

PAŃSTWOWA WYŻSZA SZKOŁA ZAWODOWA W NOWYM SĄCZU

KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2016/2017

Instytut Techniczny

Kierunek studiów: Informatyka

Profil: Praktyczny

Forma studiów: Stacjonarne

Kod kierunku: 11.3

Stopień studiów: I

Specjalności: Informatyka stosowana

1 PRZEDMIOT

NAZWA PRZEDMIOTU	Programowanie niskopoziomowe
KOD PRZEDMIOTU	IT 11.3 PIS C2 16/17
KATEGORIA PRZEDMIOTU	Przedmioty specjalnościowe
LICZBA PUNKTÓW ECTS	2
SEMESTRY	2

2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

SEMESTR	WYKŁAD	ĆWICZENIA	LABORATORIUM	PROJEKT	SEMINARIUM
2	15			15	

3 CELE PRZEDMIOTU

Cel 1 Zapoznanie studenta z podstawowymi technikami, narzędziami i algorytmami przydatnymi w optymalizacji, synchronizacji procesów i profilowaniu oprogramowania.

Cel 2 Zdobywanie wiedzy w zakresie języków assemblerowych i technik programowania.

Cel 3 Poznanie sposobów łączenia kodu napisanego w językach niskiego poziomu z kodem napisanym w językach wysokiego poziomu.

Cel 4 Nabycie umiejętności optymalizacji i rozpoznawania elementów krytycznych tworzonego oprogramowania.

Cel 5 Pozyskanie umiejętności łączenia kodu napisanego w języku assembler z kodem napisanym w językach wysokiego poziomu.

Cel 6 Nabycie umiejętności tworzenia oprogramowania w języku assembler dla różnych platform sprzętowych.



4 WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

- a Podstawy programowania w językach C, C++, elektroniki cyfrowej.
- b Znajomość architektury komputera i systemów operacyjnych.

5 EFEKTY KSZTAŁCENIA

- EK1** Wiedza: Student zna podstawowe techniki, narzędzia i algorytmy przydatne w optymalizacji, synchronizacji procesów i profilowaniu oprogramowania.
- EK2** Wiedza: Student ma szczegółową wiedzę w zakresie języków assemblerowych i technik programowania.
- EK3** Wiedza: Student zna sposoby łączenia kodu napisanego w językach niskiego poziomu z kodem zaimplementowanym w językach C i C++.
- EK4** Umiejętności: Student potrafi odnaleźć krytyczne elementy tworzonego oprogramowania i dokonać ich optymalizacji.
- EK5** Umiejętności: Student potrafi łączyć elementy kodu napisanego przy pomocy assemblera z kodem napisanym za pomocą języków wyższego poziomu.
- EK6** Umiejętności: Student potrafi napisać program w języku assembler dla różnych platform sprzętowych.

6 TREŚCI PROGRAMOWE

WYKŁAD

LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W1	Architektura komputera, ogólna charakterystyka procesora.	2
W2	Podstawy assemblera dla procesorów rodziny x86 i x86_64.	2
W3	Proces tworzenia programu, kompilacji, oprogramowanie narzędziowe, kompilator, konsolidator. Biblioteki. Tworzenie programu wielomodułowego.	2
W4	Segmentacja, praca w trybie chronionym, pamięć wirtualna. Mikroarchitektury, NetBurst, Core. Model programowy. Rejestry, przerwania, tryby adresacji.	2
W5	Obsługa instrukcji procesora koprocesor, MMX, SSE, AVX.	2
W6	Łączenie modułów ASM z językami wysokiego poziomu.	2
W7	Techniki programowania assemblerowego. Metody optymalizacji kodu możliwe do realizacji wyłącznie na poziomie assemblera. Błędy i złe nawyki w programowaniu assemblerowym.	3
	RAZEM	15

PROJEKT

LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
P1	Projekt programu wykorzystującego charakterystyczne cechy języka assembler - użycie kompilatora NASM (Linux), MASM32 (Windows)	3
P2	Projekt aplikacji wykorzystującej przerwania i instrukcje koprocesora arytmetycznego.	3
P3	Projekt aplikacji w języku assembler korzystający z instrukcji MMX i SSE.	3
P4	Projekt oprogramowania łączący kod napisany w języku assembler z elementami napisanymi w językach C i C++.	6
	RAZEM	15



7 METODY DYDAKTYCZNE

M1 Wykłady

M2 Praca w grupach

M3 Projekty

8 OBCIĄŻENIE PRACĄ STUDENTA

FORMA AKTYWNOŚCI	ŚREDNIA LICZBA GODZIN NA ZREALIZOWANIE AKTYWNOŚCI
Godziny kontaktowe z nauczycielem akademickim, w tym:	
Godziny wynikające z planu studiów	30
Konsultacje przedmiotowe	2
Egzaminy i zaliczenia w sesji	2
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	2
Opracowanie wyników	4
Przygotowanie raportu, projektu, prezentacji, dyskusji	10
SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA	50
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	2

9 SPOSOBY OCENY

OCENA FORMUJĄCA

F1 Egzamin

F2 Projekt zespołowy

F3 Projekt indywidualny

F4 Odpowiedź ustna

OCENA AKTYWNOŚCI BEZ UDZIAŁU NAUCZYCIELA AKADEMICKIEGO

1 Projekt indywidualny

KRYTERIA OCENY

EFEKT KSZTAŁCENIA 1		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 1
NA OCENĘ 3	Student zna podstawowy techniki optymalizacji oprogramowania, zna możliwości koprocatora arytmetycznego i MMX.	wykład, projekt	Egzamin pisemny, projekt, odpowiedź ustna.
NA OCENĘ 4	Student zna techniki optymalizacji oprogramowania, zna możliwości koprocatora arytmetycznego i MMX, AVX.		



NA OCENĘ 5	Student zna techniki optymalizacji oprogramowania, synchronizacji procesów. Zna możliwości koprocatora arytmetycznego i MMX, AVX, SSE.		
EFEKT KSZTAŁCENIA 2		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 2
NA OCENĘ 3	Student zna podstawy tworzenia oprogramowania dla procesorów rodziny x86 i x86_64, wykorzystania stosu, rejestrów, przerwań.	wykład, projekt	Egzamin pisemny, projekt, odpowiedź ustna.
NA OCENĘ 4	Student zna podstawy tworzenia oprogramowania dla procesorów rodziny x86 i x86_64, wykorzystania stosu, rejestrów, przerwań koprocatora, MMX i SSE, tryby adresacji.		
NA OCENĘ 5	Student zna podstawy tworzenia oprogramowania dla procesorów rodziny x86 i x86_64, wykorzystania stosu, rejestrów, przerwań koprocatora, MMX i SSE, tryby adresacji a także pamięci wirtualnej i bibliotek.		
EFEKT KSZTAŁCENIA 3		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 3
NA OCENĘ 3	Student opanował tylko podstawowe elementy umożliwiające łączenie kodu assemblerowego z językiem C++.	wykład, projekt	Egzamin pisemny, projekt, odpowiedź ustna.
NA OCENĘ 4	Student zna sposoby łączenia kodu napisanego w językach niskiego poziomu z kodem zaimplementowanym w językach C i C++.		
NA OCENĘ 5	Student zna sposoby łączenia kodu napisanego w językach niskiego poziomu z kodem zaimplementowanym w językach C i C++, opanował wiadomości pozwalające na optymalizację aplikacji tworzonych przy użyciu języków niskiego i wysokiego poziomu.		
EFEKT KSZTAŁCENIA 4		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 4
NA OCENĘ 3	Student potrafi odnaleźć krytyczne elementy kodu, nie potrafi samodzielnie dokonać optymalizacji kodu na poziomie asemblera.	projekt	Egzamin pisemny, projekt, odpowiedź ustna.
NA OCENĘ 4	Student potrafi odnaleźć krytyczne elementy kodu, ma problemy zw stosowaniem metod optymalizacji kodu możliwej do realizacji wyłącznie na poziomie asemblera.		
NA OCENĘ 5	Student potrafi odnaleźć krytyczne elementy kodu i stosować metody optymalizacji kodu możliwe do realizacji wyłącznie na poziomie asemblera.		
EFEKT KSZTAŁCENIA 5		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 5



NA OCENĘ 3	Student ma problemy - wymaga ukierunkowania - złączeniem elementów kodu napisanego przy pomocy assemblera z kodem napisanym za pomocą języków wyższego poziomu.	projekt	Projekt, odpowiedź ustna.
NA OCENĘ 4	Student potrafi łączyć elementy kodu napisanego przy pomocy assemblera z kodem napisanym za pomocą języków wyższego poziomu.		
NA OCENĘ 5	Student potrafi łączyć elementy kodu napisanego przy pomocy assemblera z kodem napisanym za pomocą języków wyższego poziomu, świadomie wybiera elementy programu które napisane w języku niskiego poziomu zwiększą jego efektywność.		
EFEKT KSZTAŁCENIA 6		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 6
NA OCENĘ 3	Student potrafi napisać program w języku assembler dla jednej wybranej platformy sprzętowej.	projekt	Projekt, odpowiedź ustna.
NA OCENĘ 4	Student potrafi napisać program w języku assembler dla wybranej wybranej platformy sprzętowej, potrafi także w ograniczonym zakresie tworzyć oprogramowanie dla innych platform.		
NA OCENĘ 5	Student potrafi napisać program w języku assembler dla różnych platform sprzętowych.		

OCENA DO INDEKSU (OCENA PODSUMOWUJĄCA)

Ocena podsumowująca wynika z analizy ocen uzyskanych z egzaminu, projektów indywidualnych i grupowych oraz odpowiedzi ustnej uzyskanych w trakcie trwania przedmiotu.

WARUNKI ZALICZENIA PRZEDMIOTU

- a Uzyskanie oceny pozytywnej z każdego efektu kształcenia.
- b Uzyskanie oceny pozytywnej z egzaminu pisemnego.

10 MACIERZ REALIZACJI PRZEDMIOTU

EFEKTY KSZTAŁCENIA DLA PRZEDMIOTU	ODNIESIENIE DO EFEKTÓW KIERUNKOWYCH	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	METODY DYDAKTYCZNE
EK1	INFP_W11	Cel1	W1, W2, W3, W4, W5, W6, W7, P1, P2, P3, P4	M1, M2, M3
EK2	INFP_W09	Cel2	W2, W3, W4, W5, W6, W7, P1, P2, P3, P4	M1, M2, M3



EFEKTY KSZTAŁCENIA DLA PRZEDMIOTU	ODNIESIENIE DO EFEKTÓW KIERUNKOWYCH	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	METODY DYDAKTYCZNE
EK3	INFP_W09	Cel3	W6, P4	M1, M2, M3
EK4	INFP_UB04	Cel4	W3, W4, P2, P3, P4	M1, M2, M3
EK5	INFP_UO03	Cel5	W6, P4	M1, M2, M3
EK6	INFP_UB08	Cel6	W1, W2, P1, P4	M1, M2, M3

11 WYKAZ LITERATURY

LITERATURA PODSTAWOWA:

- [1] K. R. Irvine — *Asembler dla procesorów intel*, Gliwice, 2003, HELION
- [2] E. Wróbel — *Asembler -ćwiczenia praktyczne*, Gliwice, 2002, HELION
- [3] R. Hyde — *Asembler. Sztuka programowania. Wydanie II*, Gliwice, 2010, HELION

LITERATURA UZUPEŁNIAJĄCA:

- [1] G. Michałek — *Asembler nie tylko dla orłów*, Warszawa, 1997, QUE
- [2] V. Pirogow — *Asembler. Podręcznik programisty*, Gliwice, 2005, HELION

12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

OSOBA ODPOWIEDZIALNA ZA KARTĘ

mgr Grzegorz Litawa (kontakt: glitawa@poczta.onet.pl)

OSOBY PROWADZĄCE PRZEDMIOT

mgr Grzegorz Litawa (kontakt: glitawa@poczta.onet.pl)

13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejscowość, data) (odpowiedzialny za przedmiot) (kierownik zakładu) (dyrektor instytutu)

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....