All information on this infosheet is also available in webpage format on the course Quercus site:

https://q.utoronto.ca/courses/334563

Overview

Welcome to CSC469H: Design and Implementation of Operating Systems / CSC2208H: Advanced 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.

For this year, lecture and tutorial material will be blended over the three class meetings per week. 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 discuss design alternatives.

Course Information

Class Meetings

Times: Monday, Wednesday, Friday @ 11 a.m.-12 noon Eastern Time
Format: In-person
Location: SF3202

Copyright Notice:

Course Staff

Instructor: Angela Demke Brown
Physical Office: BA 5228
Physical Office Hour: Fridays 1-2 p.m.
Email: `csc469-2024-09@cs.toronto.edu`
TAs: Vinicius Dantas de Lima Melo, Guy Khazma and Pawan Kumar Sanjaya

Note: Virtual or physical meetings can be scheduled at students' request outside of the posted office hours.

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 the 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 (online)
- Marshall Kirk McKusick, George V. Neville-Neil and Robert N. M. Watson: *The Design and Implementation of the FreeBSD Operating System.* Addison Wesley (2nd ed. 2015).
- K.N. King: *C Programming: A Modern Approach.* Norton and Co (2nd ed. 2008).

Website and Discussion Board

The course website will be maintained on Quercus (`https://q.utoronto.ca/courses/334563`) and is required reading. It contains lecture notes, assignment handouts, tutorial materials, policies, etc. as well as a link to a discussion board (Piazza). A shared discussion board will help you get a faster response to any questions – but this will only work if you participate! The board is the best place to get answers to your questions. Course announcements will be posted to both the discussion board and the course website so check them regularly.

Email Policy

If you are having trouble with the course material or if you need extra help, please do not hesitate to contact me. I will answer as soon as possible (usually within 24 hours, longer on weekends). Keep in mind that email volume (and hence response latency) increases as due dates approach.

Please follow these guidelines for email correspondence:

1. Read the posts on the discussion board to see if your question has already been answered.
2. If your question may be of interest to other students (e.g., a question about an assignment, the readings, or lectures), post to the discussion board instead of sending email. If your question is personal (e.g., a question about missing a test due to illness), definitely send email.
3. Use a good subject. Include the course number (to avoid the spam filter) and an informative topic (for example, `"CSC469: problem compiling libraries for A1"`).

Marking Scheme

Individual Exercises: 15%

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.

- Exercise 1 (5%) - Due Tuesday, January 23, 9:00 p.m.
- Exercise 2 (5%) - Due Tuesday, February 27, 9:00 p.m.
- Exercise 3 (5%) - Due Tuesday, March 19, 9:00 p.m.

Group Assignments: 30%

Working in teams, students will take on three more substantial assignments related to some of the core concepts in the course. CSC369 students must work in teams of 2-3, while CSC2208 students must work in teams of 1-2; CSC469 and CSC2208 students may not work together.

- Assignment 1 (10%) - Performance Evaluation - Due Friday, February 9, 9:00 p.m.
- Assignment 2 (10%) - Concurrency - Due Friday, March 8, 9:00 p.m.
- Assignment 3 (10%) - Distributed Systems - Due Friday, April 5, 9:00 p.m.

Term Tests (2 x 10%): 20%

The term tests will be held in-person, during Wednesday class times (50-minute duration, 11:10 a.m. - 12:00 p.m.).

- Test 1 (10%) - Wednesday, February 14, 10:10 a.m. - 11:00 a.m.
- Test 2 (10%) - Wednesday, March 27, 11 a.m. - 12:00 p.m.

Final Exam: 35%

The final exam will be cover all course material and will be scheduled during the exam period by the Faculty of Arts and Science.

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 can easily be avoided by ensuring all files are added to the repository well before the deadline, and that you know how to commit+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. Eastern Time** on the due date. Each student is granted **twelve 4-hour grace tokens** for the entire semester, to be used on any of the group assignments or individual exercises as you see fit. Submitting an assignment up to 4 hours late uses one token. Once your tokens have been used late assignments *will not be accepted*, except in extremely special circumstances.

For group assignments, a grace token will be deducted from **all** team members for each extra 4-hour period. As a result, you can use at most the minimum number of grace tokens between the team members.

Please note that 9:00:59 p.m. will be considered late, and ensure that your work is not submitted at the very last second. We will allow up to 59 seconds of "free" grace time, but will be unsympathetic to pleas regarding submissions that are "just a few seconds" late beyond that point. 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 me 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 instructor 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
  - 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)
  - 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.
- 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 page. It is always easier to make alternate arrangements before a due date, so please inform me 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.

Re-mark Requests

If a piece of work has been mis-marked 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 what you believe was mis-marked. 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.

Academic Integrity

All of the work you submit must be done by you (and your team members, where applicable) 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 familiarize yourself with the UofT Academic Integrity website and the UofT Code of Behaviour on Academic Matters. **An academic offense 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 offense on your record.**

Here are a few guidelines to help you avoid plagiarism in this course.

- 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**: Do not look for assignment solutions online. Places like public Github repositories may contain code that may be useful in your assignments, but **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. 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 cases you **must** summarize the information you find in your own words and you **must** properly cite the sources that you use. If you are unsure how to cite some source, ask your instructor.
- **Important**: The entire code must be written by yourself (or your team, for group assignments). Submitting code you find elsewhere or AI-generated code (e.g., ChatGPT, Github Co-pilot) is strictly forbidden and any violation will be prosecuted.
- **Important: You must discuss the assignments with your team, not just to understand the content, but also to avoid the unfortunate situation where your team members might be committing plagiarism.** If you suspect that a team member does not understand their own code, it may be a sign that they have plagiarized the code from other sources. Keep in mind that you are responsible for all the work submitted and plagiarism cases will be prosecuted for all team members, so you must be vigilant and involved in all parts of the assignment.

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 as soon as possible: disability.services@utoronto.ca or http://studentlife.utoronto.ca/accessibility.