



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Narzędzia przetwarzania rozproszonego [S2Inf1-SRC>NPR]

Przedmiot

Kierunek studiów
Informatyka

Rok/Semestr
1/1

Studia w zakresie (specjalność)
Systemy rozproszone i chmurowe

Profil studiów
ogólnoakademicki

Poziom studiów
drugiego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
stacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład
30

Laboratorium
30

Inne
0

Ćwiczenia
0

Projekty/seminaria
0

Liczba punktów ECTS

5,00

Koordynatorzy

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

Wykładowcy

Wymagania wstępne

Efekte kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP (weryfikowane w procesie rekrutacji na studia 2 stopnia, prezentowane w serwisie internetowym wydziału www.cat.put.poznan.pl lub www.fc.put.poznan.pl), w szczególności umiejętność programowania we popularnych współcześnie językach (np. C/C++, Java), znajomość zagadnień z zakresu systemów operacyjnych i przetwarzania współbieżnego, orientacja w zagadnieniach programowania zastosowań sieciowych i przetwarzania rozproszonego.

Cel przedmiotu

1. przekazanie studentom podstawowej wiedzy z zakresu narzędzi przetwarzania rozproszonego oraz współtworzących je usług i mechanizmów, a także związanych z nimi technik i paradygmatów programowania rozproszonego, 2. rozwijanie u studentów umiejętności wyboru właściwego podejścia do rozwiązywania problemów przetwarzania rozproszonego i budowy systemów rozproszonych.

Przedmiotowe efekty uczenia się

Wiedza:

1. ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną związaną z paradygmatami tworzenia

systemów rozproszonych.

2. ma zaawansowaną wiedzę szczegółową dotyczącą wybranych środowisk i narzędzi stosowanych w tworzeniu aplikacji rozproszonych.
3. ma wiedzę o trendach rozwojowych w zakresie narzędzi i środowisk konstrukcji systemów rozproszonych.
4. ma zaawansowaną i szczegółową wiedzę o procesach zachodzących w cyklu życia systemów rozproszonych.
5. zna techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich w zakresie konstrukcji systemów rozproszonych.

Umiejętności:

1. potrafi planować i przeprowadzać eksperymenty, w tym pomiary i symulacje komputerowe, interpretować uzyskane wyniki i wyciągać wnioski oraz formułować i weryfikować hipotezy związane ze złożonymi problemami inżynierskimi z zakresu narzędzi przetwarzania rozproszonego.
2. potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich z zakresu narzędzi przetwarzania rozproszonego metody analityczne i eksperymentalne.
3. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych z zakresu narzędzi przetwarzania rozproszonego.
4. potrafi dokonać krytycznej analizy istniejących rozwiązań technicznych z zakresu narzędzi przetwarzania rozproszonego oraz zaproponować ich ulepszenia.
5. potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego z zakresu narzędzi przetwarzania rozproszonego, polegającego na budowie lub ocenie systemu informatycznego lub jego składowych, w tym dostrzec ograniczenia tych metod i narzędzi.
6. potrafi — zgodnie z zadaną specyfikacją, uwzględniającą aspekty pozatechniczne — zaprojektować system rozproszony oraz zrealizować ten projekt, co najmniej w części, używając właściwych metod, technik i narzędzi, w tym przystosowując do tego celu istniejące lub opracowując nowe narzędzia z zakresu przetwarzania rozproszonego.

Kompetencje społeczne:

1. rozumie, że w informatyce wiedza i umiejętności z zakresu narzędzi przetwarzania rozproszonego bardzo szybko stają się przestarzałe.
2. rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu narzędzi przetwarzania rozproszonego w rozwiązywaniu problemów badawczych i praktycznych.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

- a) w zakresie wykładów: na podstawie odpowiedzi na pytania dotyczące materiału omówionego wcześniej lub wiedzy z innych dziedzin/przedmiotów;
- b) w zakresie laboratoriów / ćwiczeń: na podstawie oceny bieżącego postępu realizacji zadań.

Ocena podsumowująca:

- a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez ocenę wiedzy i umiejętności wykazanych na egzaminie pisemnym o charakterze problemowym, składającym się z 4 pytań otwartych, nie znanych wcześniej, punktowanych w skali od 0 do 25, co daje maksymalnie 100 pkt. za cały egzamin, przy czym próg na ocenę pozytywną wynosi 50 pkt.,
- b) w zakresie laboratoriów / ćwiczeń weryfikowanie założonych efektów kształcenia realizowane jest przez ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych oraz ocenę postępu jakości i terminowości realizacji zadań projektowych, rejestrowanych w systemie zarządzania projektem. Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za: omówienie dodatkowych aspektów zagadnienia, efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu oraz planowania lub interpretacji wyników eksperymentów, uwagi związane z udoskonaleniem materiałów dydaktycznych, wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.

Treści programowe

Program wykładu obejmuje następujące zagadnienia:

- podstawowe pojęcia i koncepcje, w tym wyodrębnienie systemu sieciowego, systemu rozproszonego, warstwy pośredniej (middleware) oraz związane z tym aspekty zagadnienia projektowe, takie jak

transparentność, otwartość, interoperacyjność, skalowalność itp.;

- klasyfikacja mechanizmów komunikacji międzyprocesowej;
 - zdalne wywoływanie procedur z uwzględnieniem problemów realizacji protokołu w kontekście semantyki awarii, idempotentność procedur i bezstanowość usług opartych na RPC;
 - podejście obiektowe do budowy systemów rozproszonych obejmujące zagadnienia klasyfikacji obiektów w przetwarzaniu rozproszonym, definicji interfejsu i jego roli w określaniu typu parametrów, dziedziczenia;
 - współbieżność w realizacji zdalnych usług w tym wielowątkowość, współdzielenie zasobów i synchronizacja wątków, kontrola stopnia współbieżności;
 - systemy klasy MOM (ang. message-oriented middleware): koncepcja, paradygmat kolejkowania oraz publikowania-subskrypcji, przykłady realizacji oparte na brokerach oraz ZeroMQ;
 - przestrzeń krotek: koncepcja przestrzeni krotek oparta na modelu Linda, adresowanie asocjacyjne, specyfikacja JavaSpaces wraz przykładem realizacji;
 - środowisko Ada: podstawy programowania (typy danych, konstrukcje programotwórcze, pakiety, obiektowość), współbieżność (zadania, spotkania, obiekty chronione), przetwarzanie rozproszone oparte na Aneksie E specyfikacji (klasy jednostek kompilacji, język opisu systemu).
- Zajęcia laboratoryjne prowadzone są w formie piętnastu 2-godzinnych ćwiczeń, odbywających się w laboratorium wyposażony w stacje robocze pracujące pod kontrolą systemu operacyjnego Linux. Ćwiczenia realizowane są indywidualnie przez studentów, a w uzasadnionych przypadkach przez zespoły 2-osobowe. Program laboratorium obejmuje następujące zagadnienia:
- współbieżność na poziomie systemu operacyjnego: wątki w standardzie POSIX i ich synchronizacja, koncepcja monitora;
 - realizacja rozproszonego monitora z wykorzystaniem ZeroMQ jako mechanizmu komunikacji;
 - serializacja danych w oparciu o XDR oraz Protocol Buffers;
 - Sun RPC, gRPC: testowanie semantyki błędu na przykładzie prostej usługi zdalnego licznika;
 - Java RMI lub Internet Communications Engine: porównanie realizacji usługi zdalnego licznika w podejściu obiektowym oraz proceduralnym, realizacja zdalnego bufora;
 - systemy klasy MOM jako uogólnienie zdalnego bufora: testowanie i porównanie paradygmatu kolejkowania oraz publikowania-subskrypcji;
 - JavaSpaces: przykład wykorzystania adresowania asocjacyjnego w komunikacji międzyprocesowej;
 - podstawy programowania w języku Ada na przykładzie implementacji prostego problemu algorytmicznego;
 - współbieżność w środowisku Ada: tworzenie zadań, synchronizacja zadań poprzez spotkania, obiekty chronione, przykłady realizacji wybranych mechanizmów synchronizacji (np. semaforów) oraz implementacji rozwiązań klasycznych problemów synchronizacji (np. producenta-konsumenta);
 - realizacja przetwarzania rozproszonego w modelu zdefiniowanym przez Aneks E: przykłady użycia wybranych klas jednostek kompilacji (RCI, RT, SP).

Tematyka zajęć

Program wykładu obejmuje następujące zagadnienia:

- podstawowe pojęcia i koncepcje, w tym wyodrębnienie systemu sieciowego, systemu rozproszonego, warstwy pośredniej (middleware) oraz związane z tym aspekty zagadnienia projektowe, takie jak transparentność, otwartość, interoperacyjność, skalowalność itp.;
 - klasyfikacja mechanizmów komunikacji międzyprocesowej;
 - zdalne wywoływanie procedur z uwzględnieniem problemów realizacji protokołu w kontekście semantyki awarii, idempotentność procedur i bezstanowość usług opartych na RPC;
 - podejście obiektowe do budowy systemów rozproszonych obejmujące zagadnienia klasyfikacji obiektów w przetwarzaniu rozproszonym, definicji interfejsu i jego roli w określaniu typu parametrów, dziedziczenia;
 - współbieżność w realizacji zdalnych usług w tym wielowątkowość, współdzielenie zasobów i synchronizacja wątków, kontrola stopnia współbieżności;
 - systemy klasy MOM (ang. message-oriented middleware): koncepcja, paradygmat kolejkowania oraz publikowania-subskrypcji, przykłady realizacji oparte na brokerach oraz ZeroMQ;
 - przestrzeń krotek: koncepcja przestrzeni krotek oparta na modelu Linda, adresowanie asocjacyjne, specyfikacja JavaSpaces wraz przykładem realizacji;
 - środowisko Ada: podstawy programowania (typy danych, konstrukcje programotwórcze, pakiety, obiektowość), współbieżność (zadania, spotkania, obiekty chronione), przetwarzanie rozproszone oparte na Aneksie E specyfikacji (klasy jednostek kompilacji, język opisu systemu).
- Zajęcia laboratoryjne prowadzone są w formie piętnastu 2-godzinnych ćwiczeń, odbywających się w

laboratorium wyposażony w stacje robocze pracujące pod kontrolą systemu operacyjnego Linux. Ćwiczenia realizowane są indywidualnie przez studentów, a w uzasadnionych przypadkach przez zespoły 2-osobowe. Program laboratorium obejmuje następujące zagadnienia:

- współbieżność na poziomie systemu operacyjnego: wątki w standardzie POSIX i ich synchronizacja, koncepcja monitora;
- realizacja rozproszonego monitora z wykorzystaniem ZeroMQ jako mechanizmu komunikacji;
- serializacja danych w oparciu o XDR oraz Protocol Buffers;
- Sun RPC, gRPC: testowanie semantyki błędu na przykładzie prostej usługi zdalnego licznika;
- Java RMI lub Internet Communications Engine: porównanie realizacji usługi zdalnego licznika w podejściu obiektowym oraz proceduralnym, realizacja zdalnego bufora;
- systemy klasy MOM jako uogólnienie zdalnego bufora: testowanie i porównanie paradygmatu kolejkowania oraz publikowania-subskrypcji;
- JavaSpaces: przykład wykorzystania adresowania asocjacyjnego w komunikacji międzyprocesowej;
- podstawy programowania w języku Ada na przykładzie implementacji prostego problemu algorytmicznego;
- współbieżność w środowisku Ada: tworzenie zadań, synchronizacja zadań poprzez spotkania, obiekty chronione, przykłady realizacji wybranych mechanizmów synchronizacji (np. semaforów) oraz implementacji rozwiązań klasycznych problemów synchronizacji (np. producenta-konsumenta);
- realizacja przetwarzania rozproszonego w modelu zdefiniowanym przez Aneks E: przykłady użycia wybranych klas jednostek kompilacji (RCI, RT, SP).

Metody dydaktyczne

1. Wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, rozwiązywanie zadań.
2. Ćwiczenia laboratoryjne: wykonywanie eksperymentów, rozwiązywanie zadań, dyskusja, praca w zespole.

Literatura

Podstawowa

1. G. Coulouris, J. Dollimore, T. Kindberg, Systemy rozproszone. Podstawy i projektowanie, WNT, W-wa, 1999.
2. A.S. Tanenbaum, M. van Steen, Systemy rozproszone. Zasady i paradygmaty, WNT, W-wa, 2006.
3. M. van Steen, A.S. Tanenbaum, Distributed Systems, 4th ed., 2023.

Uzupełniająca

1. M. Gabassi, B. Dupouy, Przetwarzanie rozproszone w systemie UNIX, Lupus, W-wa, 1995.
2. E. Freeman, S. Hupfer, K. Arnold, JavaSpaces Principles, Patterns, and Practice, Addison-Wesley, 1999.
3. Z. Huzar, Z. Fryźlewicz, I. Dubielewicz, i in., Ada 95, Helion, Gliwice, 2001.
4. J. Barnes, Ada 2012, Cambridge University Press, 2014.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	125	5,00
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)	65	2,50