



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Systemy operacyjne i aplikacje dla Systemów Wbudowanych

Przedmiot

Kierunek studiów

Informatyka

Studia w zakresie (specjalność)

Przetwarzanie brzegowe

Poziom studiów

drugiego stopnia

Forma studiów

stacjonarne

Rok/semestr

1/2

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obieralny

Liczba godzin

Wykład

15

Ćwiczenia

Laboratoria

15

Projekty/seminaria

15

Inne (np. online)

Liczba punktów ECTS

4

Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr inż. Mariusz Naumowicz

email: mariusz.naumowicz@put.poznan.pl

tel. +48 61 665-2364

Wydział Informatyki i Telekomunikacji

ul. Piotrowo 3, 60-965 Poznań

Odpowiedzialny za przedmiot/wykładowca:

Wymagania wstępne

Student rozpoczynający przedmiot powinien posiadać podstawową wiedzę z zakresu systemów operacyjnych i elektroniki. Powinien również rozumieć konieczność poszerzania swoich kompetencji oraz mieć gotowość do podjęcia współpracy w ramach zespołu.

Cel przedmiotu

-Przekazanie studentom wiedzy związanej z nowoczesnymi systemami wbudowanymi oraz systemami



operacyjnymi dedykowanymi dla tych systemów .

- Zapoznanie studentów z nowoczesnymi metodami projektowania, testowania i prototypowania systemów wbudowanych.
- Rozwijanie u studentów umiejętności rozwiązywania złożonych problemów projektowych w zakresie systemów wbudowanych i systemów operacyjnych.
- Kształtowanie u studentów umiejętności pracy zespołowej.

Przedmiotowe efekty uczenia się

Wiedza

1. Ma zaawansowaną i szczegółową wiedzę o procesach zachodzących w cyklu życia systemów informatycznych, szczególnie warstwy sprzętowej systemów - [K2st_W5]
2. Zna zaawansowane metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich i prowadzeniu prac badawczych w wybranym obszarze informatyki - [K2st_W6]

Umiejętności

1. Potrafi przy formułowaniu i rozwiązywaniu zadań inżynierskich integrować wiedzę z różnych obszarów informatyki (a w razie potrzeby także wiedzę z innych dyscyplin naukowych) oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne - [K2st_U5]
2. Potrafi poprawnie użyć wybraną metodę szacowania pracochłonności wytwarzania oprogramowania - [K2st_U7]
3. Potrafi dokonać krytycznej analizy istniejących rozwiązań technicznych oraz zaproponować ich ulepszenia (usprawnienia) - [K2st_U8]
4. Potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego, polegającego na budowie lub ocenie systemu informatycznego lub jego składowych, w tym dostrzec ograniczenia tych metod i narzędzi; - [K2st_U9]

Kompetencje społeczne

1. Rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K2st_K1]
2. Rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu informatyki w rozwiązywaniu problemów badawczych i praktycznych - [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 zaliczenie (test w postaci elektronicznej na platformie Moodle);

b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez sprawdzian projektowy i ocenę zadań realizowanych w ramach każdego spotkania laboratoryjnego;

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.

Treści programowe

Program wykładu obejmuje następujące zagadnienia:

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.

Zajęcia laboratoryjne prowadzone są w formie 2-godzinnych spotkań, odbywających się w laboratorium, poprzedzonych sesją instruktazową na początku semestru. Ćwiczenia realizowane są przez 2-osobowe zespoły studentów. Program laboratorium obejmuje następujące zagadnienia:

Przygotowanie i konfiguracja środowiska programistycznego z wykorzystaniem oprogramowania Eclipse oraz kompilatora GCC. Budowanie i konfigurowanie programów rozruchowych dla dedykowanych urządzeń. Kompilacje i uruchamianie jądra systemu Linux dla dedykowanych urządzeń. Przygotowanie własnego systemu operacyjnego w oparciu o Buildroot oraz Yocto. Tworzenie projektów i zarządzanie nimi w systemie GIT i Redmine. Programowanie urządzeń peryferyjnych w dedykowanych systemach wbudowanych.

Część wymienionych wyżej treści programowych realizowana jest w ramach pracy własnej studenta.

Metody dydaktyczne

1. wykład: prezentacja multimedialna uzupełniona przykładami podawanymi na tablicy.
2. ćwiczenia laboratoryjne: ćwiczenia praktyczne, dyskusja, praca w zespole, zawody projektowe.
3. zajęcia projektowe: studium przypadków, omawianie i demonstracja działania projektowanych systemów wbudowanych

Literatura



Podstawowa

1. Andrew S. Tanenbaum, Herbert Bos, Systemy operacyjne. Wydanie IV. Helion, 2015. ISBN: 9788328314221.
2. Daniel P. Bovet, Marco Cesati, Understanding the Linux Kernel. 3rd Edition, Helion, 2005. ISBN: 9780596554910.

Uzupełniająca

1. Alex Gonzalez, Embedded Linux Projects Using Yocto Project Cookbook, Packt Publishing, 2015. ISBN: 1784395188.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	45	2,0
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych, przygotowanie do sprawdzianów, wykonanie projektu) ¹	55	2,0

¹ niepotrzebne skreślić lub dopisać inne czynności