GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Computer Architecture, PG_00058927		
Field of study	Informatics		
Date of commencement of studies	October 2024	Academic year of realisation of subject	2025/2026
Education level	first-cycle studies	Subject group	Obligatory subject group in the field of study Subject group related to scientific research in the field of study
Mode of study	Part-time studies	Mode of delivery	at the university
Year of study	2	Language of instruction	Polish
Semester of study	3	ECTS credits	4.0
Learning profile	general academic profile	Assessment form	exam
Conducting unit	Department of Computer Architecture -> Faculty of Electronics, Telecommunications and Informatics		
Name and surname of lecturer (lecturers)	Subject supervisor	dr inż. Marcin Narloch	
	Teachers	dr inż. Marcin Narloch	

Lesson types and methods of instruction	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	15.0	0.0	15.0	0.0	0.0	30
	E-learning hours included: 0.0						

Learning activity and number of study hours	Learning activity	Participation in didactic classes included in study plan	Participation in consultation hours	Self-study	SUM
	Number of study hours	30	4.0	66.0	100

Subject objectives	The aim of the course is to provide knowledge of the concepts related to the computer architecture and knowledge of the basic mechanisms of processors at the ISA level, and to present the latest trends in the construction of the processors.

Learning outcomes	Course outcome	Subject outcome	Method of verification
	[K6_W04] knows and understands, to an advanced extent, the principles, methods and techniques of programming and the principles of computer software development or programming devices or controllers using microprocessors or programmable elements or systems specific to the field of study, and organisation of systems using computers or such devices	Students knows computer architecture, principles of operating memory management and addressing techniques, program assembly technique, operation of interrupt systems and cooperation with external systems.	[SW1] Assessment of factual knowledge
	[K6_U09] can carry out a critical analysis of the functioning of existing technical solutions and assess these solutions, as well as apply experience related to the maintenance of technical systems, devices and facilities typical for the field of studies, gained in the professional engineering environment	Student understands and integrates low-level system subprograms requiring the use of different standards, communication methods or character coding.	[SU1] Assessment of task fulfilment
	[K6_U04] can apply knowledge of programming methods and techniques as well as select and apply appropriate programming methods and tools in computer software development or programming devices or controllers using microprocessors or programmable elements or systems specific to the field of study	The student is able to integrate software modules in high and low level languages.	[SU1] Assessment of task fulfilment
	[K6_U41] can produce, test or evaluate software using modern programming platforms, tools, languages and paradigms of different levels, as well as use software packages supporting scientific and research processes as well as business decision-making processes and teamwork	Student develops software at the processor level, runs and tests own programs.	[SU1] Assessment of task fulfilment
Subject contents	1. Introduction and principles of course completion, literature. 2. General organization of the computer according to von Neumann. 3. x86 architecture elements at the register level: general purpose registers, tags, real mode and protected mode. 4. Physical and virtual memory, addressing memory, the problem of bytes order (little/big endian). 5. Development of the x86 architecture. 6. Processor software model, command cycle, instruction classification according to how it affects the instruction pointer (command counter). 7. Rules of programming at the level of processor commands, functions of typical commands. 8. Addressing modes. 9. Programming elements in assembler: instruction mnemonics, source line formats, variables and labels, directives. 10. Stack operations: call and return from subprogram (ret), transfer of parameters to subprogram. 11. Hardware support for stack-based parameter transfer, stack frame. 12. Mixed programming, ABI interface, typical function call standards (Pascal, C, StdCall). 13 System services and their calling, API interface, interrupt descriptor board in x86 architecture. 14. Data types and formats: binary numbers with and without a character. 15. Text encoding: ASCII, Windows, ISO, Unicode, UTF-8, UTF-16. 16. Arithmetic operations, identification of excess. 17. Single bit operations, logical and cyclic shifts (rotations). 18. Basic concepts for controlling external devices Controlling devices via co-addressable memory area or I/O address space. 19. Hardware interrupts and their operation, interrupt priorities, masked and unmasked interrupts. 20. Elements of hardware interrupt handling technology in PCs, mapping of interrupt lines into elements of interrupt descriptor boards. 21. CPU exceptions, hardware and software interrupts. 22. Data transmission by DMA. 23. Floating point number formats (IEEE 754 standard). 24. Arithmetic coprocessor, calculation examples. 25. Selection of calculation options, state register and coprocessor control register. 26. Computer memory hierarchy: registers, cache, main (operating) memory, mass memory. 27. The concept of virtual memory as an integration of operating and disk memory. 28. Potential processing, control conflicts, prediction of jumps. 29. CISC and RISC computers. 30. Multi-threaded and multi-core architectures, instructions for mutimedia applications (MMX, SSE)		
Prerequisites and co-requisites			

Assessment methods and criteria	Subject passing criteria	Passing threshold	Percentage of the final grade
	Exam	0.0%	50.0%
	Practical exercises	0.0%	50.0%
Recommended reading	Basic literature	Null L., Lobur J.: Struktura organizacyjna i architektura systemów komputerowych. Wyd. Helion 2004. Tanenbaum A.S.: Strukturalna organizacja systemów komputerowych, wyd. Helion Lewis D.: Między asemblerem a językiem C, wyd. RM John L. Hennessy, David A. Patterson, "Computer Architecture: A Quantitative Approach", 2014 Morgan Kaufmann	
	Supplementary literature	No requirements	
	eResources addresses	Adresy na platformie eNauczanie:	
Example issues/ example questions/ tasks being completed			
Work placement	Not applicable		