



Architektury komputerów

Karta opisu przedmiotu

Informacje podstawowe

Kierunek studiów Informatyka Techniczna	Cykl dydaktyczny 2024/2025
Specjalność -	Kod przedmiotu MIFTS.li2O.86b8adae74f5aa66ee6b0a3831667713.24
Jednostka organizacyjna Wydział Inżynierii Metali i Informatyki Przemysłowej	Języki wykładowe polski
Poziom kształcenia Studia inżynierskie I stopnia	Obligatoryjność Obowiązkowy
Forma studiów Stacjonarne	Blok zajęciowy Przedmioty ogólne
Profil studiów Ogólnoakademicki	Przedmiot powiązany z badaniami naukowymi Tak
Koordynator przedmiotu	Adam Mrozek
Prowadzący zajęcia	Adam Mrozek, Zbigniew Bublński

Okres Semestr 2	Forma zaliczenia Zaliczenie	Liczba punktów ECTS 4
	Forma prowadzenia i godziny zajęć Wykład: 28 Ćwiczenia laboratoryjne: 28	

Cele kształcenia dla przedmiotu

C1	W ramach przedmiotu studenci poznają: - budowę i zasadę działania współczesnego systemu mikroprocesorowego, - podstawy programowania w assemblerze (x86-64, Linux), - budowę plików wykonywalnych i bibliotek, - sposób wykonywania obliczeń zmiennoprzecinkowych, - wpływ podstawowych technik optymalizacji i wektoryzacji kodu oraz wykorzystania pamięci cache na wydajność programu.
----	---

Efekty uczenia się dla przedmiotu

Kod	Efekty w zakresie	Kierunkowe efekty uczenia się	Metody weryfikacji
Wiedzy - Student zna i rozumie:			
W1	Student zna podstawowe pojęcia związane z architekturą komputerów i techniką cyfrową	IFT1A_W04	Kolokwium
W2	Student zna budowę i zasady działania elementów techniki cyfrowej, procesorów i systemów mikroprocesorowych	IFT1A_W04	Kolokwium
W3	Student zna wpływ architektury procesora, komputera oraz optymalizacji kodu na wydajność obliczeń	IFT1A_W04	Kolokwium
Umiejętności - Student potrafi:			
U1	Student umie projektować proste układy cyfrowe oraz pisać, modyfikować i uruchamiać proste programy w assemblerze.	IFT1A_U01, IFT1A_U04, IFT1A_U06	Wykonanie ćwiczeń laboratoryjnych
Kompetencje społecznych - Student jest gotów do:			
K1	Student ma świadomość odpowiedzialności za wykonaną pracę.	IFT1A_U09, IFT1A_K01	Wykonanie ćwiczeń laboratoryjnych

Treści programowe zapewniające uzyskanie efektów uczenia się dla modułu zajęć

Student poznaje działanie systemu mikroprocesorowego. Od prostych układów cyfrowych, poprzez zasadę pracy współczesnych procesorów i pamięci, po współpracę z I/O, systemem operacyjnym i bibliotekami.

Nakład pracy studenta

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane aktywności
Wykład	28
Ćwiczenia laboratoryjne	28
Przygotowanie do zajęć	20
Samodzielne studiowanie tematyki zajęć	20
Egzamin lub kolokwium zaliczeniowe	2
Dodatkowe godziny kontaktowe	2
Łączny nakład pracy studenta	Liczba godzin 100
Liczba godzin kontaktowych	Liczba godzin 56

* godzina (lekcyjna) oznacza 45 minut

Treści programowe

Lp.	Treści programowe	Efekty uczenia się dla przedmiotu	Formy prowadzenia zajęć
1.	1. Zapoznanie ze środowiskiem programowania w języku asemblera. 2. Pisanie prostych programów w asemblerze, wywoływanie funkcji systemowych. 3. Analiza kodu asemblera generowanego przez kompilatory. 4. Budowa bibliotek, łączenie asemblera z C/C++. 5. Programowanie jednostki zmiennoprzecinkowej. 6. Wektoryzacja obliczeń.	W1, W2, W3, U1, K1	Ćwiczenia laboratoryjne
2.	1. Wstęp: rozwój maszyn liczących, podstawowe koncepcje i typy architektur komputerowych. 2. Podstawy teorii układów cyfrowych. 3. Budowa układów arytmetycznych. Niskopoziomowe algorytmy obliczeń, znaczniki stanu jednostki arytmetyczno-logicznej. 4. Budowa procesora, lista rozkazów, architektury CISC i RISC. 5. Przetwarzanie potokowe. 6. Procesory superskalarne. 7. Pamięć komputera: technologia, organizacja i hierarchia. 8. Pamięć podręczna i wirtualna. 9. Operacje I/O, system przerwań.	W1, W2, W3	Wykład

Informacje rozszerzone

Metody i techniki kształcenia:

Wykład, Demonstracja, instruktaż, Dyskusja

Rodzaj zajęć	Metody zaliczenia	Warunki zaliczenia przedmiotu
Wykład	Kolokwium	
Ćwiczenia laboratoryjne	Wykonanie ćwiczeń laboratoryjnych, Kolokwium	

Warunki i sposób zaliczenia poszczególnych form zajęć, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu

Wykład: punktowany test wielokrotnego wyboru (na ostatnim wykładzie).

Ocena z części wykładowej: zależna od liczby zdobytych punktów (procentowo, w odniesieniu do maksymalnej liczby punktów możliwych do zdobycia z testu).

Ćwiczenia laboratoryjne: dwa punktowane kolokwia.

Ocena z laboratoriów: zależna od liczby zdobytych punktów (procentowo, w odniesieniu do maksymalnej liczby punktów możliwych do zdobycia z obu kolokwiów).

Progi procentowe: zgodne z regulaminem studiów AGH.

Dodatkowe punkty poprawiające ocenę można zdobyć za aktywność na zajęciach.

Przewidziane są dwa terminy poprawkowe testu oraz dwa terminy poprawkowe dot. laboratoriów (wyznaczone w czasie sesji).

Warunkiem koniecznym do zaliczenia całego modułu jest otrzymanie pozytywnej oceny z części laboratoryjnej.

Sposób obliczania oceny końcowej

Średnia ważona ocen z laboratorium (66%) i testu zaliczeniowego (34%).

Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach

Gdy istnieje możliwość - opuszczone zajęcia laboratoryjne należy odrobić z inną grupą studencką.

W przypadku braku takiej możliwości - student musi wyrównać zaległości we własnym zakresie.

Wymagania wstępne i dodatkowe

Znajomość podstaw informatyki i programowania w języku C/C++.

Zasady udziału w poszczególnych zajęciach, ze wskazaniem, czy obecność studenta na zajęciach jest obowiązkowa

Wykład: studenci uczestniczą w zajęciach poznając kolejne zagadnienia zgodnie z programem przedmiotu. Studenci powinni na bieżąco zadawać pytania i wyjaśniać wątpliwości. Rejestracja audiowizualna wykładu wymaga zgody prowadzącego.

Ćwiczenia laboratoryjne: obecność jest obowiązkowa. Studenci wykonują ćwiczenia laboratoryjne zgodnie z materiałami udostępnionymi przez prowadzącego. Studenci są zobowiązani do przygotowania się do wykonywanego ćwiczenia, potrzebna jest również wiedza z zajęć poprzednich. Przygotowanie do zajęć może zostać zweryfikowane w formie ustnej (pytanie) lub pisemnej (kartkówka).

Dopuszczalne są dwie nieusprawiedliwione nieobecności na laboratoriach.

Trzecia nieusprawiedliwiona nieobecność obniża ocenę o jeden stopień.

Cztery (i kolejne) nieusprawiedliwione nieobecności na zajęciach laboratoryjnych oznaczają brak możliwości zaliczenia przedmiotu.

Literatura

Obowiązkowa

1. John L. Hennessy, David A. Patterson - Computer Architecture, Fifth Edition: A Quantitative Approach, Morgan Kaufmann; 2011
2. David A. Patterson, John L. Hennessy - Computer Organization and Design, Fourth Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design), Morgan Kaufmann, 2014.
3. Dokumentacja dostarczana przez producentów mikroprocesorów, elementów systemu komputerowego oraz oprogramowania.

Badania i publikacje

Badania

1. Analiza niezmienności zapisów w pamięci flash drukarki fiskalnej EMAR-TEMPO. Główny wykonawca, praca na zlecenie firmy EMAR, Politechnika Krakowska, dokumentacja do wglądu u prowadzącego zajęcia.

Publikacje

1. Real-Time Evolutionary Optimization of Metallurgical Processes Using ARM Microcontroller / Adam MROZEK, Waław Kuś, Tadeusz Burczyński // Computer Methods in Materials Science, 2016 vol. 16 no. 1
2. Hybrid parallel evolutionary algorithm in optimization of 2D graphene-like materials — Poszukiwanie nowych, płaskich materiałów grafenopodobnych przy użyciu hybrydowego algorytmu optymalizacji / Adam MROZEK, Waław Kuś, Tadeusz Burczyński // Computer Methods in Materials Science, 2015 vol. 15 no. 1
3. Memetic optimization of graphene-like materials on Intel PHI coprocessor / Waław Kuś, Adam MROZEK, Tadeusz Burczyński // W: Artificial intelligence and soft computing : 15th international conference, ICAISC 2016 : Zakopane, Poland, June 12-16, 2016 : proceedings, Pt. 1 / eds. Leszek Rutkowski, [et al.]. — Switzerland : Springer International Publishing, cop. 2016. — (Lecture Notes in Artificial Intelligence ; ISSN 0302-9743 ; 9692). — ISBN: 978-3-319-39377-3 ; e-ISBN: 978-3-319-39378-0

Kierunkowe efekty uczenia się

Kod	Treść
IFT1A_K01	Rozumie potrzebę krytycznej oceny posiadanej wiedzy i odbieranych treści; uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych oraz zasięgania opinii ekspertów w przypadku trudności z samodzielnym rozwiązaniem problemu
IFT1A_U01	Potrafi wykorzystywać posiadaną wiedzę związaną z informatyką techniczną i inżynierią materiałową - formułować i rozwiązywać złożone i nietypowe problemy oraz wykonywać zadania w warunkach nie w pełni przewidywalnych przez: - właściwy dobór źródeł i informacji z nich pochodzących, dokonywanie oceny, krytycznej analizy i syntezy tych informacji, - dobór oraz stosowanie właściwych metod i narzędzi, w tym zaawansowanych technik informacyjno-komunikacyjnych
IFT1A_U04	Potrafi przy identyfikacji i formułowaniu specyfikacji zadań inżynierskich związanych z informatyką techniczną oraz ich rozwiązywaniu: - wykorzystywać metody analityczne, symulacyjne i eksperymentalne, - dostrzegać ich aspekty systemowe i pozatechniczne, w tym aspekty etyczne, - dokonywać wstępnej oceny ekonomicznej proponowanych rozwiązań i podejmowanych działań inżynierskich związanych z informatyką techniczną; dokonywać krytycznej analizy sposobu funkcjonowania istniejących rozwiązań technicznych i oceniać te rozwiązania
IFT1A_U06	Potrafi projektować, zgodnie z zadaną specyfikacją, oraz wykonywać, używając odpowiednio dobranych metod, technik, narzędzi i materiałów, typowe dla informatyki technicznej proste systemy (jak np. oparte na bazach danych, zawierające interfejs graficzny, realizujące symulacje zjawisk fizyczne, wykorzystujące sprzęt równoległy)
IFT1A_U09	Potrafi samodzielnie planować i realizować własne uczenie się przez całe życie
IFT1A_W04	Zna i rozumie w zaawansowanym stopniu wybrane zagadnienia z zakresu programowania proceduralnego i obiektowego oraz równoległego i mobilnego, algorytmów i struktur danych, architektur komputerów, systemów operacyjnych, sieci komputerowych, inżynierii oprogramowania, baz danych, metod numerycznych, grafiki komputerowej, optymalizacji, systemów wbudowanych, inżynierii internetu, modelowania komputerowego