

SYLLABUS

Fall, 2016

CSC 381 Operating System Principles

3.0 credits

Prerequisite(s): CSC 260

Instructor: Beifang Yi
email: byi@salemstate.edu

Office: MH 211A
Hours: 9:40-10:40 (W/F)
12:10-1:10 (W/F)
3:30-4:30 (W/F)

Phone: (978) 542-7246
WebSite: <http://cs.salemstate.edu/~byi/>

Section	Time	Room	Final Exam
01	W & F 10:50am—12:05pm	MH 301	Tuesday 12/20, 8:00am-10:00am MH 301

Catalog description:

This course presents the evolution of computer operating systems, operating system functionalities, and current design and implementation techniques. Relationships between the operating system, computer architecture, and the user community are discussed. Three lecture hours per week.

Course Goals:

The aims of this course are:

- CG1: present a descriptive overview of modern operating systems, their purposes and design principles;
- CG2: discuss the most important ingredients, techniques, and algorithms used in their construction;
- CG3: foster an understanding of fundamental technical issues in the implementation of modern operating systems.

Course Objectives:

Upon completion of this course, the student will have demonstrated the ability to:

- CO01: summarize the development of operating systems from single-user and batch processing mainframes to modern multitasking systems
- CO02: describe the mechanisms of interrupts and Direct Memory Access;
- CO03: describe how a process or task is represented in a modern computer system;
- CO04: differentiate between the concepts of process and thread and describe the behavior of a multithreaded system;
- CO05: describe the system components, actions, and algorithms involved in scheduling and managing concurrent processes;
- CO06: describe the concept of deadlock and the common techniques for recognizing, predicting, avoiding, and recovering from it;
- CO07: describe the common techniques and problems involved in memory management, including paging and virtual memory;
- CO08: describe the common techniques and problems involved in management of disk storage;
- CO09: describe the common techniques and problems involved in file management;
- CO10: describe the common techniques and problems involved in system protection and security.

Course Topics:

A detailed topics list and a general course bibliography can be found on the Computer Science Department website at <http://cs.salemstate.edu/dept/index.php?page=184>. Select CSC 381 to access a PDF document.

The focus of the course is on a general discussion of the nature and functionality of operating systems. There is no extended treatment of specific case studies, although examples drawn from specific systems are used to illustrate major concepts. In addition to the topics listed in the above link, we may also briefly introduce the following special purpose systems:

- threads and thread programming
- virtualization/virtual machines
- Linux

Text(s): (required) **Operating System Concepts**, 9th Edition, by A. Silberschatz, P. B. Galvin, and G. Gagne. Wiley, 2013 (ISBN: 978-1-118-06333-0).

Required Material:

(Required) Thumb (flash) drive (or other portable memory devices), 4 GB minimum.

Additional references (optional):

- Course website: http://cs.salemstate.edu/~byi/CSC381_Silberschatz/index.html
- *Principles of Modern Operating System*, 2nd Edition, by J. M. Garrido, R. Schlesinger, and K. Hoganson. Jones & Barlett Learning, 2013
- *Mastering Lambdas: Java Programming in a Multicore World* (Oracle Press), 1st Edition, by Maurice Naftalin. McGraw-Hill Osborne Media, 2014.
 - <https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html>
 - <http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/Lambda-QuickStart/index.html>
- *Start Concurrent: An Introduction to Problem Solving in Java with a Focus on Concurrency*, by Barry Wittman. And Tim Korb. Purdue University Press (Kindle/2014, paperback/2013).
- *Parallel and Concurrent Programming in Haskell: Techniques for Multicore and Multi-threaded Programming*, 1st edition, by Simon Marlow. O'Reilly Media, 2013.
- *Java Concurrency in Practice*, 1st edition, by Biran Goetz & at. al. Addison-Wesley Professional, 2006.
- *Understanding Unix/Linux Programming: A Guide to Theory and Practice* (2003), by Bruce Molay. Prentice Hall.
- *Java: How to Program*, 10th edition. Deitel & Deitel. Prentice Hall, 2015.
- VMWare website: <http://www.vmware.com/>
- (Handouts will be given in class).

Cell phones:

Turn the ringer off, or, better yet, *turn the phone off*.

Class Attendance:

Class attendance is highly recommended. You are responsible for all materials presented in class, examinations, and other announcements. No excuses of any nature will be construed as relieving you from the responsibility for completion of the work assigned. Each student is responsible for completing all course requirements and for keeping up with all that goes on in the course (whether or not the student is present).

Final Grade:

Final grade will be determined using the following grading weights:

written assignments	35%
projects	30%
midterm examination	10%
final examination	25%

Attendance is not used to calculate the final grade: *however*, note that you are at all times responsible for assignments and materials presented in class.

The following table shows how the course work is assessed against the course objectives:

	Written Assignments	Projects	Examinations
CO01	✓		✓
CO02	✓		✓
CO03	✓	✓	✓
CO04	✓	✓	✓
CO05	✓	✓	✓
CO06	✓	✓	✓
CO07	✓	✓	✓
CO08	✓	✓	✓
CO09	✓		✓
CO10	✓		✓

Projects:

There will be several projects (including programming projects) to be completed throughout the semester. The programming languages and platforms will vary (C/C++, Java...Windows, Linux...) and will be given out in class. These projects will help students understand the basic concepts of modern operating systems and get hands-on experience in the implementation of some concepts and functions of the operating systems. There is a deadline to each project and **penalty** will be imposed for late submissions (see the Due Dates/Time below).

For each project, a write-up should be submitted which describes the problem, solutions, and other requirements (usually, there are additional requirements for the assignments). For a programming project, it should provide the following: (a) instructions on how to compile and run the program, (b) testing results, (b) source code, and (d) input data (if applicable).

There will be bonus projects given in the semester which are optional for bonus project credits. Bonus credits will be added to the final grade (with some adjustment).

All the projects must be submitted at Canvas (<https://elearning.salemstate.edu/>).

Homework/Written Assignments:

There will be a series of written assignments from the textbook and other sources: question-answering, short essay-writing, and/or reporting of calculation results. Reading will be a part of the written assignments. Please note that in addition to these (written) assignments there will be (Programming) Projects. There is a deadline to each assignment and **penalty** will be imposed for late submissions (see the Due Dates/Time below).

All the assignments must be submitted at Canvas (<https://elearning.salemstate.edu/>).

Each assignment may have different full score points, depending on the difficulty and the amount of the work of the exercises. There might be one or two extra assignments given in the semester and these extra assignments will be used as *make-up* assignments. All the assignment (including the extra assignments) questions might be in the examination. Any assignment may be used as the extras. The average score for the overall assignments will be the total scores received for all the assignments divided by the total scores of the **required** assignments. You may **not** move the extra points to the final grade. For example, if there are about 1100-point assignment questions throughout the semester (including the extra assignments) and the required assignment total scores will be about 1000 points, and if students A and B have completed 1050-point and 800-point assignments, their overall semester assignment (written) grades will be 100% and 80% respectively for A and B).

Examinations/Tests:

There will be one midterm and one final (comprehensive) examination. The midterm will be held in week 8 *depending* on class progress. **Note:** Make-ups are given for examinations only under exceptional circumstances and with documented circumstances.

Missed Tests:

Missed tests will be made up *only under extreme conditions/emergency with the proper documentation*. Students who know in advance that they must be absent on an exam day for an excusable reason should notify the instructor prior to the exam day. Students who are absent on the day of the exam for an excusable reason should contact the instructor immediately following their absence. Makeup work will be permitted *only when* the instructor is presented with acceptable documentation for acceptable absences. It is student's responsibility to notify the instructor of any excused absence as far in advance as possible.

Due Dates/Time:

- **There will be a 20% penalty for each week an assignment (lab/project/short-answer/written exercises...) is late;** penalties accrue at the due time of the assigned due date.
- **No assignments (lab/project/short-answer exercise...) will be accepted after the final examination.**

Study Groups:

While I strongly encourage study groups, I require that each student hand in his/her answers in her/his own words - if two answers are highly similar to each other, neither will receive credit.

When working on your programming projects, you may discuss with others the project topics, the algorithms and methodologies related to the project; but when you work on writing the code, this coding work should be 100% of your own work.

If two answers/written code segments come out exactly the same or highly similar, neither will receive credit and/or further actions will be taken (such as reporting to the department and/or university). Given the nature of most of the projects, homework questions and writing assignments, it will be almost impossible for two people to come up with highly similar answers UNLESS they copy.

Academic Integrity:

Academic Integrity Policy and Regulations can be found in the University Catalog and on the University's website ([http://catalog.salemstate.edu/content.php?catoid=13&navoid=1295#Academic Integrity](http://catalog.salemstate.edu/content.php?catoid=13&navoid=1295#Academic_Integrity)). The formal regulations are extensive and detailed - familiarize yourself with them if you have not previously done so. A concise summary of and direct quote from the regulations: "Materials (written or otherwise) submitted to fulfill academic requirements must represent a student's own efforts". *Submission of other's work as one's own without proper attribution is in direct violation of the University's Policy* and will be dealt with according to the University's formal Procedures.

"Salem State University is committed to providing equal access to the educational experience for all students in compliance with Section 504 of The Rehabilitation Act and The Americans with Disabilities Act and to providing all reasonable academic accommodations, aids and adjustments. Any student who has a documented disability requiring an accommodation, aid or adjustment should speak with the instructor immediately. Students with Disabilities who have not previously done so should provide documentation to and schedule an appointment with the Office for Students with Disabilities and obtain appropriate services."

In the event of a University declared critical emergency, Salem State University reserves the right to alter this course plan. Students should refer to <http://www.salemstate.edu> for further information and updates. The course attendance policy stays in effect until there is a university declared critical emergency. In the event of an emergency, please refer to the alternative educational plans for this course via the course link at Canvas (<https://elearning.salemstate.edu/>).

Please remember that if, for any reason, you decide to drop this course, you **MUST** do so officially through the Registrar's office. The last day to withdraw from a course this semester is **November 28th**.

Note: This syllabus represents the intended structure of the course for the semester. If changes are necessary, students will be notified in writing and via all regular class communication mechanisms (class discussion, emails, and/or the course link at Canvas <https://elearning.salemstate.edu/>).