

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Programowanie rozproszone		Kod 1010335511010335196
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) (brak)	Rok / Semestr 1 / 1
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: II stopień	Forma studiów (stacjonarna/niestacjonarna) niestacjonarna	
Godziny Wykłady: 16 Ćwiczenia: - Laboratoria: 8 Projekty/seminaria: 8		Liczba punktów 6
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) (brak)		(ogólnouczelniany, z innego kierunku) (brak)
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne nauki techniczne		Podział ECTS (liczba i %) 6 100% 6 100%
Odpowiedzialny za przedmiot / wykładowca:		
dr inż. Adam Meissner email: Adam.Meissner@put.poznan.pl tel. 61 665 37 24 Wydział Elektryczny ul. Piotrowo 3A 60-965 Poznań		dr inż. Krzysztof Zwierzyński email: Krzysztof.Zwierzynski@put.poznan.pl tel. 61 665 37 55 Wydział Elektryczny ul. Piotrowo 3A 60-965 Poznań
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie podstawowych algorytmów i ich analizy, technik projektowania algorytmów, abstrakcyjnych struktur danych i ich implementacji, problemów obliczeniowo trudnych; ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie architektury systemów komputerowych, zasad działania systemów operacyjnych i ich rodzajów; zna typowe informatyczne technologie inżynierskie.
2	Umiejętności:	Student posługuje się językiem angielskim w stopniu wystarczającym do porozumiewania się, a także czytania ze wskazanej literatury przedmiotowej; potrafi konstruować algorytmy z wykorzystaniem podstawowych technik algorytmicznych i dokonać analizy ich złożoności; potrafi posłużyć się środowiskami i platformami programistycznymi do pisania, wykonywania i testowania prostych programów kodowanych w językach programowania imperatywnego, obiektowego i deklaratywnego.
3	Kompetencje społeczne	Student rozumie potrzebę i zna możliwości ciągłego dokształcania się, podnoszenia kompetencji językowych, zawodowych, osobistych i społecznych; ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania.
Cel przedmiotu:		
przedstawienie podstawowych modeli programów rozproszonych oraz najważniejszych metod synchronizacji i komunikacji w takich programach; prezentacja wybranych problemów budowy programów rozproszonych.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
1. Student ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie technologii sieciowych - [K_W07] 2. Student ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie technologii internetowych - [K_W11]		
Umiejętności:		
1. Student potrafi pracować indywidualnie i w zespole; umie oszacować czas potrzebny na realizację zleconego zadania; potrafi opracować i zrealizować harmonogram prac zapewniający dotrzymanie terminów - [K_U02] 2. Student potrafi planować i przeprowadzać eksperymenty, wykorzystywać poznane metody i modele matematyczne, a także symulacje komputerowe do testowania, analizy i oceny działania systemów informatycznych i ich składowych - [K_U07] 3. Student potrafi dokonać krytycznej analizy sposobu funkcjonowania sprzętu komputerowego, systemu operacyjnego (lub ich fragmentów) i sieci komputerowych - [K_U11]		
Kompetencje społeczne:		

1. Student ma świadomość roli społecznej absolwenta uczelni techn., rozumie potrzebę przekazywania społeczeństwu - m.in. przez środki masowego przekazu - informacji dotyczących informatyki; podejmuje starania, by przekazać informacje w sposób zrozumiały - [K_K06]
2. Student ma świadomość ważności dokładnego wykonania projektu, zachowania standardów notacyjnych, przestrzegania poprawności językowej i terminowego oddania prac - [K_K07]

Sposoby sprawdzenia efektów kształcenia

Wykład. Egzamin pisemny obejmujący pytania teoretyczne oraz proste zadania z zakresu programowania rozproszonego. Laboratorium. Ustne lub pisemne sprawdzenie wiadomości niezbędnych do wykonania bieżących ćwiczeń, przygotowanie i terminowe składanie sprawozdań, ocena aktywności uczestników zajęć.
Projekt. Terminowa realizacja kolejnych etapów zadania projektowego. Przygotowanie dokumentacji.
Warunki zaliczenia wykładu, laboratorium i projektu: należy uzyskać co najmniej 50,1% całkowitej liczby punktów.

Treści programowe

Wykład. Programowanie rozproszone a programowanie współbieżne, rozproszony model programu współbieżnego, przezroczystość sieciowa, model klient-serwer, standard MPI, środowisko Open CL, metody synchronizacji wątków i procesów, miary wydajności systemów rozproszonych, projektowanie algorytmów rozproszonych, elementy programowania w modelu klient-serwer, problemy awaryjności i bezpieczeństwa w systemach rozproszonych. Programowanie rozproszone w języku Erlang.

Aktualizacja 2017: programowanie rozproszone w języku Erlang.

Laboratorium. Programowanie rozproszone przy użyciu biblioteki MPI. Rozpraszenie obliczeń z wykorzystaniem technologii GPGPU. Kolejowanie zadań w systemach superkomputerowych (zadanie opcjonalne).

Projekt. Realizacja zadania projektowego ilustrującego możliwości rozpraszania obliczeń za pomocą wybranej platformy programistyczno-sprzętowej.

Zastosowane metody kształcenia:

- wykład z prezentacją slajdów uzupełniany przykładami podawanymi na tablicy
- laboratorium - indywidualne rozwiązywanie przez studentów zadań programistycznych i uruchamianie ich w środowiskach rozproszonych, dostępnych także z domu, eksperymenty obliczeniowe
- projekt - praca w zespole, referowanie przez zespół postępów prac ilustrowane prezentacjami multimedialnymi, połączone z dyskusją proponowanych rozwiązań, przygotowanie przez zespół dokumentacji projektowej i szczegółowe zrecenzowanie tej dokumentacji przez prowadzącego.

Literatura podstawowa:

1. Learn You Some Erlang for Great Good! A beginner's Guide, Herbert F., <http://learnyousomeerlang.com/>
2. Programowanie współbieżne i rozproszone, Weiss Z., Gruzlewski T., WNT, 1993
3. Programowanie. Koncepcje, techniki i modele, Roy P. van, Haridi S., Wyd. Helion, Gliwice, 2005
4. Systemy rozproszone. Zasady i paradygmaty, Tanenbaum A.S., Steen M. van, WNT, Warszawa, 2006

Literatura uzupełniająca:

1. A User's Guide to MPI, Pacheco P.S., <http://www.wellesley.edu/CS/courses/CS331/notes/mpi.guide.pdf>
2. Ericsson AB, Erlang/OTP System Documentation, <http://erlang.org/doc/pdf/otp-systemdocumentation.pdf>
3. Introduction to Parallel Computing, Barney B., https://computing.llnl.gov/tutorials/parallel_comp/
4. Sztuka programowania wieloprocesorowego, Herlihy M., Shavit N., PWN, Warszawa, 2008

Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
1. Wykłady	16
2. Laboratoria	8
3. Projekt	8
4. Konsultacje i egzamin	5
5. Przygotowanie do ćwiczeń lab., wykonanie sprawozdań	15
6. Wykonanie projektu	15
7. Przygotowanie do egzaminu	30

Obciążenie pracą studenta

forma aktywności	godzin	ECTS
Łączny nakład pracy	97	6
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	37	3
Zajęcia o charakterze praktycznym	46	3

