Course Syllabus

Course Syllabus

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 understanding OS concepts helped them in their career. CSC369 is structured around a series of programming assignments covering key topics: file systems, system calls, process management, synchronization, and virtual memory. 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. I 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 at less than 3-4 days before the deadline.**

Course Information

Instructor: Karen Reid

Physical Office: BA4224

Virtual Office: Zoom/Piazza/Email

In-person office hours: Monday 3-4, Wednesday 3-4

Online office hours: Friday 3-4pm

Email: csc369-2021-09@cs.toronto.edu

Lectures:

LEC 0101 2-3pm Mondays and Wednesdays	BA 1130
LEC 5101 6-8pm Wednesdays	BA 1170

Once again, the beginning of this term is filled with uncertainties, so plans for this course may change as the guidance from the University and the province shifts. At this point in time, I plan to hold lectures in the classroom, live stream, and record lectures for those that cannot attend. I have not taught in a hybrid fashion before, so I will be relying on your help and feedback so that all students have a good experience. You should expect some technical problems from time to time, especially with the live stream.

Every week has readings and/or short videos to be reviewed before the lecture so that we can spend more time during the lecture working through some of the details, and you will be working on exercises. Students attending online using zoom should expect to use breakout rooms to work on the exercises in small groups. Lectures will be recorded and posted later.

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 instructor, the University, and/or other source 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 recording and use of videos in which you appear please contact your instructor.

Tutorials: We are still working out the details with the TAs, so these plans may change

LEC 0101 2-3pm Friday	BA 1130	BA 2135
LEC 5101 8-9pm Wednesday	BA 1130	BA 2195

Textbook(s)

Textbook(s):

We will be using a free text quite extensively **Operating Systems: Three Easy Pieces** **(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** **(http://www.pearsonhighered.com/educator/product/Modern-Operating-Systems/9780133591620.page)** Systems by Andrew Tannenbaum.

You will also want a good C reference.

Website and Discussion Board:

You will be able to find all course materials linked to from this Quercus site. Piazza has finally been integrated with Quercus and is required reading. Please use the discussion board to ask general questions, and remember to search to check if someone else has already answered the question. I will be monitoring it daily, and we will have TAs monitoring close to assignment deadlines.

Email

Please use email for personal issues and the discussion board 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.);

Marking Scheme

Tutorial Exercises 12%:

For each tutorial, you will be asked to turn in a small piece of work or to work on an activity during tutorial. These exercises will be auto-graded, and a few can be completed in pairs. 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 9 scores.

No grace periods may be used for the tutorial exercises.

In-Class Exercises 10%:

There will be in-class exercises during most lectures. 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 the exercise if we can tell that you have made a serious attempt at the exercises. You may not get credit for the exercise if you do nothing and just submit. As with tutorial exercises, we will take the best n-2 out of n scores. We strongly encourage you to do these exercises in class, to get a clearer understanding of the material, and welcome questions during these activities.

Assignments 33%:

Over the term, you will complete 3 assignments. For all three assignments, you may work individually or in pairs of two. To submit your assignment, check your work into your repository. **Both partners MUST understand ALL parts of the assignments.**

Midterm test (12%) - The midterm will be a 50 minute in-person test held during class time on November 3 in EX100.

If you are registered in the day section (LEC 0101) you will write from 2-3pm and if you are registered in the evening section (LEC 5101) you will write from 6-7 pm. There will be a lecture in place of the tutorial that week.

Final Exam (32%) - The final exam will also be in-perons and will be held during the final exam period and will be scheduled by Arts and Science. To pass the course you much achieve a grade of at least 40% on the final exam.

Survey 1%: More information to follow.

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. There will be a 10% penalty applied if your program compiles with warnings. 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. Executable files must not be committed to the repository. Compiling and testing your work on the teaching lab machines at intermediate stages will also avoid last minute problems. **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 **10 p.m. sharp** on the due date. Each student is granted **three 24-hour grace tokens** for the entire semester, to be used on any of the assignments as you see fit. Submitting an assignment up to 24 hours late uses one token. Once your tokens have been used late assignments *will not be accepted*, except in extremely special circumstances. Grace days *may not be used for exercises*.

Additionally, if you partner for assignments, for each extra 24 hours, a grace token will be deducted from **both** partners (which means that in effect, you can use up to at most the minimum number of grace tokens between the team members).

This is so that those who prefer to work individually are not at a disadvantage. Please note that 10:00:01 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 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, fill in the absence declaration form and contact me by email within 48 hours of the due date. 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.

Re-mark Requests:

If a piece of work has been marked incorrectly, 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.

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 partner, 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 read the **Rules and Regulations (http://www.governingcouncil.utoronto.ca/policies/behaveac.htm)** from the U of T Calendar (especially the Code of Behaviour on Academic Matters). 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 submitted 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.

Important: **You must discuss the assignment with your partner, not just to understand the content, but also to avoid the unfortunate situation where your partner might be committing plagiarism.** If you suspect that your partner does not understand their own code, it may be a sign that your partner has 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 both assignment partners, 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