

(pieczęć wydziału)

KARTA PRZEDMIOTU

Z1-PU7

WYDANIE N1

Strona 1 z 4

1. Nazwa przedmiotu: OBLICZENIA RÓWNOLEGŁE 2	2. Kod przedmiotu: OR2
3. Karta przedmiotu ważna od roku akademickiego: 2015/2016	
4. Forma kształcenia: studia drugiego stopnia	
5. Forma studiów: studia niestacjonarne zaoczne	
6. Kierunek studiów: INFORMATYKA (RAU)	
7. Profil studiów: ogólnoakademicki	
8. Specjalność: OPROGRAMOWANIE SYSTEMOWE	
9. Semestr: II	
10. Jednostka prowadząca przedmiot: Wydział Automatyki, Elektroniki i Informatyki	
11. Prowadzący przedmiot: dr inż. Jacek Widuch	
12. Przynależność do grupy przedmiotów: przedmioty specjalnościowe	
13. Status przedmiotu: obowiązkowy	
14. Język prowadzenia zajęć: polski	
15. Przedmioty wprowadzające oraz wymagania wstępne: Przedmioty wprowadzające: Analiza matematyczna i algebra liniowa, Programowanie komputerów, Algorytmy i struktury danych, Architektura komputerów, Obliczenia równoległe	
16. Cel przedmiotu: Celem jest wprowadzenie słuchacza w zaawansowane zagadnienia obliczeń równoległych. W ramach wykładów omawiane są zasady projektowania algorytmów równoległych, z uwzględnieniem własności bezpieczeństwa i żywotności. Omawiane są własności wzajemnego wykluczania i wzajemnej blokady związane z własnością bezpieczeństwa oraz problem zagłócenia i własność uczciwości, związane z żywotnością. Dokonywana jest analiza wybranych algorytmów, mająca na celu określenie własności bezpieczeństwa i żywotności. Prezentowane są także podstawowe informacje dotyczące programowania przy użyciu biblioteki MPI oraz OpenMP. Omawiana jest także architektura CUDA oraz jej użycie w obliczeniach równoległych. W ramach ćwiczeń dokonywana jest analiza wybranych algorytmów, mająca na celu określenie własności bezpieczeństwa i żywotności. Omawiane są zasady komunikacji i synchronizacji procesów w modelu z pamięcią wspólną i rozproszoną. Dla modelu z pamięcią wspólną omawiana jest synchronizacja i komunikacja z użyciem semaforów i monitorów. Słuchacze ćwiczą projektowanie algorytmów równoległych dla wymienionych modeli. Celem laboratorium jest zapoznanie słuchacza z zaawansowanymi zagadnieniami obliczeń równoległych oraz ich praktyczną implementacją na przykładzie wybranych bibliotek i architektur. Słuchacze poznają metody zarządzania wątkami, a także ich synchronizację i komunikację z użyciem interfejsu OpenMP. Omawiane jest także zastosowanie procesorów GPU architektury CUDA w obliczeniach równoległych. Na przykładzie biblioteki MPI słuchacze poznają także komunikację i synchronizację procesów w modelu z pamięcią rozproszoną. Podczas laboratorium student nabywa umiejętności projektowania, oceny i analizy algorytmów w modelu obliczeń równoległych z pamięcią wspólną i rozproszoną, a także użycia bibliotek programistycznych stosowanych do tego celu.	

17. Efekty kształcenia:¹				
Nr	Opis efektu kształcenia	Metoda sprawdzenia efektu kształcenia	Forma prowadzenia zajęć	Odniesienie do efektów dla kierunku studiów
1	Zawansowana wiedza o podstawowych algorytmach równoległych	EP, SP	WM, C	K2A_W12
2	Znajomość zasad korzystania z bibliotek wspierających obliczenia równoległe	EP, CL	WM, L	K2A_W12, K2A_W13
3	Umiejętność wykorzystania biblioteki do zarządzania wątkami	EP, CL	WM, L	K2A_U10, K2A_U19
4	Umiejętność użycia obiektów synchronizacji w obliczeniach równoległych w modelu ze wspólną pamięcią	EP, CL, SP	WM, C, L	K2A_U10, K2A_U19
5	Umiejętność rozwiązywania zadań inżynierskich z użyciem modelu obliczeń równoległych z pamięcią wspólną i rozproszoną	EP, CL	WM, L	K2A_U10, K2A_U15, K2A_U20
18. Formy zajęć dydaktycznych i ich wymiar (liczba godzin)				
W. 15 Ćw. 15 L. 15 P. - Sem. -				
19. Treści kształcenia:				
<u>Treść wykładów:</u>				
W ramach wykładu prezentowane są następujące zagadnienia:				
1. Projektowanie algorytmów równoległych, w szczególności dekompozycja funkcyjna i dekompozycja danych 2. Omówienie zasad korzystania z biblioteki MPI (komunikacja pomiędzy procesami, komunikacja kolektywna, funkcje redukcji, typy pochodne, komunikatory). 3. Omówienie podstaw programowania z użyciem standardu OpenMP (równoległość instrukcji iteracyjnych, równoległe regiony, synchronizacja, efektywność) 4. Omówienie architektury CUDA oraz przeprowadzenia obliczeń równoległych z jej użyciem.				
<u>Tematy ćwiczeń:</u>				
1. Projektowanie algorytmów równoległych. 2. Komunikacja i synchronizacja procesów współbieżnych z użyciem pamięci współdzielonej i semaforów. 3. Komunikacja i synchronizacja procesów współbieżnych z użyciem monitorów. 4. Komunikacja i synchronizacja procesów współbieżnych za pomocą przesyłania komunikatów.				
<u>Tematy ćwiczeń laboratoryjnych:</u>				
1. Interfejs OpenMP. 2. Biblioteka MPI. 3. Obliczenia równoległe w architekturze CUDA.				
20. Egzamin: tak				
21. Literatura podstawowa:				
1. R. Chandra, R. Menon, L. Dagum, D. Kohr, D. Maydan, J. McDonald: „Parallel programming In				

¹ należy wskazać ok. 5 – 8 efektów kształcenia

OpenMP”, Morgan Kaufmann, 2001. 2. B. Chapman, G. Jost, R. van der Pas: „Using OpenMP”. MIT Press, 2008. 3. P.S. Pacheco: „Parallel programming with MPI”, Morgan Kaufman, 1997. 4. Z. Czech: „Wprowadzenie do obliczeń równoległych”. Wydawnictwo Naukowe PWN, Warszawa 2010. 5. J. Sanders, E. Kandrot: „CUDA w przykładach. Wprowadzenie do ogólnego programowania procesorów GPU”. Wydawnictwo Helion, Gliwice 2012. 6. T. Rauber, G. Rünger: „Parallel Programming for Multicore and Cluster Systems”. Springer, 1st edition (March 10, 2010). 7. D.B. Kirk, W.W. Hwu: „Programming Massively Parallel Processors: A Hands-on Approach”. Morgan Kaufmann, 1st edition (February 5, 2010). 8. S. Meyers: „Effective Modern C++ 42 Specific Ways to Improve Your Use of C++11 and C++14”. O'Reilly Media, 2014.		
22. Literatura uzupełniająca:		
1. Praca zbiorowa pod red. Z. Czecha, Systemy operacyjne i języki obliczeń równoległych, skrypt Pol. Śl. Nr 2121, Gliwice, 1998. 2. M. Ben-Ari: „Podstawy programowania współbieżnego”, WNT, Warszawa 1996. 3. Z. Weiss, T. Gruzlewski: „Programowanie współbieżne i rozproszone w przykładach i zadaniach”, WNT, Warszawa 1993. 4. R.L. Graham, D.E. Knuth, O. Patashnik: „Matematyka Konkretna”, PWN, Warszawa, 1996. 5. Praca zbiorowa pod red. Z. Czecha, Programowanie współbieżne. Wybrane zagadnienia, skrypt Pol. Śl. nr 2191, Gliwice 1999, wyd. IV. 6. G. Coulouris, J. Dollimore, T. Kindberg: „Systemy rozproszone, podstawy i projektowanie”, WNT, Warszawa 1998. 7. B. Eckel: „Thinking in Java. Edycja polska”. Wydanie IV. Wydawnictwo Helion, Gliwice 2006.		
23. Nakład pracy studenta potrzebny do osiągnięcia efektów kształcenia		
Lp.	Forma zajęć	Liczba godzin kontaktowych / pracy studenta
1	Wykład	15 / 15
2	Ćwiczenia	15 / 15
3	Laboratorium	15 / 15
4	Projekt	- / -
5	Seminarium	- / -
6	Inne (egzamin)	- / 30
	Suma godzin	45 / 75
24. Suma wszystkich godzin: 120		
25. Liczba punktów ECTS:² 4		
26. Liczba punktów ECTS uzyskanych na zajęciach z bezpośrednim udziałem nauczyciela akademickiego: 2		
27. Liczba punktów ECTS uzyskanych na zajęciach o charakterze praktycznym (laboratoria, projekty): 2		
26. Uwagi: -		

² 1 punkt ECTS – 30 godzin.

Zatwierdzono:

.....
(data i podpis prowadzącego)

.....
(data i podpis dyrektora instytutu/kierownika katedry/
Dyrektora Kolegium Języków Obcych/kierownika lub
dyrektora jednostki międzywydziałowej)