Operating Systems

Spring 2021

Instructor:	Amir Hossein Rouhani Seraji	**Code:**	7330
Email:	ah.rouhaniseraji@gmail.com	**Unit:**	3

Course Pages: Please check this website two or three times a week

- https://rouhani-class.github.io/os

Discussion Group: Every student has to be a member of our Telegram Channel and Telegram Group for our discussion and daily announcements.

- Telegram Channel.
- Telegram Group.

Teaching Assistant:

- Mehran Moeini Jam.
- Ali Vanaki.
- Zeinab Emdadi.
- Parmida Ghamari.
- Mohammad Hossein Khoshechin.

Prerequisites: An undergraduate-level understanding of Computer Architecture & Organization, Programming in C, and Assembly.

Objectives: This course is primarily designed for undergraduate students. In this course many fundamentals concepts will be covered like: Understanding major components of OSs, Describe various ways of structuring an OS, Describe various features of processes including scheduling , creation , termination , & communication, Understand thread and its issues, Describe various CPU-Scheduling algorithms, Good Understanding of Evaluation Criteria to select a CPU-Scheduling algorithm for a system, Learn concepts of atomic excecution, Learn how to ensure consistency of shared data, Learn how to prevent deadlocks in an OS, Learn various ways of organizing memory, Understanding various memory management techniques, Learn Virtual-Management techniques, Understand functions and interface of file systems, Be able to explain structure of an OS I/O subsystem. The title of topics which will be covered are :

- Introduction
  - What is operating systems?
  - History of operating systems
  - Computer System organization & architecture
  - Operating systems operations
  - Distributed systems (optional)
- Operating Systems Structures

  - OS architecture types
  - OS services
  - System calls
  - Bootstrapping
  - Virtualization
  - OS design & implementation
  - Distributed Operating Systems (optional)
  - Real-time Operating Systems (optional)

- Processes
  - Process model
  - Process life-cycle
  - Multiprogramming & Multitasking
  - Operations on processes
  - Interprocess Communication techniques
  - Process scheduling
  - Process transition states in Solaris(optional)
  - Distributed Synchronous & Asynchronous Message-passing algorithms (optional)
- Threads
  - Thread model
  - Multicore programming
  - Posix Threads
  - Threading issues
  - Implementing Threads in user space & the kernel
  - Java Thread (optional)
- Synchronization
  - Race conditions
  - Critical section
  - Hardware support for synchronization
  - Mutual Exclusion & Busy waiting
  - Sleep & Wakeup
  - Semaphores
  - Mutex & Futex locks
  - Monitors & Barriers
  - Avoiding Locks
  - Alternative Approaches (optional)
  - Recoverable Mutual Exclusion (optional)
  - Verification of Reactive systems (optional)
  - Formalizing Mutual exclusion & Resource Allocation with I/O Automaton (optional)
- CPU Scheduling

  - Scheduling in various systems
  - Scheduling algorithms & criteria
  - Thread scheduling
  - Multi-Processor scheduling
  - Real-time CPU Scheduling
  - Solaris scheduling (optional)
  - Game-Theoritic approaches to Scheduling algorithms (optional)
  - Approximation Scheduling algorithms (optional)
  - Parallel Task Scheduling (optional)

- Memory Management
  - Memory abstraction
  - Address spaces
  - Swapping
  - Virtual memory
  - Paging
  - Structure of the page tables
  - Demand paging
  - Copy-on-Write
  - Page replacement algorithms
  - Design & Implementation issues for paging systems
  - Segmentation
  - Application-Integrated Far Memory (optional)

- Deadlock
  - Resources
  - System model
  - Deadlock modeling
  - Deadlock in multithreaded applications
  - Deadlocks handeling methods
  - Deadlock Prevention
  - Deadlock Avoidance
  - Deadlock Detection
  - Recovery from deadlock
  - Model checking of reactive systems for deadlock-freedom property (optional)

- File Systems
  - File Concepts
  - Access Methods
  - Directory structures
  - File system structures
  - File system operations

  - File system mounting
  - File system implementation
  - Distributed File Systems (optional)
- Storage & I/O Management
  - I/O devices
  - Device controllers
  - Memory-Mapped I/O
  - Direct memory access
  - Programmed I/O
  - Interrupt Handlers
  - Disk Hardware
  - Disk Formatting
  - Disk Arm Scheduling Algorithms
  - Parallel I/O operations (optional)

Main References: Our Lectuers are mainly based-on this book.

- Abraham Silberschatz, Peter Baer Galvin, Greg Gagne *Operating System Concepts*, Wiley, 10th ed, 2018.

Supplementary References: We also use these references for complementary discussion.

- Andrew S. Tanenbaum, Herbert Bos, *Modern Operating Systems*, Pearson, 4th ed, 2015.
- William Stallings, *Operating Systems Internals and Design Principles*, Pearson, 9th ed, 2018.
- Thomas Anderson, Michael Dahlin, *Operating Systems Principles & Practice*, Recursive Books, 2nd ed, 2015.
- Remzi H. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau, *Operating Systems: Three Easy Pieces*, CreateSpace Independent Publishing Platform, 2018.
- Maurice Bach, *The Design of the Unix Operating System*, Prentice-Hall, 1986.
- Richard McDougall, Jim Mauro, *Solaris Internals: Solaris 10 & OpenSolaris Kernel*, Prentice-Hall, 2nd ed, 2006.
- Marshall Mckusick, George Neville-Neil, Robert Watson, *Design & Implementation of the FreeBSD Operating System*, Addison-Wesley, 2nd ed, 2014.

Useful Links: Courses with Video Lectures.

- 6.S081: Operating System Engineering, MIT, Fall 2020.
- CS-537: Introduction to Operating Systems, University of Wisconsin, Spring 2018.
- CS124 Operating Systems, California Institute of Technology, Fall 2018.
- CS124 Operating Systems, California Institute of Technology, Spring 2017.
- CS-354: Introduction to Computer Systems, University of Wisconsin, Spring 2017.

- 15-213/18-213/15-513: Introduction to Computer Systems (ICS), Carnegie Mellon University, Fall 2017.
- COMPSCI 377: Operating Systems, University of Massachusetts Amherst , Fall 2016.
- COMPSCI 577: Operating Systems Design and Implementation, University of Massachusetts Amherst, Spring 2020.

Courses with lecture notes only.

- CPSC 423/523: Principles of Operating Systems, Yale University, Spring 2021.
- CS 422/522: Design and Implementation of Operating Systems, Yale University, Fall 2020.
- CS 537: Introduction to Operating Systems, University of Wisconsin, Fall 2020.
- CS 537 Intro to Operating Systems, University of Wisconsin, Spring 2020.
- CSE451: Introduction to Operating Systems, University of Washington, Spring 2021
- CSE551: Operating Systems, University of Washington, Spring 2019
- CS 162: Operating Systems and Systems Programming, UC Berkeley, Spring 2021.
- CS140: Operating Systems, Stanford University, Spring 2021.
- 15-410, Operating System Design and Implementation, Carnegie Mellon UniversityCarnegie Mellon, Spring 2021.
- CS 161: Operating Systems, California Institute of Technology, Spring 2021

Reasearch Groups in Operating Systems.

- Parallel & Distributed Operating Systems Group, MIT.
- The UNSAT group, University of Washington.
- Computer Systems Lab, University of Washington.
- The ADvanced Systems Laboratory (ADSL), University of Wisconsin.
- USENIX, The Advanced Computing Systems Association

Books website.

- Operating Systems: Three Easy Pieces
- Operating System Concepts
- Operating Systems: Principles and Practice

Documentaries to Watch.

- UNIX: Making Computers Easier To Use – AT&T Archives film from 1982, Bell Laboratories
- AT&T Archives: The UNIX Operating System
- Revolution OS: movie about GNU/Linux history

Our passed Semesters Audio Lectures.

- 7330 Operating Systems, IAU South Tehran Branch, Fall 2018

- 7330 Operating Systems, IAU South Tehran Branch, Spring 2017

Grading Policy:

- Assignments (40%)
- Quiz (30%)
- Final Exam (30%)
- (Optional) Presentation (10%)