Wojskowa Akademia Techniczna

Wydział Elektroniki

**Projektowanie aplikacji sieciowych**

Zadania laboratoryjne

Opracował:

ppłk dr inż. Jarosław Krygier

mgr inż. Sebastian Szwaczyk

mgr inż. Jakub Banaszek

Warszawa 2017

Spis treści

[Spis treści 2](#__RefHeading___Toc453_1008357531)

[Przygotowanie do ćwiczeń laboratoryjnych: 3](#__RefHeading___Toc455_1008357531)

[1.Laboratorium 1 – gniazdo typu RAW 4](#__RefHeading___Toc457_1008357531)

[Zadanie 1. Sniffer 4](#__RefHeading___Toc459_1008357531)

[Zadanie 2. Generator 4](#__RefHeading___Toc461_1008357531)

[2.Laboratorium 2 – opcje wspomagające wykorzystanie gniazd 5](#__RefHeading___Toc463_1008357531)

[ZADANIE 1. Gniazdo nieblokowane 5](#__RefHeading___Toc465_1008357531)

[ZADANIE 2. Wykorzystanie serwera DNS 5](#__RefHeading___Toc467_1008357531)

[ZADANIE 3. Komunikacja międzyprocesowa 5](#__RefHeading___Toc469_1008357531)

[3.Laboratorium 3 – Serwer TCP z wykorzystaniem funkcji select 6](#__RefHeading___Toc471_1008357531)

[ZADANIE 1. Serwer TCP 6](#__RefHeading___Toc473_1008357531)

[ZADANIE 2. Serwer TCP – funkcja select 6](#__RefHeading___Toc293_589627818)

[ZADANIE 3. Serwer TCP – broadcast 6](#__RefHeading___Toc295_589627818)

[4.Zadanie projektowe 7](#__RefHeading___Toc475_1008357531)

[Przygotowanie do ćwiczenia: 7](#__RefHeading___Toc477_1008357531)

[Realizacja ćwiczenia: 7](#__RefHeading___Toc479_1008357531)

[Wymagania: 7](#__RefHeading___Toc481_1008357531)

Przygotowanie do ćwiczeń laboratoryjnych:

1. Pobrać, zainstalować i zapoznać się z oprogramowaniem Eclipse.
2. Zapoznać się z materiałami z wykładów - samodzielnie przećwiczyć zadania dodatkowe.
3. Zapoznać się z podstawowymi poleceniami systemu pomocy Linux.
4. Laboratorium 1 – gniazdo typu RAW

Celem ćwiczenia jest zapoznanie studentów z wykorzystaniem gniazd typu RAW, zarówno do odbioru jak i nadawania ramek Ethernet.

Zadanie 1. Sniffer

Zadanie ma na celu odebranie i wyświetlenie zawartości ramki Ethernet.

1. Utworzyć nowy projekt dla języka C w IDE Eclipse typu "Hello world ANSI C Project".
2. Utworzyć gniazdo typu RAW pozwalające na odbiór wszystkich ramek Ethernetowych z interfejsu zadanego przez prowadzącego.
3. Przygotować struktury pozwalające na przechowanie nagłówków Ethernet i IPv4.
4. Zadeklarować pętlę nieskończoną, w której:
   * + - 1. Odbierać kolejne pakiety z interfejsu sieciowego;
         2. Filtrować tylko pakiety IPv4;
         3. Odwzorować odebrany ciąg bajtów na utworzone wcześniej struktury;
         4. Wypisać zawartość struktur;

Zadanie 2. Generator

Zadanie ma na celu wygenerowanie dowolnej ramki.

1. Utworzyć nowy projekt dla języka C w IDE Eclipse typu "Hello world ANSI C Project".
2. Utworzyć gniazdo typu RAW pozwalające na wysłanie dowolnie sformatowanej ramki.
3. Uzupełnić struktury stworzone w zadaniu nr 1, tak aby nadawana ramka była pakietem IPv4.
4. Jako dane w ramce IPv4 wpisać tekst podany przez prowadzącego.
5. Wysłać ramkę do interfejsu podanego przez prowadzącego.
6. Przechwycić wysłany pakiet w wiresharku i porównać czy przechwycone wartości zgadzają się z wartościami ustawionymi w strukturach.
7. Laboratorium 2 – opcje wspomagające wykorzystanie gniazd

Celem laboratorium jest zapoznanie studentów z dodatkowymi opcjami i funkcjami wspomagającymi pracę z gniazdami.

ZADANIE 1. Gniazdo nieblokowane

1. Utworzyć nowy projekt dla języka C w IDE Eclipse typu "Hello world ANSI C Project".
2. Stworzyć program klient\_TCP
3. Klient ma zestawiać połączenia do serwera na zdefiniowany numer portu serwera i mieć możliwość wysyłania komunikatów do serwera
4. Wykorzystaj funkcje socket(), connect(), send(), read(), write() i close()
5. Stworzyć plik serwera TCP
6. Serwer ma odbierać wiadomości przesłane od klienta
7. Wykorzystaj funkcje socket() bind() listen () write () read () i close() – serwer ma oczekiwać na połączenie z klientem, akceptować nowe połączenia i blokować po zamknięciu połączenia z klientem
8. Zmodyfikować odpowiednio utworzone gniazda wykorzystują funkcje fcntl(deskryptor gniazda, przeprowadzona operacja, argument), tak aby gniazdo było nieblokowalne, przykład:   
   ifcntl(sock, F\_GETFL),

ifcntl(sock, F\_SETFL, O\_NONBLOCK)

ZADANIE 2. Wykorzystanie serwera DNS

1. Utworzyć nowy projekt dla języka C w IDE Eclipse typu "Hello world ANSI C Project".
2. Program ma wysłać zapytanie DNS , wykorzystaj w tym celu funkcje gethostbyname() / getaddrinfo()
3. Program musi być odpowiednio zabezpieczony, jeżeli adres podany jest blednie powinien wyświetlić odpowiedni komunikat
4. Finalnym wynikiem ma być wyświetlony w konsoli adres IP podanej strony

ZADANIE 3. Komunikacja międzyprocesowa

1. Utworzyć nowy projekt dla języka C w IDE Eclipse typu "Hello world ANSI C Project".
2. Utworzyć proces
3. Utworzyć proces potomny do wcześniej wywołanego procesu
4. Do komunikacji miedzy procesami wykorzystaj mechanizm pamięci dzielonej:
   1. tworzenie obszaru
   2. ustalanie rozmiaru
   3. odwzorowanie pamięci
   4. odłączenie się od segmentu
5. Utwórz trzy przykładowe komunikaty miedzy procesem potomnym a rodzicem
6. Laboratorium 3 – Serwer TCP z wykorzystaniem funkcji select

Celem laboratorium jest zapoznanie studentów z działaniem funkcji *select* na przykładzie serwera TCP.

ZADANIE 1. Serwer TCP

1. Utworzyć nowy projekt dla języka C w IDE Eclipse typu "Hello world ANSI C Project".
2. Otworzyć gniazdo pozwalające na komunikację TCP
3. Nasłuchiwać połączeń na wybranym porcie
4. Po nawiązaniu połączenia wysłać wiadomość powitalną do klienta
5. Oczekiwać na informacje od klienta i odpowiadać echem

Poprawność działania serwera należy zwerifkować za pomocą klienta *telnet*.

ZADANIE 2. Serwer TCP – funkcja select

1. Do projektu stworzonego w zadaniu 1, dodać możliwość obsługi wielu klientów naraz, za pomocą funkcji select

Poprawność działania serwera należy zweryfikować za pomocą kilku jednoczesnych połączeń klientów *telnet*.

ZADANIE 3. Serwer TCP – broadcast

1. Do projektu stworzonego w zadaniu 2, dodać możliwość wysłania wiadomości do wszystkich aktualnie podłączonych klientów.
2. Należy jednocześnie oczekiwać na dane od klientów (i odpowiadać im echem) jak i na dane wprowadzone do konsoli serwera.
3. Po wykryciu wprowadzania danych w konsoli serwera, należy odebraną wiadomość przesłać do wszystkich aktualnie podłączonych klientów.

Poprawność działania serwera należy zweryfikować za pomocą kilku jednoczesnych połączeń klientów *telnet*.

1. Zadanie projektowe

Celem zadania projektowego jest stworzenie klienta i wieloprocesowego serwera TCP.

Przygotowanie do ćwiczenia:

W ramach przygotowania do ćwiczenia należy zapoznać się ze sposobem tworzenia zarówno aplikacji klienckich jak i serwerów, opartych o protokół TCP, w języku C.

Realizacja ćwiczenia:

1. Każdy ze studentów realizuje projekt samodzielnie.
2. Program **serwera** ma realizować następujące funkcje:
   1. możliwość podania z linii poleceń portu, na, którym ma nasłuchiwać;
   2. możliwość obsługi wielu klientów jednocześnie poprzez wykorzystanie procesów potomnych;
   3. dla każdego z klientów serwer tworzy listę wiązaną, w której przechowuje kolejne wartości liczb całkowitych przesłane przez klienta;
   4. na żądanie klienta serwer wykonuje jedną z operacji:
      1. sumowanie przesłanych liczb
      2. odejmowanie przesłanych liczb
      3. uszeregowanie rosnąco
      4. uszeregowanie malejąco

i wysyła odpowiedź do klienta;

* 1. po wysłaniu odpowiedzi serwer czeka na kolejne informacje od klienta
  2. po zakończeniu połączenia przez klienta proces obsługujący go powinien zostać zamknięty

1. Program klienta ma pozwalać na realizację następujących funkcji:
   1. możliwość podania z linii poleceń adresu IP i portu, na którym nasłuchuje serwerem
   2. możliwość nawiązania połączenia TCP z serwerem
   3. możliwość przesłania liczb wprowadzonych przez użytkownika do serwera
   4. możliwość żądania wykonania przez serwer jednej z podanych w punkcie 2.d operacji
   5. Wyświetlenie odebranego wyniku
   6. możliwość podjęcia decyzji przez użytkownika o zakończeniu lub dalszym działaniu programu

Wymagania:

1. Historia prowadzenia i dokumentacja projektu powinna znajdować się na platformie github.com (lub innej obsługującej system kontroli wersji git).