OPERATING SYSTEMS

Faculty of Computer Science				
Study programme:	**Computer Science**		Degree level:	**Engineer's degree full-time programme**
Specialization	**---**		Diploma path:	**2026/2027W - 2026/2027S**
Module name:	**Operating Systems** (Systemy operacyjne)			
Module type:	**obligatory**	**Semester: 2**	ECTS:**4**	Module ID:**FCS-00017**
No. of hrs in semester:	Lecture (L) - **30** Classes(C) - **0** Specialization workshop (SW) - **30** Project (P) - **0** Laboratory classes (LC) - **0** Seminar (S) - **0**			
Prerequisites	Computer Organization and Architecture (FCS-00011), Programming Basics (FCS-00031),			

Aims and objectives:	Skills for solving problems requiring synchronization of threads (or processes). The knowledge of an operating system fundamentals. Skills for programming using an operating system API.		
Forms of teching activities::	lecture, specialization workshop,	Assessment:	Evaluation must be relevant to the intended learning outcomes: The lecture - a written examination, The specialistic workshop - four tests and two software projects.
Module content:	Lecture: Introduction. Elements of a computer system architecture. Processes and threads. Concurrency. Fundamental synchronization problems. Semaphores and monitors. Deadlocks and starvation. CPU and disk scheduling. Memory management and virtual memory. Protection and Security. File systems. Operating systems for parallel and distributed computers. Real-time and multimedia operating systems. Workshop: System programming with Linux API. Concurrent programming with the POSIX Threads standard.		
Teaching methods:	programming, informative lecture,		

Learning outcomes		
Symbol	Specify min. 4, max. 8 learning outcomes in the following order: knowledge – skills – competence. Each learning outcome must be verifiable	Reference to the programme learning outcomes of education
LO1	knows the operating system fundamentals	INF1_W03
LO2	can solve basic synchronization problems using semaphores and monitors	INF1_U05
LO3	can program using an operating system API	INF1_U05
LO4	can experimentally evaluate the efficiency of a computer system	INF1_U06
LO5	can install and configure an operating system	INF1_U05

No. of learning outcome	Methods of assessing the learning outcome	Type of teaching activities (if more than one) during which the outcome is assessed
LO1	Written exam, about 30 questions	L
LO2	Presentation and defense of a software project, 2 tests	Sw
LO3	Presentation and defense of a software project, 2 tests	Sw
LO4	Presentation and defense of a software project, 2 tests	Sw
LO5	Homework	Sw

Student's workload (in hours)	**1** - Listening to lectures	None	30
	2 - Participation in the specialistic workshop	None	30
	3 - Preparation of software projects	None	20
	4 - Preparation for examination	None	10
	5 - Preparation for specialistic workshop	None	10
		TOTAL:	**100**
Quantitative indicators	Student's workload - activities that require direct teacher participation: (1)+(2)	60	**ECTS** 2.4
	Student's workload connected with practical classes (2)+(3)+(5)	60	2.4

Basic references:	1. A. Silberschatz, P. B. Galvin,G. Gagne, Operating Systems Concepts , 2013. 2. A. S. Tanenbaum, Modern Operating Systems, 2013. 3. W. Stallings, Operating Systems: Internals and Design Principles, 2014.
Further reading	1. R. Love, Linux System Programming: Talking Directly to the Kernel and C Library, 2013.

Unit:	Software Department	Lecturer/ instructor	dr hab. inż. Wojciech Kwedlo,dr inż. Daniel Reska

Date of issuing the programme:	31st March 2026	Author of the programme:	dr inż. Daniel Reska

L - lecture, C - classes, LC - laboratory classes, P-project, SW - specialization workshop, S - seminar