Computer Science Department
cs.salemstate.edu

CSC 381 Operating System Principles **3 cr.**

Instructor: TBA **Office:** location **Phone:** (978) 542-extension
email: TBA@salemstate.edu **Office Hours:** days and times

Section	Time	Room	Final Exam
nn	days and times	location	date and time

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. Not open to students who have received credit for CSC 280.

Prerequisite: CSC 260.

Goals:

The aims of this course are to:

CG01: present a descriptive overview of modern operating systems, their purposes and their design principles;
CG02: discuss the most important ingredients, techniques and algorithms used in their operating system construction;
CG03: foster an understanding of fundamental technical issues in the implementation of modern operating systems.

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.

Student Outcome (SO) vs. Course Objectives matrix

SO	CO01	CO02	CO03	CO04	CO05	CO06	CO07	CO08	CO09	CO10
SO-1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SO-2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SO-3										
SO-4	✓									✓
SO-5										
SO-6	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Notes:

SO-1: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to

identify solutions.

SO-2: Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

SO-3: Communicate effectively in a variety of professional contexts.

SO-4: Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.

SO-5: Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.

SO-6: Apply computer science theory and software development fundamentals to produce computing-based solutions.

Topics:

- introduction: **OS1(1)**
  - What is an operating system?
  - functions and goals of an operating system
  - principal components
- review of relevant topics in computer hardware, architecture, and organization **AR2(0.5),AR3(1.5),AR4(2),AR5(0.5)**
- evolution of operating systems **OS1(1)**
  - resident monitors (single-user systems)
  - multiprogramming
  - time-sharing
  - networks and distributed processing **AR9(0.5)**
  - multiprocessing **AR7(1)**
- operating system structures
  - system services, system calls
  - interrupts and interrupt handling **OS2(1)**
  - kernel of an operating system **OS2(0.5)**
  - layered structure **OS2(0.5)**
  - virtual machines
- evaluation of system performance **OS11(1)**
- protection and security mechanisms **OS7(2)**
- process management and threading
  - the process concept **OS2(0.5)**
  - representation of processes **OS2(0.5)**
  - concurrent processes **OS3(1)**
  - CPU scheduling and scheduling algorithms **OS4(4)**
  - multiprocessor scheduling **OS4(1)**
  - threading models
  - threading libraries
- process coordination and thread synchronization
  - classical synchronization problems **OS3(2)**
    - Critical Section Problem, Bounded Buffer Problem, Readers & Writers Problem, etc.
  - synchronization mechanisms **OS3(3)**
    - hardware
    - semaphores
    - language constructs
  - interprocess communication, message systems **OS3(0.5)**
  - thread synchronization in Java
- deadlocks **OS3(1.5)**
  - characterization, detection, prevention, avoidance, recovery
- storage management **OS5(8)**
  - memory management
  - swapping
  - paging
  - segmentation
  - virtual memory
    - demand paging
    - page replacement algorithms

        - frame allocation algorithms
        - thrashing
    - secondary storage management **OS6(3)**
        - disk structure
        - allocation methods
        - scheduling algorithms
- file management **OS8(1)**
    - file systems
    - access methods
    - file protection
- distributed systems
    - network topologies and types **AR9(0.5)**
    - coodination and deadlock in distributed systems
    - distributed file systems

The focus of the course is on a general discussion of the nature and functionality of operating systems. Extended treatment of specific case studies is not appropriate to the course, although examples drawn from specific systems are used to illustrate major concepts.

Assignments: Weekly reading assignments in the text and in other books and journals provide the background for lecture topics. It will be assumed that the reading has been done *before* the class discussion of the associated material.

Approximately eight written assignments with specific deadlines cover such topics as development, analysis, and comparison of scheduling and allocation algorithms, synchronization of concurrent processes, multi-threading issues, resource allocation graphs, problems in paging and virtual memory systems, disk scheduling, etc. Depending on the availability of appropriate hardware and software, some of the written assignments may be replaced by projects (including programming projects) on the same topics.

Examinations: A mid-term examination and a comprehensive written final examination are given..

The course grade is determined using the following approximate weights:

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

Course Objective / Assessment Mechanism matrix

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

Bibliography:

Billimoria, Kaiwan N. **Hands-On System Programming with Linux: Explore Linux system programming interfaces, theory, and practice.** Packt Publishing, 2018.

Blum, Richard. **Linux Command Line and Shell Scripting Bible. Third Edition.** Wiley, 2015.

Goetz, B; Peierls, T.; Bloch, J.; Bowbeer, J.; Holmes, D.; Lea, D. **Java Concurrency in Practice**. Addison Wesley, 2006.

Herlihy, Maurice; Shavit, Nir. **The Art of Multiprocessor Programming, Revised Reprint**. Morgan Kaufmann, 2012.

Hoover, Adam. **System Programming with C and Unix**. Addison Wesley, 2009.
Lavieri, Edward. **Mastering Java 11: Develop modular and secure Java applications using concurrency and advanced JDK libraries. Second Edition.** Packt Publishing, 2018.
McHoes, Ann; Flynn, Ida M. **Understanding Operating Systems. Eighth Edition**. Cengage Learning, 2017.
Nemeth, Evi; et. al. **UNIX and Linux System Administration Handbook. Fifth Edition.** Addison-Wesley Professional, 2017.
Newham, Cameron. **Learning the bash Shell: Unix Shell Programming (In a Nutshell). Third Edition**. O'Reilly Media, 2005.
Robbins, Kay A.; Robbins, Steve. **UNIX Systems Programming: Communication, Concurrency and Threads. Second Edition**. Prentice Hall, 2015.
Roscoe, A. W. **Understanding Concurrent Systems**. Springer, 2010.
Siever, Ellen; Figgins, Stephen; Love, Robert; Robbins, Arnold. **Linux in a Nutshell. Sixth Edition**. O'Reilly Media, 2009.
Silberschatz, Abraham; Galvin, Peter Baer. **Operating System Concepts**. **Ninth Edition.** Wiley, 2012.
Silberschatz, Abraham; Galvin, Peter Baer. **Operating System Concepts Essentials. Second Edition**. Wiley, 2013.
Stallings, William. **Operating Systems: Internals and Design Principles**. **Ninth Edition.** Pearson, 2017.
Tannenbaum, Andrew S. **Modern Operating Systems**. **Fourth Edition.** Pearson Hall, 2014.

Academic Integrity Statement:

"Salem State University assumes that all students come to the University with serious educational intent and expects them to be mature, responsible individuals who will exhibit high standards of honesty and personal conduct in their academic life. All forms of academic dishonesty are considered to be serious offences against the University community. The University will apply sanctions when student conduct interferes with the University primary responsibility of ensuring its educational objectives." Consult the University catalog for further details on Academic Integrity Regulations and, in particular, the University definition of academic dishonesty.

The Academic Integrity Policy and Regulations can be found in the University Catalog and on the University website (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. *Copying without attribution is considered cheating in an academic environment - simply put,* ***do not do it!***

University-Declared Critical Emergency Statement:

In the event of a university-declared emergency, Salem State University reserves the right to alter this course plan. Students should refer to 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, which will be distributed via standing class communication protocols. Students should review the plans and act accordingly. Any required material that may be necessary will have been previously distributed to students electronically or will be made available as needed via email and/or Internet access.

Equal Access Statement:

"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."

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 email.