

Karta przedmiotu

Nazwa i kod przedmiotu	Programowanie współbieżne i równoległe, PG_00203635						
Kierunek studiów	Informatyka (O)						
Data rozpoczęcia studiów	październik 2026 r.		Rok akademicki realizacji przedmiotu		2026/2027		
Poziom kształcenia	II stopnia		Grupa zajęć		Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	niestacjonarne		Sposób realizacji		na uczelni		
Rok studiów	1		Język wykładowy		polski		
Semestr studiów	2		Liczba punktów ECTS		5.0		
Profil kształcenia	ogólnoakademicki		Forma zaliczenia		zaliczenie		
Jednostka prowadząca							
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr Piotr Arłukowicz				
	Prowadzący zajęcia z przedmiotu		dr Piotr Arłukowicz				
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	20.0	0.0	20.0	0.0	0.0	40
	W tym liczba godzin zajęć na odległość: 0.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów		Udział w konsultacjach		Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	40		0.0		85.0	125
Cel przedmiotu	Zapoznanie studentów z modelami programowania współbieżnego i równoległego, mechanizmami synchronizacji oraz technikami projektowania wydajnych i skalowalnych programów wielowątkowych i rozproszonych z wykorzystaniem modelu pamięci współdzielonej (OpenMP) oraz modelu przesyłania komunikatów (MPI).						
Efekty uczenia się przedmiotu	Efekt kierunkowy		Efekt z przedmiotu		Sposób weryfikacji i oceny efektu		
	[INFMU2_W03] ma pogłębioną wiedzę na temat paradygmatów i architektur programowania oraz zaawansowanych konstrukcji programistycznych; zna aktualne trendy w językach programowania		Student ma pogłębioną wiedzę o konstrukcjach programowania współbieżnego i równoległego (wątki i procesy, mechanizmy synchronizacji, modele OpenMP i MPI), o wzorcach przetwarzania równoległego oraz o aktualnych trendach w językach i narzędziach wspierających współbieżność.		[SW4] test/egzamin - ustny lub pisemny [SW2] prezentacja/projekt/referat/ raport		
	[INFMU2_W04] zna i rozumie w pogłębionym stopniu zasady projektowania i implementacji złożonych systemów oprogramowania, wpływ architektury systemu na jego współbieżność, wydajność i skalowalność, modele przetwarzania współbieżnego i równoległego oraz ich zastosowanie w nowoczesnych systemach		Student zna i rozumie modele przetwarzania współbieżnego i równoległego (pamięć współdzielona oraz przesyłanie komunikatów), wpływ architektury sprzętowej i programowej na współbieżność, wydajność i skalowalność, a także prawa skalowalności (prawo Amdahla, prawo Gustafsona).		[SW4] test/egzamin - ustny lub pisemny [SW2] prezentacja/projekt/referat/ raport		

Treści przedmiotu	1. Współbieżność a równoległość — procesy i wątki, modele wykonania, architektury wielordzeniowe i rozproszone. 2. Problemy współbieżności — wyścigi (race conditions), zakleszczenie (deadlock), zagłodzenie, warunki Coffmana. 3. Mechanizmy synchronizacji — sekcje krytyczne, muteksy, semaforey, monitory, zmienne warunkowe, bariery. 4. Klasyczne problemy współbieżności — producent-konsument, czytelnicy-pisarze, uczący filozofowie. 5. Model pamięci współdzielonej — programowanie wielowątkowe, dyrektywy OpenMP (równoległe pętle, sekcje, redukcje, zakres zmiennych). 6. Model przesyłania komunikatów — programowanie z użyciem MPI (komunikacja punkt-punkt i grupowa, komunikatory, synchronizacja procesów). 7. Wzorce programowania równoległego — fork-join, map-reduce, pule wątków, równoległość danych i zadań. 8. Wydajność i skalowalność — przyspieszenie, efektywność, prawo Amdahla i Gustafsona, równoważenie obciążenia. 9. Elementy programowania równoległego na akceleratorach (GPU) — wprowadzenie. 10. Poprawność programów współbieżnych — testowanie i debugowanie, narzędzia do wykrywania błędów współbieżności.		
Wymagania wstępne i dodatkowe			
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	projekt	51.0%	60.0%
	kolokwium	51.0%	40.0%
Zalecana lista lektur	Podstawowa lista lektur	• M. Ben-Ari, <i>Podstawy programowania współbieżnego i rozproszonego</i>, WNT. • Z.J. Czech, <i>Wprowadzenie do obliczeń równoległych</i>, PWN. • P. Pacheco, <i>An Introduction to Parallel Programming</i>, Morgan Kaufmann.	
	Uzupełniająca lista lektur	• B. Wilkinson, M. Allen, <i>Parallel Programming. Techniques and Applications Using Networked Workstations and Parallel Computers</i>, Pearson. • A. Silberschatz, P.B. Galvin, G. Gagne, <i>Podstawy systemów operacyjnych</i>, WNT. • Dokumentacja standardów OpenMP (https://www.openmp.org/) oraz MPI (https://www.mpi-forum.org/).	
	Adresy eZasobów		
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania			
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy		

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.