

COMP 3500 Introduction to Operating Systems

Syllabus

Term: Fall 2015

Schedule: MWF, 11:00 – 11:50 am, in Shelby Center 1124

Instructor: Dr. Xiao Qin (xqin@auburn.edu)

Office: 3101E Shelby Center

Office Phone: 334-844-6327

Office Hours: MWF, 1:00 pm – 1:50 pm

TA: Yuanqi Chen, yzc0029@auburn.edu

Office Hour: MWF 3:00pm-3:50pm.

Office: 3139 Shelby Center

Required Text: Operating System Concepts 9th Edition, Authors: Abraham Silberschatz, Peter B. Galvin, and Greg Gagne. ISBN-13: 978-1118063330, ISBN-10: 1118063333.

Optional Text:

- Operating Systems, Publisher: Pearson, 8th Edition, Author: Stallings, Publication Date: Jan. 1, 2014. ISBN 9780133805918
- Operating Systems (3rd Edition) 3rd Edition Author: Gary Nutt. ISBN-13: 978-0201773446, ISBN-10: 0201773449

Course Description

Course Objectives

This course aims to

- educate students on the fundamental concepts and components of operating systems;
- introduce the characteristics of hardware related to operating systems; and
- present the good tradeoffs among objectives conflicting with one another.

On completion of the course, students should be able to:

- understand the essential concepts and mechanisms of operating systems;
- implement a synchronization mechanism;
- build system calls;
- construct a virtual memory management module;
- implement a simple file system;
- evaluate operating system performance;
- know the state-of-the-art technologies in operating systems research and development; and
- demonstrate ability to program at low levels.

This course typically requires at least 5 hours of time per week, on average for the average student. If you don't have it, drop.

Prerequisite:

- COMP 2710 Software Construction
- COMP 3350 Computer Organization and Assembly Language Programming

Topic List (not necessarily strictly in chronological order; these topics may change)

- Administrative Stuff
- Processes and scheduling
- Concurrency problems
- Synchronization
- Mutual exclusion
- Deadlocks
- Memory management
- Virtual memory
- Input/Output
- File systems

Assessment**Exams:** Two Midterm Exams and Final Exam

Exams will be closed book, closed notes. Questions will be derived from lectures, material taught only in class, and from assignments. Question format will be mixed.

Short Homework Assignments and Activities: 2-3 homework assignments

These activities will be take-home in nature and designed to reinforce concepts taught in class. An electronic copy must be submitted through the Canvas system. Generally, these assignments are designed to be low-risk in the sense that they are designed to assess thinking and effort, rather than to strictly punish errors.

Individual and Group Programming Projects: 5 programming projects

- There will be five programming assignments, which will make use of the OS/161 educational operating system running on a simulated MIPS R3000 computer called System/161. The simulators, which were developed at by the Systems Research group at Harvard University, are relatively platform independent (OS/161 and System/161 run on Linux and Unix as well as Windows systems).
- Assignment solutions will be submitted in C code. All projects should be made to compile under the gcc compiler on Linux.
- **You may use any development platform or compiler, but your projects will be graded ONLY on a gcc compilers running on Linux. If your project does not work in that environment, you will NOT get credit. Always test your programs on a Linux machine!**
- I reserve the right to assess other penalties for any errors not strictly covered in the above rubric. I also reserve the right to give bonus points for exceptional work.

Grades:

- Mid-term 1 10%
- Mid-term 2 10%

- Final Exam 20%
- Quizzes 10%
- Homework 10%
- Programming Projects 40%

A [90, 100], B [80,90), C [70,80), D [60,70), F [0,60)

Note: In order to pass the class, you must receive at least 60% credit on the Individual Construction Projects and Homework, regardless of performance on exams.

Course Policies

Makeup Quiz Policy: Missed quizzes may NOT be made up unless there is an official university excuse, a medical or family emergency. You must request a makeup quiz via email **prior to** the class in which the quiz is given. Make-up quizzes will only be given before the class in which the quiz is returned to the students. There is no guarantee that your request will be granted.

Makeup Policy for Midterm and Final Exams: There will be no make-up exams except through arrangement with Dr. Qin **prior to** the exams. There is no guarantee that your request will be granted.

Scaling, Curves, etc: Grades may be (not guaranteed) scaled, curved, or adjusted arbitrarily. Some opportunities for bonus points may be provided to the entire class at the instructor's discretion.

Project Due Dates: Projects will be submitted through Canvas. Projects will always be due at 11:55 pm on the due date. Deadlines will be made as generous as possible to *a priori* take into account illness, other courses, Acts of God, and nearly all conceivable excuses. If you have a documented illness preventing you from completing your assignment, you may submit all of your partial work and request an extension. **This extension is not automatic.**

Late Submission Penalty

- Twenty percent (10%) penalty per day for late submission. For example, an assignment submitted after the deadline but up to 1 day (24 hours) late can achieve a maximum of 90% of points allocated for the assignment. An assignment submitted after the deadline but up to 2 days (48 hours) late can achieve a maximum of 80% of points allocated for the assignment.
- Assignment submitted more than 3 days (72 hours) after the deadline will not be graded.

Rebuttal Period: You will be given a period of 72 hours to read and respond to the comments and grades of your homework or project assignment. The TA may use this opportunity to address any concern and question you have. The TA also may ask for additional information from you regarding your homework or project.

Announcements: E-mail is an official form of university communication. You are responsible for all announcements made in class or electronically. You should read your e-mail at least once a day. If a student asks a particularly relevant question, I may forward the response to the entire class for their benefit.

Special Accommodations: A student in need of special accommodations must bring that need to my attention within the first two weeks of class. The need must be properly documented.

Academic Integrity: Students will be expected to understand and follow Academic Honesty policies in place by the university. Programming projects and written assignment are completed in teams of two. Students may discuss with their friends about general approaches to solving problems and writing programs. Students in one group should NOT share any project code or even detailed algorithm information with students in other groups. Your programming code is exclusive to you and your group. Please do not attempt to recycle code from the Internet (plagiarism). Any instance of suspected cheating or plagiarism will be referred to student judicial affairs.

Approved references:

The following constitute acceptable references to help you complete assignments.

- The course textbook is always approved and content may be used without citation.
- My course notes, lectures, and advice I give in my office may be used without citation
- Online general web references are fine, provided you give a citation for the website at the top of your code AND clearly label any lines of code that you use (it should never be ambiguous which lines of code you used from a website)
- Other books/textbooks on the language are fine, but require citations
- You are allowed to discuss broad conceptual ideas (for example, the idea of polymorphism) with other students, but never to share code. If you discuss something with another student (even casually), you should always cite that reference in clear terms.

Unapproved references (these constitute Academic Dishonesty):

This is not a complete listing and cases of ambiguity should always be referred to the instructor for approval prior to use.

- Solution manuals for the text (or the like)
- Websites that sell custom code to individuals
- Code written by others (students or otherwise) for this class or similar classes
- Anything not listed under “Approved References” or approved by the instructor

You MUST document references clearly. If you discuss a project with another student or professor, you should indicate what you discussed and who you discussed it with clearly in the header of your project documentation (and/or code).

For example:

//Xiao Qin

//Project2.cpp

//Dr. Homer Carlisle helped me debug a syntax error in my ‘for’ loop.

//I used Wikipedia.org in order to learn how a genetic algorithm works.

//I spoke with Bob Smith in the class about identifying objects in C++.

If you don’t need any sources for an assignment, clearly state “I did not use any external sources for this assignment” in your source code. Failing to document sources is plagiarism and will be penalized.

If you are unsure whether or not to document a source, it is better to document. Breaches of Academic Honesty will be referred to the Academic Honesty Committee and the strictest sanctions possible (including expulsion and failure) will be my recommendation.

If you are ever unclear about whether or not a course of action is unacceptable, you are always encouraged to consult the instructor.

Attendance: You are responsible for all material and announcements presented in class (even if absent).

1) For exams:

- a. if you have a planned university-approved absence you must make me aware before the test in writing (with appropriate documentation).
- b. if you an unplanned absence, you must provide written, documented, and verifiable justification.
- c. Make-up exams will be different from the original exam.

2) If you are late for a test, you do not receive any extension.

3) Consistent attendance is typically essential to obtaining a good grade (C or better) in COMP 4300.

Reading: Students are expected to read the appropriate sections of the book before each lecture.

Getting Help: Project assignments are challenging and time-consuming. You are always welcome to bring questions concerning labs to the class, as well as to office hours. A good strategy is to always start early on projects, so that if you run into difficulties, you can get help as soon as possible. I will do my best to answer e-mails concerning labs within 48 hours of receiving them; however, I do not guarantee that I will always have time to debug code via e-mail (I prefer not to do so). For time-consuming problems dealing with code, office hours are always preferable. I will not help debug code via e-mail on the day an assignment is due. The Blackboard Discussion Board is a great way to ask questions so that everyone benefits from the answer to your question!

Office Hours: You are always welcome to drop by during office hours to discuss projects or general concepts. To get urgent help or advice out of office hours, it is recommended to send an email in advance to make an appointment.

Course Difficulty: The course starts off relatively easy and gets harder as time goes on. Often, students are deceived by the (slower) initial pace and develop lazy habits at the beginning of the course. By mid-semester, they have thrown away many grade opportunities and find themselves in a bad situation with respect to grades. No amount of effort at the end of the class will compensate for consistent, dedicated effort throughout the class. Whether or not you have past experience with programming (or even with C++), my strongest recommendation is that you respect the class and come to class ready to engage every single class period. If you do this, you will dramatically increase your chances of success.

Study Hints

- Ask questions in class.
- At the first sign of difficulty, talk to your instructor and the teaching assistant.
- Form a study group and meet regularly.
- Construct chapter summaries noting concepts, definitions, & procedures.