GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Assembler Programming, PG_00047620		
Field of study	Automatic Control, Cybernetics and Robotics		
Date of commencement of studies	October 2024	Academic year of realisation of subject	2026/2027
Education level	first-cycle studies	Subject group	Optional subject group Subject group related to scientific research in the field of study
Mode of study	Full-time studies	Mode of delivery	at the university
Year of study	3	Language of instruction	Polish
Semester of study	6	ECTS credits	2.0
Learning profile	general academic profile	Assessment form	assessment
Conducting unit	Department of Automatic Control -> Faculty of Electronics, Telecommunications and Informatics		
Name and surname of lecturer (lecturers)	Subject supervisor	dr inż. Paweł Raczyński	
	Teachers	dr inż. Paweł Raczyński	

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	2.0	18.0	50

Subject objectives

The main aim of the subject is to know basic techniques for assembly programming

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	The student knows the programming principles in various programming languages, including low-level programming. Knows the construction and organization of typical computer interfaces and peripherals. He knows the rules of their program support. Knows the principles of cooperation between user programs and system software.	[SW1] Assessment of factual knowledge
[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	Student describes and put to use in practice the syntax of assembly language. Student describes and put to use in practice assembly language compilers and linkers. Student describes and put to use in practice assembler techniques in using of operational and disk memories. Student describes and knows how to use it in practice a list of commands and addressing modes of the x86 family of processors. Student describes and knows how to combine techniques used in assembly language programs both written in the languages C and Pascal. Student describes and put to use in practice PC computers and BIOS programming module.	[SW3] Assessment of knowledge contained in written work and projects [SW1] Assessment of factual knowledge

<table>
<tr><td>Subject contents</td><td colspan="3">1. Simple assembler, syntax, key words 2. Name dictionary, operations on the name dictionary 3. Simple assembler directives 4. Assembly process, example 5. Macro assembler, condition assembly 6. Macroinstructions and subroutines 7. Name attributes, multi file projects 8. Linking process, example 9. Architecture x86 and its effect on assembler programming techniques 10. Memory addressing modes and its assembler implementation 11. Input and output service in assembler 12. Interrupt service routines in assembler 13. Programming techniques in assembler –TASM and MASM compilers 14. Memory models and their effect on program construction techniques 15. Static memory and dynamic memory allocation techniques 16. Effective data transfer commands 17. Stack, buffers, cyclic buffers – organization and application 18. Arithmetic commands, number formats, long number calculations 19. Logical commands, flags and its applications 20. Jump commands and its applications in different memory models 21. Functions and procedures, parameter exchange techniques 22. Interface into high level languages (C, C++, PASCAL) 23. File system and file handling techniques 24. Screen monitor, and screen service techniques 25. Basic of graphics programming techniques in assembler 26. Graphic files, basic of data compression techniques 27. Keyboard service techniques 28. BIOS, structure and application 29. Operating system, functions and services 30. Debugger idea and applications 31. Selected algorithms and its assembler implementations</td></tr>
<tr><td>Prerequisites and co-requisites</td><td colspan="3">No requirements</td></tr>
<tr><td>Assessment methods and criteria</td><td colspan="3"><table>
<tr><th>Subject passing criteria</th><th>Passing threshold</th><th>Percentage of the final grade</th></tr>
<tr><td>Practical exercise</td><td>51.0%</td><td>60.0%</td></tr>
<tr><td>Midterm colloquium</td><td>51.0%</td><td>40.0%</td></tr>
</table></td></tr>
<tr><td rowspan="3">Recommended reading</td><td>Basic literature</td><td colspan="2">A. Pyrchla, B. Danowski, BIOS. Przewodnik, Helion 2007 E. Wróbel, Asembler Praktyczny kurs asemblera, Helion 2004 J. Hollingworth, D. Buttrtfield, B. Swart, J. Allsop, C++ Builder 5 vademecum profesjonalisty tom 1 i 2, Helion 2001 K. R. Irvine, Asembler dla procesorów Intel vademecum profesjonalisty, Helion 2003 S. Kruk, Turbo asembler idee, polecenia, rozkazy procesora Pentium, Mikom 2000, Zasoby Internetu</td></tr>
<tr><td>Supplementary literature</td><td colspan="2">No requirements</td></tr>
<tr><td>eResources addresses</td><td colspan="2">Adresy na platformie eNauczanie:</td></tr>
<tr><td>Example issues/ example questions/ tasks being completed</td><td colspan="3"></td></tr>
<tr><td>Work placement</td><td colspan="3">Not applicable</td></tr>
</table>