

Course Syllabus

[Jump to Today](#)

 [Edit](#)

Overview

Welcome to CSC369H: Operating Systems. The course covers principles of operating systems with a focus on systems programming in C.

In this course, you will gain:

- Experience working with systems code written in C.
- Theoretical knowledge of and practical experience with:
 - Processes and Threads (address spaces, system calls, scheduling)
 - Synchronization (algorithms and structures like locks, semaphores, and monitors)
 - Virtual Memory (paging, page tables, eviction, segmentation)
 - File Systems (the file abstraction, directory structures, and disk I/O)

This material is core CS. Past students have indicated that the assignments in this course provided them with examples for interviews and that OS concepts featured prominently in interview problems. CSC369 is structured around a series of programming assignments covering key topics: threads, synchronization, virtual memory, and file systems. The lectures provide a historical and theoretical context for the assignments. However, they are not traditional lectures; while some material will be presented, some class time will also be spent in small group activities. The tutorial time will be used to reinforce the theoretical content through exercise, but will frequently be used to help you work on components of the assignments. We (your instructors), and your classmates, will expect you to be an active participant; to do so, *you must keep up with the readings and begin the assignments promptly*. There is solid evidence that starting early on assignments is correlated with success in the course (not to mention less stress), and higher than average grades. **None of the assignments can be completed if you start less than 3-4 days before the deadline.**

Course Information

Instructor:

Name: Angela Demke Brown

Virtual Office: <https://utoronto.zoom.us/j/81654358743>
(<https://utoronto.zoom.us/j/81654358743>) (passcode:
csc369h1s) *This is the same as the lecture zoom link.*

Physical Office: BA5228

Office Hours: Monday 2-3 p.m., Wednesday 10-11 a.m.

Office hours will be virtual via Zoom until at least February 1st. In-person office hours will start when regular on-campus activities resume. Meetings may be scheduled outside of regular office hours by request.

Communications:

Course Email: csc369-2022-01@cs.toronto.edu (<mailto:csc369-2022-01@cs.toronto.edu>)

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

Lectures:

All lectures will be online synchronous via Zoom until February 1st. When regular in-person activities resume, lectures will be held in the locations indicated.

LEC0101	Tuesday 2-3 p.m., WI 1016 (https://map.utoronto.ca/?id=1809#!m/494503)	Thursday 2-3 p.m., WB 116 (https://map.utoronto.ca/?id=1809#!m/494460)
LEC0201	Tuesday 3-4 p.m., BA 1190 (https://map.utoronto.ca/?id=1809#!m/494470)	Thursday 3-4 p.m., BA 1190 (https://map.utoronto.ca/?id=1809#!m/494470)
BOTH	https://utoronto.zoom.us/j/81654358743 (https://utoronto.zoom.us/j/81654358743) Passcode: csc369h1s	

Format and Preparation for lectures:

I plan to regularly assign readings and/or short videos to be reviewed before the synchronous lecture so that we can spend more time during the lecture 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 synchronous lectures and complete the exercises during, or shortly after, the lectures.

This course, including your participation, will be recorded on video and will be available to students in the course for viewing remotely and after each session.

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 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 instructors.

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

Tutorials:

All tutorials will be online synchronous via Zoom until February 1st. When regular in-person activities resume, tutorial room assignments will be posted as an announcement.

LEC0101	Friday 2-3 p.m.
LEC0201	Friday 3-4 p.m.
BOTH	https://utoronto.zoom.us/j/81654358743 (https://utoronto.zoom.us/j/81654358743) Passcode: csc369h1s (same as synchronous lectures)

Tutorials will be delivered as asynchronous videos to help you understand the exercises or assignments. During tutorial times, TAs will be available for questions via Zoom or in-person when regular in-person activities resume.

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/9780133591620.page) (<http://www.pearsonhighered.com/educator/product/Modern-Operating-Systems/9780133591620.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) by King (<http://knking.com/books/c2/index.html>), 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 linked to from this Quercus site. The Piazza discussion board (linked from Quercus) 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
Lecture Exercises	8 %
Tutorial Exercises	10 %
Programming Assignments (4)	36 %
Term Tests (2)	20 %
Final Exam	25 %
Research Survey	1 %

Detailed Description

Lecture Exercises 8%:

There will be in-class exercises associated with the lectures during most classes. These will typically take the form of quizzes on Quercus or small auto-checked exercises on MarkUs and will be graded on best effort. In other words, you will get full marks for an exercise if we can tell that you have made a serious attempt at it. You may not get credit for the exercise if you do nothing and just submit. We will take the best $n-2$ out of n scores. We expect that n will be 22 so that each completed lecture exercise is worth 0.4%. We strongly encourage you to do these exercises in class (i.e., during the synchronous lecture session) to get a clearer understanding of the material. We welcome questions about these activities during the synchronous lectures. All lecture exercises will be due by the end of the day following the lecture (i.e., 11:59:59 p.m. Wednesday for the Tuesday lectures, and 11:59:59 p.m. Friday for the Thursday lectures).

Tutorial Exercises 10%:

For each tutorial, you will be asked to turn in a small piece of work or to work on an activity during the tutorial. These exercises will be auto-graded and should be completed individually. Where possible, you

will be allowed to resubmit and we will take your final score. Many of these will be directly related to the assignments. We will take the best 8 out of 10 scores (for 1.25% each). There will be no tutorial exercises in the weeks when term tests are written.

No grace periods may be used for the tutorial exercises.

Programming Assignments (4 x 9% each = 36%):

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.** All assignments must be submitted by checking your work into your MarkUs repository.

Term Tests (2 x 10% each = 20%):

There will be two 50-minute tests at the end of Week 5 (Feb. 11) and Week 10 (Mar. 25) of the term, which will be completed during your regular tutorial time slot. These tests are currently scheduled as in-person assessments. Should it become necessary to conduct either or both tests online, they will be delivered using a combination of Quercus quizzes and timed assessments on MarkUs. The material covered on the term tests will not be cumulative.

Final Exam (25%):

The Final Exam will be scheduled by Arts & Sciences in the final assessment period. It will cover all course material, including topics that were tested on the term tests and questions about the assignments.

Research Survey (1%):

To support computer science education research, you will be asked to complete a short survey, which should take about 15 minutes of work. Everyone who completes the survey will receive 1%, even if they do not consent to sharing their data for the study.

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 are submitted electronically and are due at **9:00:00 p.m. sharp** on the due date. Each student is granted **twelve 4-hour grace tokens** for the entire semester, to be used on any of the assignments as you see fit. Submitting an assignment up to 4 hours late uses one token. Once your tokens have been exhausted, late assignments *will not be accepted*, except in extremely special circumstances. Grace tokens *may not be used for exercises*.

The grace tokens are provided to help you deal with minor, unforeseen disruptions to your work schedule. However, instructional staff will not 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 9:00:01 p.m. will be considered late, and ensure that your work is not submitted at the very last second. Because 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, fill out the absence declaration form on ACORN and notify us to request special consideration. 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.

Re-mark Requests:

If a piece of work has been marked incorrectly or if you believe the rubric used to evaluate the work is not appropriate, you may request a re-mark. For a re-mark to succeed, you must clearly and concisely express the marking error that you believe occurred. To request a re-mark, use the form for the assignment on MarkUs. Requests must be submitted within *1 week* of the marks being returned. Remarking may increase the original grade, leave it as is, *or possibly decrease* the original grade.

If there was a problem with your submission or your code did not compile, you may request a remark. If the remark request is accepted, there will be a 20% penalty applied.

Academic Integrity:

All of the work you submit must be done by you and your work must not be submitted by someone else. Plagiarism is academic fraud and is taken very seriously. The department uses software that compares

programs for evidence of similar code. Please refer to the UoFT [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)).

Here are a few guidelines to help you avoid plagiarism:

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 that is in preparation for submission, or submitted work, to a TA or instructor.




Important: 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.











Important: Do not look for assignment solutions online. Places like public Github repositories may contain code that may be useful in your assignments. Using someone else's code and ideas without attribution, even if making some changes, is considered plagiarism. Keep in mind that our plagiarism detection software can detect such cases.

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)).

Course Summary:

Date	Details	Due
	 L1: Privileged or not? (https://q.utoronto.ca/courses/250638/assignments/786114)	due by 11:59pm
Fri Jan 14, 2022	 L1: Why are you here? (https://q.utoronto.ca/courses/250638/assignments/786110)	due by 11:59pm
	 L2: Processes (https://q.utoronto.ca/courses/250638/assignments/786117)	due by 11:59pm

Date	Details	Due
Tue Jan 18, 2022	 Tut1: C and GDB (https://q.utoronto.ca/courses/250638/assignments/786103)	due by 12:01pm
	 Tut1: C and gdb (https://q.utoronto.ca/courses/250638/assignments/786128)	due by 12:01pm
Wed Jan 19, 2022	 L3: Threads And System Calls (https://q.utoronto.ca/courses/250638/assignments/786102)	due by 11:59pm
Thu Feb 3, 2022	 Assignment 1: Cooperative User-level Threads (https://q.utoronto.ca/courses/250638/assignments/786124)	due by 9pm
Tue Mar 1, 2022	 Assignment 2: Concurrency (https://q.utoronto.ca/courses/250638/assignments/786125)	due by 9pm
Thu Mar 17, 2022	 Assignment 3: Virtual Memory (https://q.utoronto.ca/courses/250638/assignments/786127)	due by 9pm
Tue Apr 5, 2022	 Assignment 4: File Systems (https://q.utoronto.ca/courses/250638/assignments/786126)	due by 9pm
	 TutX: File System Image (https://q.utoronto.ca/courses/250638/assignments/786129)	
	 TutX: FUSE passthrough (https://q.utoronto.ca/courses/250638/assignments/786130)	
	 TutX: Path lookup (https://q.utoronto.ca/courses/250638/assignments/786131)	