



IV. Opis programu studiów

3. KARTA PRZEDMIOTU

Kod przedmiotu	M#1-N1-IP-602
Nazwa przedmiotu	Programowanie Rozproszone i Współbieżne
Nazwa przedmiotu w języku angielskim	Concurrent and Distributed Programming
Obowiązuje od roku akademickiego	2020/2021

USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	INFORMATYKA PRZEMYSŁOWA
Poziom kształcenia	I stopień
Profil studiów	ogólnoakademicki
Forma i tryb prowadzenia studiów	studia niestacjonarne
Zakres	wszystkie
Jednostka prowadząca przedmiot	Katedra Technik Komputerowych i Uzbrojenia
Koordinator przedmiotu	Rafał Pawlikowski
Zatwierdził	

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	przedmiot kierunkowy
Status przedmiotu	obowiązkowy
Język prowadzenia zajęć	polski
Usytuowanie modułu w planie studiów - semestr	semestr 6
Wymagania wstępne	Umiejętność programowania w C i C++, znajomość budowy systemów operacyjnych, wiedza na temat sieci komputerowych
Egzamin (TAK/NIE)	NIE
Liczba punktów ECTS	3

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	seminarium
Liczba godzin w semestrze	9		18		

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Student posiada podstawową wiedzę na temat elementów programowania współbieżnego i rozproszonego.	IP1_W11
	W02	Student posiada podstawową wiedzę na temat architektur systemów współbieżnych i rozproszonych, zna różne standardy programowania systemów z pamięcią wspólną oraz rozproszoną.	IP1_W11
	W03	Student posiada podstawową wiedzę o bibliotekach, środowiskach i interfejsach wspomagających programowanie współbieżne i rozproszone.	IP1_W11
Umiejętności	U01	Student potrafi na poziomie podstawowym zarządzać procesami w systemie operacyjnym.	IP1_U21
	U02	Student potrafi napisać prosty program z wykorzystaniem biblioteki Pthreads.	IP1_U25
	U03	Student potrafi napisać prosty program z wykorzystaniem środowiska OpenMP.	IP1_U25
	U04	Student potrafi napisać prosty program z wykorzystaniem interfejsu MPI.	IP1_U25
Kompetencje społeczne	K01	Rozumie potrzebę dokształcania się.	IP1_K01

TREŚCI PROGRAMOWE

Forma zajęć*	Treści programowe
wykład	1. Podstawowe pojęcia z zakresu przetwarzania równoległego i rozproszonego.
	2. Architektury maszyn równoległych – maszyny typu SISD, SIMD, MIMD, klastry, gridy.
	3. Przetwarzanie równoległe (współbieżne) w systemach operacyjnych – procesy i wątki oraz monitoring i zarządzanie nimi.
	4. Podstawowe elementy programowania równoległego: procesy i wątki, mechanizmy komunikacji i synchronizacji: mechanizmy przesyłania komunikatów gniazda, kolejki, zamek, semafor, bariera, mechanizmy pamięci wspólnej.
	5. Elementy programowania na maszyny wieloprocessorowe z pamięcią wspólną z zastosowaniem biblioteki Pthreads.
	6. Elementy programowania na maszyny wieloprocessorowe z pamięcią wspólną w środowisku OpenMP.
	7. Elementy programowania rozproszonego w sieciach komputerowych: model klient-serwer, gniazda, komunikacja synchroniczna i asynchroniczna, interfejs MPI.
	8. Podstawowe informacje o programowaniu w opartym na wywoływaniu zdalnych procedur (RPC, CORBA, SOAP, UDDI).
laboratorium	1. Zarządzanie procesami w systemie operacyjnym.
	2. Elementy programowania w systemie Linux (kompilatory, biblioteki, edytory, środowiska programistyczne).
	3. Analiza, uruchomienie i śledzenie wykonania prostych programów napisanych w oparciu o bibliotekę wątków Pthreads.
	4. Analiza, uruchomienie i śledzenie wykonania prostych programów napisanych w oparciu o środowisko programowania OpenMP (w systemach z pamięcią wspólną).
	5. Analiza, uruchomienie i śledzenie wykonania prostych programów napisanych w oparciu o mechanizm gniazdek (w systemach z pamięcią rozproszoną).
	6. Analiza, uruchomienie i śledzenie wykonania prostych programów napisanych w oparciu o interfejs MPI (w systemach z pamięcią rozproszoną).

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów kształcenia (zaznaczyć X)					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01		X				
W02		X				
W03		X				
U01			X			
U02			X			
U03			X			
U04			X			
K01						X

FORMA I WARUNKI ZALICZENIA

Forma zajęć*	Forma zaliczenia	Warunki zaliczenia
wykład	zaliczenie z oceną	Zaliczenie kolokwium, (udzielenie minimum 50% poprawnych odpowiedzi)
laboratorium	zaliczenie z oceną	Obecność na zajęciach. Wykonanie poprawnie co najmniej 50% zadań na kolokwiach przeprowadzanych na komputerze w trakcie zajęć.

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS							
Lp.	Rodzaj aktywności	Obciążenie studenta					Jednostka
		W	C	L	P	S	
1.	Udział w zajęciach zgodnie z planem studiów	9		18			h
2.	Inne (konsultacje, egzamin)	2		2			h
3.	Razem przy bezpośrednim udziale nauczyciela akademickiego	31					h
4.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	1,2					ECTS
5.	Liczba godzin samodzielnej pracy studenta	44					h
6.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	1,8					ECTS
7.	Nakład pracy związany z zajęciami o charakterze praktycznym	50					h
8.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	2,0					ECTS
9.	Sumaryczne obciążenie pracą studenta	75					h
10.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	3					ECTS

LITERATURA

1. Zbigniew Czech, Wprowadzenie do obliczeń równoległych, PWN 2010.
2. W. Iszkowski M. Maniecki. - Programowanie współbieżne.
3. M. Ben-Ari, Podstawy programowania współbieżnego i rozproszonego, WNT
4. W. Richard Stevens – Programowanie zastosowań sieciowych”
5. A.S. Tanenbaum, Rozproszone systemy operacyjne, PWN, 1997.
6. Karbowski A., Niewiadomska-Szynkiewicz E.: Programowanie Równoległe i Rozproszone, Oficyna Wydawnicza PW, 2009.
7. M.J Bach, Budowa systemu operacyjnego UNIX, WNT, 1995
8. Grama A., Kumar V., Gupta A., Karypis G., Introduction to Parallel Computing, 2/E, Addison Wesley , 2003.
9. Andrews, G.R.: „Foundations of Multithreaded, Paralel and Distributed Programming”. Addison Wesley, 2002.
10. Michael J. Quinn, “Parallel Programming in C with MPI and OpenMP”, McGraw–Hill 2004
11. William Gropp, Ewing Lusk, Anthony Skjellum, “Using MPI: Portable Parallel Programming with the Message–Passing Interface”, MIT Press 1999
12. Internet.