Operating Systems

Module Description

Module code	CS204-S28	**Credits**	8 ECTS
Programme	Foundation programme	**Level**	Semester 4
Semester	Spring 2028	**Prerequisites**	CS102-F26 Digital Design and Computer Architecture I; CS103-S27 Data Structures and Algorithms I
Module leader	Dr. Ali Khudiyev	**Email**	ali.khudiyev@wosm.academy
Office hours	Monday–Friday, 10:00–17:00 (Baku time, UTC+4)		

1. Module Overview

An operating system turns bare hardware into a machine that many programs can share safely and efficiently. This module explains how it does so, around three ideas: *virtualisation* of the processor and memory, *concurrency*, and *persistence* of data in file systems.

The module builds on the two Digital Design and Computer Architecture modules, which explain the hardware the operating system manages: privilege modes, interrupts, caches and memory. Practical work is in C using the POSIX interface, building on the C programming from the Data Structures and Algorithms modules.

2. Aims

- Explain how an operating system virtualises the processor and memory and manages devices and storage.
- Teach the principles and pitfalls of concurrent programming.
- Show the design trade-offs in scheduling, memory management and file systems.
- Give practical experience of systems programming in C.

3. Learning Outcomes

After completing this course, you will be able to:

LO1. Explain the role and structure of an operating system, including privilege modes, interrupts, traps and system calls.

LO2. Explain processes and threads, and evaluate CPU scheduling policies.

LO3. Write correct concurrent programs using locks, semaphores and condition variables, and recognise, prevent and handle race conditions and deadlock.

LO4. Explain and analyse memory management: address spaces, segmentation, paging, translation lookaside buffers and page replacement.

LO5. Explain how file systems and I/O are implemented, including crash consistency.

LO6. Write systems programs in C that use the POSIX interface for processes, threads, memory and files.

4. Syllabus

Unit	Topic
1	Introduction: what an operating system does; operating system structures
2	Hardware support: privilege modes, interrupts, traps and system calls
3	Processes and the process API: fork, exec, wait
4	Inter-process communication: pipes, signals and shared memory
5	Threads and POSIX threads
6	CPU scheduling: FCFS, SJF, round robin, MLFQ; multiprocessor scheduling
7	Synchronisation I: race conditions, critical sections, locks
8	Synchronisation II: semaphores, condition variables, monitors; classic problems
9	Deadlock: conditions, prevention, avoidance and detection
10	Memory management: address spaces, segmentation and paging
11	Virtual memory: TLBs, multi-level page tables, page replacement
12	I/O devices, device drivers; hard disks and SSDs
13	File systems: files, directories and their implementation
14	Crash consistency: journaling and log-structured file systems
15	Protection and security
16	Virtual machines and containers
17	Course Review & Exam

5. Teaching and Learning Methods

The module is self-paced. All materials are provided on Moodle. Each unit has a lecture and exercises. Programming assignments in C let you build parts of operating-system functionality, such as a shell, a thread-safe data structure or a memory allocator.

Activity	Hours
Lectures	34
Labs / Tutorials	46
Independent study	145
Total	**225**

6. Assessment

Your knowledge will be assessed throughout the course with quizzes, programming assignments and the final exam. The exam checks whether you can explain and analyse operating-system mechanisms and policies and reason about concurrent code. The programming assignments check whether you can write working systems programs in C.

Component	Weight	Due	LOs
Quizzes	20%	Throughout	LO1–LO5
Programming assignments	40%	Throughout	LO3, LO6
Final exam	40%	End of module	LO1–LO5
Total	**100%**		

7. Reading List

Essential

- R. H. Arpaci-Dusseau, A. C. Arpaci-Dusseau: *Operating Systems: Three Easy Pieces*, Arpaci-

Dusseau Books, 2018. Free online at `https://pages.cs.wisc.edu/~remzi/OSTEP/`.

Recommended

- A. Silberschatz, P. B. Galvin, G. Gagne: *Operating System Concepts*, 10th ed., Wiley, 2018.
- A. S. Tanenbaum, H. Bos: *Modern Operating Systems*, 4th ed., Pearson, 2014.
- R. E. Bryant, D. R. O'Hallaron: *Computer Systems: A Programmer's Perspective*, 3rd ed., Pearson, 2016.
- B. W. Kernighan, D. M. Ritchie: *The C Programming Language*, 2nd ed., Prentice Hall, 1988.

8. Materials and Licence

Course materials are published under the Creative Commons Attribution-ShareAlike 4.0 licence unless stated otherwise. See wosm.academy/licence for details.

9. Policies

- **Attendance:** Attendance is not mandatory. The module is self-paced, so you can study the materials when it suits you.
- **Late submissions:** Work submitted after the deadline is penalised.
- **Academic integrity:** All submitted work must be your own. See the Code of Conduct.