2.3.1 for our policy on late submissions. You have three “work” weeks (i.e., not including reading week) to complete an assignment. We *highly recommend* you begin assignments early. You access, and submit, all assignments using the MarkUs platform.

1.5.3 Midterm Examination

A midterm examination will be held on Friday, March 1st during tutorial time. The midterm will assess you on material covered between Weeks 1 and 6, inclusive. You must attend the midterm examination for the tutorial section you are enrolled in. The location of the midterm, and additional instructions, will be announced on Quercus closer to the date.

1.5.4 Final Examination

A three-hour final examination will be scheduled by the Faculty of Arts and Science during the final assessment period (i.e., between April 10th and April 30th). The final examination is worth 45% of your overall grade. In addition, *you must get at least 30% on the final exam to pass the course*. That is, if your final exam mark is less than 30%, then your final mark in the course will be reduced (if necessary) to no more than 45%.

2 Course policies

The University of Toronto is committed to equity, human rights and respect for diversity. All members of the learning environment in this course should strive to create an atmosphere of mutual respect where all members of our community can express themselves, engage with each other, and respect one another’s differences. The University of Toronto does not condone discrimination or harassment against any persons or communities.

In the rest of this section, we detail our course policies related to academic integrity, accommodations, and intellectual property.

2.1 Academic integrity

All suspected cases of academic dishonesty will be investigated following procedures outlined in the Code of Behaviour on Academic Matters. If you have questions or concerns about what constitutes appropriate academic behaviour or appropriate research and citation methods, please reach out. Note that you are expected to seek out additional information on academic integrity from me or from other institutional resources (for example, the [University of Toronto website on Academic Integrity](#)).

All code you submit must be written by yourself. Submitting code you find elsewhere, either via the internet or generated through AI (e.g., ChatGPT, Github Co-pilot) is strictly forbidden. Here are a few guidelines to help you maintain academic integrity:

- Never look at another student's or group's assignment solution or idea for a solution, whether it is on paper or on the computer screen, and don't allow your solution to be viewed by or come into the possession of another student. Maintain absolute control of your work, including notes and partial solutions, at all times.
- We encourage you to discuss course concepts and to study for exams with other students, but any work that is submitted should be your own. The easiest way to avoid plagiarism is to only show work to a TA or instructor.
- An academic offence may significantly slow your progress through your degree. It is better to submit a partially completed assignment and receive a low mark than to face an academic offence on your record.
- While it might be tempting to look for ideas and solutions in public repositories like GitHub, remember that using someone else's work and ideas without attribution, even if making some changes, is considered plagiarism. Keep in mind that plagiarism detection software may detect such cases.

2.2 Grading Errors

If you believe there is a mistake in your grade, you can email us for clarification (csc369-2024-01@cs.toronto.edu) within two weeks of the grade being released to you. Your email should clearly and concisely describe why you believe your assessment was incorrectly graded. Please note that your inquiry may increase the original grade, leave it as is, or decrease the original grade, depending on the grading error or errors found.

2.3 Accommodations

In this section, we describe accommodations that may be available to you and situations where you may request special consideration from us or the university. We also recommend you review the Faculty of Arts and Science policy on [student absences](#), especially

if you are experiencing an emergency that impacts multiple courses. In these cases, you should also contact your [College Registrar](#) for academic and/or personal advising.

2.3.1 Grace credits

We recognize that unexpected problems sometimes make it difficult to submit a programming exercise or assignment on time. For this reason, we use grace credits to give you flexibility with these deadlines. Each student will receive 24 grace credits; each grace credit can be used for a four-hour extension.

You can only extend a programming exercise deadline up to a maximum of 12 hours (i.e., 3 grace credits). You can only extend an assignment deadline up to a maximum of 48 hours (i.e., 12 grace credits). Grace credits serve both the role of allowing you some flexibility in how you plan your time, as well as an automatic cushion for when small things just go wrong. But once your credits have been exhausted, late submissions *will not be accepted*. If you encounter an emergency (e.g., medical) during the term, please see Section [2.3.4](#).

MarkUs automatically deducts grace credits when you submit work for an assignment late; you do not need to explicitly say you are using a grace credit, just submit your work within the grace credit four-hour periods. These periods round up: for example, *if you submit an assignment just two minutes (or even one second) after the actual assignment due date, this still uses up a grace credit*.

2.3.2 Tests and exams

If you require accommodations to write the midterm, these accommodations are handled by [Accommodated Testing Services](#). These types of accommodations include, but are not limited to, additional writing time and/or alternate print formats. If, however, you miss (or will miss) the midterm, please see Section [2.3.4](#). *Please note that accommodations for final exams are handled by your registrar.*

2.3.3 Disabilities

The University provides academic accommodations for students with disabilities in accordance with the terms of the Ontario Human Rights Code. This occurs through a collaborative process that acknowledges a collective obligation to develop an accessible learning environment that both meets the needs of students and preserves the essential academic requirements of the University's courses and programs. Students with diverse learning styles and needs are welcome in this course.

If you have a disability that may require accommodations, please contact the [Accessibility Services on the St. George campus](#) office as soon as possible. If you are registered

with accessibility services, please let us know by emailing us your Letter of Academic Accommodation (csc369-2024-01@cs.toronto.edu). Alternatively, you may ask the accessibility office to send the letter. When you are in need of special consideration, please follow the process outlined in Section 2.3.4.

2.3.4 Emergencies

Students experiencing illness or other emergencies that prevent them from being able to complete homework on time, attend a lab, or write the midterm, can apply for special consideration. You will be required to affirm that you are abiding by the Code of Behaviour on Academic Matters, in particular, to be aware that it is an offence:

to engage in any form of cheating, academic dishonesty or misconduct, fraud or misrepresentation not herein otherwise described, in order to obtain academic credit or other academic advantage of any kind

That is, you must be truly experiencing an emergency, and acknowledge that to falsely claim so is an academic offence. Applying does not guarantee that you will be granted special consideration. To apply for special consideration, please see the *Syllabus* page on the Quercus course website.

Submit your request soon as possible if you find yourself in such a situation. It is easier to resolve situations earlier rather than later. If your emergency will affect your ability to complete coursework for more than a few days, or in multiple courses, we recommend you also talk to your registrar.

2.3.5 Religious observances

The University provides reasonable accommodation of the needs of students who observe religious holy days other than those already accommodated by ordinary scheduling and statutory holidays. Students have a responsibility to alert members of the teaching staff in a timely fashion to upcoming religious observances and anticipated absences. *Please reach out to us at least two weeks before the due date* to communicate any anticipated absences related to religious observances and to discuss any possible related implications for course work.

2.4 Copyright

Course materials prepared by the instructor are considered by the University to be an instructor's intellectual property covered by the Copyright Act, RSC 1985, c C-42. These materials are made available to you for your personal, and cannot be shared outside of the class or published (made publicly available) in any way. Posting course materials or any recordings you may make to other websites without the express permission of

the instructor will constitute copyright infringement. This notice applies to all course materials, including (but not limited to): course notes, lecture slides, lecture recordings, lecture and tutorial handouts, sample solutions, and assessment handouts, starter files, and solutions.

2.4.1 Lecture recordings

This course, including your participation, may be recorded on video and made available to students in the course for viewing remotely after each session. Course videos and materials belong to your instructor, the University, and/or other sources depending on the specific facts of each situation and are protected by copyright. In this course, you are permitted to download session videos and materials for your own academic use, but you should not copy, share, or use them for any other purpose without the explicit permission of the instructor. For questions about the recording and use of videos in which you appear, please contact us.

2.4.2 Your course work

Work that you complete in this course may not be shared with other students or published. This policy is to both protect the intellectual property of course staff and to protect you from committing acts of academic dishonesty. For more information on this topic, see the [Department of Computer Science website](#).

GitHub is a popular option for computer science students and professionals to both collaborate in teams and publish their work online, including to develop a portfolio for potential employers. As we said in the Academic Integrity section, you should not put your work publicly on GitHub. However, you may use GitHub's private repositories to store your own work (See GitHub's instructions for creating a repository and select "Private" in Step 4).