



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Systemy operacyjne i aplikacje dla Systemów Wbudowanych [S2Inf1-PB>SOPSW]

Przedmiot

Kierunek studiów
Informatyka

Rok/Semestr
1/2

Studia w zakresie (specjalność)
Przetwarzanie brzegowe

Profil studiów
ogólnoakademicki

Poziom studiów
drugiego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
stacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład
15

Laboratorium
15

Inne
0

Ćwiczenia
0

Projekty/seminaria
15

Liczba punktów ECTS

4,00

Koordynatorzy

dr inż. Mariusz Naumowicz
mariusz.naumowicz@put.poznan.pl

Wykładowcy

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) na podstawie przygotowanej prezentacji w ramach wykładu,,
- b) w zakresie laboratoriów: na podstawie oceny bieżącego postępu realizacji zadań,
- c) w zakresie projektu: na podstawie oceny bieżącego postępu realizacji.

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 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.

W zakresie projektu weryfikowanie założonych efektów kształcenia realizowane jest poprzez ocenę realizacji projektu.

Treści programowe

Program obejmuje zagadnienia związane z budową systemów operacyjnych dla systemów wbudowanych oraz pisanie dedykowanych aplikacji dla tych systemów.

Tematyka zajęć

Wykłady obejmują następującą tematykę:

- budowę jądra systemu w systemach wbudowanych,
- system zarządzania zasobami i procesami,
- wątki i procesy: zarządzanie, synchronizacja, komunikacja,
- wielowątkowość,
- obsługę 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,

- pracę z repozytoriami GIT,
- sterowniki, programowanie urządzeń we/wy,
- rozwijanie oprogramowania dla systemów wbudowanych, kompilacja skrośna.

Zajęcia laboratoryjne oraz projektowe prowadzone są w formie 2-godzinnych spotkań, odbywających się w laboratorium, poprzedzonych sesją instruktażową na początku semestru. Ćwiczenia oraz projekty realizowane są przez 2-osobowe zespoły studentów.

Program laboratorium obejmuje następujące zagadnienia:

- przygotowanie i konfiguracja środowiska programistycznego w systemie Linux, pod kątem budowania systemów operacyjnych,
- budowanie i konfigurowanie programów rozruchowych dla dedykowanych urządzeń,
- kompilacje i uruchamianie jądra systemu Linux dla dedykowanych urządzeń,
- przygotowanie własnego systemu plików z wykorzystaniem Busybox oraz integracja z jądrem Linuxa,
- przygotowanie własnego systemu operacyjnego w oparciu o Buildroot,
- przygotowanie własnego systemu operacyjnego w oparciu o Yocto.

Tematyka projektów powiązana jest z treścią laboratoriów i dotyczy tworzenia własnych systemów operacyjnych oraz dedykowanych aplikacji obsługujących urządzenia peryferyjne.

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 na płytach prototypowych.

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	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	45	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	55	2,00