

(pieczęć wydziału)

KARTA PRZEDMIOTU

1. Nazwa przedmiotu: JĘZYKI ASSEMBLEROWE		2. Kod przedmiotu: MK_19		
3. Karta przedmiotu ważna od roku akademickiego: 2012/2013				
4. Forma kształcenia:		studia pierwszego stopnia		
5. Forma studiów:		studia stacjonarne		
6. Kierunek studiów:		INFORMATYKA (RAu)		
7. Profil studiów:		ogólnoakademicki		
8. Specjalność:				
9. Semestr:		4 i 5		
10. Jednostka prowadząca przedmiot:		Instytut Informatyki		
11. Prowadzący przedmiot:		dr inż. Krzysztof Tokarz		
12. Przynależność do grupy przedmiotów:		przedmioty wspólne		
13. Status przedmiotu:		obowiązkowy		
14. Język prowadzenia zajęć:		polski		
15. Przedmioty wprowadzające oraz wymagania wstępne: Teoria układów cyfrowych, Arytmetyka systemów cyfrowych, Podstawy programowania komputerów, Konstrukcja układów cyfrowych				
16. Cel przedmiotu: Celem przedmiotu jest zapoznanie studenta z podstawową wiedzą i umiejętnościami w zakresie programowania w języku assemblera. Znajomość tych zagadnień ułatwi studentowi dokonywanie wyboru narzędzi programistycznych do rozwiązywania konkretnych zadań, zwłaszcza tych, które wymagają optymalizacji czasowej lub pamięciowej. Umożliwi lepsze zrozumienie działania procesora i komputera oraz bardziej świadome korzystanie z języków wyższego poziomu.				
17. Efekty kształcenia: ¹				
Nr	Opis efektu kształcenia	Metoda sprawdzenia efektu kształcenia	Forma prowadzenia zajęć	Odniesienie do efektów dla kierunku studiów
1.	Ma wiedzę w zakresie: architektury nowoczesnych procesorów i komputerów, systemów operacyjnych, mechanizmów ochrony pamięci i adresowania wirtualnego	EP	WM	K1A_W07 K1A_W11
2.	Ma wiedzę w zakresie implementacji i optymalizacji algorytmów zapisanych w języku assemblera	PS	L	K1A_W10
3.	Ma wiedzę na temat cyklu życia nowoczesnych procesorów	PS	L	K1A_W12 K1A_W13
4.	Ma umiejętność	CL, RP, PS	L, P	K1A_U01

	programowania w języku asemblera i posługiwania się systemami operacyjnymi na poziomie API			
5.	Ma umiejętność przeprowadzania testów oprogramowania niskopoziomowego	RP, PS	P	K1A_U14
6.	Ma umiejętność tworzenia, zgodnie z zadaną specyfikacją, oprogramowania niskopoziomowego	RP, PS	P	K1A_U24

18. Formy zajęć dydaktycznych i ich wymiar (liczba godzin)

Sem 4. W. 30h **Ćw.** -- **L.** 15h **P.** -- **Sem.** --

Sem 5. W. -- **Ćw.** -- **L.** -- **P.** 30h **Sem.** --

19. Treści kształcenia:

Wykład:

- Wprowadzenie. Miejsce języka symbolicznego (assemblera) we współczesnej informatyce oraz w hierarchii języków programowania. Mity i fakty na temat assemblera. MASM, TASM, NASM.
- Architektura mikroprocesorów rodziny x86-32 widziana oczami programisty. Od procesora 8086 do Core i7. Rejestry, znaczniki, organizacja pamięci operacyjnej, adres logiczny fizyczny, liniowy, tryby adresowania, praca procesora w różnych trybach, adresowanie układów we/wy, przerwania i "wyjątki", typy danych.
- Ogólna charakterystyka listy rozkazów rodziny procesorów Intelu x86-32. Formaty rozkazów, podstawowe rozkazy procesora, rozkazy sterujące.
- Koprocessor arytmetyczny, technologia MMX i SSE. Arytmetyka zmiennoprzecinkowa, rozszerzenia MMX, SSE, SSE2, SSE3, SSSE3, SSE4, AVX. (Rozszerzenia procesorów firmy AMD).
- Ogólna charakterystyka języka makroassemblera. Podstawowe elementy języka, stałe, nazwy symboliczne, składnia wiersza programu, słowa kluczowe, dyrektywy i pseudoinstrukcje.
- Wybrane dyrektywy i pseudoinstrukcje. Tworzenie segmentów logicznych, asemlacja warunkowa, makroinstrukcje, przydział danych, połączenia międzymodułowe, struktury danych i rekordy, łańcuchy, bloki powtarzane, sterowanie postacją listingu, procedury, błędy warunkowe, inne.
- Symbole, wyrażenia i operatory, symbole predefiniowane.
- Narzędzia. Program assemblera, narzędzia pomocnicze.
- MASM w wersji dla systemu operacyjnego MS DOS oraz Windows.
- Łączenie modułów assemblerowych z programem w języku wyższego poziomu.
- Optymalizacja kodu programu. Kryteria optymalizacji, zasady ogólne, czas wykonania a zajętość pamięci, Optymalizacja dla rodziny procesorów Intelu.
- Przykłady (wplecione przy okazji różnych omawianych tematów szczegółowych)

Laboratorium:

- Prosty program assemblerowy w środowisku MSDOS (Skrypt [1] ćw.1)
- Program uruchomieniowy. Przykłady użycia rozkazów procesora (Skrypt [1] ćw. 2 i 3)
- Prosty program assemblerowy w środowisku Windows (Skrypt [1] ćw. 4)
- Struktura programu z jednym oknem głównym (Skrypt [1] ćw. 5)
- Program assemblerowy w Visual Studio 2008, łączenie modułów w języku assemblerowym z programem w języku C.

Projekt:

Indywidualny kompletny projekt w języku assemblerowym przygotowany w środowisku Visual Studio

(z wykorzystaniem mechanizmu łączenia modułów w assemblerze oraz C oraz technik tworzenia bibliotek typu dll) lub w MASM32.

20. Egzamin: tak

21. Literatura podstawowa:

1. Wróbel Eugeniusz „Praktyczny kurs assemblera. Wydanie II”, Wydawnictwo Helion, Gliwice 2011.
2. Wróbel Eugeniusz „Programowanie w języku assemblera MASM. Laboratorium”, Wydawnictwo Politechniki Śląskiej, Gliwice 2005.
3. Randal Hyde „Assembler. Sztuka programowania. Wydanie II”, Wydawnictwo Helion, Gliwice 2010.
4. Wróbel Eugeniusz (pod red.) „Assembler. Ćwiczenia praktyczne”, Wydawnictwo Helion, Gliwice 2002.
5. Vlad Pirogov „Assembler. Podręcznik programisty”, Wydawnictwo Helion, Gliwice 2005.
6. Microsoft Macro Assembler Programmer's Guide (obecnie dostępne w Internecie).
7. IA-32 Intel Architecture. Software Developer's Manual (dostępne na www.intel.com):
 - a) Volume 1: Basic Architecture
 - b) Volume 2: Instruction Set Reference
 - c) Volume 3: System Programming Guide

22. Literatura uzupełniająca:

1. Sivarama P. Dandamudi: “Introduction to Assembly Language Programming”, Springer Verlag 1998.
2. Karen Miller “An Assembly Language Introduction to Computer Architecture”, Oxford University Press 1999.
3. Eugeniusz J. Wróbel: „Assembler 8086/88”, Seria "Mikrokomputery", WNT Warszawa 1992, wyd.II.
4. H. Małysiak, B. Pochopień, E. Wróbel „Procesory arytmetyczne”, Seria "Mikrokomputery". WNT Warszawa, 1993.
5. H. Małysiak, B. Pochopień, E. Wróbel „Mikrokomputery klasy IBM PC”, Seria „Mikrokomputery”, WNT Warszawa, 1992, wyd.II.

23. Nakład pracy studenta potrzebny do osiągnięcia efektów kształcenia

Lp.	Forma zajęć	Liczba godzin kontaktowych / pracy studenta
1	Wykład	30 / 10
2	Ćwiczenia	-- / --
3	Laboratorium	15 / 25
4	Projekt	30 / 40
5	Seminarium	-- / --
6	Inne	-- / --
	Suma godzin	75 / 75

24. Suma wszystkich godzin: 150

25. Liczba punktów ECTS:² 6

26. Liczba punktów ECTS uzyskanych na zajęciach z bezpośrednim udziałem nauczyciela akademickiego: 3

27. Liczba punktów ECTS uzyskanych na zajęciach o charakterze praktycznym (laboratoria, projekty): 4

28. Uwagi:

Zatwierdzono:

.....
(data i podpis prowadzącego)

.....
(data i podpis dyrektora instytutu/kierownika katedry
/Dyrektora Kolegium Języków Obcych/
kierownika lub dyrektora jednostki międzywydziałowej)

¹ należy wskazać ok. 5 – 8 efektów kształcenia

² 1 punkt ECTS – 25 - 30 godzin.