

KARTA MODUŁU / KARTA PRZEDMIOTU

Kod modułu	
Nazwa modułu	Programowanie współbieżne
Nazwa modułu w języku angielskim	Parallel programming
Obowiązuje od roku akademickiego	2012/1013

A. USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	Elektrotechnika
Poziom kształcenia	II stopień (I stopień / II stopień)
Profil studiów	ogólnoakademicki (ogólno akademicki / praktyczny)
Forma i tryb prowadzenia studiów	stacjonarne (stacjonarne / niestacjonarne)
Specjalność	Elektronika i Telekomunikacja
Jednostka prowadząca moduł	Katedra Elektroniki i Systemów Inteligentnych
Koordynator modułu	dr inż. Filip Rudziński
Zatwierdził:	

B. OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	przedmiot kierunkowy (podstawowy / kierunkowy / inny HES)
Status modułu	nieobowiązkowy (obowiązkowy / nieobowiązkowy)
Język prowadzenia zajęć	polski
Usytuowanie modułu w planie studiów - semestr	semestr II
Usytuowanie realizacji przedmiotu w roku akademickim	semestr letni (semestr zimowy / letni)
Wymagania wstępne	(kody modułów / nazwy modułów)
Egzamin	nie (tak / nie)
Liczba punktów ECTS	4

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	inne
w semestrze	30		30		

C. EFEKTY KSZTAŁCENIA I METODY SPRAWDZANIA EFEKTÓW KSZTAŁCENIA

Cel modułu	Celem modułu jest wykształcenie studentów w zakresie implementacji programów komputerowych, z wykorzystaniem języków programowania C i C++ oraz paradygmatów programowania imperatywnego, proceduralnego i obiektowego. (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	Zna koncepcje i idee programowania współbieżnego oraz podstawowe narzędzia programistyczne, stosowane w tym obszarze.	W	K_W12	T2A_W02 T2A_W04 T2A_W07
W_02	Wie jak wykorzystać narzędzia programowania współbieżnego do tworzenia aplikacji wielowątkowych dla systemów jednoprosesorowych, wielordzeniowych.	W	K_W12	T2A_W02 T2A_W04 T2A_W07
W_03	Wie jak wykorzystać narzędzia programowania współbieżnego do tworzenia aplikacji wielowątkowych dla systemów wieloprosesorowych.	W	K_W12	T2A_W02 T2A_W04 T2A_W07
U_01	Potrafi zaprojektować i zaprogramować aplikację wykorzystującą co najmniej jeden odrębny wątek oraz synchronizować pracę odrębnych wątków z wątkiem podstawowym obsługującym tzw. graficzny interfejs użytkownika.	L	K_U08 K_U14	T2A_U09 T2A_U18
U_02	Potrafi rozwiązywać podstawowe problemy współbieżności z wykorzystaniem narzędzi do programowania aplikacji wielowątkowych.	L	K_U08 K_U14	T2A_U09 T2A_U18
U_03	Potrafi implementować wybrane algorytmy obliczeniowe w systemach wieloprosesorowych o niejednorodnym dostępie do pamięci operacyjnej.	L	K_U08 K_U14	T2A_U09 T2A_U18

Treści kształcenia:

1. Treści kształcenia w zakresie wykładu

Nr wykładu	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu
1	Wprowadzenie do programowania wielowątkowego. Podstawowe pojęcia i definicje. Wątek i proces. Priorytety wątków i procesów. Operacje zależne i niezależne. Prawa Amdahla i Gustafsona.	W_01
2	Synchronizacja wątków – cz. 1. Operacje atomowe. Sekcje krytyczne, zdarzenia, muteksy i semaforey. Sterowanie pracą wątków.	W_01
3	Synchronizacja wątków – cz. 2. Synchronizacja wątków z GUI aplikacji.	W_01
4	Synchronizacja procesów. Narzędzia do synchronizacji procesów: zdarzenia, muteksy i semaforey.	W_01
5	Współdzielenie pamięci operacyjnej przez wątki i procesy – cz. 1. Komunikacja pomiędzy wątkami i procesami: pamięć współdzielona.	W_01
6	Współdzielenie pamięci operacyjnej przez wątki i procesy – cz. 2. Komunikacja pomiędzy wątkami i procesami: potoki nazwane, skrzynki pocztowe, gniazda sieciowe.	W_01
7	Klasyczne problemy współbieżności. Komunikacja synchroniczna i asynchroniczna. Wzajemne wykluczanie. Problemy: pięciu filozofów, czytelników i pisarzy, itp.	W_01 W_02
8 – 10,	Obliczenia drobnoziarniste w systemach UMA.	W_02

	Implementacja przykładowych algorytmów dla systemów komputerowych o architekturze pamięci typu UMA, z jednym lub wieloma CPU, pracujących pod kontrolą 32-bitowych systemów operacyjnych rodziny Windows (z wykorzystaniem narzędzi API).	
11,	Obliczenia drobnoziarniste w systemach NUMA – cz. 1. Podstawowe narzędzia do obsługi pamięci o architekturze typu NUMA w 32-bitowych i 64-bitowych systemach operacyjnych rodziny Windows.	W_03
12 - 14	Obliczenia drobnoziarniste w systemach NUMA – cz. 2. Implementacja przykładowych algorytmów dla systemów komputerowych o architekturze pamięci typu NUMA, pracujących pod kontrolą 32-bitowych i 64-bitowych systemów operacyjnych rodziny Windows (z wykorzystaniem narzędzi API).	W_03
15	Obliczenia gruboziarniste w systemach klastrowych. Obliczenia drobnoziarniste w systemach klastrowych rodziny LINUX.	W_03

2. Treści kształcenia w zakresie ćwiczeń

3. Treści kształcenia w zakresie zadań laboratoryjnych

Nr zajęć lab.	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu
1, 2	Wprowadzenie do programowania wielowątkowego. Zapoznanie się ze środowiskiem programistycznym. Tworzenie wątków. Przekazywanie danych do procedur wątków.	U_01
3, 4	Synchronizacja wątków. Sterowanie wątkami (wstrzymanie, wznowienie, zakończenie pracy). Synchronizacja z GUI aplikacji.	U_01
5, 6	Współdzielenie pamięci pomiędzy wątkami. Zapewnienie bezpiecznego dostępu do danych współdzielonych z wykorzystaniem sekcji krytycznych i zdarzeń.	U_01
7	Kolokwium zaliczeniowe nr 1. Implementacja programu komputerowego wykorzystującego narzędzia i techniki programistyczne przedstawione na zajęciach lab. nr 1 – 6.	
8, 9	Klasyczne problemy współbieżności. Synchronizacja procesów na przykładzie problemu producenta i konsumenta.	U_02
11, 12	Obliczenia drobnoziarniste w systemach UMA. Wielowątkowa realizacja wybranego algorytmu optymalizacji numerycznej w systemie z pamięcią typu UMA.	U_02
13, 14	Obliczenia drobnoziarniste w systemach NUMA. Wielowątkowa realizacja wybranego algorytmu optymalizacji numerycznej (z zajęć lab. nr 11, 12) w systemie z pamięcią typu NUMA.	U_03
15	Kolokwium zaliczeniowe nr 2. Implementacja programu komputerowego wykorzystującego narzędzia i techniki programistyczne przedstawione na zajęciach lab. nr 8 – 14.	

4. Charakterystyka zadań projektowych

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.)
U_01 U_02 U_03	Test z wykorzystaniem komputera na zajęciach laboratoryjnych nr 7 i 15.

D. NAKŁAD PRACY STUDENTA

Bilans punktów ECTS		
	Rodzaj aktywności	obciążenie studenta
1	Udział w wykładach	30 godz.
2	Udział w ćwiczeniach	
3	Udział w laboratoriach	30 godz.
4	Udział w konsultacjach (2-3 razy w semestrze)	1 godz.
5	Udział w zajęciach projektowych	
6	Konsultacje projektowe	
7	Udział w egzaminie	
8		
9	Liczba godzin realizowanych przy bezpośrednim udziale nauczyciela akademickiego	61 godz. (suma)
10	Liczba punktów ECTS, którą student uzyskuje na zajęciach wymagających bezpośredniego udziału nauczyciela akademickiego <i>(1 punkt ECTS=25-30 godzin obciążenia studenta)</i>	2.44
11	Samodzielne studiowanie tematyki wykładów	0.6 (15 godz.)
12	Samodzielne przygotowanie się do ćwiczeń	
13	Samodzielne przygotowanie się do kolokwium	
14	Samodzielne przygotowanie się do laboratoriów	0.36 (9 godz.)
15	Wykonanie sprawozdań	
15	Przygotowanie do kolokwium końcowego z laboratorium	0.6 (15 godz.)
17	Wykonanie projektu lub dokumentacji	
18	Przygotowanie do egzaminu	
19		
20	Liczba godzin samodzielnej pracy studenta	39 godz. (suma)
21	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy <i>(1 punkt ECTS=25-30 godzin obciążenia studenta)</i>	1.56
22	Sumaryczne obciążenie pracą studenta	100 godz.
23	Punkty ECTS za moduł <i>1 punkt ECTS=25-30 godzin obciążenia studenta</i>	4
24	Nakład pracy związany z zajęciami o charakterze praktycznym <i>Suma godzin związanych z zajęciami praktycznymi</i>	30 godz.
25	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym <i>1 punkt ECTS=25-30 godzin obciążenia studenta</i>	1.2

E. LITERATURA

Wykaz literatury	1. Bielecki J.: "Visual C++ 6. Programowanie współbieżne, obiektowe i zdarzeniowe", Helion, 1999. 2. Ben-Ari M.: "Podstawy programowania współbieżnego i rozproszonego". WNT, Warszawa 1996. 3. Weiss Z., Gruzlewski T.: "Programowanie współbieżne i rozproszone w przykładach i zadaniach". WNT, Warszawa 1993.
Witryna WWW modułu /przedmiotu	