Program	Level	First cycle
	Name of the program	Theoretical Computer Science

COURSE

Course title	**Operating Systems**			
Course code	Semester	Course status	ECTS	Contact hours (L+AE+LE)
CS 280	IV	Mandatory course	4	2+0+2
Lecturer				
Course Goals	The course aims to show the students basic operating system concepts, present high-level understanding of the processes relevant to the operating system and demonstrate how to write system programs that use operating system services.			
Learning Outcomes	Upon successful completion of the course, students will be able to: • Identify the main components of the operating system and describe their functions • Discuss the features of the operating system required for each application. • Understand the different levels of system and application software. • Be familiar with major operating system services, such as file systems, memory management, process management, device management, and the user interface. • Use and customize operating systems			

COURSE CONTENT

- Introduction: Role, functionality and structure of the operating system, historical development of operating systems: batch, multiprogramming, time-sharing
- Computer system structure: interrupts and interrupt management, input-output operations, dual processor mode, system call services.
- Operating system structure: layered operating system structure, monolithic and mirokernel, functional organization of Unix, Linux, Windows operating systems
- Process management: Concept and process states, task swittching, process operations, process representation, threads and thread management, process management in Unix, interprocess communication using taps and signals, via message forwarding: direct, indirect, buffering.
- Shared memory, process synchronization problem, critical section and mutual shutdown, traffic lights and hardware synchronization techniques: test_and_set.
- Processor scheduling: General concepts and scheduling criteria, dispatcher, scheduling algorithms: FCFS, SJF, priority, Round Robin, Multilevel feedback, stochastic algorithms, real-time scheduling algorithms, algorithms for multiprocessor systems, scheduling on UNIX operating system examples and Windows
- Resource management, downtime resolution methods, banking algorithm
- Memory management: Loaders, general concepts of address translation from logical to physical, memory allocation, continuous: with one or more partitions, static and dynamic, and non-continuous: paging and segmentation, virtual memory, memory management in Unix.
- Peripheral Management, I/O Manager, User Interface, Text, Graphics and Network
- File management: File system structures, free space management, file and directory implementation, file systems for Unix and Windows operating systems: logical organization
- Security of operating systems
- • Architecture of DOS, Windows and Linux and Android systems, a look at the source code of the small operating system kernel

LITERATURE

[1] Ribić S, "Operativni sistemi", Univerzitet u Sarajevu, 2019
[2] Tanenbaum A., "Modern Operating Systems", 4th Edition, Prentice Hall, 2014
[3] Đorđević B., Pleskonjić D, Maček N, "Operativni sistemi, teorija, praksa i rešeni zadaci", Mikro knjiga, Beograd 2005
[4] Silberschatz A, Gagne G, Galvin P., "Operating System Conceptrs", 10th Edition, Addison Wesley, 2018.

STUDENT WORKLOAD (hours in a semester)

Lectures	30	Tutorial	30	Individual work	40	T o t a l	100

GRADING			**REMARKS**
Criterion	Maximum points	Minimum points	

Midterm exams	40		
Homework assignment	10		
Project			
Laboratory assignments	10		
Final exam	40		
T o t a l	100	55	