



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Systemy operacyjne i aplikacje dla Systemów Wbudowanych

Przedmiot

Kierunek studiów

Informatyka

Studia w zakresie (specjalność)

Mikrosystemy informatyczne

Poziom studiów

drugiego stopnia

Forma studiów

stacjonarne

Rok/semestr

1/2

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obligatoryjny

Liczba godzin

Wykład

15

Laboratoria

Inne (np. online)

Ćwiczenia

Projekty/seminaria

30

Liczba punktów ECTS

4

Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr inż. Mariusz Naumowicz

email: mnaumowicz@cs.put.poznan.pl

tel. +48 61 665-2364

Wydział Informatyki i Telekomunikacji

ul. Piotrowo 3 60-965 Poznań

Odpowiedzialny za przedmiot/wykładowca:

dr hab. inż. Szymon Szczęsny

email: szymon.szczesny@cs.put.poznan.pl

tel. +48 61 665-2297

Wydział Informatyki i Telekomunikacji

ul. Piotrowo 3 60-965 Poznań

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu programowania strukturalnego i obiektowego. Wymagana jest ponadto umiejętność rozwiązywania podstawowych problemów z zakresu wykorzystania gotowych bibliotek i implementacji algorytmów oraz umiejętność pozyskiwania informacji ze wskazanych źródeł. Niezbędna jest również umiejętność poszerzania posiadanej wiedzy oraz pracy w zespole. Ze względu na kompetencje społeczne student powinien być świadomy, że wiedza w informatyce szybko staje się przestarzała i wymaga ustawicznego poszerzania. Student powinien prezentować postawę uczciwości, kreatywności, rzetelności, ciekawości poznawczej oraz okazywać szacunek dla innych ludzi.



Cel przedmiotu

1. Przekazanie studentom podstawowej wiedzy z zakresu projektowania dedykowanych systemów operacyjnych dla systemów wbudowanych.
2. Zaznajomienie z popularnymi narzędziami wspomagającymi budowanie dedykowanych systemów operacyjnych dla systemów wbudowanych.
3. Przedstawienie studiów przypadku ilustrujących różne realizacje systemów wbudowanych ze względu na ich zastosowanie.
4. Rozwijanie u studentów umiejętności praktycznego wykorzystania wiedzy z zakresu programowania i budowy systemów operacyjnych do realizacji postawionych zadań projektowych.

Przedmiotowe efekty uczenia się

Wiedza

1. Absolwent zna szczegółowo budowę systemów operacyjnych przeznaczonych dla systemów wbudowanych. - [K2st_W5]
2. Absolwent zna wybrane narzędzia wspomagające budowanie i rozwijanie systemów wbudowanych dostosowanych do wiodących trendów. - [K2st_W6]

Umiejętności

1. Absolwent potrafi wykorzystać wiedzę i umiejętności związane z programowaniem systemów operacyjnych. - [K2st_U5]
2. Absolwent potrafi analizować kod programu oraz ocenić jego przydatność w projektowanym rozwiązaniu. - [K2st_U7]
3. Absolwent posiada umiejętność rozwijania istniejących rozwiązań w dziedzinie systemów wbudowanych zgodnie z własnymi potrzebami. - [K2st_U8]
4. Absolwent zna ograniczenia wybranych narzędzi wspomagających budowę systemów operacyjnych i potrafi wybrać właściwe środowiska projektowe do realizacji powierzonego zadania. - [K2st_U9]

Kompetencje społeczne

1. Absolwent jest gotów do ciągłego poszerzania wiedzy w obszarze systemów wbudowanych, zwłaszcza w zakresie zmieniających się trendów rozwoju rynku informatycznego. - [K2st_K1]
2. Absolwent rozumie konieczność stosowania najnowszych rozwiązań podczas implementacji mikrosystemów. - [K2st_K2]

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

- a) w zakresie wykładów:



- na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach,

b) w zakresie laboratoriów:

- na podstawie oceny bieżącego postępu realizacji zadań

Ocena podsumowująca:

a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:

- egzamin pisemny

b) w zakresie projektów weryfikowanie założonych efektów kształcenia realizowane jest przez:

- ocenę sprawozdania przygotowywanego częściowo w trakcie zajęć, a częściowo po ich zakończeniu; ocena ta obejmuje także umiejętność pracy w zespole; dotyczy ćwiczeń projektowych o charakterze odtwórczym (student realizuje ćwiczenie według dostarczonej instrukcji)

- ocena realizacji złożonego zadania wymagającego integracji zdobytej w trakcie zajęć projektowych wiedzy i umiejętności; ocenie podlegają aspekty techniczne realizacji, umiejętność rozwiązania niekonwencjonalnych problemów oraz biegłość wykorzystania dostępnych narzędzi projektowych

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:

- omówienia dodatkowych aspektów zagadnienia,

- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,

- umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium,

- uwagi związane z udoskonaleniem materiałów dydaktycznych,

- wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.

Treści programowe

Program wykładu obejmuje następujące zagadnienia:

Standard POSIX. Systemy czasu rzeczywistego: QNX Neutrino, RT Linux). System Windows Embedded CE. Budowa jądra systemu w systemach wbudowanych. System zarządzania zasobami i procesami. Wątki i procesy: zarządzanie, synchronizacja, komunikacja. Wielowątkowość. Obsługa przerwań. Sprzętowo zależne systemy operacyjne. Budowanie systemów operacyjnych ze źródeł. Ograniczanie i rozszerzanie funkcjonalności systemów operacyjnych. Ładowanie systemu operacyjnego do urządzenia: firmware, bootloader, BIOS, UEFI. Praca z repozytoriami GIT. Sterowniki, programowanie urządzeń we/wy. Rozwijanie oprogramowania dla systemów wbudowanych, kompilacja skrośna. Rozszerzanie języka Python za pomocą modułów napisanych w języku C. Wskaźniki wydajności: przepustowość, czas wykonywania, czas oczekiwania.



Zajęcia projektowe prowadzone są w formie piętnastu 2-godzinnych ćwiczeń, odbywających się w laboratorium, poprzedzonych 2-godzinną sesją instruktazową na początku semestru. Ćwiczenia realizowane są przez 2-osobowe zespoły studentów.

Ćwiczenia projektowe obejmują: Tworzenie i konfiguracja środowiska programistycznego z wykorzystaniem oprogramowania Eclipse oraz kompilatora GCC. Kompilacje i uruchamianie jądra systemu Linux dla dedykowanych urządzeń. Tworzenie projektów i zarządzanie nimi w systemie GIT i Redmine. Programowanie urządzeń peryferyjnych w dedykowanych systemach wbudowanych.

Metody dydaktyczne

Wykład: prezentacja multimedialna, wykład tradycyjny.

Zajęcia projektowe: wykonywanie instrukcji, zadania praktyczne, samodzielne rozwiązywanie problemów, praca w zespole.

Literatura

Podstawowa

1. Systemy operacyjne, Andrew S. Tanenbaum
2. Linux Device Drivers, 3rd Edition, Jonathan Corbet
3. Understanding the Linux Kernel, Third Edition, Daniel P. Bovet
4. Embedded Linux Projects Using Yocto Project Cookbook, Alex Gonzalez, PACKPUB

Uzupełniająca

1. Programming the Raspberry Pi, Second Edition: Getting Started with Python, Simon Monk
2. Raspberry Pi User Guide 3rd Edition, Eben Upton, Gareth Halfacree

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	50	2
Praca własna studenta (studia literaturowe, przygotowanie do egzaminu, wykonanie projektu) ¹	50	2

¹niepotrzebne skreślić lub dopisać inne czynności