KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu
Programowanie sieciowe [S2Inf1-SRC>PSIEC]

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 (np. online)
0

Ćwiczenia
0

Projekty/seminaria
0

Liczba punktów ECTS

4,00

Koordynatorzy

dr inż. Michał Kalewski
michal.kalewski@put.poznan.pl

Wykładowcy

dr inż. Michał Kalewski
michal.kalewski@put.poznan.pl

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu systemów operacyjnych, programowania współbieżnego oraz sieci komputerowych. Student rozpoczynający ten przedmiot powinien posiadać umiejętność pozyskiwania informacji z literatury oraz innych źródeł (w języku ojczystym i angielskim); powinien potrafić wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody analityczne, symulacyjne oraz eksperymentalne; powinien potrafić integrować wiedzę z różnych obszarów informatyki (a w razie potrzeby także wiedzę z innych dyscyplin naukowych) oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne. 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

1. Przekazanie studentom wiedzy z dziedziny programowania sieciowego w zakresie prezentacji teoretycznych i praktycznych aspektów implementacji programów i systemów wykorzystujących współczesne sieci komputerowe. 2. Zapoznanie studentów z problematyką rozwiązań technicznych stosowanych obecnie przy implementacji aplikacji sieciowych, a szczególności z problematyką architektur i konstrukcji serwerów i klientów sieciowych oraz programowalnych sieci komputerowych. 3. Rozwijanie u studentów umiejętności rozwiązywania problemów, z jakimi spotyka się programista aplikacji sieciowych. 4. Zdobycie przez studentów umiejętności wykorzystania i konfigurowania istniejących protokołów i narzędzi systemowych do projektowania i tworzenia oprogramowania sieciowego.

Przedmiotowe efekty uczenia się

Wiedza:
ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną związaną z kluczowymi zagadnieniami z zakresu programowania sieciowego
ma zaawansowaną wiedzę szczegółową dotyczącą wybranych zagadnień z zakresu sieci komputerowych oraz tworzenia aplikacji sieciowych
ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach informatyki i innych, wybranych, pokrewnych dyscyplin naukowych
ma zaawansowaną i szczegółową wiedzę o procesach zachodzących w cyklu życia aplikacji serwerów i klientów sieciowych
zna zaawansowane metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich i prowadzeniu prac badawczych w zakresie programowania sieciowego i tworzenia aplikacji sieciowych

Umiejętności:
potrafi pozyskiwać informacje z literatury, baz danych oraz innych źródeł (w języku polskim i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie
potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody analityczne, symulacyjne oraz eksperymentalne
potrafi — przy formułowaniu i rozwiązywaniu zadań inżynierskich — integrować wiedzę z różnych obszarów informatyki (a w razie potrzeby także wiedzę z innych dyscyplin naukowych) oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne
potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych
potrafi dokonać krytycznej analizy istniejących rozwiązań technicznych oraz zaproponować ich ulepszenia (usprawnienia)
potrafi współdziałać w zespole, przyjmując w nim różne role
potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia, w tym innych osób

Kompetencje społeczne:
rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe
rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu informatyki w rozwiązywaniu problemów badawczych i praktycznych
rozumie znaczenie działalności popularyzatorskiej dotyczącej najnowszych osiągnięć z zakresu informatyki

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Ocena podsumowująca:
a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:
- ocenę wiedzy i umiejętności wykazanych na zaliczeniu pisemnym obejmującym około 5 zagadnień omawianych w ramach wykładów o różnej wartości punktowej wg następującej skali:
- 0 - 50% ocena: 2,0
- 51 - 60% ocena: 3,0
- 61 - 70% ocena: 3,5
- 71 - 80% ocena: 4,0
- 81 - 90% ocena: 4,5

- 91 - 100% ocena: 5,0
- omówienie wyników zaliczenia
b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:
- ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych i projektów zaliczeniowych,
- ocenianie ciągłe, na każdych zajęciach (odpowiedzi ustne) - premiowanie przyrostu umiejętności posługiwania się poznanymi zasadami i metodami,
- ocenę wiedzy i umiejętności związanych z realizacją zadań projektowych;
uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:
- omówienia dodatkowych aspektów zagadnienia,
- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,
- umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium,
- uwagi związane z udoskonaleniem materiałów dydaktycznych,
- wskazywanie trudności percepcyjnych studentów, umożliwiające bieżące doskonalenia procesu dydaktycznego.

Treści programowe

Program wykładu obejmuje następujące zagadnienia:
- Wprowadzenie: przypomnienie materiału dotyczącego modelu warstwowego sieci komputerowych oraz podstawowych datagramowych (UDP) i strumieniowych (TCP) gniazd sieciowych protokołu IPv4.
- Bufory nadawcze i odbiorcze gniazd sieciowych oraz metody i narzędzia kapsułkowania danych aplikacyjnych w transmisjach strumieniowych.
- Opcje systemowe i funkcje pomocnicze dla komunikacji internetowej interfejsu gniazd sieciowych oraz obsługa systemu nazw domenowych DNS.
- Dostęp programistyczny do ramek warstwy łącza danych (biblioteki libpcap oraz libnet) oraz pakietów sieciowych w jądrze systemu operacyjnego (biblioteki libnetfilter).
- Surowe gniazda sieciowe dla protokołów IPv4 i IPv6 oraz gniazda typu SOCK_PACKET i PF_PACKET.
- Gniazda sieciowe protokołu SCTP (Stream Control Transmission Protocol) oraz implementacje komunikacji wielo-strumieniowych.
- Datagramowe i strumieniowe gniazda sieciowe protokołu IPv6; współdziałanie procesów wykorzystujących protokoły IPv4 i IPv6; konstrukcje dualnych serwerów sieciowych.
- Mechanizmy i architektury współbieżnych serwerów sieciowych: nieblokujące funkcje sieciowe; multipleksacja wejścia-wyjścia (funkcje systemowe select, poll, epoll i kqueue); funkcje sieciowe wyzwalane sygnałami systemowymi; współbieżność z zastosowaniem procesów potomnych oraz wątków; pule procesów potomnych oraz wątków i mechanizmy ich aktywacji do obsługi komunikacji sieciowej; mechanizm SO_REUSEPORT.
- Metody i mechanizmy detekcji rozłączeń w strumieniowej komunikacji internetowej.
- Implementacje internetowej komunikacji rozgłoszeniowej oraz multicastowej z użyciem interfejsu gniazd sieciowych.
- Interfejsy programistyczne do obsługi tablic routingu i pamięci podręcznej protokołu ARP oraz gniazda sieciowe zarządzania kluczami (typu PF_KEY).
- Interfejs gniazd sieciowych w systemach operacyjnych Windows (winsock), Windows Phone, Android i iOS; wykorzystanie interfejsu gniazd sieciowych do realizacji komunikacji w bezprzewodowych sieciach mobilnych.
- Mechanizmy obsługi komunikacji sieciowej w aplikacjach z graficznym interfejsem użytkownika.
- Implementacje komunikacji internetowej z zastosowaniem protokołów warstwy aplikacji (biblioteka libcurl).
- Wprowadzenie do programowalnych sieci komputerowych SDN (Software-Defined Networking), protokołu OpenFlow oraz emulatora Mininet; wykorzystanie protokołu OpenFlow do implementacji kontrolerów urządzeń sieciowych w sieciach SDN.
Program laboratorium obejmuje następujące zagadnienia:
- Wykorzystanie bibliotek libpcap, libnet oraz libnetfilter do implementacji programów przechwytujących ramki warstwy łącza danych oraz pakiety sieciowe w jądrze systemu operacyjnego.
- Implementacje prostych narzędzi trasowania w sieciach IPv4 i IPv6 z użyciem surowych gniazd sieciowych (ICMPv4 i ICMPv6).
- Implementacje aplikacji klienta i serwera wykorzystujących gniazda sieciowe protokołu SCTP i połączenia wielo-strumieniowe oraz testy efektywnościowe takich połączeń.
- Implementacje aplikacji klientów i serwerów z użyciem datagramowych i strumieniowych gniazd sieciowych protokołu IPv6; modyfikacje aplikacji serwerów do rozwiązań dualnych (IPv4/IPv6).

- Implementacja wybranych mechanizmów i architektur serwerów współbieżnych i ich porównawcze testy efektywnościowe.
- Implementacje komunikacji rozgłoszeniowych i multicastowych w sieciach internetowych.
- Implementacje programów manipulujących tablicami routingu i pamięcią podręczną protokołu ARP.
- Implementacje aplikacji klientów sieciowych dla wybranych systemów operacyjnych urządzeń mobilnych, obsługa graficznych interfejsów użytkownika oraz komunikacja w bezprzewodowych sieciach mobilnych.
- Wykorzystanie biblioteki libcurl do implementacji komunikacji internetowej z zastosowaniem protokołów warstwy aplikacji.
- Obsługa emulatora Mininet oraz implementacje kontrolerów urządzeń sieciowych w sieciach SDN z zastosowaniem protokołu OpenFlow.

Metody dydaktyczne

Wykład: prezentacja multimedialna, ilustrowana przykładami podawanymi na tablicy.
Ćwiczenia laboratoryjne: prezentacja multimedialna prezentacja ilustrowana przykładami podawanymi na tablicy oraz wykonanie zadań podanych przez prowadzącego - ćwiczenia praktyczne.

Literatura

Podstawowa
1. UNIX - programowanie usług sieciowych. 1, API: gniazd a i XTI, Stevens W. R., Wydawnictwa Naukowo - Techniczne, 2002
2. Sieci komputerowe TCP/IP. 3, Programowanie w trybie klient- serwer, wersja BSD, Comer D., Stevens D., Wydawnictwa Naukowo-Techniczne, 1997
3. Computer Networks, A. S. Tanenbaum, Pearson, 2014
Uzupełniająca
1. Data and Computer Communications. Networking and Internetworking, Hura G. S., Singhal M., CRC Press LLC, 2001
2. Client/Server Survival Guide, Harkey D., Wiley, 1999
3. Wireless Communications and Networks, Stallings W., Pearson, Prentice Hall, 2002
4. Algorithms and Protocols for Wireless, Mobile Ad Hoc Networks, Boukerche A., Wiley-IEEE Press, 2008

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 kolokwiów/egzaminu, wykonanie projektu)	40	1,50