



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez
Unię Europejską w ramach
Europejskiego Funduszu
Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



| | | | |
|--|-----------------|---|---------------------------|
| Nazwa przedmiotu | | Kod ECTS | |
| Programowanie współbieżne | | 11.3.0835 | |
| Nazwa jednostki prowadzącej przedmiot | | | |
| Instytut Informatyki | | | |
| Studia | | | |
| wydział | kierunek | poziom | pierwszego stopnia |
| Wydział Matematyki, Fizyki i Informatyki | Informatyka | forma | stacjonarne |
| | | moduł | wszystkie |
| | | specjalnościowy | wszystkie |
| | | specjalizacja | wszystkie |
| Nazwisko osoby prowadzącej (osób prowadzących) | | | |
| dr inż. Jerzy Skurczyński | | | |
| Formy zajęć, sposób ich realizacji i przypisana im liczba godzin | | Liczba punktów ECTS | |
| Formy zajęć | | 5 Przedmiot w wymiarze 30h wykładu i 30h lab. + praca własna studenta | |
| Wykład, Ćw. laboratoryjne | | | |
| Sposób realizacji zajęć | | | |
| zajęcia w sali dydaktycznej | | | |
| Liczba godzin | | | |
| Wykład: 30 godz., Ćw. laboratoryjne: 30 godz. | | | |
| Cykl dydaktyczny | | | |
| 2017/2018 zimowy | | | |
| Status przedmiotu | | Język wykładowy | |
| fakultatywny (do wyboru) | | polski | |
| Metody dydaktyczne | | Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne | |
| <ul style="list-style-type: none"> - Projektowanie doświadczeń - Wykonywanie doświadczeń - Wykład z prezentacją multimedialną | | Sposób zaliczenia | |
| | | <ul style="list-style-type: none"> - Zaliczenie na ocenę - Egzamin | |
| | | Formy zaliczenia | |
| | | <ul style="list-style-type: none"> - egzamin pisemny testowy - ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru | |
| | | Podstawowe kryteria oceny | |
| | | Wykład - liczba punktów uzyskanych na egzaminie. Laboratorium - liczba punktów uzyskanych za napisane i zaliczone programy oraz ocena za projekt końcowy. | |
| Sposób weryfikacji założonych efektów kształcenia | | | |
| | | | |

| zakładany efekt kształcenia | egzamin | odpowiedź ustna | projekt | obserwacja |
|-----------------------------|--------------|-----------------|---------|------------|
| | Wiedza | | | |
| KW_02 | x | x | x | |
| KW_06 | x | x | x | |
| | Umiejętności | | | |
| KU_07 | x | x | x | |
| KU_10 | x | x | x | |
| | Kompetencje | | | |
| KK_04 | | | x | x |
| KK_05 | | | x | x |

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

A. Wymagania formalne

Zaliczone przedmioty: 1) Systemy operacyjne; 2) Sieci komputerowe.

B. Wymagania wstępne

Umiejętność programowania w języku C pod systemem Linux oraz tworzenia skryptów w bash'u.

Cele kształcenia

Nauczenie studentów projektowania i tworzenia prawidłowo zbudowanych programów wieloprotokolowych i wielowątkowych oraz analizy ich działania przy użyciu narzędzi teoretycznych.

Treści programowe

1. Modele teoretyczne zbiorów procesów współbieżnych.
2. Rodzaje komunikacji pomiędzy procesami.
3. Synchronizacja, unikanie blokady i głodzenia procesów.
4. Logika temporalna jako narzędzie dowodzenia poprawności programów współbieżnych.
5. Mechanizmy komunikacji i synchronizacji procesów w systemie Unix / Linux: kolejki, semafony, pamięć dzielona.
6. Programowanie rozproszone przy użyciu gniazd.
7. Mechanizmy koordynacji wątków.

Wykaz literatury

1. M. Ben-Ari, Podstawy programowania współbieżnego i rozproszonego, WNT, 1996.
2. J.S. Gray, Komunikacja między procesami w UNIXie, RM, 1998.

Efekty kształcenia

(obszarowe i kierunkowe)

Student rozumie różnicę pomiędzy programowaniem sekwencyjnym a współbieżnym, zna kryteria oceny poprawności programu współbieżnego i umie je zastosować. Zdaje sobie sprawę z niskiej użyteczności testowania działania programów współbieżnych i docenia rolę formalnego dowodzenia ich poprawności.

KW_02

ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną

w zakresie programowania, algorytmów i złożoności, architektury systemów komputerowych, systemów operacyjnych, technologii sieciowych, języków i paradygmatów programowania, baz danych, inżynierii oprogramowania, języków formalnych

KW_o6

zna zasady działania systemów operacyjnych ze szczególnym

uwzględnieniem procesów, współbieżności, szeregowania zadań i zarządzania pamięcią

KU_07

Wiedza

Student zna wybrane narzędzia programistyczne służące do tworzenia programów współbieżnych oraz wybrane narzędzia teoretyczne służące do dowodzenia ich poprawności.

Umiejętności

Student potrafi zaprojektować i zbudować nieduży program współbieżny przy użyciu narzędzi dostępnych w systemie Linux oraz udowodnić jego poprawność przy użyciu logiki temporalnej.

Kompetencje społeczne (postawy)

Student jest w stanie współpracować z zespołem programistów tworzących wspólnie większy program współbieżny, a w razie potrzeby pełnić funkcję kierownika takiego zespołu.

| | |
|---|--|
| <p>rozumie niskopoziomowe zasady wykonywania programów KU_10</p> <p>rozumie mechanizmy synchronizacji programów współbieżnych KK_04</p> <p>rozumie i docenia znaczenie uczciwości intelektualnej w działaniach własnych i innych osób; postępuje etycznie</p> <p>KK_05</p> <p>potrafi samodzielnie wyszukiwać informacje w literaturze, także w językach obcych</p> | |
| Kontakt | |
| jsk@inf.ug.edu.pl | |