



Karta przedmiotu

Nazwa i kod przedmiotu	Architektura komputerów, PG_00047659						
Kierunek studiów	Informatyka						
Data rozpoczęcia studiów	październik 2026 r.	Rok akademicki realizacji przedmiotu			2027/2028		
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć			Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	2	Język wykładowy			polski		
Semestr studiów	3	Liczba punktów ECTS			6.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			egzamin		
Jednostka prowadząca	Wydziały Politechniki Gdańskiej -> Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Architektury Systemów Komputerowych						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr inż. Tomasz Dziubich				
	Prowadzący zajęcia z przedmiotu		dr inż. Tomasz Dziubich				
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	30.0	15.0	15.0	0.0	0.0	60
	W tym liczba godzin zajęć na odległość: 0.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów		Udział w konsultacjach		Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	60		7.0		83.0	150
Cel przedmiotu	Celem przedmiotu jest przekazanie wiedzy w zakresie pojęć związanych z architekturą komputerów oraz wiedzy dotyczącej podstawowych mechanizmów funkcjonowania procesorów na poziomie ISA, a także przedstawienie najnowszych trendów w konstrukcjach wewnętrznych procesorów.						
Efekty uczenia się przedmiotu	Efekt kierunkowy		Efekt z przedmiotu			Sposób weryfikacji i oceny efektu	
	[K6_U03] potrafi zaprojektować, zgodnie z zadaną specyfikacją, oraz wykonać typowe dla kierunku studiów proste urządzenie, obiekt, system lub zrealizować proces, używając odpowiednio dobranych metod, technik, narzędzi i materiałów, korzystając ze standardów i norm inżynierskich, stosując właściwe dla kierunków studiów technologie i wykorzystując doświadczenie zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską		Student koduje programy na poziomie instrukcji procesora, uruchamia i testuje programy; Student potrafi dokonać integracji modułów oprogramowania w języku wysokiego i niskiego poziomu			[SU1] Ocena realizacji zadania	

Treści przedmiotu	<p>Treści przedmiotu - wykład</p> <p>1. Wprowadzenie, zasady zaliczania, literatura 2. Model komputera von Neumanna, język maszynowy i asembler 3. Ewolucja sprzętu i oprogramowania komputerów, architektury Intel i AMD 32/64 4. Tryby pracy procesora (tryb jądra, tryb użytkownika) 5. Pamięć główna (operacyjna) 6. Fizyczne struktury informacji 7. Rejestry ogólnego przeznaczenia, rejestry sterujące i stanu 8. Pobieranie i wykonywanie instrukcji, cykl rozkazowy 9. Instrukcje sterujące przepływem sterowania 10. Zasady programowania na poziomie rozkazów procesora, funkcje typowych rozkazów 11. Modyfikacje adresowe bezpośrednio i pośrednio 12. Elementy programowania w asemblerze: mnemoniki instrukcji, formaty wierszy źródłowych, zmienne i etykiety, dyrektywy, porównanie składni asemblerów typu Intel i AT&T 13. Makroprzetwarzanie 14. Technika asemblacji programów, licznik lokacji, asemblacja jedno- i dwuprzbiegowa, sprawozdanie z asemblacji 15. Organizacja stosu 16. Instrukcje sterujące bezwarunkowe, wywołanie (call) i powrót z podprogramu (ret) 17. Przekazywanie parametrów do podprogramów 18. Sprzętowe wspomaganie przekazywania parametrów z użyciem stosu, ramka stosu 19. Techniki dostępu do zmiennych statycznych i dynamicznych 20. Programowanie mieszane, interfejs ABI, typowe standardy wywoływania funkcji (Pascal, C, StdCall) 21. Usługi systemowe i ich wywoływanie, interfejs API, tablica deskryptorów przerwań w architekturze IA32 22. Interfejsy programowania w systemach Windows i Linux, przykłady funkcji usługowych 23. Inicjalizacja pracy komputera, rola systemu BIOS, funkcje usługowe systemu BIOS 24. Ogólne zasady i przesłanki kodowania instrukcji procesora 25. Podstawowy format rozkazów w architekturze IA32 26. Kodowanie instrukcji sterujących 27. Typy i formaty danych: liczby binarne ze znakiem i bez znaku, kodowanie BCD 28. Kodowanie tekstów: kody ASCII, Windows, ISO, Unicode, UTF8 29. Operacje arytmetyczne, identyfikacja nadmiaru 30. Działania na liczbach wielokrotnej długości 31. Technika porównywania zawartości rejestrów i komórek pamięci, przegląd instrukcji sterujących warunkowych 32. Operacje na pojedynczych bitach, przesunięcia logiczne i cykliczne (obroty) 33. Organizacja pętli rozkazowych, realizacja sprzętowa operacji na blokach danych 34. Podstawowe koncepcje sterowania pracą urządzeń zewnętrznych 35. Sterowanie urządzeń poprzez współadresowalny obszar pamięci lub poprzez przestrzeń adresową we/wy 36. Pamięć ekranu w trybie tekstowym i graficznym jako przykład obszaru współadresowalnego 37. Przykłady komunikacji szeregowej i równoległej 38. Przerwania sprzętowe i ich obsługa, priorytety przerwań, przerwania maskowalne i niemaskowalne 39. Elementy techniki obsługi przerwań sprzętowych w komputerach PC, odwzorowanie linii przerwań w elementy tablicy deskryptorów przerwań 40. Obsługa zegara systemowego, zegar czasu rzeczywistego (RTC) 41. Wyjątki procesora, przerwania sprzętowe a programowe 42. Przesyłanie danych za pomocą DMA 43. Formaty liczb zmiennoprzecinkowych (standard IEEE 754) 44. Koprocessor arytmetyczny jako maszyna stosowa, przykłady obliczeń 45. Wybór opcji obliczeń, rejestr stanu i rejestr sterujący koprocessora 46. Obsługa wyjątków w trakcie obliczeń (niedomiary, nadmiar, niedokładny wynik, nieliczby) 47. Hierarchia pamięci w komputerach: rejestry, pamięć podręczna, pamięć główna (operacyjna), pamięć masowa 48. Zasady działania, własności, porównanie różnych typów pamięci używanych w komputerach 49. Koncepcja pamięci wirtualnej jako integracji pamięci operacyjnej i dyskowej 50. Realizacja pamięci wirtualnej za pomocą stronicowania, sprzętowa transformacja adresów w architekturze Intel/AMD 51. Sprzętowe wspomaganie transformacji adresów za pomocą pamięci asocjacyjnej (TLB) 52. Pamięci podręczne dla instrukcji i danych 53. Algorytmy dostępu do pamięci podręcznej 54. Przetwarzanie potokowe, konflikty sterowania, przewidywanie skoków, optymalizacja kodu programu 55. Komputery CISC i RISC, okna rejestrów w procesorach RISC 56. Architektury wielowątkowe i wielordzeniowe 57. Prawo Amdahla skalowalność systemu komputerowego</p>														
Wymagania wstępne i dodatkowe	Nie ma wymagań														
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	<table border="1"> <thead> <tr> <th>Sposób oceniania (składowe)</th> <th>Próg zaliczeniowy</th> <th>Składowa oceny końcowej</th> </tr> </thead> <tbody> <tr> <td>Ćwiczenia</td> <td>32.0%</td> <td>25.0%</td> </tr> <tr> <td>Egzamin</td> <td>10.0%</td> <td>50.0%</td> </tr> <tr> <td>Laboratorium</td> <td>76.0%</td> <td>25.0%</td> </tr> </tbody> </table>	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej	Ćwiczenia	32.0%	25.0%	Egzamin	10.0%	50.0%	Laboratorium	76.0%	25.0%		
Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej													
Ćwiczenia	32.0%	25.0%													
Egzamin	10.0%	50.0%													
Laboratorium	76.0%	25.0%													
Zalecana lista lektur	<p>Podstawowa lista lektur</p> <p>Uzupełniająca lista lektur</p> <p>Adresy eZasobów</p>	<p>Null L., Lobur J.: Struktura organizacyjna i architektura systemów komputerowych. Wyd. Helion 2004. Tanenbaum A.S.: Strukturalna organizacja systemów komputerowych, wyd. Helion Lewis D.: Między asemblerem a językiem C, wyd. RM Wróbel E.: Asembler. Ćwiczenia praktyczne.: Wyd. Helion</p> <p>Nie ma wymagań</p>													
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania															
Zajęcia praktyczne w ramach przedmiotu	Nie dotyczy														

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.