Operating Systems

Syllabus

Last update: January 2025

1. General information

Course name	Operating Systems
Code	11310079
Course type	Mandatory
Credits	3
Credit type	A - Theoretical
Direct contact weekly hours	48 hours
Independent weekly hours	96 hours
Prerequisites	None
Co-requisites	None
Schedule	Group 1: Monday and Wednesday, 15:00 – 17:00 Ed. Calatraba, Hipatia Group 2: Tuesday and Thursday, 9:00 – 11:00 Ed. Calatraba, Hipatia

2. Información del profesor y monitor

Nombre del profesor	**Pedro Wightman**
Perfil profesional	Systems Engineer from the Universidad del Norte. Master and Doctor in Computer Science from the University of South Florida. Principal professor and coordinator of international accreditation of the School of Engineering, Science and Technology at Universidad del Rosario. Senior member of IEEE. Research lines: Location-based information

	systems, with emphasis on the area of data privacy, blockchain applications, application of multiple realities in industrial environments, communication infrastructure for the Internet of Things.
Correo electrónico institucional	pedro.wightman@urosario.edu.co
Lugar y horario de atención	Oficina 508 – Edificio Cabal
Página web u otros medios (opcional)	https://pure.urosario.edu.co/es/persons/pedro-mario-wightman-rojas

3. Course summary and purpose

Knowledge of operating systems in terms of their design, operation and configuration are essential for a computer science professional, mainly in two areas: knowledge of the internal processes of operational systems, allows their correct parameterization to benefit the performance of the applications that run on such platform; and knowing the mechanisms of communication and synchronization of processes, allows the development of applications that benefit from these mechanisms for effective execution.

4. Fundamental concepts

This course covers the different mechanisms and techniques that make up the basic functions of modern operating systems. The basic points that will be developed include the study of the operational evolution of computer systems, the process management and coordination; the management of physical and virtual memory; and secondary storage and file systems management mechanisms.

5. Student learning outcomes

At the end of the course, students will be able to:

1. Understand the impact of the basic elements that make up the environment of an operating system, such as its structure, processor registers, instruction cycles, interrupts, memory hierarchy, peripherals, etc.
2. Know the different strategies to solve problems related to process planning, synchronization, and deadlocks, and apply them in test scenarios.
3. Understand the memory addressing process, and its implications in the management of real and virtual memory.

4. Implement the solution to problems related to the management of secondary storage in traditional hard disk and solid-state disks.
5. Implement the solution of practical problems through simple programs in programming languages or within real operating systems.
6. Evaluate the application of current operating systems topics, through the review of related research articles and presentation of their own project.
7. Acquire and apply new knowledge, applying appropriate learning strategies.

6. Coure modality

In-person +

7. Learning strategies

The course learning methodology is designed so that a student can receive the basic knowledge of each of the course modules, put into practice the concepts learned in the development of modules for the UR-OS project during the course, and other software applications that take advantage of the tools that Oss offer developers.

8. Evaluation activities

Evaluation	**%**	**Topics**
First evaluation	20	Process Management module for UR-OS
Second Exam	20	Main memory management module for UR-OS (Allocation and Free-space)
Third Exam	20	Virtual memory management module for UR-OS
Exams	15	Theoretical exams
Class project	25	Linux configuration benchmarking UR-OS challenges

9. Sessions schedule

Week	Topic	Activity
Week 1 July 29 – July 31	1. Introduction to Operational Systems 1.1. History	Lecture classes
Week 2 August 5 August 7 (Holiday)	1.2. Structure 1.3. Services and Features	Lecture classes
Week 3 August 12 (Remote class) August 14 (Remote class)	2. Process Management 2.1. Processes 2.2. Process planning algorithms	Lecture classes
Week 4 August 19 – August 21	2.3. Intro to UR-OS and Process Planning In-class work – UR-OS	Lecture classes
Week 5 August 26 – August 28	2.4. Threads 2.5 Week 8 Sept 15 – Sept 17. Communication between processes First deliverable – UR_OS Process Planning	Demo of threads Demo of sockets Implement Process management in UR-OS – Part 1. Each group develops one algorithm, following the standard
Week 6 Sept 2 – Sept 4	3. Mutual Exclusion, Synchronization and Deadlocks 3.1. Definition of critical area and basic synchronization algorithms 3.2. Classic synchronization algorithms	
Week 7 Sept 9 – Sept 11	Presentation - Second deliverable – UR_OS Process Planning + Presentation	Implement Process management in UR-OS – Part 2.

Week 8 Sept 16 – Sept 18	3.3 Implementation of semaphores 3.3. Definition of Deadlocks Online quiz – Process planning + Synchronization	Demo of semaphores Lecture classes
Week 9 Sept 23 Sept 25 (No class)	3.4. Deadlock prevention algorithms – Banker's Algorithm	Lecture classes
Week 10 Sept 30 – Oct 2	3.5. Algorithms for deadlock detection 3.6. Algorithms for handling deadlocks 4. Main and Virtual Memory Management 4.1. Contiguous memory	Lecture classes
Week 11 Oct 7 Oct 9 (No class)	4.2. Free memory management 4.3. Intro to UR-OS Memory Management Online quiz – Contiguous Memory and Free Memory Management Independent UR-OS work First deliverable – UR_OS Free memory management	Lecture classes Implement Memory management in UR_OS – Part 1
Oct 14 – Oct 16	Rosarista Break week	
Week 12 Oct 21 – Oct 23	4.3. Paging 4.4. Segmentation	Lecture classes
Week 13 Oct 28 – Oct 30	In-class work – UR-OS 4.5. Introduction to the virtual memory model 4.6. Virtual memory management algorithms	

Week 14 Nov 4 – Nov 7	Presentation Second deliverable – UR_OS Free memory management 4.5. Introduction to the virtual memory model 4.6. Virtual memory management algorithms	Implement Main Memory Management in UR-OS – Part 2
Week 15 Nov 11 – Nov 13	4.6. Virtual memory management algorithms exercises Intro to UR-OS Virtual memory management 5. File systems 5.1. Introduction to file systems 5.2. File systems user interface 5.3. File system implementation	Lecture classes
Week 16 Nov 18 – Nov 20	6. Secondary Memory Management 6.1. Introduction to mass storage 6.2. Traditional disk scheduling algorithms 6.3. Solid state drives 6.4. RAID - Protection of data on disks Online quiz – Virtual memory In-class work on UR-OS	Implement Virtual Memory Management in UR-OS
	Project presentations	

10. Success factors

Below are a series of suggested actions that can contribute significantly to achieving goals and consequently foster a successful experience in this course:

1. Plan and organize the individual work time that you will dedicate to the course
2. Organize the site and study materials
3. Have a study group, seek the support of classmates
4. Cultivate discipline and perseverance, work weekly, do not allow topics or work to

accumulate

5. Constantly perform a self-evaluation, determine if the actions carried out are productive or if, on the contrary, strategies should be changed
8. Attend the professor's consultation hours, participate in class, never remain with doubts
9. Use the spaces designated for consultations and resolution of doubts, such as the Gauss Room and the Knuth Room
10. Provide spaces for rest and mental hygiene, try to have good sleeping habits
11. Always keep in mind values such as honesty and sincerity, in the end it is not just about passing an exam, it is about to learn and acquire knowledge. Fraud is self-deception.

11. References and resources

- SILVERSCHATZ, A; PETERSON, J y GALVIN, P. Operating System Concepts. 10th. Ed. Wilie. 2018.
- TANENBAUM, Andrew S y WOODHULL Albert S. Operating Systems Design and Implementation 3rd. Ed. Pearson. 2007.
- GARG, R., & VERMA, G.. Operating Systems: An Introduction. Mercury Learning & Information. 2017.

- OSDev.org. https://wiki.osdev.org/Main_Page
- Raspberry Pi 4 OS Tutorial. https://www.rpi4os.com/
- Java Documentation. https://docs.oracle.com/en/java/
- Operating Systems: Three Easy Pieces. https://pages.cs.wisc.edu/~remzi/OSTEP/
- A Journey in Creating an Operating System Kernel. https://539kernel.com/

12. Agreements for the development of the course

The School of Engineering, Science and Technology does not exempt any student from the final exam, so all students enrolled in its courses must take it. No type of bonus will be used in the course.

13. Non-discrimination statement

If you have a disability, whether visible or not, and require some type of support to be on equal terms with other students, please inform your professor so that reasonable adjustments can be made to the course as soon as possible. Likewise, if you do not have the technological resources required for the development of the course, please inform the Academic Secretary of your program or the Student Office in a timely manner, so that your request can be attended to in time.

Remember that it is the duty of all people to respect the rights of those who are part of the Rosarista community. Any situation of harassment, sexual harassment, discrimination or bullying, whether in person or virtual, is unacceptable. Anyone who feels in any of these situations can report its occurrence by contacting the team of the Coordination of Psychology and Quality of Life of the Dean's Office of the University Environment (Decanatura del Medio Universitario) (Telephone or WhatsApp 322 2485756).