



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Podstawy programowania współbieżnego [S1Bioinf1>PPW]

Przedmiot

Kierunek studiów

Bioinformatyka

Rok/Semestr

2/3

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obieralny

Liczba godzin

Wykład

30

Laboratorium

30

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

4,00

Koordynatorzy

dr hab. inż. Anna Kobusińska prof. PP
anna.kobusinska@put.poznan.pl

dr inż. Dariusz Wawrzyniak
dariusz.wawrzyniak@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student rozpoczynający ten moduł powinien posiadać podstawową wiedzę z zakresu funkcjonowania komputera i programowania imperatywnego. Powinien posiadać umiejętność rozwiązywania podstawowych problemów z zakresu implementacji i oceny kosztu działania prostych algorytmów oraz umiejętność pozyskiwania informacji ze wskazanych źródeł. Powinien również rozumieć konieczność poszerzania swoich kompetencji i wykazywać gotowość do podjęcia współpracy w ramach zespołu. Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.

Cel przedmiotu

Przekazanie studentom podstawowej wiedzy teoretycznej związanej z przetwarzaniem współbieżnym w systemach komputerowych oraz praktycznych aspektów realizacji przetwarzania współbieżnego w tego typu systemach. Rozwijanie u studentów umiejętności rozwiązywania problemów związanych z przetwarzaniem współbieżnym w systemach komputerowych.

Przedmiotowe efekty uczenia się

Wiedza:

1. Rozumie podstawowe problemy przetwarzania współbieżnego w systemach operacyjnych (np. niedeterminizm, zakleszczenie).
2. Zna wybrane zagadnienia z zakresu programowania strukturalnego i obiektowego związane z synchronizacją procesów.
3. Zna wybrane zagadnienia optymalizacji kombinatorycznej w przetwarzaniu współbieżnym.
4. Zna wybrane zagadnienia z zakresu cyklu życia systemów informatycznych.

Umiejętności:

1. Potrafi projektować i tworzyć oprogramowanie współbieżne zgodnie z zadaną specyfikacją, używając właściwych metod, technik i narzędzi.
2. Potrafi dokonać analizy funkcjonalności i analizy wymagań systemów informatycznych w kontekście zagadnień przetwarzania współbieżnego.
3. Potrafi pozyskiwać informacje z literatury, baz danych oraz innych właściwie dobranych źródeł, także w języku angielskim.

Kompetencje społeczne:

1. Rozumie potrzebę uczenia się przez całe życie i podnoszenia swoich kompetencji.
2. Potrafi współdziałać i pracować w grupie, przyjmując w niej różne role.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca

- a) W zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:
- odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach.
- b) W zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:
- ocenę przygotowania studenta do poszczególnych sesji zajęć laboratoryjnych (sprawdzian „wejściowy”) oraz ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych,
 - ocenę wiedzy i umiejętności związanych z realizacją zadań laboratoryjnych poprzez dwa kolokwia w semestrze.

Ocena podsumowująca

- a) W zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:
- ocenę wiedzy i umiejętności wykazanych na zaliczeniu pisemnym o charakterze problemowym składającym się z 4 – 5 pytań otwartych, z możliwością uzyskania 20 – 30 punktów za każde z nich oraz w sumie 100 punktów; aby uzyskać ocenę pozytywną, należy zdobyć minimum 50 punktów;
 - omówienie wyników zaliczenia.
- b) W zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:
- ocenę wiedzy i umiejętności związanych z treściami przekazywanymi na laboratoriach poprzez kolokwium końcowe;
 - zestawienie ocen wystawionych w trakcie semestru w postaci średniej.

Aktywność podczas zajęć premiowana jest dodatkowymi punktami, w szczególności za:

- omówienie dodatkowych aspektów zagadnienia,
- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,
- uwagi prowadzące do udoskonalenia materiałów dydaktycznych lub procesu dydaktycznego.

Treści programowe

Program modułu obejmuje następujące zagadnienia:

- 1.
2. Abstrakcja programowania współbieżnego.
- 3.
4. Pojęcia bezpieczeństwa i żywotności.
- 5.
6. Problem wzajemnego wykluczania.
- 7.
8. Koncepcja i rola systemu operacyjnego (w tym procesy, zasoby i wątki).
- 9.
10. Mechanizmy synchronizacji i komunikacji wspierane przez system operacyjny.

- 11.
12. Klasyczne problemy synchronizacji.
- 13.
14. Wsparcie językowe dla mechanizmów synchronizacji.
- 15.
16. Problem zakleszczenia.
- 17.
18. Planowanie przydziału czasu procesora.
- 19.

Tematyka zajęć

Na wykładzie omawiane są następujące tematy:

- 1.
2. Abstrakcja programowania współbieżnego: wprowadzenie pojęcia operacji atomowych i ich przepływu.
- 3.
4. Problem wzajemnego wykluczania oraz jego rozwiązanie oparte na atomowych operacjach odczytu i zapisu współdzielonych zmiennych (np. algorytmy: Dekkera, Dijkstry, Petersona, Lamporty).
- 5.
6. Złożone operacje atomowe, np.: test-and-set, exchange i ich zastosowanie w realizacji wzajemnego wykluczania.
- 7.
8. Koncepcja systemu operacyjnego jako zarządcy zasobów (w tym pojęcia procesu i wątku) oraz jego rola w realizacji przetwarzania współbieżnego.
- 9.
10. Mechanizmy synchronizacji wspierane przez system operacyjny: semafore binarne i zliczające, mechanizmy standardu POSIX (zamki i zmienne warunkowe).
- 11.
12. Klasyczne problemy synchronizacji: producent-konsument, czytelnicy i pisarze, pięciu filozofów, śpiący fryzjerzy.
- 13.
14. Wsparcie językowe dla mechanizmów synchronizacji: monitory i warunkowe regiony krytyczne.
- 15.
16. Problem zakleszczenia: definicja zakleszczenia, warunki konieczne i dostateczne zakleszczenia, przeciwdziałanie zakleszczeniom (zapobieganie, unikanie, detekcja i likwidacja).
- 17.
18. Planowanie przydziału czasu procesora: koncepcja i algorytmy planowania, kryteria oceny.
- 19.

Program zajęć laboratoryjnych obejmuje następujące zagadnienia na poziomie interfejsu usług systemowych w języku C:

- 1.
2. Dostęp do plików.
- 3.
4. Zarządzanie procesami.
- 5.
6. Obsługa sygnałów.
- 7.
8. Komunikacja międzyprocesowa i synchronizacja za pośrednictwem łączy.
- 9.
10. Komunikacja międzyprocesowa i synchronizacja za pośrednictwem mechanizmów IPC.
- 11.

Metody dydaktyczne

1. Wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, rozwiązywanie zadań.
2. Ćwiczenia laboratoryjne: implementacja programów współbieżnych, dyskusja.

Literatura

Podstawowa

1. M. Ben-Ari, Podstawy programowania współbieżnego i rozproszonego, WNT, W-wa, 2016.
2. A. Silberschatz, G. Gagne, P.B. Galvin Podstawy systemów operacyjnych, WN PWN, W-wa, 2021.
3. M. J. Rochkind, Programowanie w systemie Unix dla zaawansowanych, WNT, Warszawa, 2007.

Uzupełniająca

1. Z. Weiss, T. Gruzlewski, Programowanie współbieżne i rozproszone w przykładach i zadaniach, WNT, W-wa, 1993.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	60	2,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	40	1,50