

Course Syllabus

 Edit

Welcome to CSC469H: Design and Implementation of Operating Systems.

This course builds on the concepts introduced in a standard first course on operating systems (such as CSC369H) to provide students with a deeper understanding of the internal workings of operating systems, and the impact of system-level implementation choices on user-level applications. These insights are important both for students embarking on a research program in computer systems, and for computing professionals who will work with the development and deployment of computer systems. Topics include operating system design and internal structure, benchmarking and performance evaluation, alternatives for inter-process communication, advanced synchronization strategies including non-blocking synchronization, virtual memory solutions for large address spaces and multiprocessors, multiprocessor scheduling, distributed systems, reliable storage, and security.

Course Information

Instructor: Kuei (Jack) Sun

Personal Office: BA 4231

Office Hour: Mondays and Wednesdays 1:30pm to 3pm, **BA 4290**, starting from week 2.

Virtual or physical meetings may be scheduled outside of regular office hours by request.

Course Email: csc469-2026-01@cs.toronto.edu (<mailto:csc469-2026-01@cs.toronto.edu>)

Please use email for personal issues and Piazza for all other course-related questions. We will try to respond to email by the end of the next day. However, due to volume, it may take longer, especially on weekends. (We are often not able to answer email more than once on the weekend.)

Class Meetings

All lectures will be held in-person at the locations indicated.

Afternoon Section LEC0101		Afternoon Section LEC0201	
Monday 12-1pm	BA 1190 https://map.utoronto.ca/?id=1809#m/982031?share	Monday 3-4pm	GB 244 https://map.utoronto.ca/?id=1809#m/494473?share
Wednesday 12-1pm	BA 1180 https://map.utoronto.ca/?id=1809#m/982031?share	Wednesday 3-4pm	MP 137 https://map.utoronto.ca/?id=1809#m/494490?share
Friday 12-1pm	MP 137 https://map.utoronto.ca/?id=1809#m/494490?share	Friday 3-4pm	BA 1190 https://map.utoronto.ca/?id=1809#m/982031?share

Civil Classroom Behaviour

We will switch between presentations of core course material, discussions of assignments, and demonstrations of tools, with occasional breakouts into small groups to work on exercises or evaluate design alternatives. During group discussions, please follow these expectations:

- Behave as you would in a professional software or engineering environment.
- Communicate with respect. Critique ideas, not people. No interruptions or personal attacks.
- Participate actively without dominating. Encourage others and rotate roles when possible.
- Stay on topic. No disruptive side conversations during lectures or group work.
- Collaborate equitably. Help peers learn instead of finishing alone.

Course staff may intervene to maintain a respectful environment. Repeated disruption may lead to removal from the class and/or group activities.

Copyright Notice

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 videos and materials for your academic use, but you should not copy, share, or use them for any other purpose without the explicit permission of the instructor.


TA Office Hours:

TAs are available to help students with assignments and other deliverables during TA office hours. See Office Hour Schedule (TBA) for more information.

Readings and Textbook(s)

There is no required textbook for this course, however, readings will be assigned from the research literature and the open source community. These readings are a key part of the course -- make sure you keep up with them! All assigned readings will be posted on this Quercus course website. Background on concepts can be found in any standard operating systems text.

Recommended texts include:

- Jerome H. Saltzer and M. Frans Kaashoek, *Principles of Computer System Design*. Morgan Kaufmann (2009)
- Andrew Tanenbaum, *Modern Operating Systems*. Prentice Hall (3rd ed. 2007 or 4th ed. 2016)
- Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau, [Operating Systems: Three Easy Pieces](http://pages.cs.wisc.edu/~remzi/OSTEP/)  (<http://pages.cs.wisc.edu/~remzi/OSTEP/>), (online)
- Marshall Kirk McKusick, George V. Neville-Neil and Robert N. M. Watson, *The Design and Implementation of the FreeBSD Operating System* (2nd ed. 2015)
- K.N. King, *C Programming: A Modern Approach*. Norton and Co (2nd ed. 2008).

Communications

The course website will be maintained on Quercus and is required reading. It contains official announcements, lecture notes, assignment handouts, tutorial materials, and course policies, along with a link to the Piazza discussion board. Piazza is the best place to ask questions and receive fast responses, especially when you participate actively.

Please follow the guidelines below for appropriate use of the discussion board:

- **Check existing posts first.** Use search to avoid duplicate topics before creating a new post or emailing the course team.
- **Do not share any part of your assignment solution on Piazza.** Doing so can lead to plagiarism allegations and may result in an Academic Offence.
- **Keep posts course-relevant and non-personal.** Avoid off-topic or personal content. If your concern is personal (e.g., illness-related absence), email the course team directly instead.
- **Use clear, unique email subjects when emailing.** If you need to send an email, include your student number in the subject (example: Midterm Remark Request - 1000123456) to prevent email clients from collapsing different student requests into one thread.

For all emails, include your full name, student number, tutor, and lecture section in the message body.

Marking Scheme

Overview

ITEM	EACH WORTH	WEIGHT (SCHEME 1)	WEIGHT (SCHEME 2)	WEIGHT (SCHEME 3)
Individual Exercises (3)	3.3 %	10 %	10 %	10 %
Group Assignments (3)	10 %	30 %	30 %	30 %
Midterms (2)	9 or 12 %	24 %	21 %	18 %
Final Exam	--	35 %	38 %	41 %
Surveys (4)	0.25 %	1 %	1 %	1 %

You do not need to choose a grading scheme. Your final grade will automatically be calculated under all three schemes, and you will receive the highest score.

The purpose of multiple schemes is to give you a safety net. A weak performance on one midterm (or both) will not count heavily against you and will not prevent you from doing well in the course overall.

Detailed Description

Participation Surveys (1%):

This course includes participation in four short research surveys aimed at evaluating the course content and delivery. However, you have the option to opt out of having your responses included in the research study. If you choose to opt out, your data will not be used for research purposes, but you will still receive the participation points for completing the survey. Your feedback is invaluable in shaping tools that support student learning, and all responses will remain confidential. Opting out will not affect your grade or standing in the course. If you wish to opt out, please send the request to the course email.

Lecture Exercises (10%):

There will be three relatively short exercises to be completed individually. These exercises are intended to ensure each student demonstrates proficiency with some of the key tools or skills of the OS developer.

Group Assignments (30%):

Working in teams, students will complete three substantial assignments covering core course concepts. Teams may include up to 3 students, and teams with fewer than 3 students must complete the same amount of work as all other teams and will not receive special consideration.

All group assignments must be submitted by committing your work to your group MarkUs repository.

As a responsible team member, you must help ensure that your group's submission complies with the academic integrity policies of this course. If you suspect plagiarism, unauthorized aid, or inequitable work distribution, you must report it to the course team by email immediately. Work that is submitted for grading and violates course policy may result in all group members being held accountable for academic offence.

Midterm Tests (18-24%):

There will be two 90-minute tests, on Wednesday, **February 4** and Wednesday, **March 11**. Each test covers material up to the week before the test date, but is not cumulative. The first midterm will cover Weeks 1–4, and the second midterm will cover Weeks 5-9. Content may include lectures, exercises, activities, and assignments due before the test date. You must write the midterm in your assigned lecture section unless approved to write in another section. To request to write in another section (including the makeup section) due to scheduling conflict, please use the [Midterm Scheduling Change Request Form](https://forms.office.com/r/1k4ZYpbs1U) [☞](https://forms.office.com/r/1k4ZYpbs1U). Requests must be made **one week** before the scheduled midterm date to be considered.

	Midterm 1 - February 4th	Midterm 2 - March 11th
LEC0101	Wednesday 12pm - 2pm, EX 310	Wednesday 12pm - 2pm, EX 310
LEC0201	Wednesday 2pm - 4pm, EX 310	Wednesday 2pm - 4pm, EX 310
MAKEUP	Wednesday 6pm - 8pm, Room TBA	Wednesday 6pm - 8pm, Room TBA
Schedule Change Request Deadline	January 28th, 11:59pm	March 4th, 11:59pm

The midterm will be closed book. You may only bring a non-programmable calculator and a self-made, one-sided, letter-sized physical reference sheet (minimum font size of 10pt, printed or hand written). If you cannot write a midterm test due to extraordinary circumstances beyond your control, please submit the [Special Consideration Form for Missed Midterm](https://forms.office.com/r/SJAjCdrd6y) [☞](https://forms.office.com/r/SJAjCdrd6y) with the supporting documentation to request a special consideration as soon as possible. Special consideration requests will be evaluated on a case-by-case basis.

Each midterm is worth 9-12% of your final grade.

Final Exam (35-41%):

The final exam will be scheduled by the Faculty of Arts & Sciences sometime in March. It is comprehensive, covering all course material, including topics tested on the midterms and material from the assignments. The exam will be closed book. You may bring only a non-programmable calculator and one self-prepared, two-sided reference sheet (minimum font size 10pt).

A minimum grade of 30% on the final exam is required to pass the course (post-curve).

The weight of the final exam will depend on the grading scheme that produces the highest overall grade for you at the end of the term.

Policies

Minimum Standards for Submitted Work:

For your assignment to be graded, it must meet the minimum standards of a professional computer scientist. **All files** required to build the program must be submitted, and the program **must** compile cleanly, without errors or warnings on the teaching labs machines. Written reports must (i) use professional language, (ii) be neatly typeset using legible fonts and graphics, (iii) be spellchecked, and (iv) properly cite all referenced works. Last-minute difficulties with git and/or MarkUs can easily be avoided by ensuring all files are added to the repository well before the deadline, and that you know how to commit and push them. Compiling and testing your work on the teaching lab machines at intermediate stages will avoid last-minute problems as well. **Submissions that are missing files or do not compile will receive a grade of 0.**

Late Work:

All assignments (individual exercises and group assignments) must be submitted electronically by **11:59:59 p.m. sharp** on the due date. Each student is allotted **seven 1-day grace tokens** for the semester, which can be distributed across assignments however you choose. For example, you could use all seven tokens on a single assignment to gain a one-week extension or spread them across multiple assignments. Submitting an assignment even one second late will automatically use one grace token. Once your tokens are exhausted, late submissions will not be accepted unless under exceptional circumstances. **If you find yourself in a serious medical or emergency situation where use of grace tokens is not sufficient, you will have the opportunity to submit the [Assignment Special Consideration Request Form](https://forms.office.com/r/aHTV5nWwbs) to explain your circumstances and your specific special consideration request. Special consideration requests will be evaluated on a case-by-case basis.** Please be warned that once an extension is granted for an assignment, you **may not use grace tokens on top of the extension**, unless you are registered with Accessibility Services.

For group assignments, a grace token will be deducted from all team members for each extra 1-day period. As a result, you can use at most the minimum number of grace tokens between the team members.

The grace tokens are provided to help you deal with minor, unforeseen disruptions to your work schedule. However, instructional staff may not provide office hours or answer questions about the assignments after the deadline has passed. Make sure you start early and have a good understanding of the assignment requirements, even if you expect to use the grace period to complete your work.

Please note that a submission made at 12:00:00 a.m. after the due date will be considered late. You should ensure that your work is not submitted at the very last second. Since you will be using version control, it is very easy to commit regularly to avoid running into the deadline.

Religious Holidays:

If a religious holiday will keep you from completing any assigned work, please let us know as soon as possible (but no later than two weeks before the due date), and we will work out a mutually agreeable accommodation.

Emergencies:

In the event of an illness or other catastrophe that affects your ability to do your academic work, consult the course instructors right away. Normally, you will be asked for documentation in support of your specific circumstances. This documentation may take the following forms:

- Absence declaration via [ACORN \(https://www.acorn.utoronto.ca/\)](https://www.acorn.utoronto.ca/)
 - The University has updated its policies on the use of the Absence Declaration, which can now only be used in case of (a) a health condition or personal injury, (b) a personal or family emergency, or (c) bereavement. Students may submit **one absence declaration per academic term**, to declare an absence for a **maximum period of seven consecutive calendar days**. The seven-day declaration period can be retroactive for up to six days in the past, or up to six days in the future, but it must cover the period in which the missed academic obligation occurred.
- [U of T Verification of Illness or Injury Form \(VOI\) \(http://www.illnessverification.utoronto.ca/index.php\)](http://www.illnessverification.utoronto.ca/index.php)
 - The VOI indicates the impact and severity of the illness, while protecting your privacy about the details of the nature of the illness. If you cannot submit a VOI due to limits on terms of use, you can submit a different form (like a letter from a doctor), as long as it is an original document, and it contains the same information as the VOI (including dates, academic impact, practitioner's signature, phone and registration number). To download a copy of the VOI, please see <http://www.illnessverification.utoronto.ca> (<http://www.illnessverification.utoronto.ca/index.php>).
- College Registrar's letter
- Letter of Academic Accommodation from Accessibility Services

For more information on documentation of absences for Arts and Science students, including limitations on the use of the Absence Declaration tool, refer to the [A&S Student Absences \(http://www.artsci.utoronto.ca/absence\)](http://www.artsci.utoronto.ca/absence) page. It is always easier to make alternate arrangements before a due date, so please inform us as soon as you know that you will need accommodation. If you get a concussion, break your hand, or suffer some other acute injury, you should register with Accessibility Services as soon as possible.

Remark Requests:

If you believe there was an error in the grading of your assignment, you may submit a remark request through MarkUs. Your request must clearly state one of the following:

- Marking error:** Clearly and concisely describe the specific error you believe occurred in the grading.
- Minor submission issue:** Identify a minor problem in your submission that can be resolved with no more than one line of code, e.g., so that your code would compile again. If your request is accepted under this category, a 20% penalty will be applied.

Remark requests must be submitted within **two weeks** of the marks being released. Please note that remarking may result in an increased grade, no change, or a *decreased grade*.

Academic Integrity:

All of the work you submit must be your own. Submitting work written by someone else, or sharing your own work for others to submit, is considered plagiarism and is treated as academic fraud. The department uses plagiarism-detection software that compares code across current and past submissions, as well as code available online. Please review the University of Toronto [Academic Integrity website \(https://www.academicintegrity.utoronto.ca/\)](https://www.academicintegrity.utoronto.ca/), and read the [Code of Behaviour on Academic Matters \(https://governingcouncil.utoronto.ca/secretariat/policies/code-behaviour-academic-matters-july-1-2019\)](https://governingcouncil.utoronto.ca/secretariat/policies/code-behaviour-academic-matters-july-1-2019).

To help you avoid academic offences, keep the following in mind:

- Do not share or view solutions.** Never look at another student's or group's assignment solution, in whole or in part, and never allow your work to be shared or copied. Maintain strict control of your notes, drafts, and code at all times.
- Discussion is fine, copying is not.** You are encouraged to discuss concepts and study collaboratively, but all submitted work must be written entirely by you. If you want feedback on your code or solutions, only share it with a TA or instructor.
- When in doubt, submit your own work.** It is always better to hand in a partial or incomplete solution and receive a lower grade than to commit an academic offence. An offence can have severe consequences and may delay your progress through your degree.
- Do not look for or reuse solutions online.** Public sources (e.g., GitHub repositories, forums, or blogs) may contain code similar to our assignments. Submitting such code, even with modifications, is plagiarism. Our detection tools can identify these cases.
 - Note: Some assignments in this course may encourage you to look online for resources that help you explain experimental results or discuss the rationale behind an OS design decision. In these situations, you must summarize the information you find in your own words and you must properly cite the sources that you use.
- Do not use AI-generated code.** The entire code and/or report you submit must be written by you. Submitting any work generated by tools such as ChatGPT, GitHub Copilot, or any other AI system is strictly prohibited and will be treated as an academic offence, with penalties applied to the fullest extent of University regulations.

Protect Yourself

Students sometimes receive an Academic Offence (AO) not because they copied, but because they did not protect their own work. To avoid this:

- Do not share accounts.** Never share accounts (e.g., Google Drive, GitHub, ChatGPT). We have caught cases where one student was charged with an AO because a friend copied their work from a shared chat log.
- Secure your devices.** Password-protect your computer and phone. Always log out of public or shared computers to prevent others from accessing your files.
- Keep your repository private.** Do not publish your assignment solutions online. The assignment handout and starter code are the intellectual property of the course instructors. Making completed assignments publicly accessible, even in modified form, is considered copyright infringement and a breach of academic integrity.
- Know everything your group submits.** Discuss the assignment with your team not only to understand the content, but also to ensure all work is original. If a teammate cannot explain their own code or writing, this may indicate unauthorized or copied work. All group members share responsibility for the final submission, and plagiarism cases may result in an academic offence for the entire team. Stay actively involved and review all components before the submission deadlines.

Use of Generative AI

You may use GenAI tools to support learning and build understanding, including:

- Asking for explanations of concepts
- Asking AI to identify issues in your code or reasoning, e.g., "Why might deadlock happen in this scenario?" or "Point out logic bugs here."

Using GenAI in the ways described above is strongly encouraged in this course. It can act as a study partner that is always available to discuss ideas and help you test your understanding. However, be careful not to cross the line into using AI-generated solutions in your submission, which counts as unauthorized aid in this course.

- Allowed:** Asking AI to analyze your code and describe possible problems, e.g., "What invariants could my filesystem be violating?" or "Why is my cache design inefficient?"
- Not allowed:** Copying AI-generated code or applying its suggested fix verbatim into your submission, even if lightly edited, e.g., asking "Fix this for me" and copy-pasting the corrected code into your submission.

All work you submit must be written, typed, and validated by you (or your group members). The following are prohibited:

- Submitting any code or text produced by GenAI, **this includes comments**.
- Copying AI output into your assignment, even in modified or rewritten form.
- Using AI to complete assignment tasks on your behalf.

The rule of thumb is to use AI to understand the problem, not to produce the solution you submit. Treat GenAI the way you would a peer, and ask yourself: "Is this friend giving me an implementation to copy, or explaining the root cause of the issue?"

Think into the future: if a potential employer or supervisor can get the same result from an AI tool as from you, why should they hire you? Our goal is to help you reach a level of understanding that goes far beyond what automated tools can provide. The assignments in this course are carefully designed to guide you toward that mastery.

If you have questions or get stuck, please come to office hours! we'll be glad to help.

Accessibility Needs:

The University of Toronto is committed to accessibility. If you require accommodations for a disability or have any accessibility concerns about the course, the classroom, or course materials, please contact [Accessibility Services \(https://studentlife.utoronto.ca/department/accessibility-services\)](https://studentlife.utoronto.ca/department/accessibility-services), as soon as possible via email ([accessibility.services@utoronto.ca \(mailto:accessibility.services@utoronto.ca\)](mailto:accessibility.services@utoronto.ca)), or phone ([416-978-8060 \(tel:416-978-8060\)](tel:416-978-8060)).