University of Gdańsk

Subject card

Subject name and code	Concurrent Programming, PG_00179347		
Field of study	Informatics		
Date of commencement of studies	October 2023	Academic year of realisation of subject	2025/2026
Education level	Bachelor's studies	Subject group	Obligatory subject group in the field of study Optional subject group
Mode of study	full-time studies	Mode of delivery	at the university
Year of study	3	Language of instruction	Polish
Semester of study	5	ECTS credits	5.0
Learning profile	practical	Assessment form	exam
Conducting unit			
Name and surname of lecturer (lecturers)	Subject supervisor	dr Paweł Pączkowski	
	Teachers		

Lesson types

Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
Number of study hours	30.0	0.0	30.0	0.0	0.0	60

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	60	0.0	0.0	60

Subject objectives

Familiarizing students with the design and creation of multiprocess and multithreaded programs, as well as analyzing their behaviour.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[INFL3_W03] has ordered knowledge in software engineering and project management methodologies, IT project life cycle, software specification, validation and verification, design patterns	The student knows methods of creation of concurrent processes and threads as well as selected methods of their communication (pipes, IPC, sockets) and synchronization (semaphores, mutexes).	[SW4] test/exam - oral or written
[INFL3_U08] assesses the usefulness of various paradigms and related programming tools to solve various types of problems	Using the documentation, the student is able to use the mechanisms for creating processes and concurrent threads in programs, as well as methods for their communication (pipes, IPC, sockets) and synchronization (semaphores, mutexes). The student can explain the communication and synchronization mechanisms used in the programs.	[SU5] implementation of a problem task
[INFL3_W04] has ordered, theoretically founded knowledge in programming, algorithms and complexity, programming languages and paradigms	The student knows the classical problems of concurrent programming (mutual exclusion, producer-consumer, readers and writers problems, the dining philosophers problem). The student knows the concepts of deadlock and starvation.	[SW4] test/exam - oral or written

<table>
<tr><td>Subject contents</td><td>

- Classical problems of concurrent programming: mutual exclusion, producer-consumer, the dining philosophers, readers and writers.
- Synchronization of processes, thhe notions of deadlock and starvation.
- Inference about brhaviour of concurrent processes.
- Communication and synchronization mechanisms for processes: pipes, queues, semaphores, shared memory, sockets.
- Creation of concurrent threads and mechanisms of coordination thereof.
- High-level mechanisms for concurrent programming in an abstract perspective - monitors and in functioning libraries (e.g., Python multiprocessing).
- Terminology in the English language.

</td></tr>
<tr><td>Prerequisites and co-requisites</td><td>Programming skills in Python or C/C++</td></tr>
<tr><td>Assessment methods and criteria</td><td>
<table>
<tr><th>Subject passing criteria</th><th>Passing threshold</th><th>Percentage of the final grade</th></tr>
<tr><td>written and discussed computer programs</td><td>51.0%</td><td>50.0%</td></tr>
<tr><td>written exam</td><td>51.0%</td><td>50.0%</td></tr>
</table>
</td></tr>
<tr><td>Recommended reading</td><td>
<table>
<tr><td>Basic literature</td><td>1. M. Ben-Ari, Principles of Concurrent and Distributed Programming, Pearson Education, 2006.
2. J.S. Gray, Interprocess communications in Linux, Prentice Hall, 2003.
3. Z.J. Czech, Wprowadzenie do obliczeń równoległych, Wydawnictwo Naukowe PWN, 2013.</td></tr>
<tr><td>Supplementary literature</td><td>1. K. Barteczko, JAVA Uniwersalne techniki programowania, PWN, 2015.</td></tr>
<tr><td>eResources addresses</td><td>Basic
https://docs.python.org/3/library/ - Python Library Documentation
Supplementary
https://www.man7.org/linux/man-pages/ - Linux manual</td></tr>
</table>
</td></tr>
<tr><td>Example issues/ example questions/ tasks being completed</td><td>

- Explain the concepts of deadlock and starvation in concurrent programming.
- Using sockets for communication, write and test client-server programs that allow playing the game rock-paper-scissors.

</td></tr>
<tr><td>Work placement</td><td>Not applicable</td></tr>
</table>

Document generated electronically. Does not require a seal or signature.