

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Przetwarzanie równoległe</b>		Kod <b>1010514351010510100</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>3 / 5</b>
Ścieżka obieralności/specjalność <b>-</b>	Przedmiot oferowany w języku: <b>polski</b>	Kurs (obligatoryjny/obieralny) <b>obligatoryjny</b>
Stopień studiów: <b>I stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>niestacjonarna</b>	
Godziny Wykłady: <b>12</b> Ćwiczenia: <b>-</b> Laboratoria: <b>12</b> Projekty/seminaria: <b>-</b>		Liczba punktów <b>2</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) <b>kierunkowy</b>		(ogólnouczelniany, z innego kierunku) <b>z danego kierunku</b>
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki <b>nauki techniczne</b> <b>nauki techniczne</b>		Podział ECTS (liczba i %) <b>2 100%</b> <b>2 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b>		
dr inż. Rafał Walkowiak email: rafal.walkowiak@cs.put.poznan.pl tel. 616652574 Wydział Informatyki ul. Piotrowo 2, 60-965 Poznań		
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	Student rozpoczynający ten przedmiot powinien posiadać wiedzę z zakresu organizacji maszyn cyfrowych, algorytmów i struktur danych, programowania w języku C i programowania matematycznego.
2	<b>Umiejętności:</b>	Powinien posiadać umiejętność rozwiązywania podstawowych problemów z dziedziny budowy i oceny funkcjonowania systemu obliczeniowego.
3	<b>Kompetencje społeczne</b>	Student musi prezentować takie postawy jak uczciwość, punktualność, odpowiedzialność za pracę wykonywaną w grupie, umiejętność prezentacji w ramach tekstu pisanego toku rozumowania, zasad działania sprzętu, sposobu realizacji kodu i zależności pomiędzy elementami systemu.
<b>Cel przedmiotu:</b>		
1. Przekazanie studentom podstawowej wiedzy z przetwarzania równoległego, w zakresie obejmującym modele, systemy obliczeniowe, środowiska i języki, problemy i metody ich rozwiązywania. 2. Rozwijanie u studentów umiejętności rozwiązywania zadań z dziedziny przetwarzania i optymalizacji przetwarzania w równoległym systemie obliczeniowym, porównywania efektywności przetwarzania równoległego realizowanego przy zastosowaniu różnych środowisk i sprzętu. 3. Rozwijanie u studentów świadomości potrzeby wykorzystania, zrozumienia zasad działania i umiejętności korzystania z narzędzi pozwalających na ocenę efektywności przetwarzania w równoległych systemach przetwarzających.		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
1. Student ma uporządkowaną, podbudowaną teoretycznie wiedzę w zakresie równoległych systemów komputerowych, algorytmów przetwarzania równoległego, ich złożoności oraz paradygmatów programowania równoległego. - [K1st_W4] 2. Student ma wiedzę o kierunkach rozwoju architektur równoległych systemów komputerowych. - [K1st_W5] 3. Student zna metody, techniki i narzędzia stosowane przy rozwiązywaniu zadań informatycznych z zakresu budowy systemów komputerowych i języków programowania równoległego. - [K1st_W7]		
<b>Umiejętności:</b>		

1. Student potrafi właściwie zaplanować oraz wykonać eksperymenty oceny efektywności przetwarzania równoległego, dokonać interpretacji rezultatów oraz poprawnie wyciągnąć płynące z nich wnioski. - [K1st\_U3]
2. Student potrafi rozwiązując zadania z dziedziny przetwarzania równoległego zastosować odpowiednie metody optymalizacyjne (programowanie matematyczne) i eksperymentalne (narzędzia Code XL, Cuda profiler). - [K1st\_U4]
3. Student potrafi ocenić złożoność obliczeniową algorytmów równoległych. - [K1st\_U8]
4. Student posiada umiejętność formułowania algorytmów równoległych i ich implementacji w środowisku OpenMP oraz CUDA. - [K1st\_U11]
5. Student potrafi współdziałać i pracować w grupie realizując projekty i badania nad implementacją i oceną efektywności algorytmów równoległych. - [K1st\_U18]

#### Kompetencje społeczne:

1. Student rozumie konieczność poszerzania wiedzy i umiejętności wynikającą z postępu technologicznego w dziedzinie sprzętu przetwarzania równoległego. - [K1st\_K1]
2. Student ma świadomość znaczenia wiedzy w informatycznej (w zakresie sprzętu i oprogramowania) w rozwiązywaniu problemów z dziedziny optymalizacji przetwarzania równoległego. - [K1st\_K2]

#### Sposoby sprawdzenia efektów kształcenia

Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

- a) w zakresie wykładów na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach;
- b) w zakresie laboratorium na podstawie dyskusji i oceny bieżącego postępu realizacji zadań.

Ocena podsumowująca:

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

- ocenę przez prowadzącego zajęcia laboratoryjne i obronę przez studentów sprawozdań z realizacji zadań i projektów,
- sprawdzian z określonego zakresu wiadomości i umiejętności wynikających z realizacji ćwiczenia laboratoryjnego,
- ocenę wiedzy i umiejętności wykazanych na zaliczeniu pisemnym materiału wykładowego (zaliczenie polega na przygotowaniu odpowiedzi pisemnych na ok. 6 pytań problemowych, na zaliczenie wymagane jest uzyskanie 50% punktów, zadania z lat ubiegłych dostępne są dla studentów, poprawne rozwiązania zadań są prezentowane na konsultacjach studentom przedstawiającym swoje próby rozwiązania zadań.)

Możliwe jest również uzyskiwanie punktów dodatkowych za aktywność studentów polegającą na:

- przygotowaniu poprawnych odpowiedzi na problemowe zadania domowe przedstawiane podczas wykładów,
- pomysłowość i zaangażowanie w rozwiązywaniu zadanego problemu (laboratorium),
- wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenie procesu dydaktycznego.

#### Treści programowe

Program przedmiotu obejmuje następujące zagadnienia:

- współbieżność wewnętrzna systemów obliczeniowych, superskalarność,
- klasyfikacje i przykłady systemów równoległych (wielordzeniowe CPU, GPU),
- sieci połączeń komputerów równoległych,
- pamięć podręczna i problem spójności pamięci podręcznej w systemach wieloprocesorowych,
- komunikacja w systemach równoległych: metody transmisji komunikatów, porównanie kosztów komunikacji w różnych architekturach przetwarzania, algorytmy komunikacji grupowej,
- podstawy oceny efektywności systemów i algorytmów równoległych - skalowalność, prawa Amdahla i Gustafsona
- modele przetwarzania równoległego (pamięć współdzielona, przekazywanie komunikatów, równoległość danych),
- algorytmy równoległe: ogólna metoda konstrukcji algorytmów równoległych - techniki podziału problemu, metody przydziału zadań do procesorów (metoda zadania jednorodnego, zagnieżdżenie grafów procesorów i architektury systemu równoległego),
- przykładowe środowiska przetwarzania równoległego (Open MP, CUDA, MPI ),
- przykładowe proste algorytmy równoległe (sortowanie, maksimum, mnożenie macierzy, znajdowanie liczb pierwszych, składowe spójne).

W ramach laboratorium studenci realizują poniższe zadania.

- Studenci poznają praktycznie środowisko OpenMP realizując zadania dotyczące sposobów współdzielenia danych w prostym kodzie równoległym i przydziału zadań do rdzeni procesora, oceniają jakość przetwarzania równoległego.
- Dla zadanego zagadnienia studenci przygotowują wersje kodu aplikacji równoległej dla komputera z procesorem wielordzeniowym, oceniają i porównują efektywność wersji kodu i analizują kluczowe dla efektywności zdarzenia procesora. Dla realizacji zadania studenci poznają zasady oceny efektywności przetwarzania i program profilujący przetwarzanie na poziomie zdarzeń procesora.
- Studenci poznają praktycznie zasady wykorzystania (GPU) Graphical Processing Unit dla obliczeń równoległych, analizowane zadania dotyczą optymalizacji konfiguracji kodu i optymalizacji dostępu do pamięci karty GPU.

Metody dydaktyczne:

<p>1. Wykład: prezentacja ilustrowana przykładami wyjaśnianymi na tablicy, rozwiązywanie zadań, zadania domowe - rozwiązywanie zadań problemowych związanych z materiałem wykładu</p> <p>2. Ćwiczenia laboratoryjne: prezentacja działania, konfiguracji i wykorzystania środowisk i narzędzi, rozwiązywanie przykładowych zadań modelowania systemu przy tablicy, przeprowadzanie i omawianie w grupach wyników szerokich eksperymentów obliczeniowych nad przygotowanym przez studentów kodem.</p>		
<p><b>Literatura podstawowa:</b></p> <p>1. Wprowadzenie do obliczeń równoległych, Z. Czech, PWN, Warszawa, 2013.</p> <p>2. Cuda w przykładach : wprowadzenie do ogólnego programowania procesorów GPU, J.Sanders, E.Kandrot, Helion, 2012.</p> <p>3. Introduction to Parallel Computing, A.Grama, A.Gupta, G.Karypis,V.Kumar, Addison Wesley, 2003.</p> <p>4. Designing and Building Parallel Programs, I. Foster , www.mcs.anl.gov/dbpp/text/book.html</p>		
<p><b>Literatura uzupełniająca:</b></p> <p>1. Specyfikacje i podręczniki do OpenMP, CUDA.</p> <p>2. A general purpose lossless data compression method for GPU, M.Chłopkowski, R.Walkowiak, JPDC/75, 40-52, (2015), <a href="http://dx.doi.org/10.1016/j.jpdc.2014.09.016">http://dx.doi.org/10.1016/j.jpdc.2014.09.016</a></p>		
<p><b>Bilans nakładu pracy przeciętnego studenta</b></p>		
<p><b>Czynność</b></p>	<p><b>Czas (godz.)</b></p>	
1. udział w wykładach	12	
2. udział w zajęciach laboratoryjnych:	12	
3. przygotowanie do ćwiczeń laboratoryjnych opracowanie koncepcji, modeli, przygotowanie programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	12	
4. dokończenie (w ramach pracy własnej) sprawozdań z projektów laboratoryjnych:	2	
5. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych i projektów	4	
6. przygotowanie do zaliczenia wykładów i udział w kolokwium zaliczeniowym		
<p><b>Obciążenie pracą studenta</b></p>		
<p><b>forma aktywności</b></p>	<p><b>godzin</b></p>	<p><b>ECTS</b></p>
Łączny nakład pracy	54	2
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	26	1
Zajęcia o charakterze praktycznym	36	1