**Nazwa przedmiotu:**

Architektura komputerów

**Koordynator przedmiotu:**

Rajmund Kożuszek

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne

**Kod przedmiotu:**

ARKO

**Semestr nominalny:**

2 / rok ak. 2021/2022

**Liczba punktów ECTS:**

4

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

1. liczba godzin kontaktowych – 64 h, w tym
 obecność na wykładach: 45 h,
 obecność na zajęciach laboratoryjnych: 15 h,
 udział w konsultacjach związanych z problematyką poruszaną na wykładzie/laboratorium: 4 h,

1. praca własna studenta – 56 godz., w tym
 lektura dokumentacji i rozwiązywanie zadań w ramach przygotowań do zajęć laboratoryjnych 30 godz.,
 udział w dyskusji w trakcie wykładu: 1 godz.,
 analiza literatury i materiałów wykładowych związana z przygotowaniem do kolejnych wykładów, zajęć laboratoryjnych, projektu, instalacja oprogramowania: 10 godz.,
 przygotowanie do kolokwiów: 15 godz.

Łączny nakład pracy studenta wynosi 120 godz., co odpowiada 4 pkt. ECTS.

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

2,5 pkt. ECTS, co odpowiada 64 godz. kontaktowym.

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

1 pkt. ECTS, co odpowiada 25 godz.

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład:  | 45h |
| Ćwiczenia:  | 0h |
| Laboratorium:  | 15h |
| Projekt:  | 0h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

Podstawy programowania.

**Limit liczby studentów:**

150

**Cel przedmiotu:**

Przedmiot ma za zadanie zapoznanie studentów z architekturą współczesnych procesorów i komputerów. W trakcie wykładu są omawiane podstawy teoretyczne architektury komputerów, zasady konstruowania modeli programowych procesorów oraz implementacji języków wysokiego poziomu, zasady działania procesorów, konstrukcja hierarchii pamięci oraz systemowe aspekty architektury procesora. Wykład prezentuje również podstawowe zasady współpracy komputera z urządzeniami peryferyjnymi oraz architektury współczesnych komputerów.

**Treści kształcenia:**

Wykład

1. (3h)Regulamin przedmiotu i organizacja zajęć. Pojęcia podstawowe. Taksonomie architektur komputerów. Maszyna von Neumanna, architektury Harvard, Princeton, Harvard-Princeton. Pojęcie hierarchii pamięci.
2. (3h)Dane w pamięci komputera. Adresowanie pamięci. Dane wielobajtowe w pamięci - konwencje Big i Little Endian. Wyrównanie danych i jego skutek dla alokacji struktur zajętości pamięci. Dane wektorowe.
3. (3h)Wymagania języków wysokiego poziomu wobec architektury procesora. Identyfikacja klas pamięci. Stos. Języki maszynowe i asemblerowe. Przekazywanie sterowania pomiędzy procedurami – konwencje wołania. Mapa przestrzeni adresowej programu. Ramka stosu w rejestrach i w pamięci. Model procesora – realizacja przekazywania sterownia. Odwzorowanie modelu w x86. Dostęp do ramki stosu przy użyciu wskaźnika stosu i wskaźnika ramki.
4. (6h)Model programowy procesora — zestawy rejestrów, tryby adresowania, model operacji warunkowych. Konstrukcja listy instrukcji - podejścia CISC, RISC, post-RISC. Formaty instrukcji. Prezentacja modelu programowego RISC (MIPS) i CISC (x86, x86-64). Jednostki wektorowe. ARMv8 jako przykład architektury post-RISC.
5. (6h)Wprowadzenie do zajęć laboratoryjnych. Program w komputerze – proces tworzenia i wykonanie – kompilacja, asemblacja, konsolidacja, ładowanie do pamięci. Programowanie asemblerowe. Programowanie hybrydowe. Techniki programowania asemblerowego – realizacja wybranych struktur sterowania i algorytmów.
6. (1h)Kolokwium sprawdzające.
7. (4h)Procesor jednocyklowy – projekt. Wykonanie instrukcji przez procesor jednofazowy – ograniczenia modelu programowego. Procesor wielocyklowy z jednostką sterującą, mechanizmy przyspieszające - prefetch i kolejka instrukcji. Potok. Synchronizacja potoku – hazardy i opóźnienia. Superpotok. Potokowa implementacja procesora CISC – potok CISC, potok z transkodowaniem.
8. (2h)Architektury superskalarne. Synchronizacja wielopotokowej jednostki wykonawczej. Prezentacja współczesnych architektur wielopotokowych z transkodowaniem instrukcji.
9. (3h)Kieszenie jako warstwa hierarchii pamięci — zasada działania, warianty organizacji. Ilościowy model wydajności hierarchii pamięci. Zachowanie kieszeni przy zapisie danych. Współpraca kieszeni w wielowarstwowej hierarchii pamięci. Problem spójności zawartości pamięci operacyjnej i kieszeni.
10. (2h)Problemy wydajnościowe w architekturach superskalarnych i superpotokowych – opóźnienie skoków, opóźnienie danych. Przewidywanie skoków. Sterowanie działaniem kieszeni. Mechanizmy przyspieszające odwołania do danych. Procesory wielowątkowe i wielordzeniowe. Fuzja instrukcji.
11. (1h)Wymagania systemu operacyjnego - zasady ochrony zasobów. Systemowy model programowy procesora. Poziomy zaufania. Maszyny wirtualne.
12. (3h)Zarządzanie pamięcią. Funkcje systemu zarządzania pamięcią. Metody zarządzania pamięcią — prosta relokacja, segmentacja, stronicowanie. Optymalizacja stronicowania. Stronicowanie w procesorach 64-bitowych. Zarządzanie działaniem hierarchii pamięci — zarządzanie pamięcią a kieszenie - sprzężenie kieszeni i jednostki zarządzania pamięcią.
13. (3h)Wyjątki – definicja, klasyfikacja, zastosowanie poszczególnych klas wyjątków. Zasady obsługi wyjątków. Obsługa przerwań – priorytety. Specyfika obsługi błędów – mechanizmy umożliwiające obsługę błędu i kontynuacje wykonania programu. Specyficzne wyjątki — podwójny błąd i restart. Implementacja systemu pamięci wirtualnej. Przechwycenia w maszynie wirtualnej.
14. (2h)Wejście-wyjście. Model logiczny i programowy sterownika wejścia wyjścia. Obsługa urządzeń wejścia-wyjścia z aktywnym oczekiwaniem, z przerwaniami i przy użyciu bezpośredniego dostępu do pamięci. Urządzenia pamięci masowej – działanie, charakterystyka użytkowa pamięci dyskowych i nieulotnych pamięci półprzewodnikowych.
15. (1h)Struktury komputerów – przegląd. Architektury: pamięciocentryczna, jedno-, dwu- i trójszynowe, architektury z połączeniami punkt-punkt. Ewolucja struktur komputerów osobistych.
16. (1h)Wprowadzenie do architektury systemowej x86.
17. (1h)Kolokwium sprawdzające.

 Laboratorium
Zasady programowania w językach asemblerowych. Programowanie asemblerowe MIPS. Programowanie hybrydowe x86. Konwencje wołania procedur. Przekazywanie parametrów.
Zadania projektowe: procedury asemblerowe dołączane do programu w języku wysokiego poziomu; algorytmy operacji arytmetycznych, optymalizacja kodu.

1. (1h) Ćwiczenie wprowadzające – zapoznanie ze środowiskiem.
1. (2h) Prosty program w środowisku symulatora procesora MIPS.
2. (3h) Prosty program hybrydowy x86.
3. (4h) Program, asemblerowy w środowisku symulatora MIPS.
4. (5h) Program hybrydowy dla x86-64. Procedura asemblerowa realizująca nietrywialne przetwarzanie (np. implementacja wybranych algorytmów obliczeniowych z arytmetyki komputerów, przetwarzanie przy użyciu jednostki wektorowej) łączona z programem w języku C.

**Metody oceny:**

 Uwagi realizacyjne
Wykład:
- realizowany w formie tradycyjnej, materiały wykładowe udostępnione studentom.

Projekty laboratoryjne:
- przy ocenianiu należy zwrócić uwagę na poprawność programów i ich zgodność z wymaganiami standardów ABI.

 Formy weryfikacji efektów kształcenia
Treści wykładowe:
Weryfikacja poprze kolokwia sprawdzające mające postać zbioru krótkich pytań otwartych i zadań.
Projekty laboratoryjne:
ocena poprawności rozwiązań programowych, zgodności ze standardami, umiejętności związanych z translacją i uruchamianiem programów oraz umiejętności wprowadzania modyfikacji do projektów.

**Egzamin:**

nie

**Literatura:**

1. Patterson, Hennessy: Computer Patterson, Hennessy: Computer Organization and Design, The Hardware/Software Interface, Elsevier
2. MIPS32 Architecture for Programmers, MIPS Technologies (www.mips.com)
3. Intel® 64 and IA-32 Architectures Software Developer's Manual, Intel Corp.
4. AMD64 Architecture Programmer's Manual, AMD Inc.
5. SYSTEM V APPLICATION BINARY INTERFACE, Intel386 Architecture Processor Supplement, Fourth Edition, 1990-1996 The Santa Cruz Operation, Inc.
6. J. Biernat, Arytmetyka Komputerów, Warszawa, PWN 1996.
7. Prezentacje wykładowe
8. Przykładowe pytania testowe
9. Przykłady programów – zadań laboratoryjnych
10. MARS Simulator, Missouri State University
11. GNU CC
12. NASM x86 assembler, sf.net

**Witryna www przedmiotu:**

https://usosweb.usos.pw.edu.pl/kontroler.php?\_action=katalog2/przedmioty/pokazPrzedmiot&prz\_kod=103C-INxxx-ISP-ARKO

**Uwagi:**

(-)

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka W01:**

ma podstawową wiedzę w zakresie organizacji transmisji danych w sprzęcie komputerowym i mechanizmów sterujących działaniem komputera

Weryfikacja:

laboratorium, sprawdziany

**Powiązane charakterystyki kierunkowe:** W03

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o, III.P7S\_WG

**Charakterystyka W02:**

rozumie działanie komputera jako złożonego systemu cyfrowego

Weryfikacja:

laboratorium, sprawdziany

**Powiązane charakterystyki kierunkowe:** W04

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o, III.P6S\_WG

**Charakterystyka W03:**

rozumie związek pomiędzy projektem danych w oprogramowaniu i wydajnością programu

Weryfikacja:

laboratorium, sprawdziany

**Powiązane charakterystyki kierunkowe:** W05

**Powiązane charakterystyki obszarowe:** I.P6S\_WG.o, P6U\_W

**Charakterystyka W04:**

rozumie związek pomiędzy projektem oprogramowania i jego wydajnością jako konsekwencją architektury komputera

Weryfikacja:

laboratorium, sprawdziany

**Powiązane charakterystyki kierunkowe:** W08

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o, III.P6S\_WG

**Charakterystyka W05:**

rozumie mechanizmy umożliwiające działanie systemu operacyjnego we współczesnym komputerze

Weryfikacja:

laboratorium, sprawdziany

**Powiązane charakterystyki kierunkowe:** W10

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o, III.P6S\_WG

**Charakterystyka W06:**

zna metody reprezentacji danych dźwiękowych i obrazowych i aspekty logiczne i wydajnościowe ich przetwarzania przez komputer

Weryfikacja:

laboratorium, sprawdziany

**Powiązane charakterystyki kierunkowe:** W12

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o

### Profil ogólnoakademicki - umiejętności

**Charakterystyka U01:**

potrafi posługiwać się dokumentacją techniczną i na jej podstawie tworzyć projekty oprogramowania i sprzętu

Weryfikacja:

laboratorium

**Powiązane charakterystyki kierunkowe:** U01, U07, U11, U12

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o, I.P6S\_UK, I.P6S\_UU

**Charakterystyka U02:**

potrafi przeprowadzić analizę wydajności i optymalizację struktury sprzętu i oprogramowania

Weryfikacja:

laboratorium

**Powiązane charakterystyki kierunkowe:** U02, U04

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o

**Charakterystyka U03:**

potrafi przetestować zaprojektowany sprzęt i oprogramowanie

Weryfikacja:

laboratorium

**Powiązane charakterystyki kierunkowe:** U05, U06, U03

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o

**Charakterystyka U04:**

potrafi tworzyć projekt w zadanych ramach czasowych, indywidualnie i w zespole

Weryfikacja:

laboratorium

**Powiązane charakterystyki kierunkowe:** U08, U10

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UO, I.P6S\_UK, III.P6S\_UW.o

**Charakterystyka U05:**

potrafi stworzyć dokumentację projektu

Weryfikacja:

laboratorium

**Powiązane charakterystyki kierunkowe:** U09

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UK

### Profil ogólnoakademicki - kompetencje społeczne

**Charakterystyka K01:**

ma świadomość zakresu własnej wiedzy i postępu w dziedzinie techniki, jaką jest budowa komputerów

Weryfikacja:

sprawdziany

**Powiązane charakterystyki kierunkowe:** K01

**Powiązane charakterystyki obszarowe:** P6U\_K, I.P6S\_KK

**Charakterystyka K02:**

rozumie wpływ decyzji projektowych na własności rozwiązania finalnego

Weryfikacja:

laboratorium

**Powiązane charakterystyki kierunkowe:** K02, K03

**Powiązane charakterystyki obszarowe:** P6U\_K, I.P6S\_KR, I.P6S\_KK

**Charakterystyka K03:**

rozumie i potrafi objaśniać funkcjonowanie komputera i jego składników

Weryfikacja:

laboratorium, sprawdziany

**Powiązane charakterystyki kierunkowe:** K05

**Powiązane charakterystyki obszarowe:** P6U\_K, I.P6S\_KO