

## 1 Klijentski deo koda

client.c

```
/*
 * Vrlo prost primer klijenta
 *
 * Koristi se bafer fiksne velicine
 * da bi se u njega upisao sadrzaj
 * jednog od argumenata predatih prilikom
 * pokretanja programa
 *
 * Komunikacija se odvija tako sto se
 * posalje sadrzaj serveru i od servera
 * ocekuje odgovor
 *
 */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

void error(const char *msg)
{
    //funkcija koja ispisuje predatu poruku
    //i izlazi iz programa
    perror(msg);
    exit(1);
}

int main(int argc, char *argv[])
{
    int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server;

    char buffer[256];

    if (argc < 4) {
        fprintf(stderr,
            "usage %s hostname port message(without spaces)\n",
            argv[0]);
        exit(0);
    }
}
```

```
portno = atoi(argv[2]); //ASCII TO INTEGER

sockfd = socket(AF_INET, SOCK_STREAM, 0); //pravljenje socket-a
//ADDRESS FAMILY je INTERNET a tip socket-a je STREAM za
//uspostavljanje
//direktne veze

if (sockfd < 0)
    error("ERROR opening socket");

server = gethostbyname(argv[1]);
//Prihvatanje podataka o serveru na
//osnovu njegove adrese predate pri
//pokretanju programa

perror("Connecting to host");

bzero((char *)&serv_addr, sizeof(serv_addr)); // nuliranje cele
//strukture

serv_addr.sin_family = AF_INET; //odabir Internet adresa
bcopy((char *)server->h_addr, (char *)&serv_addr.sin_addr.
    s_addr,
    server->h_length); //kopiranje adrese servera
serv_addr.sin_port = htons(portno); //navodjenje porta na
//serveru

if (connect(sockfd, (struct sockaddr *)&serv_addr, sizeof(
    serv_addr)) <
    0)
    error("ERROR connecting");
//konektovanje na zeljeni server preko sockfd socket
//deskriptora

bzero(buffer, 256);
//nuliranje buffera

strcpy(buffer, argv[3]);
//smestanje stringa u buffer
n = write(sockfd, buffer, strlen(buffer));
//slanje stringa serveru

bzero(buffer, 256);
n = read(sockfd, buffer, 255);
//prihvatanje odziva

printf("%s\n", buffer);
```

```
close(sockfd);  
//zatvaranje socketa i zavrsetak rada klijenta  
return 0;  
}
```

## 2 Serverski deo koda

server.c

```
/*
 * Vrlo prost ECHO server koji
 * ima fiksni bafer u koji skladišti
 * poruku od klijenta i vraća
 * je klijentu, a zatim zatvara tu
 * konekciju i postaje slobodan
 * za uspostavljanje novih konekcija
 *
 */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

void error(const char *msg)
{
    perror(msg);
    exit(1);
}

int main(int argc, char **argv)
{
    int sockfd, newsockfd, portno;
    socklen_t clilen;

    char buffer[256];
    struct sockaddr_in serv_addr, cli_addr;
    int n;

    if (argc < 2) {
        fprintf(stderr, "ERROR, no port provided\n");
        exit(1);
    }

    sockfd = socket(AF_INET, SOCK_STREAM, 0);

    bzero((char *)&serv_addr, sizeof(serv_addr));

    portno = atoi(argv[1]);
```

```
serv_addr.sin_family = AF_INET;
serv_addr.sin_addr.s_addr = INADDR_ANY;
serv_addr.sin_port = htons(portno);

if (bind(sockfd, (struct sockaddr *)&serv_addr, sizeof(
    serv_addr)) < 0)
    error("ERROR on binding");

listen(sockfd, 5);

clilen = sizeof(cli_addr);

while (1) {
    newsockfd =
        accept(sockfd, (struct sockaddr *)&cli_addr, &clilen);

    if (newsockfd < 0)
        error("ERROR on accept");

    bzero(buffer, 256);
    n = read(newsockfd, buffer, 255);

    if (n < 0)
        error("ERROR reading from socket");

    printf("Here is the message: %s\n", buffer);

    n = write(newsockfd, buffer, sizeof(buffer));

    if (n < 0)
        error("ERROR writing to socket");

    close(newsockfd);
}

close(sockfd);
return 0;
}
```

## Literatura

- [1] UNDERSTANDING BIG AND LITTLE ENDIAN BYTE ORDER, [\*http://betterexplained.com/articles/understanding-big-and-little-endian-byte-order/\*](http://betterexplained.com/articles/understanding-big-and-little-endian-byte-order/)
- [2] BEEJ'S GUIDE TO SOCKET PROGRAMMING IN C, [\*http://beej.us/guide/bgnet/output/html/singlepage/bgnet.html\*](http://beej.us/guide/bgnet/output/html/singlepage/bgnet.html)
- [3] ERRORS: ERRNO IN UNIX PROGRAMS, CHRIS HERBORTH, [\*http://www.ibm.com/developerworks/aix/library/au-errnovariable/\*](http://www.ibm.com/developerworks/aix/library/au-errnovariable/)