**Nazwa przedmiotu:**

UNIX

**Koordynator przedmiotu:**

mgr inż. Marcin Borkowski

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Wspólne

**Kod przedmiotu:**

**Semestr nominalny:**

6 / rok ak. 2012/2013

**Liczba punktów ECTS:**

3

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

1. godziny kontaktowe - 50h; w tym
a. obecność na wykładach – 15h
b. obecność na laboratoriach – 30h
c. konsultacje – 5h
2. przygotowanie do zajęć – 55h, w tym
a. przygotowanie do wykładów – 10h
b. przygotowanie do zajęć laboratoryjnych – 45h
Razem nakład pracy studenta 105h = 4 pkt. ECTS

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

1. obecność na wykładach – 15h
2. obecność na laboratoriach – 30h
3. konsultacje z prowadzącymi zajęcia – 5h
Razem 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 – 30h
2. przygotowanie do laboratorium – 45h
Razem 75h, co odpowiada 3 pkt. ECTS

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

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

**Wymagania wstępne:**

Podstawy Programowania Strukturalnego I semestr
Programowanie Obiektowe III semestr

**Limit liczby studentów:**

Bez limitu

**Cel przedmiotu:**

Poznanie zaawansowanych technik programowania wywołań funkcji systemowych w standardzie POSIX. Nabranie wprawy w programowaniu dla platformy UNIX/Linux. Poznanie problemów związanych z pisaniem przenośnego kodu oraz metod ich rozwiązywania.

**Treści kształcenia:**

Wykład:
1. Omówienie specyfiki programowania w systemie UNIX i programowania rozproszonego
2. Omówienie standardów POSIX i SYSTEM V
3. Omówienie standardów kodowania
4. Procesy i sygnały, sygnały czasu rzeczywistego, omówienie mechanizmu działania wraz z typowymi rozwiązaniami (przykłady kodu) oraz z omówieniem typowych błędów programistycznych
5. Katalogi i pliki, blokowanie plików, omówienie mechanizmu działania wraz z typowymi rozwiązaniami (przykłady kodu) oraz z omówieniem typowych błędów programistycznych
6. Kolejki FIFO i pipe, omówienie mechanizmu działania wraz z typowymi rozwiązaniami (przykłady kodu) oraz z omówieniem typowych błędów programistycznych
7. mechanizmu IPC z SYSTEM V, omówienie mechanizmu działania wraz z typowymi rozwiązaniami (przykłady kodu) oraz z omówieniem typowych błędów programistycznych
8. Asynchroniczny dostęp do plików, omówienie mechanizmu działania wraz z typowymi rozwiązaniami (przykłady kodu) oraz z omówieniem typowych błędów programistycznych
9. Gniazda sieciowe (lokalne,tcp,udp,raw), omówienie mechanizmu działania wraz z typowymi rozwiązaniami (przykłady kodu) oraz z omówieniem typowych błędów programistycznych
10. Wątki posix'owe oraz posix'owe mechanizmy synchronizacji, omówienie mechanizmu działania wraz z typowymi rozwiązaniami (przykłady kodu) oraz z omówieniem typowych błędów programistycznych
 
Laboratorium:
Ćwiczenia sprawdzające wiedze z wykładu. Każdemu tematowi wykładowemu odpowiada jedno oceniane zadanie. Zadania polegają na samodzielnym rozwiązaniu zadania podanego przez prowadzącego. Ramy czasowe są ograniczone do 90 minut. Ostatnie zadanie ma postać projektu wykonywanego pod nadzorem prowadzącego bez limitów czasowych.

**Metody oceny:**

Ocena z laboratorium jest oceną z całego przedmiotu. Kryteria oceny laboratorium:
• Do wykonania jest 8 zadań po 8 punktów (tematy tak jak w archiwum) co daje 64 oraz projekt za 44 punkty
• Najgorszy wynik z laboratoriów jest odrzucany zatem suma do zdobycia wynosi 100
• Aby zaliczyć przedmiot należy zdobyć minimum 51 punktów w sumie i nie mniej niż 32 punkty z zadań

**Egzamin:**

nie

**Literatura:**

1. “UNIX. Programowanie usług sieciowych” - tom 1 i II Richard W. Stevens WNT
2. “Programowanie w środowisku systemu UNIX” W. Richard Stevens WNT
3. “Programowanie w systemie UNIX dla zaawansowanych” Marc J. Rochkind WNT

**Witryna www przedmiotu:**

brak

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt W01:**

Ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie architektury systemów operacyjnych z rodziny UNIX oraz technologii sieciowych

Weryfikacja:

Wejściówki na laboratoriach

**Powiązane efekty kierunkowe:** K\_W05

**Powiązane efekty obszarowe:** T1A\_W03

**Efekt W02:**

Zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu prostych zadań informatycznych z zakresu budowy systemów komputerowych opartych o system Linux/Unix, sieci komputerowych i technologii sieciowych, w tym: metody komunikacji za pomocą sygnałów, kolejek FIFO, pipe, mechanizmów IPC systemu V, gniazd sieciowych; wątki i procesy; dostęp do systemu plików synchroniczny i asynchroniczny; znajomość aspektów inter-systemowych komunikacji sieciowej

Weryfikacja:

Ocena z zadań wykonywanych podczas laboratorium , ocena z projektu

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

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

### Profil ogólnoakademicki - umiejętności

**Efekt U01:**

Potrafi pozyskiwać informacje z dokumentacji technicznej systemu (man) oraz z standardu posix (głównie w języku angielskim), potrafi przekształcić uzyskaną wiedzę na pisanie kodu niezależnego od platformy uniksowej

Weryfikacja:

Ocena z zadań wykonywanych podczas laboratorium , ocena z projektu

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

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

**Efekt U02:**

Potrafi przygotować, skompilować, sprawdzić i uruchomić program w języku C przy pomocy prostych narzędzi linii poleceń systemu Linux takich jak edytor tekstowy, gcc , make i gdb

Weryfikacja:

Ocena z zadań wykonywanych podczas laboratorium , ocena z projektu

**Powiązane efekty kierunkowe:** K\_U11

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

**Efekt U03:**

Ma umiejętność posługiwania się systemem Linux/Unix na poziomie API

Weryfikacja:

Ocena z zadań wykonywanych podczas laboratorium , ocena z projektu

**Powiązane efekty kierunkowe:** K\_U15

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

**Efekt U04:**

Ma umiejętność pisania prostych aplikacji do komunikacji sieciowej takich jak komunikatory (tzw. chat) oraz proste serwery (plików i www)

Weryfikacja:

Ocena z projektu

**Powiązane efekty kierunkowe:** K\_U18, K\_U30

**Powiązane efekty obszarowe:** T1A\_U16, T1A\_U16

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt K01:**

Rozumie konieczność ciągłego śledzenia zmian w dokumentacji nowych wersji bibliotek systemowego API oraz standardów takich jak POSIX

Weryfikacja:

Wejściówki na laboratoriach

**Powiązane efekty kierunkowe:** K\_K01

**Powiązane efekty obszarowe:** T1A\_K01

**Efekt K02:**

Rozumie potrzebę pisania kodu przenośnego oraz poprawnego podziału programu na biblioteki i funkcję umożliwiającego łatwe ponowne wykorzystanie kodu

Weryfikacja:

Ocena z zadań wykonywanych podczas laboratorium , ocena z projektu

**Powiązane efekty kierunkowe:** K\_K04

**Powiązane efekty obszarowe:** T1A\_K02, T1A\_K05

**Efekt K03:**

Potrafi samodzielnie wykonać mały projekt informatyczny związany z programowaniem na poziomie API systemu Linux

Weryfikacja:

Ocena z projektu

**Powiązane efekty kierunkowe:** K\_K05

**Powiązane efekty obszarowe:** T1A\_K03, T1A\_K04