



Programowanie niskopoziomowe

Karta opisu przedmiotu

## Informacje podstawowe

Kierunek studiów : Informatyka analityczna

Ścieżka : -

Jednostka organizacyjna : Wydział Matematyki i Informatyki

Poziom kształcenia : pierwszego stopnia

Forma studiów : studia stacjonarne

Profil studiów : ogólnoakademicki

Obligatoryjność : obowiązkowy

Cykl kształcenia : 2022/23

Kod przedmiotu : UJ.WMIIANS.180.03299.22

Języki wykładowe : polski

Dyscypliny : Informatyka

Klasyfikacja ISCED : 0613 Tworzenie i analiza oprogramowania i aplikacji

Kod USOS : WMI.TCS.PN.OL

Koordinator przedmiotu

Jakub Kozik

Prowadzący zajęcia

Jakub Kozik, Jan Derbisz

---

Okres Semestr 4	Forma weryfikacji uzyskanych efektów uczenia się	Liczba punktów ECTS 6.0
	egzamin	
	Forma prowadzenia i godziny zajęć	
	wykład: 30 ćwiczenia laboratoryjne: 30	

## Cele kształcenia dla przedmiotu

---

- Kiedy wyczerpiemy algorytmiczne i projektowe sposoby przyspieszenia programu komputerowego pozostaje nam jedynie optymalizacja kodu na niskim poziomie. Kompilatory, maszyny wirtualne i generatory kodu z roku na rok stają się coraz to potężniejszymi narzędziami, ale aby osiągnąć najwyższą wydajność często musimy optymalizować kod własnoręcznie. Na kursie dowiesz się jak
- C1 dobrze zarządzać pamięcią, dlaczego warto korzystać ze struktur kursorowych i jak żyć w zgodzie z pamięcią cache. Pokażemy ci jak nie zepsuć wydajności programu wielowątkowego złą synchronizacją. Nauczmy cię jak wycisnąć ósme poty z nowoczesnych procesorów wykorzystując ich wektorowe możliwości i wielu innych przydatnych technik, które poszerzą twój programistyczny warsztat.

## Efekty uczenia się dla przedmiotu

Kod	Efekty w zakresie	Kierunkowe efekty uczenia się	Metody weryfikacji
Wiedzy – Student zna i rozumie:			
W1	podstawy architektury współczesnych procesorów.	IAN_K1_W13	egzamin pisemny, zaliczenie na ocenę
Umiejętności – Student potrafi:			
U1	programować w assemblerze wykorzystując zaawansowane funkcjonalności procesora (w tym instrukcje SIMD oraz atomowe).	IAN_K1_U03, IAN_K1_U09	egzamin pisemny, zaliczenie na ocenę, zadania programistyczne
U2	student wykorzystuje elementy niskopoziomowej optymalizacji w programowaniu w językach wysokiego poziomu (C/C++).	IAN_K1_U03, IAN_K1_U05, IAN_K1_U09	egzamin pisemny, zaliczenie na ocenę, zadania programistyczne

## Bilans punktów ECTS

Forma aktywności studenta	Średnia liczba godzin* przeznaczonych na zrealizowane rodzaje zajęć
wykład	30
ćwiczenia laboratoryjne	30
samodzielne rozwiązywanie zadań komputerowych	75

przygotowanie do egzaminu	40	
Łączny nakład pracy studenta	Liczba godzin 175	ECTS 6.0

\* godzina (lekcyjna) oznacza 45 minut

## Treści programowe

Lp.	Treści programowe	Efekty uczenia się dla przedmiotu
1.	Podstawy architektury x86_64.	W1
2.	Niskopoziomowe interfejsy systemu Linux oraz języków C i C++.	W1, U1
3.	Elementy mikroarchitektury procesora (w tym przetwarzanie potokowe oraz działanie cache'u).	W1, U2
4.	Instrukcje SIMD (Single Instruction Multiple Data).	W1, U1, U2
5.	Kod binarny oraz programy modyfikujące kod (Self Modifying Code).	U1
6.	Niskopoziomowe aspekty programowania wielowątkowego.	W1, U2

## Informacje rozszerzone

Metody nauczania :

wykład z prezentacją multimedialną, dyskusja, rozwiązywanie zadań, ćwiczenia laboratoryjne

Rodzaj zajęć	Formy zaliczenia	Warunki zaliczenia przedmiotu
wykład	egzamin pisemny	Warunkiem zaliczenia przedmiotu jest uzyskanie pozytywnej oceny z ćwiczeń oraz zdanie egzaminu na ponad 50% punktów.
ćwiczenia laboratoryjne	zaliczenie na ocenę, zadania programistyczne	Warunkiem zaliczenia ćwiczeń jest uzyskanie ponad 50% punktów za realizację zadań programistycznych. Aktywność na ćwiczeniach może podwyższyć ocenę ale nie wpływa na fakt zaliczenia.

## Wymagania wstępne i dodatkowe

- Umiejętność programowania w C oraz C++ - Znajomość podstaw budowy systemów operacyjnych.

## Literatura

### Obowiązkowa

- Dokumentacja techniczna dla architektury x86\_64 dla procesorów AMD lub Intel, dostępne na stronie producenta.

**Dodatkowa**

1. Randall Hyde, The Art of Assembly Language Programming (dostępna na stronie autora)
2. Agner Fog, Optimization manuals (dostępne na stronie autora)