**Nazwa przedmiotu:**

Programowanie sieciowe

**Koordynator przedmiotu:**

Rajmund Kożuszek

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne

**Kod przedmiotu:**

PSI

**Semestr nominalny:**

5 / rok ak. 2021/2022

**Liczba punktów ECTS:**

4

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

1. liczba godzin kontaktowych – 53 godz., w tym
 obecność na wykładach: 26 godz.,
 obecność na kolokwiach: 4 godz.
 obecność na zajęciach laboratoryjnych:15 godz.,
 udział w konsultacjach związanych z problematyką poruszaną na wykładzie//laboratorium/zajęcia wprowadzające do projektu: 4 godz.,
 udział w konsultacjach projektowych: 4 godz.,
2. praca własna studenta – 51 godz., w tym
 przygotowanie raportów z zajęć laboratoryjnych: 5 godz.
 analiza literatury i materiałów wykładowych związana z przygotowaniem do kolejnych wykładów, zajęć laboratoryjnych, projektu, instalacja oprogramowania: 10 godz.,
 realizacja zadań projektowych, przygotowanie raportu: 30 godz. ,
 przygotowanie do kolokwium: 6 godz.
Łączny nakład pracy studenta wynosi 104 godz., co odpowiada 4 pkt. ECTS.

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

2,0 pkt. ECTS, co odpowiada 53 godz. kontaktowym

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

2 pkt. ECTS, co odpowiada 50 godz. zajęć laboratoryjnych i projektowych przygotowaniu do tych zajęć oraz przygotowanie raportu

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

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

**Wymagania wstępne:**

Podstawy programowania, Sieci komputerowe

**Limit liczby studentów:**

150

**Cel przedmiotu:**

Celem przedmiotu jest zdobycie umiejętności tworzenia programowania realizującego komunikację sieciową na poziomie warstw 4 – 7 modelu OSI.
W ramach wykładu omawiane są podstawowe metody programowania sieciowego: gniazda, zdalne wywołanie procedur (RPC) oraz metody tworzenia serwisów sieciowych, poruszane są też kwestie bezpieczeństwa tworzonych rozwiązań. Na ćwiczeniach laboratoryjnych omawiane są praktyczne aspekty tworzenia oprogramowania sieciowego a zasadniczym sprawdzianem nabytych umiejętności jest projekt.

**Treści kształcenia:**

WYKŁADY:
1. Założenia programowania sieciowego w kontekście warstw ISO/OSI - rys historyczny, przypomnienie podstawowych kwestii związanych z protokołami internetowymi; przypomnienie podstawowych założeń protokołów TCP/UDP/IP/ICMP; model klient-serwer (2 godziny). 2
2. Podstawy programowania z wykorzystaniem interfejsu gniazd (BSD sockets) - koncepcja asocjacji sieciowej i gniazda; posługiwanie się adresami IPv4 i IPv6; podstawowe funkcje systemowe: socket(), bind(), listen(), connect(), accept(); funkcje we/wy w kontekście sieciowym (różnice w stosunku do obsługi plików); obsługa biblioteki resolvera (4 godziny) 6
3. Średnio-zaawansowane aspekty interfejsu gniazd – funkcje recv...(), send...(); protokoły konwersacyjne; obsługa OOB, praca z wieloma gniazdami równolegle: (poll/select); wykorzystanie "opcji" gniazd; rozłączanie połączenia i obsługa błędów; rozgłaszanie i rozsiewanie; programowanie serwera iteracyjnego i współpbieżnego. (3 godziny) 9
4. Programowanie serwisów sieciowych z wykorzystaniem gniazd; szkielety programu klienta i serwera dla protokołów transportowych UDP i TCP. (2 godziny) 11
5. Zaawansowane kwestie związane z programowaniem z wykorzystaniem gniazd: wielowątkowosć i wydajność; optymalizacja kodu; pseudo-terminale; interfejs XTI jako alternatywa do gniazd – podobieństwa i różnice; przenośność - gniazda w innych środowiskach systemowo-językowych: winsock, programowanie sieciowe w języku Java i Python. (4 godziny) 15
6. Bezpieczeństwo warstwy transportowej – krótkie przypomnienie tematyki PKI, kostrukcja certyfikatu X509v3; korzystanie z protokółu SSL na poziomie API. (3 godziny) 18
7. Programowanie w modelu RPC (zdalne wywołanie procedur) - koncepcja RPC, podstawowe założenia i filozofia programowania RPC. ONC RPC: program rpcgen i standard XDR. Inne system RPC – RMI (Java) oraz RPC w systemach z rodziny Windows. Przykłady programowania RPC: analiza prostego systemu zrealizowany z wykorzystaniem rpcgen, analiza kodu i przykady zaawansowanych funkcji RPC - kontrola transportu, rozgłaszanie, oprogramowanie pętli serwisowej. (4 godziny) 22
8. Programowanie serwisow sieciowych warstwy 7: serwery aplikacyjne, serwlety, sesje, bezpieczeństwo sesji; przetwarzanie danych MIME, model AJAX i JSON; wybrane aspekty oprogramowania komunikacji w protokole HTTP 1.1 i 2.0 oraz CoAP (2 godziny) 24
9. Inne paradygmaty programowania sieciowego; obliczenia masywnie równoległe; modele OpenMP i MPI; model map-reduce; inne wybrane modele obliczeniowe. (2 godziny) 26
LABORATORIUM:
A. Zajęcia praktyczne
1. Podstawy programowania z wykorzystaniem gniazd: narzędzia, środowisko, nagłówki, zasady tworzenia kodu; Wykorzystanie biblioteki resolvera
2. Tworzenie i analiza prostego kodu klienta serwer dla transportu UDP i TCP
3. Rozbudowa kodu z p.3 – wprowadzenie wielowątkowości; obsługa IPv6
4. Rozbudowa kodu z p.3 – jednoczesna obsługa wiekszej liczby połączeń (select/poll)
5. Opcja - analiza fragmentów kodu wybrane popularnego pakietu open source
6. Tworzenie i testowanie kodu RPC; studium przypadku – od programu lokalnego do rozproszonego – jak przejść z programu stacjonarnego do rozproszonego (krok-po-kroku)
PROJEKT:
W ramach projektu zespół 3-4 osobowy ma za zadanie opracować system klient-serwer bazujący na modelu gniazd lub modelu RPC. Celem jest praktyczna weryfikacja zdobytej na wykładzie wiedzy w stosunkowo szerokim zakresie poruszanych zagadnień: komunikacji sieciowej, jej aspektów średniozaawansowanych (multipleksowanie gniazd, wielowątkowosć, wydajność, itd.). Studenci muszą wykazać sie umiejętnościa projektowania protokołu a następnie jego poprawnej implementacji. Nacisk położony będzie na stronę dokumentacyjną w zakresie opisu interfejsu komunikacji oraz opisu samego protokołu. Istotny jest też aspekt testowania tworzonego kodu, które powinno być wielopoziomowe (testy jednostkowe, testy poprawnosci implementacji, testowanie sytuacji wyjątkowych, np. nieoczekiwanego rozłączenia sesji w wyniku wystąpienia błędu, itd).

**Metody oceny:**

Realizacja przedmiotu obejmuje następujące formy zajęć:
 wykład prowadzony w wymiarze 2 godz. tygodniowo;
 zajęcia laboratoryjne; w ramach tych zajęć student, korzystając z oprogramowania i sprzętu będzie – pod opieka prowadzącego zajęcia – realizował wskazane zadania związane tematycznie z treścią wykładu;
 zajęcia projektowe; w ramach tych zajęć student – korzystając z konsultacji prowadzącego zajęcia – będzie wykonywał zadanie związane ze specyfikacją i realizacją prostego systemu cyfrowego.
Sprawdzanie założonych efektów kształcenia realizowane jest przez:
 ocenę wiedzy i umiejętności związanych z realizacją zadań laboratoryjnych – ocenę sprawozdań z realizacji zadań;.
 ocenę wiedzy i umiejętności związanych z realizacją zadań projektowych – ocenę prezentacji i raport;
 ocenę wiedzy i umiejętności wykazanych na dwóch kolokwiach o charakterze testowo/problemowym.

**Egzamin:**

nie

**Literatura:**

1. Richard Stevens, "UNIX: programowanie usług sieciowych", Wydawnictwa Naukowo-Techniczne, Warszawa (2002);
2. Andrew Tanenbaum, et al., "Sieci komputerowe", Wydawnictwa Naukowo-Techniczne, 1988.
3. Douglas Comer, Grzegorz Grudziński, Aleksy Schubert, "Sieci komputerowe i intersieci", Wydawnictwa Naukowo-Techniczne, 1999.
4. David Butenhof, "Programming with POSIX Threads", Addison-Weslery, 2006
5. Marc Rochkind, "Programowanie w systemie Unix dla zaawansowanych", Wydawnictwa Naukowo-Techniczne, Warszawa 1997.

**Witryna www przedmiotu:**

https://usosweb.usos.pw.edu.pl/kontroler.php?\_action=katalog2/przedmioty/pokazPrzedmiot&prz\_kod=103A-INxxx-ISP-PSI

**Uwagi:**

(-)

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka W01:**

zna techniki programowania protokołów sieciowych

Weryfikacja:

projekt, laboratorium, kolokwium

**Powiązane charakterystyki kierunkowe:** W08, W10

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o, III.P6S\_WG

**Charakterystyka W02:**

zna podstawowe narzędzia służące do tworzenia oprogramowania sieciowego

Weryfikacja:

projekt, laboratorium, kolokwium

**Powiązane charakterystyki kierunkowe:** W08

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o, III.P6S\_WG

### Profil ogólnoakademicki - umiejętności

**Charakterystyka U01:**

potrafi przeanalizować, zmodyfikować stworzyć samodzielnie oraz przetestować kod systemu klient–serwer działającego w środowisku sieciowym

Weryfikacja:

laboratorium, projekt

**Powiązane charakterystyki kierunkowe:** U01, U02, U07, U08, U09

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o, I.P6S\_UO, I.P6S\_UK

**Charakterystyka U02:**

potrafi przeanalizować, zmodyfikować stworzyć samodzielnie oraz przetestować kod systemu RPC

Weryfikacja:

laboratorium, projekt

**Powiązane charakterystyki kierunkowe:** U07, U08, U09, U01, U02

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o, I.P6S\_UO, I.P6S\_UK

**Charakterystyka U03:**

potrafi tworzyć bezpieczny kod sieciowy

Weryfikacja:

projekt

**Powiązane charakterystyki kierunkowe:** U02, U07, U09

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o, I.P6S\_UK

**Charakterystyka U04:**

potrafi zaplanować i przeprowadzić eksperyment, przedstawić wyniki z badań i pomiarów w formie czytelnego sprawozdania

Weryfikacja:

laboratorium, projekt

**Powiązane charakterystyki kierunkowe:** U03, U09, U10

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o, I.P6S\_UK

### Profil ogólnoakademicki - kompetencje społeczne

**Charakterystyka K01:**

rozumie znaczenie wiedzy w rozwiązywaniu problemów poznawczych i praktycznych oraz potrzebę zasięgania opinii ekspertów w przypadku trudności w samodzielnym rozwiązywaniu problemu

Weryfikacja:

laboratorium, projekt

**Powiązane charakterystyki kierunkowe:** K03

**Powiązane charakterystyki obszarowe:** P6U\_K, I.P6S\_KK, I.P6S\_KR

**Charakterystyka K02:**

jest świadomy procesu uczenia się w kierunku zwiększania kompetencji w tym obszarze

Weryfikacja:

projekt

**Powiązane charakterystyki kierunkowe:** K01

**Powiązane charakterystyki obszarowe:** P6U\_K, I.P6S\_KK