



IV. Opis programu studiów

3. KARTA PRZEDMIOTU

Kod przedmiotu	E-1IZ2-1002-s1
Nazwa przedmiotu	Programowanie systemów rozproszonych
Nazwa przedmiotu w języku angielskim	Programming of distributed systems
Obowiązuje od roku akademickiego	2019/20

USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	
Poziom kształcenia	
Profil studiów	
Forma i tryb prowadzenia studiów	
Zakres	
Jednostka prowadząca przedmiot	
Koordinator przedmiotu	Dr inż. Karol Tomaszewski
Zatwierdził	Dziekan Wydziału Elektrotechniki Automatyki i Informatyki Dr hab. inż. Antoni Różowicz, prof. PŚk

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	
Status przedmiotu	
Język prowadzenia zajęć	
Usytuowanie modułu w planie studiów - semestr	
Wymagania wstępne	Programowanie w języku C 2 Architektura systemów komputerowych 1
Egzamin (TAK/NIE)	
Liczba punktów ECTS	

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	Inne
Liczba godzin w semestrze	18	0	9	12	0

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Znajomość problematyki konstruowania (architektura) oraz programowania (komunikacja i synchronizacja) systemów rozproszonych/ przetwarzających równolegle.	INF2_W02 INF2_W03 INF2_W09
	W02	Wiedza na temat dowodzenia poprawności programów współbieżnych. Znajomość aparatu matematyczno-pojęciowego.	INF2_W02
	W03	Poznanie różnych mechanizmów synchronizowania przetwarzania rozproszonego, w szczególności API wielozadaniowego systemu operacyjnego (język C).	INF2_W02 INF2_W03
Umiejętności	U01	Umiejętność rozwiązania typowych problemów programowania współbieżnego.	INF2_U01 INF2_U07
	U02	Umiejętność przekształcenia problemu programistycznego do postaci przetwarzania równoległego i poprawnej synchronizacji poznanymi mechanizmami.	INF2_U01
Kompetencje społeczne	K01	Umiejętność pracy zespołowej.	INF2_U01 INF2_U02 INF2_U03 INF2_K02

TREŚCI PROGRAMOWE

Forma zajęć*	Treści programowe
wykład	1. Podział systemów rozproszonych/równoległych – taksonomia Flynna. Przykładowe architektury multikomputerów i zasada ich działania.
	2. Architektury wieloprocesorów: SMP, NUMA, COMA. Problem spójności pamięci podręcznych i skalowalność.
	3. Programowanie wieloprocesorów – pojęcie wątku i procesu, zalety i wady.
	4. Programowanie wieloprocesorów – podstawy API wybranych technologii w zakresie wielowątkowości. Metody synchronizacji: współdzielona zmienna globalna, obiekt sekcji krytycznej, semafor, obiekt zdarzenia.
	5. Algorytmy współbieżne, warunki poprawności programu współbieżnego, pojęcie przepływu, blokady, zagłodzenia, hazardu (wyścigu) oraz wzajemnego wykluczania.
	6-7. Modelowanie problemu wzajemnego wykluczania (synchronizacja). Algorytm Dekkera.
	8. Problem producenta-konsumenta: analiza, modele i przykładowe rozwiązanie w języku C.
	9. Problem uczujących filozofów: analiza, modele i przykładowe rozwiązanie w języku C.
	laboratorium
2. Zrównoleglenie programu sortującego przez scalanie (poprzez wielowątkowość) z synchronizacją za pomocą zmiennych globalnych.	
3. Zrównoleglenie programu sortującego przez scalanie (poprzez wielowątkowość) z synchronizacją za pomocą obiektów sekcji krytycznej.	
4. Zrównoleglenie programu sortującego przez scalanie (poprzez wielowątkowość) z synchronizacją za pomocą semaforów oraz obiektów zdarzeń.	
5. Implementacja programu rozwiązującego problem 2ch producentów i 2ch konsumentów współdzielących ten sam bufor.	

projekt	1. Projekt realizowany w zespołach 2-3 osobowych. Tematyka obejmuje równoległą implementację wybranego zagadnienia, reprezentującego konkretny problem obliczeniowy lub symulację rzeczywistej sytuacji (np. ruch drogowy), w postaci programu wielowątkowego w dowolnym języku programowania wspierającym wielowątkowość. Weryfikacja poprawności implementacji poprzez symulację (brak blokad i zagłódzeń). W sprawozdaniu wymagany opis i uzasadnienie wybranej implementacji i mechanizmu synchronizacji.
---------	---

*) zostawić tylko realizowane formy zajęć

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów uczenia się					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01		X				
W02		X				
W03		X				
U01			X			
U02			X			
K01				X		

FORMA I WARUNKI ZALICZENIA

Forma zajęć*	Forma zaliczenia	Warunki zaliczenia
wykład		Uzyskanie co najmniej 50% punktów podczas egzaminu.
laboratorium		Uzyskanie co najmniej 50% punktów za wykonane zadania laboratoryjne oraz z kolokwium końcowego, kolokwium w trakcie zajęć lub kolokwium końcowego i kolokwium w trakcie zajęć.
projekt		Uzyskanie co najmniej 50% punktów za wykonane zadanie projektowe oraz sprawozdanie.

*) zostawić tylko realizowane formy 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	18		9	12		h
3.	Inne (konsultacje, egzamin)*	2		2	2		h
4.	Razem przy bezpośrednim udziale nauczyciela akademickiego	45					h
5.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	1,8					ECTS
6.	Liczba godzin samodzielnej pracy studenta	80					h
7.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	3,2					ECTS
8.	Nakład pracy związany z zajęciami o charakterze praktycznym	21					h

9.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	2,33	ECTS
10.	Sumaryczne godzinowe obciążenie pracą studenta	125	h
11.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	5	

** wszelkie formy weryfikacji efektów, w tym egzaminy oraz nie więcej niż 2 godziny konsultacji dla każdej formy zajęć*

LITERATURA

1. M. Ben-Ari, Podstawy programowania współbieżnego i rozproszonego, WNT
2. A.S. Tanenbaum, Rozproszone systemy operacyjne, PWN
3. Ch. Petzold, Programowanie Windows, Microsoft Press

Uwaga: wykaz literatury winien uwzględniać aktualne i dostępne publikacje