

Course Syllabus



Welcome to CSC369H: Operating Systems! This course explores the core principles of operating systems, with a focus on systems programming in C.

By the end of the course, you will gain:

- Hands-on experience working with systems-level C code.
- Theoretical and practical knowledge of:
 - **Processes and Threads**: address spaces, system calls, scheduling
 - **Synchronization**: algorithms and primitives such as locks, condition variables, and semaphores
 - **Virtual Memory**: paging, page tables, eviction, segmentation
 - **File Systems**: file abstraction, directory structures, disk I/O

Operating systems are *core computer science*. Students consistently report that the assignments provide excellent material for technical interviews, and that OS concepts appear frequently in interview questions.

The course is structured around a sequence of programming assignments on threads, synchronization, virtual memory, and file systems. Lectures provide both theoretical and historical context, but are not purely traditional. Class time will also include small-group activities and interactive discussion. To benefit from this format, it is essential that you complete all assigned readings before each lecture. Coming prepared allows you to engage in higher-order learning, rather than using lecture time for first exposure to the material.

To succeed, you must keep up with course material and also start assignments early. Students who begin promptly consistently achieve higher grades and experience less stress. **No assignment can be completed successfully if you begin fewer than 3 to 4 days before the deadline.**

Course Information

Instructor:

Name: Kuel (Jack) Sun	Name: Robin Li
Office: BA 4231 (not for office hour)	Office: PT 372 (not for office hour)
Office Hours: Thursdays 2:30-3:30pm, BA4242	Office Hours: Wednesdays 4:30-5:30pm, BA4242

Meetings may be scheduled outside of regular office hours by request. Instructor office hours run from week 2 to the final week of the term.

Communications:

Course Email: csc369-2025-09@cs.toronto.edu (<mailto:csc369-2025-09@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.)

Lectures:

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

Afternoon Section LEC0101		Afternoon Section LEC0201		Evening Section LEC5101	
Monday 1-2pm	BA 1190 (https://map.utoronto.ca/?id=1809#m/9820317sshare)	Monday 3-4pm	KP 108 (https://map.utoronto.ca/?id=1809#m/4944887sshare)	Wednesday 6-9pm	BA 1180 (https://map.utoronto.ca/?id=1809#m/9820317sshare)
Wednesday 1-2pm	FE 230 (https://map.utoronto.ca/?id=1809#m/9832357sshare)	Wednesday 3-4pm	MP 103 (https://map.utoronto.ca/?id=1809#m/4944907sshare)		
Friday 1-2pm	BA 1190 (https://map.utoronto.ca/?id=1809#m/9820317sshare)	Friday 3-4pm	BA 1160 (https://map.utoronto.ca/?id=1809#m/9820317sshare)		

Format and Preparation for lectures:

You will regularly be assigned readings and/or short videos. They should be reviewed before the lectures so that we can spend more time in class working through some of the details. You will also be working on exercises. Lectures will be recorded and one of the two recordings will be posted as soon as possible, for asynchronous viewing. However, students are expected to attend the lectures in person and complete the exercises during, or shortly after, the lectures.

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

For questions about recording and use of videos in which you appear please contact your instructors.

Note that there are no lab hours in this offering of the course.

TA Office Hours:

TAs are available to help students with assignments and other deliverables during TA office hours. See [Office Hour Schedule \(https://q.utoronto.ca/courses/395880/pages/office-hour-schedule\)](https://q.utoronto.ca/courses/395880/pages/office-hour-schedule) for more information.

Textbook(s):

We will be using a free text quite extensively [Operating Systems: Three Easy Pieces \(http://pages.cs.wisc.edu/~remzi/OSTEP\)](http://pages.cs.wisc.edu/~remzi/OSTEP) by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau.

You may also find the following traditional text useful, but it is not required: [Modern Operating Systems \(4th Ed.\) \(http://www.pearsonhighered.com/educator/product/Modern-Operating-Systems/9780133581620.page\)](http://www.pearsonhighered.com/educator/product/Modern-Operating-Systems/9780133581620.page) by Tanenbaum and Bos.

You will also want a good C reference, such as [C Programming: A Modern Approach \(2nd Ed.\) \(http://knking.com/books/c2/index.html\)](http://knking.com/books/c2/index.html) by King, or [C: A Reference Manual \(5th Ed.\) \(https://www.pearson.com/us/higher-education/program/Harbison-C-A-Reference-Manual-5th-Edition/PGM220244.html\)](https://www.pearson.com/us/higher-education/program/Harbison-C-A-Reference-Manual-5th-Edition/PGM220244.html) by Harbison and Steele.

Website and Discussion Board:

You will be able to find all course materials from this Quercus site. The Piazza discussion board is required reading. Please use Piazza to ask general questions, and remember to search to check if someone else has already answered the question. The instructor and/or TAs will be monitoring it daily.

Marking Scheme

Overview

ITEM	WEIGHT (SCHEME 1)	WEIGHT (SCHEME 2)	WEIGHT (SCHEME 3)
Warm-up Assignment	1 %	1 %	1 %
Group Activities (6/8)	3 %	3 %	3 %
Assignment Preparation (4)	4 %	4 %	4 %
Programming Assignments (4)	32 %	32 %	32 %
Midterms (2) [worth 9 or 12% each]	24 %	21 %	18 %
Final Exam	35 %	38 %	41 %

ITEM	WEIGHT (SCHEME 1)	WEIGHT (SCHEME 2)	WEIGHT (SCHEME 3)
Surveys (4)	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 effectiveness of a new exam study platform. Each survey is worth 0.25% of your final grade.

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.

Lecture Exercises (0%):

There will be practice exercises associated with most lectures. These will be delivered as quizzes on Quercus and will not be graded. They are intended to help you strengthen your understanding of the material. We strongly encourage you to complete them promptly after each class, since they are designed to prepare you for assignments, tests, and exams. You are welcome to bring questions about these activities to the synchronous lectures.

Group Activities (3%):

During regular class hours, you will complete in-class group activities (for example, think–pair–share exercises). Each student must submit their own solution individually on Quercus. To receive full marks, you must indicate the names and student numbers of your group partners. Many of these activities will connect directly to the programming assignments.

We will count the best 6 out of 8 submissions (each worth 0.5%). Deliverables for these activities are due on Sunday of the same week at 11:59 p.m. Students who cannot form groups during class may use Piazza to connect with peers asynchronously in order to meet the group requirement.

Exception: G2 will be due on Sunday, September 21 at 11:59 p.m. to allow waitlisted students to complete them.

No grace periods may be used for these in-class activities.

Programming Assignments (32%):

Over the term, you will complete 4 assignments that require you to write C code to implement some subsystem or solve some problem, related to operating system concepts. **All assignments must be completed individually.** Each assignment is worth 8% of your course grade. All assignments must be submitted by checking your work into your MarkUs repository.

Warm-up Assignment (1%):

The warm-up assignment (A0) is designed to refresh your C programming skills. The code you write for A0 will be used in the next assignment, so it is important to complete it as soon as possible. This will allow you to assess your readiness and seek help early if you are struggling, since coding is a significant part of this course.

The warm-up assignment is due Sunday, September 21 at 11:59 p.m. No grace periods may be used for this assignment.

Assignment Preparation (4%):


Each assignment will be accompanied by a small tutorial exercise designed to help you understand the assignment more deeply. These exercises are due one week after the corresponding assignment is released. For example, if Assignment 1 is released in Week 1, its tutorial exercise will be due on Sunday of Week 2 at 11:59 p.m.

If you find the tutorial exercises challenging, you are strongly encouraged to seek help during TA or instructor office hours. Each tutorial exercise is worth 1% of your final grade, and grace periods may not be used for them.

Midterm Tests (18-24%):

There will be two 50-minute tests, on Wednesday, **October 1** and Wednesday, **November 5**, held during your regular lecture time. 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 5-8. Content may include lectures, exercises, activities, and assignments due before the test date. You must write the midterm in your assigned lecture section.

	Midterm 1 - October 1st	Midterm 2 - November 5th
LEC0101	Wednesday 1pm - 2pm, Room EX100	Wednesday 1pm - 2pm, Room FE 230, SS 1071 and SS2135
LEC0201	Wednesday 3pm - 4pm, Room EX100	Wednesday 3pm - 4pm, Room GB 248 and MP 103
LEC5101	Wednesday 6pm - 7pm, Room EX100	Wednesday 6pm - 7pm, Room BA 1180 and MC 252

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 [Midterm Special Consideration Form](https://forms.office.com/r/BXXf7A49wfl)  (<https://forms.office.com/r/BXXf7A49wfl>) 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 25% on the final exam is required to pass the course.


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 teach.cs lab machines. Last-minute difficulties with git 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 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 Extension Form](https://forms.office.com/r/khH9zCem4g)  (<https://forms.office.com/r/khH9zCem4g>) 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. *Grace tokens may also not be used for A0 or group activities.*

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>)
 - 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:

1. **Marking error:** Clearly and concisely describe the specific error you believe occurred in the grading.
2. **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 **one week** 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.
- **Do not use AI-generated code.** The entire code you submit must be written by you. Submitting code 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.

Use of Generative AI

The use of generative artificial intelligence (AI) tools in assignments and practice work is strongly discouraged. This includes, but is not limited to, ChatGPT, Copilot, and open-source models you may have trained or deployed yourself.

While it may be tempting to use AI tools when doing assignments, practice tests, or exercises, doing so will inhibit your learning, and you will not be able to use them on midterms or the final exam.

Think of it this way: 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>)) or phone (**416-978-8060** (tel:416-978-8060)).