Course Outline Template

COMP 3000A/B Term Fall 2024 (Preliminary Version)

Operating Systems

Course Information

Instructor: Zinovi Rabinovich Contact: Zinovi.Rabinovich@carleton.ca Classroom: Check your schedule on Carleton Central Lectures: Section A: Tuesdays & Thursdays, 16:05 – 17:25 (in person) Section B: Tuesdays & Thursdays, 08:35 – 09:55 (in person)

Tutorials: Check your schedule on Carleton Central Course Website: <u>Brightspace</u>

For information about Carleton's academic year, including registration and withdrawal dates, see <u>Carleton's Academic Calendar</u>.

Teaching Assistants

A list of teaching assistants and their contact/office hours information will be posted once the course starts.

Course Calendar Description

Operating system implementation course stressing fundamental issues in design and how they relate to modern computer architectures. Assignments involve the modification and extension of a multitasking operating system.

Includes: Experiential Learning Activity

Precludes additional credit for SYSC 4001.

Prerequisite(s): COMP 2401 with a minimum grade of C- and COMP 2402.

Lectures three hours a week, tutorial one and a half hours a week. alendar)

Required Textbook(s) and Other Resources

Course lectures are a based on the <u>"Operating System Concepts"</u> book by Silberschatz, Galvin, Gagne (10th ed).

Additional reading source is "Operating Systems: Three Easy Pieces".

SCS Laptop Requirement (only applies to on-campus courses)

Course mid/end-term tests will use a laptop.

Please notice that every student that has been enrolled in a 1000-level (i.e., first year) course offered by the School of Computer Science after the 2020/2021 school year is required to have

a laptop. This includes COMP1001, COMP1005, and COMP1006. For more information, please visit <u>https://carleton.ca/scs/scs-laptop-requirement/</u> and then review the requirements at <u>h</u>ttps://carleton.ca/scs/scs-laptop-requirement/laptop-specs/..

Topics Covered and Learning Outcomes

Learning Outcomes

- Have a strong conceptual model of how an operating system works that can facilitate software development/testing and answer questions pertaining to an operating system's everyday use.
- Be able to write/modify C code that uses low-level Linux services and implement simple Linux kernel extensions (modules).
- Understand the basic use and architecture of virtual-machine based and container based cloud architectures.

Topics Covered:

Tentative lecture schedule by topics (subject to change):

- Week 1: Introduction (Ch1)
- Week 2: Operating System Structures (Ch 2) and Processes (Ch 3)
- Week 3: Threads and Concurrency (Ch 4) and CPU Scheduling (Ch 5)
- Week 4: Synchronisation Tools (Ch 6) and Examples (Ch 7)
- Week 5: Deadlocks (Ch 8) and Main Memory (Ch 9)
- Week 6: Virtual Memory (Ch 10) and Mid-term test
- Week 7: Mass-Storage Structure (Ch 11) and I/O (Ch 12)
- Week 8: File System Interface (Ch 13) and Implementation (Ch 14)
- Week 9: File System Internals (Ch 15) and Security (Ch 16)
- Week 10: Protection (Ch 17) and Virtual Machines (Ch 18)
- Week 11: Networks and Distributed Systems (Ch 19) and/or Case Studies
- Week 12: Case Studies and End-term test
- Week 13: Material review and Exam prep

Important dates and deadlines can be found here:

<u>https://carleton.ca/registrar/registration/dates/academic-dates/</u>, including class suspension for fall, winter breaks, and statutory holidays.

Assessment Scheme (number of tutorials and assignments is subject to change)

- Tutorials (theoretical, best 10 out of 12) 12%
- Self-study assignments (best 7 out 8) 14%
- Mid-term test (Week 6, closed book, in class) 17%
- End-term test (Week 12/13, closed book, in class) 17%
- Final exam (during the final exam period, **closed book**) 40%

Assignment due dates will be made available in <u>Brightspace</u>. The tutorials will be published in parallel with the lecture materials. You must submit your tutorial solution no later than one hour after the end of your tutorial session the week after the tutorial definition is published.

Self-study assignments (with a how-to attached) are practical exercises on code modification, may not fully interlock with the pace of lectures, and do require self-study. Tutorial sessions will be dedicated both to the review of theoretical tutorial exercises and assistance with self-study assignments, but this assistance does not replace self-study.

Up to one of the in-term tests (mid-term or end-term) can be missed, in which case the missed test's mark will be replaced by the average of the other in-term test and the final exam. You must pass the final exam. You are to take due diligence to attend both in-term tests.

IMPORTANT CONSIDERATIONS

Assignments/tutorials submissions are handled electronically (i.e., through <u>Brightspace</u>) and there is no "grace period" with respect to a deadline - an assignment submitted even one minute after the deadline is late and will not be accepted by the system. Any extenuating situation or academic accommodation request must be received by the instructor up to <u>24 hours following the due time</u>, and will be considered on a case-by-case basis. Technical problems do **not** exempt you from this requirement, so if you wait until the last minute and then have issues with your connection, you will still receive a mark of zero. Consequently, you are advised to submit your work several hours in advance of the due time. You can also submit a work-in-progress version and override it with the completed version before the deadline. Contact the TAs/instructor in case of any problems.

Format errors, missing files, and other technical/non-technical upload issues will not constitute the justification for another attempt. Only what has been uploaded by the due time will be graded. Consequently, after you upload your submission to Brightspace you should download it immediately for verification and ensure that all needed files are there in the right format.

AUTHORIZED AND UNAUTHORIZED COLLABORATION

Collaboration on all work is allowed **except** for the in-term tests and the final exams. Collaboration, however, should be **clearly acknowledged**, by mentioning the name(s) of who you have collaborated with.

For assignments, while you may get help from others and even collaboratively solve technical problems, the <u>code and answers submitted</u> should all be your own work. For example, you may not divide an assignment into parts, give a part to another student or anyone else to solve, and then submit that work as your own. You must have participated in the creation of every part of your submitted work. An easy way to enure this happens is to never share files regarding coursework or copy and paste answers.

Instead, only meet together (virtually) to work on an assignment (e.g., to discuss your understanding/confusion) and then **separate** to write up your own solutions.

Similarity between submitted assignments that has not been appropriately documented will be treated as plagiarism - the same as copying on a midterm or a final - and will be submitted to the Dean for disciplinary action.

Academic Integrity:

If you are unsure of the expectations regarding academic integrity (how to use and cite references, if unauthorized collaboration with lab- or classmates is permitted (and, if so, to what degree), then you must ASK your instructor. Sharing assignment or quiz specifications or posting them online (to sites like Chegg, CourseHero, OneClass, etc.) is ALWAYS considered academic misconduct. You are NEVER permitted to post, share, or upload course materials without explicit permission from your instructor. Academic integrity offences are reported to the office of the Dean of Science. Information, process and penalties for such offences can be found on the ODS webpage: https://science.carleton.ca/students/academic-integrity/.

Many of the assessed activities in this course were designed to be completed by an individual working alone. Unless it is explicitly stated otherwise, the use of any AI system will be considered academic misconduct. This includes, but is not limited to, chatbots or code generators (e.g., ChatGPT, Google Gemini, Microsoft Copilot), research assistants (e.g., Elicit), and image generators (e.g., Stable Diffusion, Dall-E), etc."

An exception to the above rule is made for automated grammar and punctuation checking tools (such as Grammarly).

References to any material you use but did not originate must use the IEEE/APA/MLA citation style. Failure to reference materials correctly can result in severe penalties, and the use of manufactured (i.e., falsified) or misleading references will be treated as evidence of plagiarism and considered academic misconduct.

Undergraduate Academic Advisors

The Undergraduate Advisors for the School of Computer Science are available in Room 5302HP; or by email at <u>scs.ug.advisor@cunet.carleton.ca</u>. The undergraduate advisors can assist with information about prerequisites and preclusions, course substitutions/equivalencies, understanding your academic audit and the remaining requirements for graduation. The undergraduate advisors will also refer students to appropriate resources such as the Science Student Success Centre, Learning Support Services and Writing Tutorial Services.

Graduate Academic Advisors

The Graduate Advisors for the School of Computer Science are available in Room 5302 HP; or by email at grad.scs@carleton.ca The graduate advisors can assist with understanding your academic audit and the remaining courses required to meet graduation requirements.

SCS Computer Laboratory

Students taking a COMP course can access the SCS computer labs. The lab schedule and location can be found at: <u>https://carleton.ca/scs/tech-support/computer-laboratories/</u>. All SCS computer lab and technical support information can be found at: <u>https://carleton.ca/scs/tech-support/</u>. Technical support staff may be contacted in-person or virtually, see this page for details: <u>https://carleton.ca/scs/tech-support/contact-it-support/</u>.

University Policies:

• Academic Accommodations

Carleton is committed to providing academic accessibility for all individuals. Please review the academic accommodation available to students here: <u>https://students.carleton.ca/course-outline/</u>.

• Academic Integrity

Student Academic Integrity Policy. Every student should be familiar with the Carleton University Student Academic Integrity policy. A student found in violation of academic integrity standards may be sanctioned with penalties which range from a reprimand to receiving a grade of F in the course, or even being suspended or expelled from the University. Examples of punishable offences include plagiarism and unauthorized collaboration. Any such reported offences will be reviewed by the office of the Dean of Science. More information on this policy may be found on the ODS Academic Integrity page: <u>Academic Integrity | Faculty of Science</u> (carleton.ca).

Plagiarism. As defined by Senate, "plagiarism is presenting, whether intentional or not, the ideas, expression of ideas or work of others as one's own". Such reported offences will be reviewed by the office of the Dean of Science. More information and standard sanction guidelines can be found here: <u>https://science.carleton.ca/students/academic-integrity/</u>. Please note that content generated by an unauthorized A.I.-based tool *is* considered plagiarized material.

Unauthorized Collaboration. Senate policy states that "to ensure fairness and equity in assessment of term work, students shall not co-operate or collaborate in the completion of an academic assignment, in whole or in part, when the instructor has indicated that the assignment is to be completed on an individual basis".