



KARTA MODUŁU / KARTA PRZEDMIOTU

Kod modułu	E-1IZ2-1002-s1
Nazwa modułu	Programowanie systemów rozproszonych_In2N
Nazwa modułu w języku angielskim	Programming of distributed systems
Obowiązuje od roku akademickiego	2012/13 (aktualizacja 2017/2018)

A. USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	Informatyka
Poziom kształcenia	II stopień (I stopień / II stopień)
Profil studiów	Ogólnoakademicki (ogólnoakademicki / praktyczny)
Forma i tryb prowadzenia studiów	niestacjonarne (stacjonarne / niestacjonarne)
Specjalność	Systemy informacyjne
Jednostka prowadząca moduł	Katedra Systemów Informatycznych Zakład Informatyki
Koordinator modułu	Dr inż. Karol Tomaszewski
Zatwierdził:	Dziekan WEAiI Dr hab. inż. Antoni Różowicz, prof. PŚk

B. OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	Kierunkowy (podstawowy / kierunkowy / inny HES)
Status modułu	Obowiązkowy (obowiązkowy / nieobowiązkowy)
Język prowadzenia zajęć	Polski
Usytuowanie modułu w planie studiów - semestr	1
Usytuowanie realizacji przedmiotu w roku akademickim	Semestr zimowy (semestr zimowy / letni)
Wymagania wstępne	Programowanie w języku C 2 Architektura systemów komputerowych 1 (kody modułów / nazwy modułów)
Egzamin	tak (tak / nie)
Liczba punktów ECTS	5

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	inne
w semestrze	18		9	9	



C. EFEKTY KSZTAŁCENIA I METODY SPRAWDZANIA EFEKTÓW KSZTAŁCENIA

Cel modułu	Usystematyzowanie systemów rozproszonych, poznanie podstawowych problemów konstruowania i programowania (komunikacji i synchronizacji), omówienie wybranych algorytmów rozproszonych/współbieżnych, nabycie umiejętności poprawnej synchronizacji pracy wątków i procesów (3-4 linijki)
-------------------	--

Symbol efektu	Efekty kształcenia	Forma prowadzenia zajęć (w/ć/l/p/inne)	odniesienie do efektów kierunkowych	odniesienie do efektów obszarowych
W_01	Znajomość problematyki konstruowania (architektura) oraz programowania (komunikacja i synchronizacja) systemów rozproszonych/przetwarzających równolegle.	W	K_W02 K_W03 K_W09	T2A_W03 T2A_W04 T2A_W07
W_02	Wiedza na temat dowodzenia poprawności programów współbieżnych. Znajomość aparatu matematyczno-pojęciowego.	W	K_W02	T2A_W03 T2A_W07
W_03	Poznanie różnych mechanizmów synchronizowania przetwarzania rozproszonego, w szczególności API wielozadaniowego systemu operacyjnego (język C).	W, L	K_W02 K_W03	T2A_W03 T2A_W07
U_01	Umiejętność rozwiązywania typowych problemów programowania współbieżnego.	L	K_U01 K_U07	T2A_U01 T2A_U08
U_02	Umiejętność przekształcania problemu programistycznego do postaci przetwarzania równoległego i poprawnej synchronizacji poznanymi mechanizmami.	L, P	K_U01	T2A_U01
K_01	Praca w zespole.	P	K_U01 K_U02 K_U03 K_K02	T2A_U01 T2A_U02 T2A_U03 T2A_K03

Treści kształcenia:

1. Treści kształcenia w zakresie wykładu

Nr wykładu	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu
1	Podział systemów rozproszonych/równoległych – taksonomia Flynna. Przykładowe architektury multikomputerów i zasada ich działania.	W_01
2	Architektury wieloprocesorów: SMP, NUMA, COMA. Problem spójności pamięci podręcznych i skalowalność.	W_01
3	Programowanie wieloprocesorów – pojęcie wątku i procesu, zalety i wady.	W_01 W_03
4	Programowanie wieloprocesorów – podstawy API Microsoft Windows w zakresie wielowątkowości. Metody synchronizacji: współdzielona zmienna globalna, obiekt sekcji krytycznej, semafor, obiekt zdarzenia.	W_01 W_03
5	Algorytmy współbieżne, warunki poprawności programu współbieżnego, pojęcie przeplotu, blokady, zagłodzenia, hazardu (wyścigu) oraz wzajemnego wykluczania.	W_02
6-7	Modelowanie problemu wzajemnego wykluczania (synchronizacja). Algorytm Dekkera.	W_02
8	Problem producenta-konsumenta: analiza, modele i przykładowe rozwiązanie w języku C.	W_02 W_03 U_01
9	Problem uczujących filozofów: analiza, modele i przykładowe rozwiązanie w	W_02



	języku C.	W_03 U_01
--	-----------	--------------

2. Treści kształcenia w zakresie ćwiczeń

Nr zajęć ćwic.	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu
1		
2		
3		
4		

3. Treści kształcenia w zakresie zadań laboratoryjnych

Nr zajęć lab.	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu
1	Implementacja programu sortującego przez scalanie z podziałem na podprogramy.	U_01 U_02
2	Zrównoleglenie programu sortującego przez scalanie (poprzez wielowątkowość) z synchronizacją za pomocą zmiennych globalnych.	U_01 U_02
3	Zrównoleglenie programu sortującego przez scalanie (poprzez wielowątkowość) z synchronizacją za pomocą obiektów sekcji krytycznej.	U_01 U_02
4	Zrównoleglenie programu sortującego przez scalanie (poprzez wielowątkowość) z synchronizacją za pomocą semaforów oraz obiektów zdarzeń.	U_01 U_02
5	Implementacja programu rozwiązującego problem 2ch producentów i 2ch konsumentów współdzielących ten sam bufor.	U_01 U_02

4. Charakterystyka zadań projektowych

Projekt realizowany w zespołach 2-3 osobowych zespołach. Tematyka obejmuje równoległą implementację zadanego problemu, 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.

5. Charakterystyka zadań w ramach innych typów zajęć dydaktycznych

Metody sprawdzania efektów kształcenia

Symbol efektu	Metody sprawdzania efektów kształcenia (sposób sprawdzenia, w tym dla umiejętności – odwołanie do konkretnych zadań projektowych, laboratoryjnych, itp.)
W_01	Egzamin
W_02	Egzamin
W_03	Egzamin, kolokwium (z zakresu zajęć laboratoryjnych)
U_01	Kolokwium
U_02	Kolokwium, ocena z projektu
K_01	Ocena z projektu



D. NAKŁAD PRACY STUDENTA

Bilans punktów ECTS		
	Rodzaj aktywności	obciążenie studenta
1	Udział w wykładach	18
2	Udział w ćwiczeniach	
3	Udział w laboratoriach	9
4	Udział w konsultacjach (2-3 razy w semestrze)	2
5	Udział w zajęciach projektowych	9
6	Konsultacje projektowe	9
7	Udział w egzaminie	2
8		
9	Liczba godzin realizowanych przy bezpośrednim udziale nauczyciela akademickiego	51 (suma)
10	Liczba punktów ECTS, którą student uzyskuje na zajęciach wymagających bezpośredniego udziału nauczyciela akademickiego (1 punkt ECTS=25-30 godzin obciążenia studenta)	2
11	Samodzielne studiowanie tematyki wykładów	9
12	Samodzielne przygotowanie się do ćwiczeń	
13	Samodzielne przygotowanie się do kolokwium	
14	Samodzielne przygotowanie się do laboratoriów	5
15	Wykonanie sprawozdań	
15	Przygotowanie do kolokwium końcowego z laboratorium	4
17	Wykonanie projektu lub dokumentacji	20
18	Przygotowanie do egzaminu	15
19		
20	Liczba godzin samodzielnej pracy studenta	52 (suma)
21	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy (1 punkt ECTS=25-30 godzin obciążenia studenta)	2
22	Sumaryczne obciążenie pracą studenta	103
23	Punkty ECTS za moduł 1 punkt ECTS=25-30 godzin obciążenia studenta	5
24	Nakład pracy związany z zajęciami o charakterze praktycznym Suma godzin związanych z zajęciami praktycznymi	43
25	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym 1 punkt ECTS=25-30 godzin obciążenia studenta	2

E. LITERATURA

Wykaz literatury	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
Witryna WWW modułu/przedmiotu	http://achilles.tu.kielce.pl/Members/rtomaszewski/studia-zaoczne/materia142y-dydaktyczne-dla-studiow-zaocznych