**Nazwa przedmiotu:**

Sterowniki urządzeń - podstawy programowania

**Koordynator przedmiotu:**

dr inż. Wojciech Zabołotny

**Status przedmiotu:**

Fakultatywny dowolnego wyboru

**Poziom kształcenia:**

Studia II stopnia

**Program:**

Elektronika

**Grupa przedmiotów:**

Przedmioty techniczne - podstawowe

**Kod przedmiotu:**

STER

**Semestr nominalny:**

1 / rok ak. 2019/2020

**Liczba punktów ECTS:**

5

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

Obecność na wykładzie: 30 godzin
Konsultacje projektowe: 30 godzin
Zapoznanie się z literaturą: 20 godzin
Samodzielna praca nad projektem: 45 godzin

Łączny nakład pracy studenta: 125 godzin (5 ECTS)

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

Obecność na wykładzie: 30 godzin
Konsultacje projektowe: 30 godzin

Łącznie: 60 godzin (2 ECTS)

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

Samodzielna praca nad projektem: 45 godzin
Konsultacje projektowe: 30 godzin

Łącznie: 75 godzin (3 ECTS)

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład:  | 30h |
| Ćwiczenia:  | 0h |
| Laboratorium:  | 0h |
| Projekt:  | 30h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

Studenci powinni posiadać umiejętność programowania w języku C oraz podstawową znajomość systemu UNIX (ewentualnie Linux). Wskazana jest znajomość języka angielskiego na poziomie pozwalającym korzystać z literatury anglojęzycznej.

**Limit liczby studentów:**

30

**Cel przedmiotu:**

Celem przedmiotu jest poznanie podstawowych zagadnień związanych z tworzeniem sterowników urządzeń dla współczesnych systemów operacyjnych oraz zdobycie praktycznej umiejętności tworzenia sterowników dla jednego z systemów operacyjnych dostępnych dla platformy sprzętowej PC lub typowych systemów wbudowanych.

**Treści kształcenia:**

W trakcie wykładu jest omawiana rola i struktura sterowników urządzeń w typowych systemach operacyjnych o różnej architekturze. Są uwzględnione różne rodzaje urządzeń (znakowe, blokowe, interfejsy sieciowe). Jest omawiana implementacja procedur obsługujących różne metody komunikacji między procesorem systemu a urządzeniami peryferyjnymi. Szczególna uwaga jest zwracana na problemy związane z wielozadaniowością i wielowątkowością oraz obsługą przerwań i DMA.
 Zdobyta w trakcie wykładu wiedza jest weryfikowana i ugruntowana praktycznie w trakcie realizacji projektu, który obejmuje stworzenie nowego sterownika urządzenia dla wybranego systemu operacyjnego.

Treść wykładu:
1. Rola sterowników urządzeń. Architektura współczesnych systemów operacyjnych, systemy z jądrem monolitycznym, z mikrojądrem i z jądrem modularnym.
2. Współczesne systemy operacyjne, sprzętowe mechanizmy ochrony udostępniane przez procesory i ich wykorzystanie przez systemy operacyjne. Techniki dostępu do urządzeń zewnętrznych z uwzględnieniem mechanizmów ochrony.
3. Podstawowa struktura sterowników w wybranych systemach operacyjnych.
Klasy urządzeń – znakowe, blokowe, interfejsy sieciowe.
Przykładowy sterownik urządzenia znakowego.
Interfejs aplikacji sterownika urządzeń. Korzystanie z urządzenia przez programy użytkownika.
4. Techniki uruchamiania i testowania sterowników urządzeń w różnych konfiguracjach sprzętowych. Wykorzystanie narzędzi symulacyjnych (QEMU, OVP) do uruchamiania i testowania sterowników
5. Zarządzanie pamięcią we współczesnych systemach operacyjnych. Korzystanie z mechanizmów zarządzania pamięcią przez sterowniki urządzeń.
Przekazywanie danych między przestrzenią jądra a przestrzenią użytkownika.
6. Podstawowe metody komunikacji urządzeń peryferyjnych z systemami komputerowymi (pamięć, porty I/O).
Metody implementacji procedur obsługi przerwań w systemach operacyjnych. Priorytet przerwań. Problemy związane z czasem reakcji systemu na przerwanie.
7. Realizacja zależności czasowych w sterownikach urządzeń.
Wielozadaniowość, współbieżność, wielowątkowość i ich wpływ na programowanie sterowników urządzeń. Problemy związane z tworzeniem sterowników dla systemów wieloprocesorowych.
8. Zaawansowane mechanizmy wymiany danych. Problemy związane z DMA i dzieleniem dostępu do magistrali systemowej.
9. Przydzielanie zasobów (przestrzeni adresowej, przerwań itp.) urządzeniom peryferyjnym. Magistrale PCI i PCI Express jako przykład rozwiązania
10. Sterowniki urządzeń blokowych.
11. “Wielowarstwowe” sterowniki urządzeń. Komunikacja między sterownikami w obrębie jądra systemu.
12. Sterowniki urządzeń sieciowych.
13. Sterowniki systemów plików.
14. Sterowniki urządzeń współpracujących z interfejsem USB.
15. Sterowniki urządzeń podłączanych do magistral typowych dla systemów wbudowanych (I2C, SPI itp.). Narzędzia i biblioteki usprawniające tworzenie sterowników urządzeń.

Projekt obejmuje stworzenie i uruchomienie nowego sterownika urządzenia dla jednego z omawianych systemów operacyjnych (Linux, FreeBSD). Projekt może być realizowany na typowej platformie PC, lub na systemie wbudowanym. W uzasadnionych przypadkach możliwa jest realizacja projektu w środowisku emulowanym (QEMU lub OVP + model urządzenia).

Projekt może także dotyczyć urządzeń wirtualnych lub np. sterownika nowego systemu plików, albo poważnej modyfikacji istniejących sterowników.
 Niezależnie od przygotowanego zestawu tematów projektów, możliwe będzie zgłoszenie przez studenta własnego tematu, pod warunkiem, że będzie on oryginalny i będzie możliwa jego weryfikacja (ze względu na możliwości sprzętowe).

**Metody oceny:**

Ocena wystawiana jest na podstawie zrealizowanego projektu. Ocena projektu obejmuje: demonstrację działającego projektu, analizę implementacji i dokumentacji oraz dyskusję ze studentem nad zastosowanymi rozwiązaniami

**Egzamin:**

nie

**Literatura:**

1. Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman, "Linux Device Drivers, 3rd Edition", O'Reilly Media, 2005, dostępne on-line: http://lwn.net/Kernel/LDD3/
2. Robert Lowe, “Linux Kernel Development”, Sams Publishing, 2003
3. Robert Lowe, "Linux Kernel. Przewodnik programisty", Helion, 2004
4. Daniel P. Bovet, Marco Cesati, "Understanding the Linux Kernel Third Edition", O'Reilly Media, 2005,

5. Oracle Solaris "Writing Device Drivers", dostępne on-line: http://docs.oracle.com/cd/E19082-01/819-3196/index.html

6. "Windows Driver Development", Microsoft, dostępne on-line:
http://msdn.microsoft.com/en-us/library/windows/hardware/ff557573

**Witryna www przedmiotu:**

https://eres.elka.pw.edu.pl/eres/wwersje$.startup?Z\_ID\_PRZEDMIOTU=STER

**Uwagi:**

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka W1:**

Posiada wiedzę na temat funkcji sterowników urządzeń oraz typowych klas urządzeń (znakowe, blokowe, interfejsy sieciowe) oraz systemów plików

Weryfikacja:

Projekt, omówienie projektu

**Powiązane charakterystyki kierunkowe:** K\_W05, K\_W02, K\_W03, K\_W04

**Powiązane charakterystyki obszarowe:**

**Charakterystyka W2:**

Posiada wiedzę na temat mechanizmów zapewniających wydajną komunikację między urządzeniami peryferyjnymi i jądrem systemu oraz aplikacjami użytkownika

Weryfikacja:

Projekt, omówienie projektu

**Powiązane charakterystyki kierunkowe:** K\_W02, K\_W03, K\_W04, K\_W05

**Powiązane charakterystyki obszarowe:**

**Charakterystyka W3:**

Posiada wiedzę na temat mechanizmów zapewniających właściwą synchronizację wątków przy obsłudze przerwań oraz w systemach wieloprocesorowych

Weryfikacja:

Projekt, omówienie projektu

**Powiązane charakterystyki kierunkowe:** K\_W02, K\_W03, K\_W04, K\_W05

**Powiązane charakterystyki obszarowe:**

**Charakterystyka W4:**

Posiada wiedzę o mechanizmach ochrony oraz mechanizmach zarządzania pamięcią oraz sposobach ich wykorzystania przy tworzeniu sterowników urządzeń

Weryfikacja:

Projekt, omówienie projektu

**Powiązane charakterystyki kierunkowe:** K\_W04

**Powiązane charakterystyki obszarowe:**

**Charakterystyka W5:**

Posiada wiedzę o typowych magistralach wejścia/wyjścia we współczesnych komputerach i systemach wbudowanych i o zasadach tworzenia sterowników urządzeń podłączonych do tych magistral

Weryfikacja:

Projekt, omówienie projektu

**Powiązane charakterystyki kierunkowe:** K\_W02, K\_W03, K\_W04, K\_W05

**Powiązane charakterystyki obszarowe:**

**Charakterystyka W6:**

Posiada wiedzę o narzędziach umożliwiających efektywne testowanie i uruchamianie kodu sterowników

Weryfikacja:

Projekt, omówienie projektu

**Powiązane charakterystyki kierunkowe:** K\_W02, K\_W03, K\_W04, K\_W06

**Powiązane charakterystyki obszarowe:**

### Profil ogólnoakademicki - umiejętności

**Charakterystyka U1:**

Potrafi studiować literaturę i dokumentację zarówno polskojęzyczną, jak i anglojęzyczną w celu uzupełnienia wiedzy

Weryfikacja:

Projekt, omówienie projektu

**Powiązane charakterystyki kierunkowe:** K\_U01, K\_U05

**Powiązane charakterystyki obszarowe:**

**Charakterystyka U2:**

Potrafi skorzystać z forów i list dyskusyjnych w celu uzyskania wiedzy wykraczającej poza dokumentację i literaturę

Weryfikacja:

Projekt, omówienie projektu

**Powiązane charakterystyki kierunkowe:** K\_U02, K\_U05, K\_U12

**Powiązane charakterystyki obszarowe:**

**Charakterystyka U3:**

Potrafi analizować kod jądra systemu i istniejących sterowników w celu właściwego wyboru rozwiązań

Weryfikacja:

Projekt, omówienie projektu

**Powiązane charakterystyki kierunkowe:** K\_U01, K\_U12, K\_U13

**Powiązane charakterystyki obszarowe:**

**Charakterystyka U4:**

Potrafi wybrać właściwe rozwiązania do realizacji kodu projektowanego sterownika, potrafi go zaimplementować i uruchomić

Weryfikacja:

Projekt, omówienie projektu

**Powiązane charakterystyki kierunkowe:** K\_U10, K\_U13, K\_U07, K\_U08

**Powiązane charakterystyki obszarowe:**

**Charakterystyka U5:**

Potrafi stworzyć aplikację umożliwiającą przetestowanie tworzonego sterownika i opracować scenariusz jego testowania

Weryfikacja:

Projekt, omówienie projektu

**Powiązane charakterystyki kierunkowe:** K\_U07, K\_U08, K\_U09, K\_U10, K\_U12, K\_U13, K\_U14, K\_U15, K\_U16

**Powiązane charakterystyki obszarowe:**

### Profil ogólnoakademicki - kompetencje społeczne

**Charakterystyka K1:**

Potrafi rozwiązać zadanie projektowe, rozumiejąc wymagania, potrafiąc wybrać rozwiązania pasujące do wymagań i istniejących ograniczeń

Weryfikacja:

Projekt, omówienie projektu

**Powiązane charakterystyki kierunkowe:** K\_K01

**Powiązane charakterystyki obszarowe:**