

SYLABUS
DOTYCZY CYKLU KSZTAŁCENIA 2020-2024
 Rok akademicki 2021/2022

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	<i>architektura systemów komputerowych</i>
Kod przedmiotu	
Nazwa jednostki prowadzącej kierunek	<i>Kolegium Nauk Przyrodniczych</i>
Nazwa jednostki realizującej przedmiot	<i>Kolegium Nauk Przyrodniczych</i>
Kierunek studiów	<i>informatyka</i>
Poziom studiów	<i>studia inżynierskie I-go stopnia</i>
Profil	<i>ogólnoakademicki</i>
Forma studiów	<i>stacjonarne</i>
Rok i semestr studiów	<i>II rok, 4 semestr</i>
Rodzaj przedmiotu	<i>inżynierski przedmiot kierunkowy</i>
Język wykładowy	<i>język polski, język angielski</i>
Koordinator	<i>dr Krzysztof Balicki</i>
Imię i nazwisko osoby prowadzącej / osób prowadzących	<i>dr Krzysztof Balicki, mgr inż. Jarosław Szkoła</i>

1.1. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Inne (jakie?)	Liczba pkt ECTS
4	15			30					4

1.2. Sposób realizacji zajęć

zajęcia w formie tradycyjnej i z wykorzystaniem metod i technik kształcenia na odległość

1.3 Forma zaliczenia przedmiotu (z toku)

zaliczenie z oceną

2. WYMAGANIA WSTĘPNE

Podstawy programowania w języku C.

3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE

3.1 Cele przedmiotu

C1	Celem przedmiotu jest zapoznanie słuchaczy z architekturą współczesnych systemów komputerowych ze szczególnym uwzględnieniem procesora i koprocessora arytmetycznego.
C2	Ważną cechą tych zajęć jest duża liczba przykładów i ćwiczeń praktycznych, które umożliwiają studentom poznanie podstaw programowania niskopoziomowego.

3.2 Efekty uczenia się dla przedmiotu/ modułu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu	Odniesienie do efektów kierunkowych
EK_01	Student posiada wiedzę na temat organizacji systemu komputerowego oraz architektury mikroprocesora i innych podzespołów jednostki systemowej.	K_W03
EK_02	Student zna narzędzia i techniki wykorzystywane w programowaniu niskopoziomym takie jak asembler, linker, disassembler, debugger i asmloder. Potrafi łączyć kod w asemblerze z programami w języku C. Zna sposoby komunikacji procesora z koprocessorem arytmetycznym i jednostkami wektorowymi.	K_W07
EK_03	Student posiada umiejętność programowania procesora i koprocessora arytmetycznego w stopniu podstawowym.	K_U05

3.3 Treści programowe

A. Problematyka wykładu

Wprowadzenie – system komputerowy, klasyfikacja architektur komputerowych, hierarchia pamięci. Maszyna von Neumanna, architektury Harvard, Princeton, Harvard-Princeton.
Dane – typy, reprezentacje, organizacja i adresowanie pamięci. Porządek Big-Endian i Little-Endian. Wyrównanie naturalne. Dane wektorowe.
Budowa modelu programowego — rejestry, tryby adresowania, model operacji warunkowych, lista instrukcji. Porównanie modelu programowego w podejściu CISC i RISC.
Model programowy procesorów 16, 32 i 64 bitowych rodziny Intel.
Arytmetyka stała i zmiennopozycyjna. Jednostki zmiennopozycyjne i wektorowe.
Wybrane konwencje wywołań dla kodu 32-bitowego.
Konwencje wywołań dla kodu 64-bitowego.
Analiza stanu stosu procesora i koprocessora arytmetycznego.
Analiza programów rekurencyjnych, nierekurencyjnych i niskopoziomowych.
Asembler NASM, asmloder, disassembler, debugger GDB, linker.

B. Problematyka ćwiczeń laboratoryjnych

Konwersje liczb między różnymi systemami liczbowymi.
Reprezentacja danych, konwencje little-endian i big-endian.

Mapa pamięci procesu w języku C.
Obsługa programów: NASM, asmloder, linker, disassembler, debugger DBG.
Pseudoinstrukcje asemblera NASM.
Operacje przesłań.
Tryby adresowania.
Operacje arytmetyczne, logiczne, porównań, bitowe, przesunięcia i rotacje bitów.
Instrukcje skoków, skoków warunkowych, wywołania procedur.
Analiza stanu stosu procesora i koprocesora arytmetycznego.
Analiza programów rekurencyjnych, nierekurencyjnych i niskopoziomowych.
Łączenie kodu w asemblerze z programami w języku C.
Mierzenie czasu wykonania procedur i funkcji.

3.4 Metody dydaktyczne

Wykład: wykład z prezentacją multimedialną
 Laboratorium: rozwiązywanie zadań, projekt.

4. METODY I KRYTERIA OCENY

4.1 Sposoby weryfikacji efektów uczenia się

Symbol efektu	Metody oceny efektów uczenia się (np.: kolokwium, egzamin ustny, egzamin pisemny, projekt, sprawozdanie, obserwacja w trakcie zajęć)	Forma zajęć dydaktycznych (w, ćw, ...)
EK_01	obserwacja w trakcie zajęć, projekt	lab
EK_02	obserwacja w trakcie zajęć, projekt	lab
EK_03	kolokwium, projekt	lab

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

<p>Wykład</p> <p>Zaliczenie bez oceny, efekty kierunkowe EK_01, EK_02 właściwe dla wykładu weryfikowane są w trakcie zajęć laboratoryjnych oraz na obronach projektów.</p> <p>Laboratorium</p> <p>Warunkiem zaliczenia laboratorium jest zaliczenie kolokwium i wykonanie projektu. Ocena końcowa jest średnią ocen z kolokwium i projektu. Aby zaliczyć kolokwium, należy zdobyć przynajmniej połowę maksymalnej liczby punktów. Oceny z kolokwiów przyznawane są proporcjonalnie do liczby zdobytych punktów. Do tematów projektów przypisane są oceny referencyjne zależne od stopnia ich trudności. Ocena z obrony projektu może różnić się od oceny referencyjnej o pół stopnia. Pod uwagę brana jest również aktywność na zajęciach, która może obniżyć lub podwyższyć ocenę końcową o pół stopnia.</p>

5. CAŁKOWITY NAKŁAD PRACY STUDENTA POTRZEBNY DO OSIĄGNIĘCIA ZAŁOŻONYCH EFEKTÓW W GODZINACH ORAZ PUNKTACH ECTS

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe wynikające z harmonogramu studiów	45
Inne z udziałem nauczyciela (udział w konsultacjach, egzaminie)	2
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	53
SUMA GODZIN	100
SUMARYCZNA LICZBA PUNKTÓW ECTS	4

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

7. LITERATURA

<p>Literatura podstawowa:</p> <ol style="list-style-type: none"> 1. D. Patterson, J. Hennessy, Computer Organization and design, Elsevier 2005 2. J. Biernat, Architektura komputerów, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2005
<p>Literatura uzupełniająca:</p> <ol style="list-style-type: none"> 1. P. Metzger, M. Siemieniecki, Anatomia PC : architektura komputerów zgodnych z IBM PC, Helion, Gliwice 2003 2. Materiały firmowe - dokumenty techniczne dostępne w sieci WWW – MIPS, Intel, AMD 3. Specyfikacje: Application Binary Interface