**Nazwa przedmiotu:**

Systemy operacyjne

**Koordynator przedmiotu:**

Prof. nzw. dr hab. inż. Leszek Jan Opalski

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Wspólne

**Kod przedmiotu:**

**Semestr nominalny:**

3 / rok ak. 2012/2013

**Liczba punktów ECTS:**

4

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

1. godziny kontaktowe - 45 h; w tym
a. obecność na wykładach – 30 h
b. obecność na laboratoriach – 15 h
2. przygotowanie do wykładów – 6h
3. przygotowanie do sprawdzianów audytoryjnych (w tym konsultacje) – 10 h (2 x 5h)
4. przygotowanie do zajęć laboratoryjnych (w tym konsultacje) – 48 h
Razem nakład pracy studenta 109 h = 4pkt. ECTS

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

1. obecność na wykładach – 30 h
2. obecność na laboratoriach – 15 h
3. konsultacje (do wykładu i laboratorium) – 5h
Razem 30+15+5 = 50h, co odpowiada 2 pkt. ECTS

**Język prowadzenia zajęć:**

polski

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

1. obecność na laboratoriach: 15 h
2. przygotowanie do zajęć laboratoryjnych, w tym lektura dokumentacji AP, konsultacje i rozwiązanie zadań przykładowych: 48 h (6 zajęć lab. X 8 h)
Razem 15+48= 73 h, co odpowiada 3 pkt. ECTS

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

|  |  |
| --- | --- |
| Wykład:  | 30h |
| Ćwiczenia:  | 0h |
| Laboratorium:  | 15h |
| Projekt:  | 0h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

Podstawy programowania, Podstawy programowania strukturalnego

**Limit liczby studentów:**

Bez limitu

**Cel przedmiotu:**

Celem przedmiotu jest zapoznanie z budową i działaniem współczesnych systemów operacyjnych, a także kształtowanie umiejętności wykorzystania mechanizmów systemowych do poprawnej realizacji prostych aplikacji wieloprocesowych/wielowątkowych, w tym aplikacji sieciowych (klient-serwer). Po ukończeniu kursu studenci powinni:
posiadać wiedzę na temat
zadań, właściwości i budowy systemów operacyjnych
podstawowych modeli i sposobów realizacji współbieżności przez system komputerowy
planowania przydziału czasu procesora/procesorów
zarządzania pamięcią
budowy i własności podstawowych typów systemów plików
środków komunikacji pomiędzy procesami (pamięci współdzielonej, kolejek komunikatów, łączy), podstawowych środków i schematów synchronizacji oraz problemu zakleszczeń
bezpieczeństwa i ochrony
powinni umieć :
wyznaczać wskaźniki jakości algorytmów przydziału czasu procesora, czy średni czas dostępu do pamięci
wykrywać zakleszczenia algorytmów przydziału zasobów za pomocą algorytmu bankiera
zaprojektować, napisać w języku C i przetestować stworzoną przez siebie prostą aplikację o zadanej funkcjonalności, wykorzystując mechanizmy i funkcje systemowe (API POSIX)
zaprojektować, napisać w języku C i przetestować stworzoną przez siebie prostą aplikację sieciową w układzie klient-serwer, wykorzystując interfejs gniazd

**Treści kształcenia:**

Wykład:
1.  Wprowadzenie. Zadania i właściwości systemów operacyjnych. Struktury systemów komputerowych i systemów operacyjnych. Struktura i interfejs jądra systemu.
2.  Procesy i wątki. Koncepcje procesu, procesu lekkiego i wątku; cykl życia i interfejs programisty. Realizacja współbieżności procesów i wątków, szeregowanie, wywłaszczanie. Sygnały i wyjątki oraz ich obsłu­ga.
3.  Komunikacja międzyprocesowa i synchronizacja. Komunikacja między procesami: pamięć dzielona, system komunikatów; synchronizacja współpracujących procesów i wątków: Metody realizacji i interfejs programisty. Problem zak­leszczania: model i metody postępowania.
4.  Zarządzanie pamięcią pomocniczą. Plik, organizacja systemu plików. Operacje na systemie plików i interfejs programisty. Dostęp i ochrona plików, tryby dostępu do plików.
5.  Zarządzanie systemem wejścia/wyjścia. Urządzenia znakowe i blokowe; tryby dostępu. Moduły obsługi urządzeń wejścia/wyjścia.
6.  Komunikacja sieciowa. Model komunikacji OSI. Interfejs gniazd w programowaniu połączeniowych i bezpołączeniowych aplikacji sieciowych. Problemy komunikacji i synchronizacji aplikacji sieciowych. Przykłady usług sieciowych.
7.  Ochrona i bezpieczeństwo systemów komputerowych. Cele ochrony. Uwierzytelnianie, kontrola dostępu do zasobów, integralność, niezaprzeczalność, poufność– koncepcje i realizacje. Standardy oceny bezpieczeństwa.

Laboratorium:
1.  Procesy i sygnały.
2.  Komunikacja i synchronizacja międzyprocesowa (cz. 1 i 2)
3.  Operacje wejścia/wyjścia
4.  Komunikacja sieciowa (interfejs gniazd)
5.  Wątki POSIX – tworzenie, synchronizacja

**Metody oceny:**

Ocena z przedmiotu wynika z sumy punktów uzyskanych przez studenta w laboratorium (maksymalnie 6\*7p) oraz za sprawdziany wykładowe (58p). Pozytywna ocena końcowa wymaga uzyskania co najmniej 50p, w tym 20p. z laboratorium.
Laboratorium – zasady szczegółowe:

Do dyspozycji studenta są zajęcia wprowadzające i 6 terminów zajęć ocenianych.
W każdym terminie zajęć ocenianych można uzyskać 7p  (2p – wejściówka, 5p – rozwiązania zadań programistycznych), stąd łącznie 42p.
Rozwiązania wszystkich zadań podlegających ocenie muszą być przesłane prowadzącemu zajęcia w czasie zajęć z własnego konta studenta na serwerze Wydziale MiNI, bądź być przekazane (w czasie zajęć) prowadzącemu na dyskietce. Student ma przekazać prowadzącemu kody źródłowe wszystkich opracowanych programów oraz skrypt powłoki, bądź plik Makefile, który można użyć do uzyskania programu wykonywalnego.W treści zadania laboratoryjnego może być zawarte wymaganie testowania opracowanego programu. W tym przypadku student przekazuje prowadzącemu dodatkowo pliki danych testujących, a wyniki testów przedstawia w krótkim sprawozdaniu. Wszystkie wymienione powyżej pliki tekstowe muszą zawierać nazwisko i grupę studencką autora.
Zajęcia odbywają się (bez przerw) w s. 221/222 wg harmonogramu (dostępnego pod URL: http://alpha.mini.pw.edu.pl/~opalskil/SOP/harmonogram.html) uzgodnionego z reprezentacją studentów w ciągu pierwszych 2 tygodni semestru.
Nie ma dodatkowego terminu zajęć laboratoryjnych na odrabianie nieobecności, czy poprawianie oceny.
Jedna usprawiedliwiona nieobecność powoduje zastąpienie brakującej oceny przez średnią pozostałych.

Sprawdziany wykładowe – zasady szczegółowe:

W semestrze odbędą się 2 sprawdziany wykładowe w terminach określonych w harmonogramie.
Usprawiedliwiona nieobecność daje prawo do odbycia sprawdzianu w dodatkowym terminie (terminie sprawdzianu poprawkowego). Obowiązuje ten sam zakres materiału co w pierwotnym terminie. Student musi zgłosić wykładowcy zamiar przystąpienia do tego sprawdzianu najpóźniej jedną dobę przed tym terminem
Osoby, które uzyskały co najmniej 20p z laboratorium - mogą odbyć pisemny sprawdzian poprawkowy w terminie dodatkowym (patrz harmonogram). Zakres tego sprawdzianu (za 58p) obejmuje całość materiału wykładowego. Wynik tego sprawdzianu zastępuje sumę punktów z dwóch sprawdzianów semestralnych (uwaga: wynik sprawdzianu poprawkowego może być mniejszy od sumy punktów ze sprawdzianów w semestrze). Student musi zgłosić wykładowcy zamiar przystąpienia do sprawdzianu poprawkowego najpóźniej jedną dobę przed tym sprawdzianem.

Uwagi ogólne:

Wszystkie oceniane prace muszą być wykonywane samodzielnie. Niesamodzielność pracy, bądź korzystanie przez studenta z niedozwolonych materiałów powoduje uzyskanie z danej pracy/sprawdzianu 0pt.
W czasie wykonywania ocenianych prac pisemnych (jak wejściówki laboratoryjne i sprawdziany wykładowe) nie można korzystać z żadnych materiałów pisanych, nagrań dźwiękowych, środków komunikacji elektronicznej.
W czasie realizacji zadań laboratoryjnych (po wejściówce) można korzystać z własnych materiałów, rozwiązań zadań przykładowych, dokumentacji systemowej oraz Internetu jednak tak, by oceniana praca była dziełem samodzielnym.
Wszelkie usprawiedliwienia nieobecności powinny być zgłoszone prowadzącemu przedmiot przed zakończeniem semestru, nie później niż w terminie następnego najbliższego wykładu od ustania przyczyny nieobecności.

**Egzamin:**

nie

**Literatura:**

1.  A. Silberschatz,  P.B. Galvin, G. Gagne, Podstawy systemów operacyjnych, wyd. 6 zm. i rozsz., WNT 2005
2.  W. Stallings, Systemy Operacyjne, Wyd. Robomatic, 2004
3.  U. Vahalia, Jądro systemu UNIX. Nowe horyzonty, WNT 2001
4.  R.W. Stevens, Programowanie w środowisku systemu UNIX, WNT 2002

**Witryna www przedmiotu:**

brak

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt W01:**

Ma uporządkowaną, podbudowaną teoretycznie wiedzę nt. zadań, właściwości i budowy systemów operacyjnych

Weryfikacja:

testy audytoryjne, wejściówki lab.

**Powiązane efekty kierunkowe:** K\_W05

**Powiązane efekty obszarowe:** T1A\_W03

**Efekt W02:**

Zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu prostych zadań z zakresu systemów operacyjnych. W szczególności: - zna zasady i algorytmy przydziału czasu procesora, modele współbieżności oferowane przez SO - ma podstawową wiedzę dot. zarządzania pamięcią oraz realizacji systemów plików - ma uporządkowaną wiedzę dotyczącą środków komunikacji pomiędzy procesami (pamięć współdzielona, kolejki komunikatów, łącza, gniazdka), podstawowych środków i schematów synchronizacji oraz problemu zakleszczeń

Weryfikacja:

testy audytoryjne, wejściówki lab.

**Powiązane efekty kierunkowe:** K\_W11, K\_W13

**Powiązane efekty obszarowe:** T1A\_W07, T1A\_W07

**Efekt W03:**

Zna podstawowe problemy ochrony, wie o podstawowych środkach zapewnienia bezpieczeństwa systemów komputerowych

Weryfikacja:

test audyt., wejściówka lab.

**Powiązane efekty kierunkowe:** K\_W14

**Powiązane efekty obszarowe:** T1A\_W08

### Profil ogólnoakademicki - umiejętności

**Efekt U01:**

Potrafi rozwiązywać proste zadania z zakresu systemów operacyjnych za pomocą metod analitycznych i symulacyjnych. Przykładowe zadania: - obliczanie wskaźników jakości algorytmów szeregowania zadań, czy średniego czasu dostępu do pamięci - określanie bezpieczeństwa przydziału zasobów za pomocą algorytmu bankiera

Weryfikacja:

test audyt.

**Powiązane efekty kierunkowe:** K\_U09

**Powiązane efekty obszarowe:** T1A\_U09

**Efekt U02:**

Potrafi pozyskiwać informacje z dokumentacji technicznej systemu operacyjnego (man) i standardu POSIX oraz wykorzystywać te informacje do pisania kodu przenośnego pomiędzy plaformami Unix/Linux.

Weryfikacja:

zaliczenie zadania lab.

**Powiązane efekty kierunkowe:** K\_U05, K\_U07

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U03, T1A\_U04, T1A\_U06

**Efekt U03:**

Potrafi zrealizować w języku C proste aplikacje o zadanej funkcjonalności, wykorzystując narzędzia linii poleceń systemu Linux.

Weryfikacja:

zaliczenie zadania lab.

**Powiązane efekty kierunkowe:** K\_U11

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U14, T1A\_U15

**Efekt U04:**

Potrafi wykorzystać nabytą wiedzę o własnościach SO oraz znajomość API POSIX do realizacji prostych aplikacji wieloprocesowych/ wielowątkowych o zadanej funkcjonalności, w tym aplikacji sieciowych (klient-serwer)

Weryfikacja:

zaliczenie zadania lab.

**Powiązane efekty kierunkowe:** K\_U11, K\_U15, K\_U18

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U14, T1A\_U15, T1A\_U09, T1A\_U15, T1A\_U16

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt K01:**

Rozumie konieczność ciągłego śledzenia zmian w obszarze systemów operacyjnych

Weryfikacja:

testy audytoryjne

**Powiązane efekty kierunkowe:** K\_K01

**Powiązane efekty obszarowe:** T1A\_K01