



Karta przedmiotu

Nazwa i kod przedmiotu	Parallel Programming for Multi-Core Architectures, PG_00064510						
Kierunek studiów	Informatyka (studia w jęz. angielskim)						
Data rozpoczęcia studiów	luty 2027 r.	Rok akademicki realizacji przedmiotu			2027/2028		
Poziom kształcenia	II stopnia	Grupa zajęć			Grupa zajęć fakultatywnych Grupa zajęć specjalnościowych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	1	Język wykładowy			angielski		
Semestr studiów	2	Liczba punktów ECTS			4.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			egzamin		
Jednostka prowadząca	Wydziały Politechniki Gdańskiej -> Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Architektury Systemów Komputerowych						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr hab. inż. Paweł Czarnul				
	Prowadzący zajęcia z przedmiotu		dr hab. inż. Zdzisław Czarnul				
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	30.0	0.0	15.0	15.0	0.0	60
	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	60		9.0		31.0	100
Cel przedmiotu	poznanie technik programowania równoległego i API służących do wykorzystania nowoczesnych platform wielordzeniowych						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K7_U04] potrafi wykorzystywać posiadaną wiedzę z zakresu metod i technik programowania oraz dobrać i zastosować właściwe metody i narzędzia programistyczne w tworzeniu oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, charakterystycznych dla danego kierunku studiów, dokonując oceny i krytycznej analizy wykonanego oprogramowania, a także syntezy i twórczej interpretacji prezentowanych za jego pomocą informacji	student potrafi dobrać właściwe API i metody optymalizacji aplikacji na systemach wielordzeniowych	[SU1] Ocena realizacji zadania
	[K7_W04] zna i rozumie w pogłębionym stopniu zasady, metody i techniki programowania oraz zasady tworzenia oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo inne elementy lub układy programowalne, specyficznych dla kierunku studiów, a także organizację pracy systemów wykorzystujących komputery lub te urządzenia	student zna podstawowe zasady i techniki programowania wielowątkowego na architektury wielordzeniowe	[SW1] Ocena wiedzy faktograficznej
	[K7_U12] potrafi w pogłębionym stopniu analizować działanie elementów, układów i systemów związanych z kierunkiem studiów oraz mierzyć ich parametry i badać charakterystyki techniczne, a także planować i przeprowadzać eksperymenty związane z kierunkiem studiów, w tym symulacje komputerowe, interpretować uzyskane wyniki i wyciągać wnioski	potrafi analizować wykonanie i profilować wykonanie aplikacji równoległej	[SU1] Ocena realizacji zadania
	[K7_W10] zna i rozumie w pogłębionym stopniu podstawowe procesy zachodzące w cyklu życia urządzeń, obiektów i systemów technicznych oraz metody wspomagania procesów i funkcji, specyficzne dla kierunku studiów	zna procesy i zależności dotyczące wykonania aplikacji wielowątkowych w systemach wielordzeniowych	[SW1] Ocena wiedzy faktograficznej
Treści przedmiotu	<p>Treści przedmiotu - wykład</p> <ol style="list-style-type: none"> 1. Podanie zasad zaliczenia 2. Aktualne systemy HPC 3. Cele programowania równoległego 4. GPU jako równoległy komputer 5. Architektura GPU 6. Dekompozycja danych 7. Algorytmy zrównoleglania operacji na danych 8. Model programowania CUDA 9. Architektura GPU 10. Wątki w CUDA 11. Dostęp do pamięci w CUDA 12. Optymalizacje z wykorzystaniem NVIDIA CUDA. 13. Obsługa wielu urządzeń GPU. 14. Debugowanie aplikacji. 15. Unified Memory 16. OpenCL dla GPU/CPU 17. Procesory wielordzeniowe 18. Architektury multi i manycore 19. OpenMP 20. Optymalizacja (równoważenie obciążenia, synchronizacji) 21. Modele zrównoleglania różnych pradygmatów w OpenMP. 22. Wektoryzacja. 23. False sharing 24. Thread affinity 25. Optymalizacja divide-and-conquer 26. Optymalizacja liczenie podobieństwa wektorów 27. Programowanie CPU+GPU 		

Wymagania wstępne i dodatkowe	podstawowa umiejętność programowania równoległego		
	znajomość języka C		
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	laboratorium	50.0%	25.0%
	projekt	50.0%	25.0%
	egzamin	50.0%	30.0%
	kolokwium 1 + 2	50.0%	20.0%
Zalecana lista lektur	Podstawowa lista lektur	[1] Paweł Czarnul. Parallel Programming for Modern High Performance Computing Systems. Taylor & Francis. 2018 ISBN 9781138305953 [2] CUDA C programming guide. NVIDIA [3] OpenMP specification [4] OpenCL specification	
	Uzupełniająca lista lektur	Dokumentacja CUDA - prezentacje NVIDIA	
	Adresy eZasobów		
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania			
Zajęcia praktyczne w ramach przedmiotu	Nie dotyczy		

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.