



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Programowanie równoległe [S2Teleinf2-STRC>PR]

Przedmiot

Kierunek studiów
Teleinformatyka

Rok/Semestr
2/3

Studia w zakresie (specjalność)
Sieci teleinformatyczne i rozwiązania chmurowe

Profil studiów
ogólnoakademicki

Poziom studiów
drugiego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
stacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład
14

Laboratorium
24

Inne
14

Ćwiczenia
0

Projekty/seminaria
0

Liczba punktów ECTS

4,00

Koordynatorzy

dr hab. inż. Olgierd Stankiewicz prof. PP
olgierd.stankiewicz@put.poznan.pl

Wykładowcy

Wymagania wstępne

Wiedza: - zna podstawowe struktury danych oraz algorytmy wykorzystywane w językach programowania. - ma praktyczną wiedzę w zakresie metodyki i technik programowania w językach wysokiego poziomu. - ma podstawową wiedzę z zakresu metod cyfrowego przetwarzania sygnałów w teleinformatyce. Umiejętności: - potrafi pozyskiwać informacje z literatury i baz danych oraz innych źródeł w języku polskim lub angielskim. - potrafi wykorzystywać mechanizmy programowania i środowiska programistyczne języków obiektowych oraz dostępne oprogramowanie biblioteczne. Kompetencje społeczne: - zna ograniczenia własnej wiedzy i rozumie konieczność jej uaktualniania. - rozumie wpływ pracy własnej na wyniki zespołu i konieczność podporządkowania się zasadom pracy w zespole oraz ponoszenia odpowiedzialności za wspólnie realizowane zadania

Cel przedmiotu

Zapoznanie z podstawowymi cechami przetwarzania równoległego, zaznajomienie się z istniejącymi rozwiązaniami technicznymi dotyczącymi metodyki projektowania algorytmów przetwarzania równoległego, przygotowanie własnych realizacji wybranych algorytmów oraz rozwijanie umiejętności pozyskiwania wiedzy na temat aktualnych rozwiązań w zakresie systemów przetwarzania równoległego.

Przedmiotowe efekty uczenia się

Wiedza:

K2_W02 Posiada zaawansowaną wiedzę w obszarze programowania równoległego, w tym:

- projektowania i implementacji równoległych algorytmów obliczeniowych
- optymalizacji wydajności zadań równoległych
- wykorzystania nowoczesnych technik programowania równoległego dla złożonych systemów

K2_W04 Ma pogłębioną wiedzę w zakresie zaawansowanych metodologii tworzenia złożonych systemów informatycznych, w tym:

- rozumie wykorzystanie zaawansowanych narzędzi modelowania i symulacji systemów informatycznych
- ma znajomość współczesnych języków programowania i ich zastosowań w tworzeniu oprogramowania
- rozumie kluczowe zasady inżynierii oprogramowania, które umożliwiają skuteczny rozwój i utrzymanie złożonych systemów

K2_W07 Posiada dogłębną wiedzę na temat najnowszych kierunków rozwoju oraz kluczowych innowacji w dziedzinie technologii informacyjno-komunikacyjnych, w tym:

- ma świadomość aktualnych trendów i nowych rozwiązań w obszarze technologii informacyjnych
- rozumie dynamiczne zmiany i innowacje rozwiązania w dziedzinie komunikacji cyfrowej
- orientuje się w najnowszych osiągnięciach technologicznych z zakresu telekomunikacji i informatyki

Umiejętności:

K2_U01 Demonstruje umiejętność skutecznego pozyskiwania informacji z różnorodnych źródeł, w tym z literatury specjalistycznej, baz danych i innych zasobów, oraz umiejętność:

- integracji i analizy zgromadzonych danych w celu formułowania kompleksowych wniosków
- krytycznej oceny informacji oraz tworzenia uzasadnionych i trafnych opinii
- interpretacji danych i wniosków w celu wyciągania trafnych i zrównoważonych wniosków

K2_U09 Posiada umiejętność proponowania innowacyjnych usprawnień lub alternatywnych rozwiązań dla istniejących projektów oraz systemów informatyczno-komunikacyjnych, uwzględniając:

- kreatywność w opracowywaniu nowatorskich koncepcji usprawnień systemowych
- umiejętność identyfikacji możliwości optymalizacji projektów informatycznych
- zdolność do formułowania alternatywnych strategii rozwoju systemów teleinformatycznych

K2_U10 Demonstruje umiejętność analizy i oceny zastosowania nowych technologii oraz metod projektowania w tworzeniu innowacyjnych układów i systemów informatyczno-komunikacyjnych, uwzględniając:

- zdolność do krytycznej oceny przydatności nowych osiągnięć technologicznych w praktyce
- umiejętność identyfikacji potencjalnych zastosowań nowych rozwiązań technicznych w projektach teleinformatycznych
- wrażliwość na innowacyjne podejścia do tworzenia nowoczesnych systemów informatycznych

Kompetencje społeczne:

K2_K01 Demonstruje gotowość do doceniania roli wiedzy w rozwiązywaniu zarówno teoretycznych, jak i praktycznych problemów, oraz do:

- krytycznej analizy i oceny informacji oraz treści otrzymywanych z różnych źródeł
- doceniania znaczenia wiedzy teoretycznej i praktycznej w podejmowaniu decyzji
- wykorzystywania wiedzy w celu podejmowania trafnych i zrównoważonych decyzji w różnych dziedzinach

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

1. Wykład

Egzamin pisemny i/lub ustny. Egzamin składa się z kilku - kilkunastu pytań (w zależności od przyjętego charakteru pytań) i dotyczy treści przedstawionych podczas wykładów. Dokładny charakter pytań egzaminacyjnych zostanie studentom przedstawiony podczas jednego z ostatnich wykładów. Próg zdania egzaminu: 50% punktów.

Skala ocen: <50% - 2,0 (ndst); 50% do 59% - 3,0 (dst); 60% do 69% - 3,5 (dst+) ; 70% do 79% - 4,0 (db); 80% do 89% - 4,5 (db+); 90% do 100% - 5,0 (bdb).

2. Laboratoria

Sprawozdania z jednolitych tematycznie bloków ćwiczeń laboratoryjnych. Bieżąca ocena stopnia opanowania materiału, postępu realizacji zadań oraz aktywności na zajęciach. Próg zaliczeniowy: 50% punktów.

Skala ocen: <50% - 2,0 (ndst); 50% do 59% - 3,0 (dst); 60% do 69% - 3,5 (dst+) ; 70% do 79% - 4,0 (db); 80% do 89% - 4,5 (db+); 90% do 100% - 5,0 (bdb).

Treści programowe

Przetwarzanie zrównoleglone:

- podział na zadania nie wymagające synchronizacji,
- podział na zadania wymagające synchronizacji,
- bariery,
- punkty synchronizacji,
- kolizje w dostępie do danych,

Programowanie wielowątkowe:

Przegląd bibliotek umożliwiających tworzenia oprogramowania przetwarzania równoległego:

- CPU: C++ threads, pthreads, MPI, OpenMP.
- GPU: OpenCL, Cuda, Vulkan.

Zajęcia laboratoryjne będą polegały na przygotowywaniu przez studentów programów realizujących wybrane algorytmy przetwarzania równoległego w tym na procesorach graficznych wraz z ekperymentalną weryfikacją poprawności ich działania.

Tematyka zajęć

Przetwarzanie zrównoleglone:

- podział na zadania nie wymagające synchronizacji,
- podział na zadania wymagające synchronizacji,
- bariery,
- punkty synchronizacji,
- kolizje w dostępie do danych,

Programowanie wielowątkowe:

Przegląd bibliotek umożliwiających tworzenia oprogramowania przetwarzania równoległego:

- CPU: C++ threads, pthreads, MPI, OpenMP.
- GPU: OpenCL, Cuda, Vulkan.

Zajęcia laboratoryjne będą polegały na przygotowywaniu przez studentów programów realizujących wybrane algorytmy przetwarzania równoległego w tym na procesorach graficznych wraz z ekperymentalną weryfikacją poprawności ich działania.

Metody dydaktyczne

1. Wykład

Zajęcia z wyraźnymi elementami wykładu tradycyjnego, wykładu problemowego (dyskusja ze studentami określonego problemu) oraz wykładu konwersatoryjnego (mobilizowanie studentów do dyskusji na określony temat), zależnie od treści prezentowanego materiału. Wybrane treści wykładu są prezentowane na rzutniku multimedialnym bądź tablicy. Omówieniu zagadnień towarzyszy informacja o ich praktycznym zastosowaniu.

2. Laboratoria

Zajęcia komputerowe z wykorzystaniem oprogramowania do programowania zrównoleglonego. Rozwiązywanie problemów podanych przez prowadzącego i/lub zdefiniowanych w instrukcji laboratoryjnej. Interpretacja otrzymanego rozwiązania oraz sformułowanie wniosków. Dyskusja możliwości zastosowania praktycznego zagadnień będących przedmiotem laboratorium.

Literatura

Podstawowa:

- Z. Czech, "Wprowadzenie do obliczeń równoległych", PWN, Warszawa 2013
- Foster I., "Designing and Building Parallel Programs", książka dostępna w internecie <http://www.unix.mcs.anl.gov/dbpp>
- M. Herlihy, N. Shavit „The Art of Multiprocessor Programming” Elsevier, 2008 (wydanie polskie „Sztuka programowania wieloprocesorowego”, PWN 2010)

Uzupełniająca:

- Grama A. et al., „Introduction to Parallel Computing” (wyd. 2), Addison-Wesley, 2003
- Strony internetowe: www.openmp.org, www.mpi-forum.org

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	103	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	38	1,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	65	2,50