

1. Nazwa przedmiotu: ARCHITEKTURA KOMPUTERÓW		2. Kod przedmiotu: MK_23		
3. Karta przedmiotu ważna od roku akademickiego: 2012/2013				
4. Forma kształcenia:	studia pierwszego stopnia			
5. Forma studiów:	studia stacjonarne			
6. Kierunek studiów:	INFORMATYKA; WYDZIAŁ AEII			
7. Profil studiów:	ogólnoakademicki			
8. Specjalność:				
9. Semestr:	5, 6			
10. Jednostka prowadząca przedmiot:	Instytut Informatyki, RAu2			
11. Prowadzący przedmiot:	prof. dr hab. inż. Stanisław Kozielski			
12. Przynależność do grupy przedmiotów:	przedmioty wspólne			
13. Status przedmiotu:	obowiązkowy			
14. Język prowadzenia zajęć:	polski			
15. Przedmioty wprowadzające oraz wymagania wstępne: Podstawy informatyki, Mikroinformatyka.				
16. Cel przedmiotu: Celem wykładu jest zapoznanie studentów z najważniejszymi pojęciami dotyczącymi architektury komputerów, przedstawienie głównych kierunków rozwoju architektury komputerów, przedstawienie reprezentatywnych przykładów organizacji komputerów. Główną część wykładu stanowi prezentacja architektury współczesnych procesorów i komputerów równoległych. Celem laboratorium jest praktyczne zapoznanie studentów z różnymi architekturami komputerowymi, różnymi systemami operacyjnymi i technologiami programowania równoległego i rozproszonego.. Studenci zapoznawani są z komputerami opartymi o procesorach Sparc, PowerPC , x86 pracujących pod kontrolą różnych systemów operacyjnych (Windows, Linux, Sun Solaris, OS/400).				
17. Efekty kształcenia:				
Nr	Opis efektu kształcenia	Metoda sprawdzenia efektu kształcenia	Forma prowadzenia zajęć	Odniesienie do efektów dla kierunku studiów
W1	Zna podstawowe elementy budowy procesorów i komputerów. Rozumie różnice w pomiędzy komputerami o rozwiniętej liście rozkazów (CISC) i komputerami o zredukowanej liście rozkazów (RISC).	SP	WM, WT	K1A_W07
W2	Rozumie koncepcję i problemy potokowej realizacji rozkazów skoków oraz potokowej realizacji rozkazów o zależnych argumentach. Zna metody przewidywania rozgałęzień.	SP	WM, WT	K1A_W11
W3	Zna koncepcję architektury superskalarnej oraz architektury VLIW. Zna mechanizmy wspierające na poziomie sprzętowym współbieżną realizację wielu wątków w jednym procesorze. Rozumie ogólną koncepcję architektury CUDA.	SP	WM, WT	K1A_W11, K1A_W12

W4	Ma wiedzę o klasyfikacji komputerów równoległych, zna klasyfikację Flynna. Zna podstawowe architektury komputerów równoległych: komputerów wektorowych i macierzowych, systemów wieloprocesorowych i klastrów.	SP	WM, WT	K1A_W12
W5	Zna cechy systemów wieloprocesorowych z pamięcią wspólną, pamięcią rozproszoną oraz systemów z niejednorodnym dostępem do pamięci.	SP	WM, WT	K1A_W11, K1A_W12
W6	Zna organizację klastrów o wysokiej wydajności oraz klastrów o wysokiej niezawodności.	SP	WM, WT	K1A_W11, K1A_W12
U1	Potrafi ocenić wpływ parametrów procesora na efektywność realizacji programów.	SP, CL	WT	K1A_U09, K1A_U13
U2	Potrafi wykorzystać środowiska programowania równoległego PVM i JavaSpaces do realizacji wybranych algorytmów równoległych.	SP, CL, PS	L	K1A_U13
U3	Potrafi wykorzystać klastery stacji roboczych do obliczeń równoległych i zrównoważenia obciążenia	SP, CL, PS	L	K1A_U13
K1	Posiada kompetencje w zakresie doboru architektury i parametrów systemu komputerowego stosownie do potrzeb projektowanego systemu informatycznego.	CL, PS	WT, L	K1A_U14

18. Formy zajęć dydaktycznych i ich wymiar (liczba godzin)

W. : 30 L.: 30

19. Treści kształcenia:

Wykład

Historia rozwoju architektury komputerów: pierwsze komputery, wpływ rozwoju technologii elektronicznej na architekturę komputerów, kolejne generacje komputerów. Komputery o rozwiniętej liście rozkazów (CISC). Komputery o zredukowanej liście rozkazów (RISC): założenia, przykład komputera RISC I. Potokowa realizacja rozkazów - mechanizmy przetwarzania potokowego: problem realizacji rozkazów skoków (skoki opóźnione, przewidywanie rozgałęzień) oraz problem zależności między danymi (zmiana kolejności wykonania rozkazów, szyna zwrotna). Architektura superskalarna oraz architektura VLIW. Specyfika problemów potokowej realizacji rozkazów o zależnych argumentach w architekturze superskalarnej. Przemianowanie rejestrów. Przykłady procesorów o architekturze superskalarnej: procesory UltraSPARC, Motorola, PowerPC, POWER. Sprzętowe wsparcie wielowątkowości: wielowątkowość drobnoziarnista, gruboziarnista i współbieżna.

Architektura komputerów równoległych. Klasyfikacja systemów równoległych – formy równoległości: równoległość na poziomie rozkazów i równoległość na poziomie procesorów, klasyfikacja Flynna; inne klasyfikacje. Komputery wektorowe: rozkazy skalarnie a rozkazy wektorowe - idea komputera wektorowego, przegląd rozkazów wektorowych. Przykłady komputerów wektorowych, zastosowania komputerów wektorowych. Komputery macierzowe: ogólna koncepcja, model SIMD w realizacji rozkazów, sieci łączące, przykłady komputerów macierzowych. Model SIMD we współczesnych procesorach superskalarnych. Karty graficzne i architektura CUDA. Przetwarzanie wielowątkowe – model SIMT. Systemy wieloprocesorowe: Systemy ze wspólną pamięcią: utrzymywanie spójności pamięci podręcznych, protokołów MESI sposoby łączenia elementów systemu - wspólna magistrala, systemy wielomagistralowe, przełącznica krzyżowa, pamięć wieloportowa, wielostopniowe sieci połączeń, sieci nieblokujące

Closa. Systemy o architekturze NUMA. Przykłady komercyjnych systemów z pamięcią wspólną. Systemy z pamięcią rozproszoną: model MPP, sieci łączące, rola procesora w pośrednictwie przesyłu komunikatów - systemy pierwszej i drugiej generacji, rozwój systemów z pamięcią rozproszoną na przykładzie systemów wykorzystujących procesory firmy Intel i firmy IBM. Systemy MPP na liście Top500. Klastry komputerów: definicja i własności. Sieci łączące klastrów: topologia „grubego drzewa”, sieci Infiniband. Klastry Beowulf. Klastry o wysokiej wydajności. Fizyczna budowa klastrów: systemy stelażowe i kasetowe. Przykłady klastrów. Klastry na liście Top500. Klastry o wysokiej niezawodności. Czynniki tworzące wysoką niezawodność klastrów: redundancja węzłów, dostęp do wspólnych zasobów, mechanizmy kontrolujące funkcjonowanie węzłów. Heterogeniczne systemy komputerowe – wsparcie procesorów klasycznych CPU przez procesory graficzne GPU.

Zajęcia laboratoryjne

Szczegółowy zestaw ćwiczeń laboratoryjnych:

SPARC – Programowanie niskopoziomowe procesora SPARC

CUDA – Programowanie równoległe na procesorach graficznych (GPU) w języku C/C++ z odpowiednimi rozszerzeniami

PVM – Programowanie równoległe z dynamicznym podziałem zadań oparte o przesył komunikatów z wykorzystaniem równoległej maszyny wirtualnej

Mosix – Wykorzystanie klastra stacji roboczych do obliczeń równoległych i zrównoważenia obciążenia

JavaSpaces – Programowanie równoległe z wykorzystaniem pamięci współdzielonej i rozproszonej w języku Java

AS/400 – Komunikacja i metody dostępu do danych w systemie AS/400

20. Egzamin: tak; pisemny, dwuczęściowy.

21. Literatura podstawowa:

A.S. Tanenbaum, Strukturalna organizacja systemów komputerowych. Helion, 2006

W. Stallings, Organizacja i architektura systemu komputerowego. Projektowanie systemu a jego wydajność. WNT, W-wa, 2004.

22. Literatura uzupełniająca:

L. Null, J. Lobur, Struktura organizacyjna i architektura systemów komputerowych, Helion, 2004

J. Kitowski, Współczesne systemy komputerowe, CCNS, Kraków, 2000.

S. Kozielski, Z. Szczerbiński, Komputery równoległe, architektura, elementy programowania. WNT, Warszawa, 1994

23. Nakład pracy studenta potrzebny do osiągnięcia efektów kształcenia		
Lp.	Forma zajęć	Liczba godzin kontaktowych / pracy studenta
1	Wykład	30/20
2	Ćwiczenia	0/0
3	Laboratorium	30/25
4	Projekt	0/0
5	Seminarium	0/0
6	Inne	10/15
	Suma godzin	70/60
24. Suma wszystkich godzin: 130		
25. Liczba punktów ECTS:¹ 5 (sem. 5 – 2, sem. 6 – 3)		
26. Liczba punktów ECTS uzyskanych na zajęciach z bezpośrednim udziałem nauczyciela akademickiego: 3		
27. Liczba punktów ECTS uzyskanych na zajęciach o charakterze praktycznym (laboratoria, projekty): 1		
26. Uwagi:		

Zatwierdzono:

.....
 (data i podpis prowadzącego) (data i podpis dyrektora instytutu/kierownika katedry/
 Dyrektora Kolegium Języków Obcych/kierownika lub
 dyrektora jednostki międzywydziałowej)

¹ 1 punkt ECTS – 25-30 godzin.