



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez  
Unię Europejską w ramach  
Europejskiego Funduszu  
Społecznego

**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



<b>Nazwa przedmiotu</b>		<b>Kod ECTS</b>	
Programowanie współbieżne		11.3.0835	
<b>Nazwa jednostki prowadzącej przedmiot</b>			
Instytut Informatyki			
<b>Studia</b>			
<b>wydział</b>	<b>kierunek</b>	<b>poziom</b>	<b>pierwszego stopnia</b>
Wydział Matematyki, Fizyki i Informatyki	Informatyka	<b>forma</b>	stacjonarne
		<b>moduł</b>	wszystkie
		<b>specjalnościowy</b>	wszystkie
		<b>specjalizacja</b>	wszystkie
<b>Nazwisko osoby prowadzącej (osób prowadzących)</b>			
dr inż. Jerzy Skurczyński			
<b>Formy zajęć, sposób ich realizacji i przypisana im liczba godzin</b>		<b>Liczba punktów ECTS</b>	
<b>Formy zajęć</b>		5 Przedmiot w wymiarze 30h wykładu i 30h lab. + praca własna studenta	
Wykład, Ćw. laboratoryjne			
<b>Sposób realizacji zajęć</b>			
zajęcia w sali dydaktycznej			
<b>Liczba godzin</b>			
Wykład: 30 godz., Ćw. laboratoryjne: 30 godz.			
<b>Cykl dydaktyczny</b>			
2018/2019 letni			
<b>Status przedmiotu</b>		<b>Język wykładowy</b>	
fakultatywny (do wyboru)		polski	
<b>Metody dydaktyczne</b>		<b>Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne</b>	
<ul style="list-style-type: none"> <li>- Projektowanie doświadczeń</li> <li>- Wykonywanie doświadczeń</li> <li>- Wykład z prezentacją multimedialną</li> </ul>		<b>Sposób zaliczenia</b>	
		<ul style="list-style-type: none"> <li>- Zaliczenie na ocenę</li> <li>- Egzamin</li> </ul>	
		<b>Formy zaliczenia</b>	
		<ul style="list-style-type: none"> <li>- egzamin pisemny testowy</li> <li>- ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru</li> </ul>	
		<b>Podstawowe kryteria oceny</b>	
		Wykład - liczba punktów uzyskanych na egzaminie. Laboratorium - liczba punktów uzyskanych za napisane i zaliczone programy oraz ocena za projekt końcowy.	
<b>Sposób weryfikacji założonych efektów kształcenia</b>			
<b>Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi</b>			
<b>A. Wymagania formalne</b>			
Zaliczone przedmioty: 1) Systemy operacyjne; 2) Sieci komputerowe.			
<b>B. Wymagania wstępne</b>			
Umiejętność programowania w języku C pod systemem Linux oraz tworzenia skryptów w bash'u.			
<b>Cele kształcenia</b>			
Nauczenie studentów projektowania i tworzenia prawidłowo zbudowanych programów wieloprotocowych i wielowątkowych oraz analizy ich działania przy użyciu narzędzi teoretycznych.			
<b>Treści programowe</b>			

1. Modele teoretyczne zbiorów procesów współbieżnych.
2. Rodzaje komunikacji pomiędzy procesami.
3. Synchronizacja, unikanie blokady i głodzenia procesów.
4. Logika temporalna jako narzędzie dowodzenia poprawności programów współbieżnych.
5. Mechanizmy komunikacji i synchronizacji procesów w systemie Unix / Linux: kolejki, semafony, pamięć dzielona.
6. Programowanie rozproszone przy użyciu gniazd.
7. Mechanizmy koordynacji wątków.

**Wykaz literatury**

1. M. Ben-Ari, Podstawy programowania współbieżnego i rozproszonego, WNT, 1996.
2. J.S. Gray, Komunikacja między procesami w UNIXie, RM, 1998.

**Efekty kształcenia**

**(obszarowe i kierunkowe)**

Student rozumie różnicę pomiędzy programowaniem sekwencyjnym a współbieżnym, zna kryteria oceny poprawności programu współbieżnego i umie je zastosować. Zdaje sobie sprawę z niskiej użyteczności testowania działania programów współbieżnych i docenia rolę formalnego dowodzenia ich poprawności.

KW\_02

ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną

w zakresie programowania, algorytmów i złożoności, architektury systemów komputerowych, systemów operacyjnych, technologii sieciowych, języków i paradygmatów programowania, baz danych, inżynierii oprogramowania, języków formalnych

KW\_o6

zna zasady działania systemów operacyjnych ze szczególnym uwzględnieniem procesów, współbieżności, szeregowania zadań i zarządzania pamięcią

KU\_07

rozumie niskopoziomowe zasady wykonywania programów

KU\_10

rozumie mechanizmy synchronizacji programów

współbieżnych

KK\_04

rozumie i docenia znaczenie uczciwości intelektualnej w działaniach

własnych i innych osób; postępuje etycznie

KK\_05

potrafi samodzielnie wyszukiwać informacje w literaturze,

także w językach

obcych

**Wiedza**

Student zna wybrane narzędzia programistyczne służące do tworzenia programów współbieżnych oraz wybrane narzędzia teoretyczne służące do dowodzenia ich poprawności.

**Umiejętności**

Student potrafi zaprojektować i zbudować nieduży program współbieżny przy użyciu narzędzi dostępnych w systemie Linux oraz udowodnić jego poprawność przy użyciu logiki temporalnej.

**Kompetencje społeczne (postawy)**

Student jest w stanie współpracować z zespołem programistów tworzących wspólnie większy program współbieżny, a w razie potrzeby pełnić funkcję kierownika takiego zespołu.

**Kontakt**

jsk@inf.ug.edu.pl