**Nazwa przedmiotu:**

Zaawansowane architektury procesorów

**Koordynator przedmiotu:**

Grzegorz MAZUR

**Status przedmiotu:**

Fakultatywny dowolnego wyboru

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne

**Kod przedmiotu:**

ZAP

**Semestr nominalny:**

7 / rok ak. 2018/2019

**Liczba punktów ECTS:**

4

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

30 - wykłady
30 - studium literatury
15 - zajęcia projektowe - prezentacja i dyskusja projektów
40 - praca własna nad projektami
5 - przygotowanie do sprawdzianów

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

2

**Język prowadzenia zajęć:**

polski

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

2

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

|  |  |
| --- | --- |
| Wykład:  | 30h |
| Ćwiczenia:  | 0h |
| Laboratorium:  | 0h |
| Projekt:  | 15h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

Umiejętność programowania w języku C oraz programowania hybrydowego C+assembler. Znajomość architektury komputerów na poziomie wykładu w kursie podstawowym. (Przedmioty: Programowanie komputerów, Architektura komputerów.)

**Limit liczby studentów:**

32

**Cel przedmiotu:**

Przedmiot zapoznaje studentów ze współczesnymi rozwiązaniami w zakresie architektury procesorów uniwersalnych, w tym szczegółów systemowego modelu programowego procesorów x86 i wersji 64-bitowej - x86-64, maszyn wirtualnych i mechanizmów podnoszenia wydajności procesora i komputera.

**Treści kształcenia:**

Treść wykładu

Rozszerzenia aplikacyjnego modelu programowego x86

Instrukcje przesłań ze sterowaniem hierarchią pamięci. Instrukcje
operacji bitowych BMI i TBM. Jednostki wektorowe - rejestry, formaty
danych i instrukcje MMX, 3DNow!, SSE, AVX.

Architektura systemowa procesorów rodziny x86

Ewolucja architektury systemowej 286-386-486-Pentium-P6.
Systemowy model programowy procesorów x86. Rola jednostki segmentacji -
segmenty systemowe GDT, LDT, IDT, TSS. Tablice IOPM, IRM. Praca
jednostki segmentacji w trybie rzeczywistym, chronionym i V86. Tryb
FRM. Struktura systemu operacyjnego udostępniającego tryb V86.
Jednostka stronicowania. Tryb V86 - metody obsługi sytuacji
wyjątkowych. Rozszerzenia trybu V86. Tryb SMM. Rozszerzenia modelu
programowego użytkownika i ich odbicie w modelu systemowym.

Architektura x86-64

Zestaw rejestrów i tryby adresowania. Zmiany w liście instrukcji
w porównaniu z x86. Systemowy model programowy w trybie 64-bitowym -
praca jednostki segmentacji, TSS, IDT. Stronicowanie w trybie
64-bitowym.

Maszyny wirtualne

Metody realizacji maszyny wirtualnej. Interakcja pomiędzy maszyną
wirtualną i hypervisorem. Zarządzanie zasobami maszyny wirtualnej.
Sprzętowa obsługa wirtualizacji. Start maszyny wirtualnej, przełączanie
"światów", przechwycenia. Zarządzanie pamięcią w systemie z
wirtualizacją. Dwupoziomowa translacja adresów. Migracja żywej maszyny
wirtualnej - problemy, implementacja.

Architektura ARMv7 i ARMv8

Zestaw rejestrów, lista instrukcji. Dostosowania do programowania
w językach wysokiego poziomu. Model systemowy. System przerwań.

Wybrane aspekty implementacji procesorów

Współczesne metody przewidywania skoków. Predyktory
wielopoziomowe. Predyktory pętli. Przewidywanie pobierania danych.
Instrukcje pobierania danych.

Architektura IBM POWER

Model programowy aplikacyjny i systemowy. Stronicowanie z
odwrotną tablicą stron. Implementacja POWER6 i POWER7.

Zakres projektu
Realizacja modułu oprogramowania o charakterze systemowym (sytuacje wyjątkowe, zarządzanie pamięcią, wirtualizacja) lub eksperymentalnym (optymalizacja kodu, testowanie działania i wydajności procesora, symulacja działania wybranych struktur) na komputerze klasy PC.

**Metody oceny:**

Testy z wiadomości wykładowych, projekt oprogramowania.

**Egzamin:**

nie

**Literatura:**

1. Patterson, Hennessy: Computer Patterson, Hennessy: Computer Organization and Design, The Hardware/Software Interface, Elsevier
2. MIPS32 Architecture for Programmers (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. The microarchitecture of Intel, AMD and
VIA CPUs; An optimization guide for assembly programmers and
compiler makers; Agner Fog. Copenhagen University College of Engineering. Copyright © 1996 - 2012

**Witryna www przedmiotu:**

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

**Uwagi:**

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka ZAP\_W01:**

Ma wiedzę z zakresu problemów i współczesnych tendencji w rozwoju architektur komputerowych.

Weryfikacja:

Testy

**Powiązane charakterystyki kierunkowe:** K\_W08, K\_W14

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

**Charakterystyka ZAP\_W02:**

Rozumie źródła ograniczeń wydajności wynikające z budowy komputera i mechanizmy umożliwiające zwiększenie wydajności.

Weryfikacja:

Testy, zadania projektowe.

**Powiązane charakterystyki kierunkowe:** K\_W14

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

**Charakterystyka ZAP\_W03:**

Rozumie mechanizmy systemowe występujące w komputerach, a w szczególności zarządzanie pamięcią, mechanizmy wyjątków i wirtualizacji.

Weryfikacja:

Testy, zadania projektowe.

**Powiązane charakterystyki kierunkowe:** K\_W08, K\_W09, K\_W10, K\_W14, K\_W19

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

### Profil ogólnoakademicki - umiejętności

**Charakterystyka ZAP\_U01:**

Potrafi zidentyfikować ograniczenia wydajności oprogramowania, zbadać wydajność oprogramowania i zwiększyć ją przy użyciu stosownych technik.

Weryfikacja:

Testy, zadania projektowe.

**Powiązane charakterystyki kierunkowe:** K\_U05, K\_U07, K\_U09, K\_U13, K\_U15, K\_U20, K\_U21

**Powiązane charakterystyki obszarowe:** I.P6S\_UW, III.P6S\_UW.1.o, I.P6S\_UK, I.P6S\_UU, III.P6S\_UW.4.o, III.P6S\_UW.3.o

**Charakterystyka ZAP\_U02:**

Potrafi tworzyć fragmenty oprogramowania systemowego.

Weryfikacja:

Projekt.

**Powiązane charakterystyki kierunkowe:** K\_U13, K\_U05

**Powiązane charakterystyki obszarowe:** III.P6S\_UW.4.o, I.P6S\_UW, III.P6S\_UW.1.o

**Charakterystyka ZAP\_U03:**

Rozumie potrzebę pogłębiania wiedzy i korzystania z literatury technicznej.

Weryfikacja:

kolokwia, projekt

**Powiązane charakterystyki kierunkowe:** K\_U09, K\_UK01

**Powiązane charakterystyki obszarowe:** I.P6S\_UU