

**AGH**AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Nazwa modułu: Zaawansowane programowanie niskopoziomowe

Rok akademicki: 2018/2019 Kod: JIS-2-029-GK-s Punkty ECTS: 4

Wydział: Fizyki i Informatyki Stosowanej

Kierunek: Informatyka Stosowana Specjalność: Grafika komputerowa i przetwarzanie obrazów

Poziom studiów: Studia II stopnia Forma i tryb studiów: Stacjonarne

Język wykładowy: Polski Profil kształcenia: Ogólnoakademicki (A) Semestr: 0

Strona www: —

Osoba odpowiedzialna: dr inż. Świątek
Krzysztof (swientek@agh.edu.pl)Osoby prowadzące: dr hab. inż. Mindur
Bartosz (mindur@agh.edu.pl)
dr inż. Skoczeń
Andrzej (skoczen@fis.agh.edu.pl)
dr inż. Świątek
Krzysztof (swientek@agh.edu.pl)

Krótką charakterystyka modułu

Przedmiot ma zapoznać studenta z zaawansowanym programowaniem niskopoziomowym w systemach mikroprocesorowych np. komunikacją przez interfejsy USB, pisanie sterowników itp..

Opis efektów kształcenia dla modułu zajęć

Kod EKM	Student, który zaliczył moduł zajęć wie/umie/potrafi	Powiązania z EKK	Sposób weryfikacji efektów kształcenia (forma zaliczeń)
Wiedza			
M_W001	Student zna podstawowe cechy i elementy charakterystyczne dla niskopoziomowych języków programowania wraz z budową wewnętrzną typowych mikroprocesorów 8, 32 i 64 bitowych	IS2A_W04	Aktywność na zajęciach, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych
M_W002	Student zna proces kompilacji i konsolidacji kodów źródłowych do kodu maszynowego, z wykorzystaniem narzędzi odpowiednich dla danego typu mikroprocesora	IS2A_W05, IS2A_W04	Aktywność na zajęciach, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych

M_W003	Student zna podstawowe techniki i narzędzia przydatne w optymalizacji i profilowaniu oprogramowania	IS2A_W08, IS2A_W05	Aktywność na zajęciach, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych
Umiejętności			
M_U001	Student potrafi napisać prosty program w języku assembler z wykorzystaniem podstawowych instrukcji procesora w wersjach 8, 32 i 64 bitowych	IS2A_U02, IS2A_U06, IS2A_U01	Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych
M_U002	Student potrafi łączyć elementy asemblera z oprogramowaniem napisanym za pomocą innych języków	IS2A_U02, IS2A_U06, IS2A_U04, IS2A_U01	Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych
M_U003	Student potrafi znaleźć krytyczne elementy tworzonego oprogramowania i dokonać ich optymalizacji	IS2A_U02, IS2A_U04, IS2A_U01	Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych
Kompetencje społeczne			
M_K001	Student potrafi pracować w zespole projektowym. Potrafi samodzielnie zdobyć odpowiednią wiedzę i umiejętności niezbędne do realizacji jego części zadania projektowego.	IS2A_K01, IS2A_K02	Udział w dyskusji, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych
M_K002	Student umie przedstawić wykonany projekt w komunikatywnej prezentacji	IS2A_K01, IS2A_K03, IS2A_K02	Sprawozdanie, Wykonanie projektu

Matryca efektów kształcenia w odniesieniu do form zajęć

Kod EKM	Student, który zaliczył moduł zajęć wie/umie/potrafi	Forma zajęć										
		Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Inne	E-learning
Wiedza												
M_W001	Student zna podstawowe cechy i elementy charakterystyczne dla niskopoziomowych języków programowania wraz z budową wewnętrzną typowych mikroprocesorów 8, 32 i 64 bitowych	-	-	+	-	-	+	-	-	-	-	-
M_W002	Student zna proces kompilacji i konsolidacji kodów źródłowych do kodu maszynowego, z wykorzystaniem narzędzi odpowiednich dla danego typu mikroprocesora	-	-	+	-	-	+	-	-	-	-	-
M_W003	Student zna podstawowe techniki i narzędzia przydatne w optymalizacji i profilowaniu oprogramowania	-	-	+	-	-	+	-	-	-	-	-

Umiejętności													
M_U001	Student potrafi napisać prosty program w języku assembler z wykorzystaniem podstawowych instrukcji procesora w wersjach 8, 32 i 64 bitowych	-	-	+	+	-	-	-	-	-	-	-	-
M_U002	Student potrafi łączyć elementy asemblera z oprogramowaniem napisanym za pomocą innych języków	-	-	+	+	-	-	-	-	-	-	-	-
M_U003	Student potrafi znaleźć krytyczne elementy tworzonego oprogramowania i dokonać ich optymalizacji	-	-	+	+	-	-	-	-	-	-	-	-
Kompetencje społeczne													
M_K001	Student potrafi pracować w zespole projektowym. Potrafi samodzielnie zdobyć odpowiednią wiedzę i umiejętności niezbędne do realizacji jego części zadania projektowego.	-	-	-	+	-	-	-	-	-	-	-	-
M_K002	Student umie przedstawić wykonany projekt w komunikatywnej prezentacji	-	-	-	+	-	-	-	-	-	-	-	-

Treść modułu zajęć (program wykładów i pozostałych zajęć)

Ćwiczenia laboratoryjne

Studenci w ramach prowadzonych zajęć laboratoryjnych będą samodzielnie pisać oraz testować i weryfikować krótkie programy komputerowe ilustrujące tematykę poruszaną w ramach seminarium.

Efekty kształcenia:

student potrafi przedstawić cechy charakterystyczne niskopoziomowych języków programowania

student potrafi przedstawić podstawowe komponenty składowe mikroprocesorów w wersjach 8, 32 i 64 bitowych

student potrafi napisać proste programy wykorzystując podstawowe elementy charakterystyczne dla języka assembler

student potrafi wykorzystywać dostępne oprogramowanie do tworzenia i weryfikacji kodów napisanych w języku asembler.

Student potrafi wykorzystać narzędzia debugujące i profilujące do poprawy jakości swojego oprogramowania.

Ćwiczenia projektowe

Studenci będą w zespołach realizować przydzielane losowo projekty. W ramach projektu należało będzie wykonać oprogramowanie realizujące określone w temacie zadanie z wykorzystaniem dostępnych narzędzi służących do tworzenia, testowania i weryfikacji kodów źródłowych w asemblerze. Dodatkowo zrealizowany projekt powinien posiadać pełną dokumentację funkcjonalności, zastosowanych technik

programowania, algorytmów oraz wszystkie kody źródłowe z odpowiednimi komentarzami.

Efekty kształcenia:

student potrafi przedstawić analizę projektu w raz z odpowiednim podziałem na moduły składowe

student potrafi współpracować w grupie realizując swoją część zadania

student potrafi zademonstrować funkcjonalność stworzonego oprogramowania, która jest zgodna ze specyfikacją oraz przygotować dokumentację.

Zajęcia seminaryjne

Seminarium (czasem wykład w zależności od liczby studentów) będzie obejmowało zagadnienia oprogramowania niskopoziomowego mikroprocesorów 8, 32 i 64 bitowych. Przedstawione zostaną narzędzia pozwalające na tworzenie samodzielnego oprogramowania w języku assembler, a także wykorzystanie jego elementów w językach wyższego poziomu. Seminarium ma także na celu pogłębienie rozumienia działania procesora i systemu operacyjnego, aby wykorzystać tę wiedzę do optymalizowania własnego oprogramowania.

Sposób obliczania oceny końcowej

Ocena końcowa (OK) obliczana jest jako średnia ważona ocen z laboratorium wraz z projektem (LA) i seminarium (SE):

$$OK = 0.7 \times LA + 0.3 \times SE$$

Uzyskanie pozytywnej oceny (OK) wymaga uzyskania wszystkich pozytywnych ocen cząstkowych (LA, SE). Każda ocena cząstkowa liczona jest jako średnia ważona z wszystkich terminów.

Wymagania wstępne i dodatkowe

Znajomość programowania w języku C oraz C++.

Zalecana literatura i pomoce naukowe

Richard Blum "Professional Assembly Language", Wiley 2005

Richard Detmer "Introduction to 80x86 Assembly Language and Computer Architecture", 2001 by Jones and Bartlett Publishers, Inc

Publikacje naukowe osób prowadzących zajęcia związane z tematyką modułu

K. Swientek, M. Banachowicz,

Design of Memory Subsystem for Wide Input Data Range in the SALT ASIC, Proceedings of the 24th International Conference "Mixed Design of Integrated Circuits and Systems", June 22-24, 2017, Bydgoszcz, Poland.

S. Bugiel, R. Dasgupta, M. Firlej, T. Fiutowski, M. Idzik, M. Kopec, J. Moron, K. Swientek, Ultra-Low Power Fast Multi-Channel 10-Bit ADC ASIC for Readout of Particle Physics Detectors IEEE Trans. Nucl. Sci. vol. 63, 2016, pp. 2622-2631.

M. Idzik, K. Swientek, T. Fiutowski, S. Kulis, D. Przyborowski, A 10-Bit Multichannel Digitizer ASIC for Detectors in Particle Physics Experiments, IEEE Trans. Nucl. Sci. vol. 59, 2012, pp. 294-302.

S. Kulis, A. Matoga, M. Idzik, K. Swientek, T. Fiutowski, D. Przyborowski, A general purpose multichannel readout system for radiation detectors, JINST 7 T01004 2012, pp. 294-302.

Informacje dodatkowe

I - Sposób i tryb wyrównania zaległości powstałych wskutek nieobecności studenta na zajęciach: zajęcia laboratoryjne, komputerowe, projektowe, seminaryjne:

Nieobecność na jednych zajęciach laboratoryjnych wymaga od studenta samodzielnego opanowania przerobionego na tych zajęciach materiału. Nieobecność na więcej niż jednych zajęciach wymaga od

studenta samodzielnego opanowania przerabianego na tych zajęciach materiału i jego zaliczenia w formie pisemnej w wyznaczonym przez prowadzącego terminie, lecz nie później jak w ostatnim tygodniu trwania zajęć.

Student który bez usprawiedliwienia opuścił więcej niż 10% zajęć i jego cząstkowe wyniki w nauce były negatywne może zostać pozbawiony, przez prowadzącego zajęcia, możliwości wyrównania zaległości.

Obecność na wykładzie: zgodnie z Regulaminem Studiów AGH.

II - Zasady zaliczania zajęć:

zajęcia laboratoryjne, komputerowe, projektowe: Podstawowym terminem uzyskania zaliczenia jest koniec zajęć w danym semestrze. Student może przystąpić do poprawkowego zaliczenia.

Student który bez usprawiedliwienia opuścił więcej niż 10% zajęć i jego cząstkowe wyniki w nauce były negatywne może zostać pozbawiony, przez prowadzącego zajęcia, możliwości poprawkowego zaliczenia zajęć.

seminarium: warunkiem zaliczenia jest wygłoszenie seminarium z oceną pozytywną i obecność na conajmniej 90% zajęć.

Nakład pracy studenta (bilans punktów ECTS)

Forma aktywności studenta	Obciążenie studenta
Udział w zajęciach seminaryjnych	20 godz
Przygotowanie do zajęć	30 godz
Udział w ćwiczeniach laboratoryjnych	20 godz
Udział w ćwiczeniach projektowych	20 godz
Wykonanie projektu	30 godz
Sumaryczne obciążenie pracą studenta	120 godz
Punkty ECTS za moduł	4 ECTS