COMP 346
Operating Systems
Course Outline - Winter 2019

Instructor	Dr. Abbas Javadtalab E-mail: abbas.javadtalab@concordia.ca Office: EV 3-233 Tel: (514) 848-2424 ext. 4079 **Lecture:** Section YY: Monday 5:45 PM - 8:15 PM at H-520 Office Hours: 1) Wednesday 10:00AM – 12:00 noon 2) By appointment (send e-mail).
Instructor	Dr. Bahareh Goodarzi E-mail: bahareh.goodarzi@.concordia.ca Office: EV 3-231 Tel: (514) 848-2424 ext 4106 **Lecture:** Section WW: Tuesday 5:45 PM - 8:15 PM at H-520 Office Hours: 1) Thursday 5:30 PM – 6:30 PM 2) By appointment (send e-mail).
Instructor & Coordinator	Dr. Aiman Hanna E-mail: contact@AimanHanna.com Office: EV 3-257 Tel: (514) 848-2424 ext 7878 **Lecture:** Section UU: Monday 5:45 PM - 8:15 PM at H-820 Section NN: Wednesday 5:45 PM - 8:15 PM at H-620 Office Hours: 1) Thursday: 4:30 PM - 5:30 PM; 2) By appointment anytime.
Lab Instructors & Markers	→ Please check the course websites for information on lab instructors/tutors/markers as well as lab/tutorials times and locations.

Objectives

This course has two components: a *theory component* to teach you the concepts and principles that underlie modern operating systems, and a *practice component* to relate theoretical principles with operating system implementation. In the theory component, you will learn about processes and processor management, concurrency and synchronization, memory management schemes, file system and secondary storage management, security and protection, etc. The practice component will complement the theory component through programming assignments illustrating the use and implementation of these concepts. At the end of the course you should:

- Understand fundamental operating system abstractions such as *processes*, *threads*, *files*, *semaphores*, *IPC abstractions*, *shared memory regions*, etc.,

- Understand how the operating system abstractions can be used in the development of application programs, or to build higher level abstractions,
- Understand how the operating system abstractions can be implemented,
- Understand the principles of concurrency and synchronization, and apply them to write correct concurrent programs/software,
- Understand basic resource management techniques (scheduling or time management, space management) and principles and how they can be implemented. These also include issues of performance and fairness objectives, avoiding deadlocks, as well as security and protection.

Textbook and Syllabus

The textbook for the course is: ***Operating System Concepts, 9th Edition***, by: S**ilberschatz/Galvin/Gagne**, published by John Wiley & Sons, 2013. **ISBN 978-1-118-06333-0**. (6th edition ISBN **0-471-41743-2**, and later ones, are also okay).

The exact syllabus for the course will depend on how quickly we can cover the introductory chapters. In general, our aim for this semester is to cover chapters 1 through 14, and parts of the following chapters. Any exclusion will be announced in class, and made available through the web page.

Other References:

Operating Systems, 3rd edition By: ***Gary Nutt***, published by *Addison-Wesley.* ISBN 0201773449 (ISBN-10: 0201773449 | ISBN-13: 9780201773446). *Second edition of this book (Operating Systems: A Modern Perspective) is also okay.*

Applied Operating System Concepts*, First Edition,* By: *Silberschatz/Galvin/Gagne* - published by *John Wiley & Sons, 2000.*

Background

The official pre-requisite for this course is Data Structures & Algorithms (COMP 352, and Comp 228 or Soen 228). Much of the material covered in these two courses is directly relevant for this course, and therefore it is critical that you understand and remember the knowledge that you acquired in these courses. Some of the critical concepts that you should know well to do well in this course include:

- The hardware structure of modern computers, the role of the different parts (CPU, main memory, secondary memory, system bus, peripheral devices), the inter-connections between the different parts, and how the different parts interact to support computation, communication, and storage.
- The binary number system, basic arithmetic and logical operations on binary numbers, and converting binary numbers to decimal and vice-versa.
- The hardware-software interface including assembly language programming, accessing registers and main memory from software, I/O instructions, interrupts and interrupt handling, etc.
- Common data structures such as queues, stacks, lists, prioritized lists, etc. Implementation of such data structures using arrays, dynamic arrays, and linked-lists. Evaluation of algorithmic complexity of the operations.

The important skills needed to do well in this course include: (1) good programming skills, including the ability to understand, design, implement, and debug programs with non-obvious flow of control, (2) the ability to understand tradeoffs using a mix of qualitative and quantitative reasoning of design choices, (3) the ability to relate the theoretical material covered in class to the practical aspects of implementation, and (4) the ability to abstract the knowledge learnt and apply it to a wide-range of problems (not necessarily related to operating systems, or even computer science for that matter).

Workload and Grading

This is a reasonably heavy course with several new concepts, and a fair bit of programming workload. Therefore, you should be prepared to spend adequate time and effort on this course. The practice/programming component of the course will utilize Java programming language. Although you are not required to have a strong previous knowledge of Java, it is assumed that you have a reasonably solid knowledge with object oriented programming. Sufficient introduction of Java will be provided during the early tutorials, so it is very important that you attend the tutorials.

1. **Programming Assignments:** 20%

   There will be 3 or 4 programming assignments. These assignments are designed to give you more understanding of the course material and to give you practices into aspects of operating system design and implementation. The assignments are using the Java language under Linux operating system. More information on the specific compiler version as well as where it is located in the system will be provided during lab times. Reasonable attempts of these assignments are necessary for passing the course.

   IMPORTANT: A demo will be required for each of these assignments. The marker will communicate with you through the mailing list and you must book a demo time with him/her. You **must** perform the demo (if working in a group, both members of a team must be present during the demo). Failing to demo the assignment, or failing to attend the demo at the reserved time, will result in a 0 mark regardless of your submission. There will be no replacement for a missed demo time.

2. **Mid-Term Exam:** 25%

   There will be one mid term exam. The date for the exam will be announced at the beginning of the term. The midterm will cover all material presented in the lectures, the textbook, and in the assignments and labs, up to and including the lecture preceding the exam.

3. **Final Exam:** 55%

   There will be one common final exam for all sections. The final exam will be scheduled by the University Exam's office. The exam will cover material from the whole semester, including lectures, textbook, and assignments. Passing the final exam is necessary for passing the course.

Theoretical Assignments:

(Please read carefully): There will be a total of 3 or 4 theoretical assignments. These assignments will not be marked (or only selected questions may be marked) and hence may not evaluate to any load of the course. The main purpose behind these assignments is to provide you with good preparation for the final. **You are required to submit all of these assignments before the term's last day of classes. Although these assignments may not represent any load after all, failing to submit any of these assignments will cost you some significant marks. Additionally, these assignments must be typed and submitted online to the EAS system. Scanned Hand-written assignments will be discarded.**

⇨ Please note that all assignments will be placed on the website (see below); no hardcopies of the assignments will be distributed in class. All assignments (theoretical and programming) must be submitted electronically either at https://fis.encs.concordia.ca/eas or on Moodle, depending on your section. Your professor will indicate the particular location of the submission for your section.

Submission format: All assignment-related submissions must be adequately archived in a ZIP file using your ID(s) and last name(s) as file name. The submission itself must also contain your name(s) and student ID(s). Use your "official" name only - no abbreviations or nick names; capitalize the usual "last" name. Inappropriate submissions will be heavily penalized. Only electronic submissions will be accepted. Students will have to submit their assignments (one copy per group for the programming assignments; theory assignments are individual assignments) using the Moodle/EAS system (please check for your section submission). Assignments must be submitted in the right DropBox/folder of the assignments. **Assignments uploaded to an incorrect DropBox/folder will not be marked and result in a zero mark. No resubmissions will be allowed.**

The grading of the course will be done based on the relative percentages assigned to the assignments and the exams. For reasons of fairness, we may choose to scale up/down the marks in a particular exam or assignment to ensure that all aspects of the course receive a fair weight. Any such "fine-tuning" will be made known to you before the final grades are assessed. Finally, there are no pre-set cutoff points for the final grades; the cutoff points will be decided based on an assessment of difficulty level, class performance, fairness, and instructor's wisdom from teaching and grading the course in the past. That is, there is no definite rule for translation of number grades to letter grades.

Graduate Attributes

This course emphasizes and develops the following CEAB graduate attributes:

- **Knowledge-base:** *Knowledge of fundamentals of operating system functionalities, design and implementation. Multiprogramming: processes and threads, context switching, queuing models and scheduling. Inter-process communication and synchronization. Principles of concurrency. Synchronization primitives. Deadlock detection and recovery, prevention and avoidance schemes. Memory management. Device management. File systems. Protection models and schemes.*
  - **Indicators:** Indicator 1.3: Knowledge-base in a specific domain.

- **Problem analysis:** *Analyze multithreaded programs to check for deadlocks and race conditions. Design race-condition and deadlock-free solutions for concurrency.*
  - **Indicators:** Indicator 2.1: Problem identification and formulation.
    Indicator 2.2: Modeling.

- **Design:** *Design and implement programs that interact with operating systems kernels. Develop programs using concurrency principles.*
  - **Indicators:** Indicator 4.3: Architectural and detailed design.
    Indicator 4.4: Implementation and validation.

- **Use of Engineering tools:** *Use appropriate system kernel resources to develop system software. Use semaphores and Java's built-in synchronization primitives and add-on utilities (e.g., locks and condition variables) to implement concurrent programs.*
  - **Indicators:** Indicator 5.1: Ability to use appropriate tools, techniques and resources.

The evaluation of these attributes will be based on: 1) Programming Assignments, 2) Midterm, and 2) Final exam questions. This evaluation is used to indicate your proficiency in all of the attributes as per accreditation requirements.

Course Structure

Lectures

The lectures are a key component of the course, and you are advised to attend the lectures regularly and attentively. The course material is extensive and includes several difficult concepts. Accordingly, we will try to utilize the lecture time in doing things that you would not get by simply reading the book (explaining difficult concepts, giving you alternate perspectives, relating course material to other fields, giving you tools to solve problems, etc.) rather than merely repeating facts that you can simply read from the text book. It is strongly advised that you stay current in your reading of the textbook, and attend the lectures regularly. That will enhance your learning experience and prevent you from being lost in the lectures. In fact, we suggest that you casually go through the textbook sections once before the lecture. You are also strongly advised to go and read the material thoroughly after the lecture. Discussing the material with your fellow classmates, solving problems, and asking questions in the lectures are also likely to help you.

Tutorials

Two tutorials will take place every week. During the tutorial, your tutor will explain the Lab assignments and answer questions related to the course, and more specifically related to the programming assignments. We strongly encourage you to attend these tutorials.

Labs

These lab hours are designated to provide you with personalized help on questions related to the lab assignments. You can also seek the lab instructor's help on general Java/Unix questions. Please make the best use of these lab hours. Due to extreme time limitations, no discussions in relation to the lab assignments will take place during the lecture or with the

course instructors, so you must take advantage of these lab hours for any questions related to your lab assignments.

Exams

We strive hard to give exams where your success does not depend on rote memorization. You will probably find it difficult to answer many of the questions if you have memorized the concepts without understanding them. Moreover, many exam questions will not only test your understanding of concepts, but also your ability to apply them to solve problems. The assignments should help you in preparing for the exams. There will be no substitution of any missing exam.

Important Lecture Guidelines

Laptops are **STRICTLY PROHIBITED** in classroom during the lectures. Other communications devices, such as cellular phones, communication watches, and text/video messaging devices, tablets, pads, and similar devices are also **STRICTLY PROHIBITED**. The usage of any of these materials during the class will result in you being asked to immediately leave the class.

Website and other Resources

Some resources for the course (assignments, tutorials, etc.) will be available online.

For sections YY & WW (Dr. Javadtalab & Dr. Goodarzi), please use the Moodle Web site available through the MyConcordia portal www.myconcordia.ca.

For Sections UU and NN (Dr. Hanna), the webpage for the course is: **www.AimanHanna.com** (follow Concordia links afterwards). The web pages will contain announcements related to the class, pointers to documents, your theory and lab assignments, etc.

Additionally, a mailing list will be established for the course. You should register to this mailing list ASAP. To register, please link to:

https://mail.encs.concordia.ca:444/mailman/listinfo/comp346-w19.

Finally, the faculty web pages have a wealth of information pertaining to our computer systems and software, which includes simple user guides, and answers to many standard questions. You should explore these help pages. Begin your exploration from the URL: http://www.encs.concordia.ca/helpdesk/faq/faq.php