

### Karta modułu/przedmiotu

Wypełnia Zespól Kierunku	Nazwa modułu (bloku przedmiotów): <b>PRZETWARZANIE RÓWNOLEGŁE I ROZPROSZONE</b>				Kod modułu: M7		
	Nazwa przedmiotu: <b>Przetwarzanie równoległe i rozproszone</b>				Kod przedmiotu: M7-5		
	Nazwa jednostki organizacyjnej prowadzącej przedmiot / moduł: <b>INSTYTUT INFORMATYKI STOSOWANEJ im. Krzysztofa Brzeskiego</b>						
	Nazwa kierunku: <b>INFORMATYKA</b> (w zakresie: <b>Administracja systemów i sieci komputerowych, Projektowanie baz danych i oprogramowanie użytkowe, Grafika komputerowa i multimedia</b> )						
	Forma studiów: <b>niestacjonarne</b>		Profil kształcenia: <b>PRAKTYCZNY</b>			Poziom kształcenia: <b>STUDIA I STOPNIA</b>	
	Rok / semestr: <b>2/4</b>		Status przedmiotu / modułu: <b>OBOWIĄZKOWY</b>			Język przedmiotu / modułu: <b>POLSKI</b>	
	Forma zajęć	wykład	ćwiczenia	laboratorium	projekt	seminarium	inne (wpisać jakie)
	Wymiar zajęć (godz.)	<b>15</b>		<b>30</b>			

Koordinator przedmiotu / modułu	<b>dr inż. Zdzisław Szczerbiński</b>
Prowadzący zajęcia	<b>dr inż. Zdzisław Szczerbiński</b>
Cel kształcenia przedmiotu / modułu	1. Przekazanie podstawowej wiedzy w zakresie architektur i oprogramowania systemów równoległych. 2. Nauczenie głównych zasad programowania równoległego.
Wymagania wstępne	1. Zaliczenie przedmiotu Podstawy programowania I (semestr I). 2. Umiejętność klasycznego, sekwencyjnego programowania w języku proceduralnym.

EFEKTY UCZENIA SIĘ		
Nr efektu uczenia się/ grupy efektów	Opis efektu uczenia się	Kod kierunkowego efektu uczenia się
<b>Wiedza</b>		
01	zna zasady budowy i działania współczesnych systemów równoległych (komputerów wieloprocesorowych/wielordzeniowych, klastrów).	K_W05, K_W08, K_W09
02	zna podstawowe konstrukcje programistyczne związane z programowaniem równoległym.	K_W07
03	ma wiedzę na temat przeszkód dla bezproblemowego zrównoleglania programów.	K_W15
<b>Umiejętności</b>		

04	klasyfikuje istniejące architektury systemów równoległych oraz wyprowadza wnioski na podstawie twierdzeń dotyczących efektywności wykorzystania tych systemów.	K_U06
05	pisze, uruchamia i testuje programy równoległe w środowisku programistycznym dostosowanym do systemów równoległych z pamięcią wspólną.	K_U16
06	programuje proste algorytmy równoległe.	K_U17
07	jest świadomy konieczności nieustannej aktualizacji wiedzy na temat najszybciej rozwijających się dziedzin techniki (informatyka) oraz ich nowatorskich gałęzi (przetwarzanie równoległe i rozproszone).	K_U06

<b>TREŚCI PROGRAMOWE</b>	
<b>Wykład</b>	
<p>Wprowadzenie do tematyki komputerów, programowania i obliczeń równoległych.            Klasyfikacje architektur komputerowych pod względem mechanizmu sterowania i organizacji przestrzeni adresowej.            Systemy wieloprocessorowe z pamięcią wspólną. Sposoby połączeń procesory-pamięć.            Systemy wieloprocessorowe z pamięcią rozproszoną. Struktura sieci połączeń międzyprocesorowych.            Sieci statyczne i dynamiczne.            Klastry komputerów i sieciowe systemy obliczeniowe.            Miary efektywności obliczeń równoległych. Prawo Amdahla.            Algorytmy SIMD. Ilustracja pracy w trybie SIMD w systemie o procesorach połączonych w szereg.            Algorytmy MIMD. Ilustracja pracy w trybie MIMD w systemie o procesorach połączonych w drzewo binarne.            Komunikacja i synchronizacja międzyprocesorowa przy użyciu zmiennych współdzielonych.            Metoda aktywnego czekania. Mechanizm semaforów.            Procesy współbieżne. Sterowanie dostępem procesów do wspólnego zasobu za pomocą semaforów.            Blokada wzajemna procesów. Zagłodzenie procesu. Zadanie o pięciu uczujących filozofach.            Wykorzystanie semaforów do synchronizacji zdarzeń.            Przykład użycia semaforów ogólnych: zadanie o producencie i konsumencie.            Przykład użycia semaforów binarnych: zadanie o niedźwiedziu i pszczołach.            Komunikacja i synchronizacja międzyprocesorowa przez przesyłanie komunikatów.</p>	
<b>Ćwiczenia</b>	
<b>Laboratorium</b>	
<p>Głównym celem laboratorium jest przyswojenie pojęć i metod programowania równoległego.            Platformę sprzętową stanowi system dwuprocessorowy SunFire; jako końcówki interaktywne studentów używane są komputery SunBlade; platformę programową stanowi język programowania C wzbogacony o dyrektywy zrównoleglające standardu OpenMP. Na zajęciach omawia się konkretne programy pomagające zrozumieć podstawy programowania systemów wieloprocessorowych z pamięcią wspólną. Po, poprzedzonym wstępem tablicowym, zapoznaniu się z tekstem programu, studenci samodzielnie kompilują i uruchamiają program a następnie analizują wyniki jego wykonania, sporządzając stosowne notatki. Studenci mają stały dostęp (również poza zajęciami, z zewnątrz, przez sieć Internet) do realizowanych przykładów programowych, co korzystnie wpływa na opanowanie przez nich praktyki programowania równoległego.            W końcowej części semestru, studenci poznają zasady programowania wektorowego oraz podstawy programowania równoległego w językach programowania Python i Julia.</p>	

<b>Procentowy udział:</b> treść teoretyczna – 30%, treść praktyczna – 70%.
<b>Projekt</b>
<b>Seminarium</b>
<b>Inne</b>

Literatura podstawowa	<ol style="list-style-type: none"> <li>1. <i>Programowanie równoległe i rozproszone</i>, praca zbiorowa pod red. A. Karbowskiego i E. Niewiadomskiej-Szynkiewicz, Oficyna Wydawnicza Politechniki Warszawskiej, 2009.</li> <li>2. Z. Weiss, T. Gruzlewski – <i>Programowanie współbieżne i rozproszone w przykładach i zadaniach</i>, WNT, 1993.</li> </ol>
Literatura uzupełniająca	<ol style="list-style-type: none"> <li>1. S. Kozielski, Z. Szczerbiński – <i>Komputery równoległe: architektura, elementy programowania</i>, WNT, 1994.</li> <li>2. B. E. Borowik – <i>Programowanie równoległe w zastosowaniach</i>, MIKOM, 2001.</li> </ol>
Metody kształcenia	wykład, ćwiczenia laboratoryjne

Metody weryfikacji efektów uczenia się		Nr efektu uczenia się/grupy efektów
egzamin pisemny (60 minut) z zakresu zagadnień teoretycznych (4 zadania) i praktycznych (2 zadania)		01, 02, 04
indywidualne rozwiązywanie zadań laboratoryjnych z programowania równoległego (dla przerabianych programów komputerowych oceniane są: umiejętność dokonywania samodzielnych zmian w treści, sensownego testowania, oraz interpretacji wyników wykonania)		03, 05, 06
dyskusja, na ostatnim wykładzie oraz zajęciach laboratoryjnych, na temat treści wyniesionych z zajęć wcześniejszych (od początku semestru), ze szczególnym uwzględnieniem przyswojonej wiedzy na temat najnowszych trendów w konstrukcji architektur i programowaniu równoległym		07
Formy i warunki zaliczenia	Egzamin końcowy, pisemny, obejmujący materiał z wykładu i laboratorium. Procentowy udział w ocenie z egzaminu: wykład – 70%, laboratorium – 30%. Na ocenę końcową wpływają dodatkowo ewentualnie: wysoka aktywność oraz niska frekwencja (nieusprawiedliwione nieobecności) w czasie zajęć laboratoryjnych.	

NAKŁAD PRACY STUDENTA		
Rodzaj działań/zajęć	Liczba godzin	
	Ogółem	W tym zajęcia powiązane z praktycznym

		przygotowaniem zawodowym
Udział w wykładach	15	
Samodzielne studiowanie	5	
Udział w ćwiczeniach audytoryjnych i laboratoryjnych, warsztatach, seminariach	30	21
Samodzielne przygotowywanie się do ćwiczeń	30	30
Przygotowanie projektu / eseju / itp.		
Przygotowanie się do egzaminu / zaliczenia	20	5
Udział w konsultacjach	2	
Inne - egzamin	2	
<b>ŁĄCZNY nakład pracy studenta w godz.</b>	104	56
<b>Liczba punktów ECTS za przedmiot</b>	<b>4 ECTS</b>	
<b>Liczba punktów ECTS przypisana do dyscypliny naukowej</b>	<b>Informatyka techniczna i telekomunikacja 4 ECTS</b>	
Liczba punktów ECTS związana z zajęciami praktycznymi	<b>2,2 ECTS</b>	
Liczba punktów ECTS za zajęcia wymagające bezpośredniego udziału nauczycieli akademickich	<b>1,9 ECTS</b>	