



Karta przedmiotu

Nazwa i kod przedmiotu	Architektura komputerów, PG_00058927						
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	niestacjonarne	Sposób realizacji			na uczelni		
Rok studiów	2	Język wykładowy			polski		
Semestr studiów	3	Liczba punktów ECTS			4.0		
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. Marcin Narloch				
	Prowadzący zajęcia z przedmiotu		dr inż. Marcin Narloch				
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	15.0	0.0	0.0	30
	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	30		4.0		66.0	100
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_U09] potrafi dokonać krytycznej analizy sposobu funkcjonowania istniejących rozwiązań technicznych związanych z kierunkiem studiów i ocenić te rozwiązania, a także wykorzystać zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską doświadczenie związane z utrzymaniem urządzeń, obiektów i systemów technicznych typowych dla kierunku studiów	Student rozumie i integruje podprogramy systemowe niskiego poziomu wymagające użycia różnych standardów, sposobów komunikacji lub kodowania znaków.	[SU1] Ocena realizacji zadania
	[K6_U04] potrafi wykorzystywać posiadaną wiedzę z zakresu metod i techniki programowania oraz dobrać i zastosować właściwe metody i narzędzia programistyczne w tworzeniu oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, charakterystycznych dla danego kierunku studiów	Student potrafi dokonać integracji modułów oprogramowania w języku wysokiego i niskiego poziomu.	[SU1] Ocena realizacji zadania
	[K6_W04] zna i rozumie w zaawansowanym stopniu zasady, metody i techniki programowania oraz zasady tworzenia oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, specyficznych dla kierunku studiów, a także organizację pracy systemów wykorzystujących komputery lub te urządzenia	Student zna: architektury komputerów, zasady zarządzania i adresacji pamięci operacyjnej, technikę asemblacji programu, działanie systemów przerwań i współpracy z systemami zewnętrznymi.	[SW1] Ocena wiedzy faktograficznej
Treści przedmiotu	<p>Treści przedmiotu - wykład</p> <ol style="list-style-type: none"> <li>1. Wprowadzenie i zasady zaliczenia przedmiotu, literatura.</li> <li>2. Organizacja ogólna komputera wg von Neumanna.</li> <li>3. Elementy architektury x86 na poziomie rejestrów: rejestry ogólnego przeznaczenia, znaczniki, tryb rzeczywisty i chroniony.</li> <li>4. Pamięć fizyczna i wirtualna, adresowanie pamięci, problem kolejności bajtów (little/big endian)).</li> <li>5. Rozwój architektury x86.</li> <li>6. Model programowy procesora, cykl rozkazowy, klasyfikacja instrukcji wg sposobu oddziaływania na wskaźnik instrukcji (licznik rozkazów).</li> <li>7. Zasady programowania na poziomie rozkazów procesora, funkcje typowych rozkazów.</li> <li>8. Tryby adresowania.</li> <li>9. Elementy programowania w assemblerze: mnemoniki instrukcji, formaty wierszy źródłowych, zmienne i etykiety, dyrektywy.</li> <li>10. Operacje stosu Wywołanie (call) i powrót z podprogramu (ret), przekazywanie parametrów do podprogramów.</li> <li>11. Sprzętowe wspomaganie przekazywania parametrów z użyciem stosu, ramka stosu.</li> <li>12. Programowanie mieszane, interfejs ABI, typowe standardy wywoływania funkcji (Pascal, C, StdCall).</li> <li>13. Usługi systemowe i ich wywoływanie, interfejs API, tablica deskryptorów przerwań w architekturze x86.</li> <li>14. Typy i formaty danych: liczby binarne ze znakiem i bez znaku.</li> <li>15. Kodowanie tekstów: kody ASCII, Windows, ISO, Unicode, UTF-8, UTF-16.</li> <li>16. Operacje arytmetyczne, identyfikacja nadmiaru.</li> <li>17. Operacje na pojedynczych bitach, przesunięcia logiczne i cykliczne (obroty).</li> <li>18. Podstawowe koncepcje sterowania pracą urządzeń zewnętrznych Sterowanie urządzeń poprzez współadresowalny. obszar pamięci lub poprzez przestrzeń adresową we/wy.</li> <li>19. Przerwania sprzętowe i ich obsługa, priorytety przerwań, przerwania maskowalne i niemaskowane.</li> <li>20. Elementy techniki obsługi przerwań sprzętowych w komputerach PC, odwzorowanie linii przerwań w elementy tablicy deskryptorów przerwań.</li> <li>21. Wyjątki procesora, przerwania sprzętowe a programowe.</li> <li>22. Przesyłanie danych za pomocą DMA.</li> <li>23. Formaty liczb zmiennoprzecinkowych (standard IEEE 754).</li> <li>24. Koprocesor arytmetyczny, przykłady obliczeń.</li> <li>25. Wybór opcji obliczeń, rejestr stanu i rejestr sterujący koprocesora.</li> <li>26. Hierarchia pamięci w komputerach: rejestry, pamięć podręczna, pamięć główna (operacyjna), pamięć masowa.</li> <li>27. Koncepcja pamięci wirtualnej jako integracji pamięci operacyjnej i dyskowej.</li> <li>28. Przetwarzanie potokowe, konflikty sterowania, przewidywanie skoków.</li> <li>29. Komputery CISC i RISC.</li> <li>30. Architektury wielowątkowe i wielordzeniowe, instrukcje dla zastosowań multimedialnych (MMX, SSE).</li> </ol>		
Wymagania wstępne i dodatkowe			

Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Ćwiczenia praktyczne	0.0%	50.0%
	Egzamin pisemny	0.0%	50.0%
Zalecana lista lektur	Podstawowa lista lektur	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 assemblerem a językiem C, wyd. RM  John L. Hennessy, David A. Patterson, "Computer Architecture: A Quantitative Approach", 2014 Morgan Kaufmann	
	Uzupełniająca lista lektur	Nie ma wymagan	
	Adresy eZasobów		
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania			
Zajęcia praktyczne w ramach przedmiotu	Nie dotyczy		

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.