**Wojskowa Akademia Techniczna**

**Wydział Elektroniki**

**Instytut Telekomunikacji**

**Studia II°**

**Projektowanie aplikacji sieciowych**

Materiały pomocnicze do zajęć

Cz.4.

**dr inż. Jarosław Krygier**

Warszawa 2017

Spis treści

[1 Gniazdo strumieniowe – serwer TCP 3](#_Toc496606169)

[2 Gniazdo strumieniowe – klient TCP 4](#_Toc496606170)

# Gniazdo strumieniowe – serwer TCP

/\*

============================================================================

Name : PAS\_5\_gniazda\_serwer\_tcp.c

Author :

Version :

Copyright :

Description : Gniazda: serwer - gniazdo strumieniowe (TCP)

============================================================================

\*/

#include <stdio.h>

#include <string.h> //strlen

#include <sys/socket.h>

#include <arpa/inet.h> //inet\_addr

#include <unistd.h> //write

int main(int argc , char \*argv[])

{

int socket\_desc , new\_socket , c;

struct sockaddr\_in server , client;

int read\_size;

char \*message , client\_message[3000];

//Utworz gniazdo

socket\_desc = socket(AF\_INET , SOCK\_STREAM , 0);

if (socket\_desc == -1)

{

printf("Could not create socket");

}

//Przygotuj addres gniazda

server.sin\_family = AF\_INET;

server.sin\_addr.s\_addr = INADDR\_ANY;

server.sin\_port = htons( 8888 );

//Polacz deskrypor gniazda (gniazdo) z przypisanym adresem

if( bind(socket\_desc,(struct sockaddr \*)&server , sizeof(server)) < 0)

{

puts("Problem dolaczenia gniazda");

}

puts("Dolaczono gniazdo strumieniowe");

//Nasluchiwanie na gniezdzie (serwer)

listen(socket\_desc , 3);

//Akceptuj polaczenie przychodzace

puts("SERWER TCP: Czekam na polaczenie...");

c = sizeof(struct sockaddr\_in);

new\_socket = accept(socket\_desc, (struct sockaddr \*)&client, (socklen\_t\*)&c);

if (new\_socket<0)

{

perror("Problem z akceptacja poloczenia");

}

puts("Polaczenie zaakceptowane");

//Odpowiedz do klienta

message = "Witam, poloczeyles sie z serwerem TCP. Czekam na dane...\n";

write(new\_socket , message , strlen(message));

#if 1

while (1) {

while( (read\_size = recv(new\_socket, client\_message , 3000 , 0)) > 0 ){

puts("SERWER: Dane odebrano");

puts (client\_message);

message = "Info od serwera: odebralem dane";

//Wyslij echo

write(new\_socket , message , strlen(message));

}

}

#endif

return 0;

}

# Gniazdo strumieniowe – klient TCP

/\*

============================================================================

Name : PAS\_4\_gniazda\_klient\_tcp.c

Author : J. Krygier

Version :

Copyright : WAT

Description : Gniazda: kilent - gniazdo strumieniowe (TCP)

============================================================================

\*/

#include <stdio.h>

#include <sys/socket.h> //socket

#include<arpa/inet.h> //inet\_addr

#include<string.h> //strlen

#include <unistd.h> //close

int main(int argc , char \*argv[])

{

int socket\_desc = -1;

struct sockaddr\_in server;

char \*message , server\_reply[2000];

socket\_desc = socket(AF\_INET , SOCK\_STREAM , 0); // gniazdo strumieniowe (transmisja TCP)

if (socket\_desc == -1){

printf("Nie moge utworzyc gniazda \n");

} else {

printf("Utworzono gniazdo o numerze: %d \n", socket\_desc);

}

//adres serwera zdalnego

//server.sin\_addr.s\_addr = inet\_addr("172.217.20.206");

server.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

server.sin\_family = AF\_INET;

//server.sin\_port = htons( 80 );

server.sin\_port = htons( 8888 );

//Polacz z serwerem zdalnym (TCP)

if (connect(socket\_desc , (struct sockaddr \*)&server , sizeof(server)) < 0) {

puts("Blad polaczenia");

return 1;

}

puts("Polaczono");

sleep (5);

#if 1

//Wyslij jakies dane

message = "GET / HTTP/1.1\r\n\r\n";

// funkcja do wysyania danych (TCP)

if( send(socket\_desc , message , strlen(message) , 0) < 0) {

puts("Blad wyslania danych");

return 1;

}

puts("Dane wyslano\n");

#endif

while (1) {

//Nasluchiwanie i odbior danych z serwera (TCP)

if( recv(socket\_desc, server\_reply , 2000 , 0) < 0){

puts("Blad odbioru danych");

}

puts("Dane odebrano\n");

puts(server\_reply);

memset (server\_reply, 0, sizeof (server\_reply));

sleep (1);

//wysylanie

if( send(socket\_desc , message , strlen(message) , 0) < 0) {

puts("Blad wyslania danych");

return 1;

}

puts("Dane wyslano\n");

}

return 0;

}