



**UNIVERSITETI POLITEKNIK I TIRANËS
FAKULTETI I TEKNOLOGJISË DHE INFORMACIONIT
DEPARTAMENTI I INXHINIERISË INFORMATIKE**

Punë Laboratori nr. 3

Tema: Përdorimi I socket ne C

Lënda: Programim në Sisteme Të Shpërndara

Grupi: III-B

Punoi:
Piro Gjikhima

Pranoi:
MSc.Megi Tartari

Ushtrimi 1

Implementoni ne gjuhën C një program që realizon komunikimin e klientëve dhe serverit me anë të socket ku klientit dërgon uri e një file dhe serveri i kthen përmbajtjen e file.

KODI

server.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

#define PORT 12345
#define BUFFER_SIZE 1024

void handle_client(int client_socket) {
    char filename[BUFFER_SIZE];
    char buffer[BUFFER_SIZE];

    int bytes_received = recv(client_socket, filename, BUFFER_SIZE, 0);
    if (bytes_received <= 0) {
        perror("Failed to receive filename");
        close(client_socket);
        return;
    }
    filename[bytes_received] = '\0';

    printf("Client requested file: %s\n", filename);

    FILE *file = fopen(filename, "r");
    if (file == NULL) {
        char *error_msg = "ERROR: File not found.";
        send(client_socket, error_msg, strlen(error_msg), 0);
    } else {
        while (fgets(buffer, BUFFER_SIZE, file) != NULL) {
            send(client_socket, buffer, strlen(buffer), 0);
        }
        fclose(file);
    }

    close(client_socket);
}

int main() {
    int server_socket, client_socket;
    struct sockaddr_in server_addr, client_addr;
    socklen_t client_len = sizeof(client_addr);
```

```

server_socket = socket(AF_INET, SOCK_STREAM, 0);
if (server_socket == -1) {
    perror("Socket creation failed");
    exit(EXIT_FAILURE);
}

server_addr.sin_family = AF_INET;
server_addr.sin_addr.s_addr = INADDR_ANY;
server_addr.sin_port = htons(PORT);

if (bind(server_socket, (struct sockaddr *)&server_addr, sizeof(server_addr)) ==
-1) {
    perror("Bind failed");
    close(server_socket);
    exit(EXIT_FAILURE);
}

if (listen(server_socket, 5) == -1) {
    perror("Listen failed");
    close(server_socket);
    exit(EXIT_FAILURE);
}

printf("Server listening on port %d...\n", PORT);

while (1) {
    client_socket = accept(server_socket, (struct sockaddr *)&client_addr,
&client_len);
    if (client_socket == -1) {
        perror("Client accept failed");
        continue;
    }
    printf("Client connected.\n");
    handle_client(client_socket);
}

close(server_socket);
return 0;
}

```

client.c

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

```

```

#define PORT 12345
#define BUFFER_SIZE 1024

int main() {
    int client_socket;
    struct sockaddr_in server_addr;
    char filename[BUFFER_SIZE];
    char buffer[BUFFER_SIZE];

    client_socket = socket(AF_INET, SOCK_STREAM, 0);
    if (client_socket == -1) {
        perror("Socket creation failed");
        exit(EXIT_FAILURE);
    }

    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(PORT);
    server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");

    if (connect(client_socket, (struct sockaddr *)&server_addr, sizeof(server_addr))
    == -1) {
        perror("Connection failed");
        close(client_socket);
        exit(EXIT_FAILURE);
    }

    printf("Enter filename: ");
    fgets(filename, BUFFER_SIZE, stdin);
    filename[strcspn(filename, "\n")] = 0;

    send(client_socket, filename, strlen(filename), 0);

    printf("\nReceived from server:\n");
    while (1) {
        int bytes_received = recv(client_socket, buffer, BUFFER_SIZE - 1, 0);
        if (bytes_received <= 0) break;
        buffer[bytes_received] = '\0';
        printf("%s", buffer);
    }

    printf("\n");
    close(client_socket);
    return 0;
}

```

Rezultate

Ne kete kod kemi implementimin e një sistem komunikimi mes një klienti dhe një serveri duke përdorur protokollin TCP/IP.

Kjo implementim mundëson që klienti të dërgojë një kërkesë për një skedar te serveri, ndërsa serveri e lexon këtë kërkesë, gjen skedarin e kërkuar dhe e dërgon përsëri te klienti.

Klienti (client.c)

- **Argumentet e linjës së komandës:** Klienti pranon një emër skedari nga përdoruesi përmes inputit.
- **Krijimi i socket-it:** Përdoret `socket(AF_INET, SOCK_STREAM, 0)` për të krijuar një socket TCP.
- **Lidhja me serverin:** `connect(client_socket, (struct sockaddr *)&server_addr, sizeof(server_addr))` përdoret për të krijuar lidhjen me serverin.
- **Dërgimi i emrit të skedarit:** Klienti merr emrin e skedarit nga përdoruesi dhe e dërgon tek serveri duke përdorur `send(client_socket, filename, strlen(filename), 0)`.
- **Marrja e përmbajtjes së skedarit:** Klienti lexon të dhënat e pranuar nga serveri dhe i shfaq në ekran.
- **Mbyllja e socket-it:** Pas marrjes së të dhënave, lidhja mbyllet me `close(client_socket)`.

Serveri (server.c)

- **Krijimi i socket-it:** `socket(AF_INET, SOCK_STREAM, 0)` krijon një socket TCP.
- **Lidhja me portën:** `bind(server_socket, (struct sockaddr *)&server_addr, sizeof(server_addr))` e lidh socket-in me një portë specifike (12345).
- **Vendosja në modalitet dëgjimi:** `listen(server_socket, 5)` lejon deri në 5 lidhje njëkohësisht.
- **Pranimi i klientëve:** `accept(server_socket, (struct sockaddr *)&client_addr, &client_len)` pranon një lidhje nga një klient.
- **Marrja e emrit të skedarit:** Serveri lexon emrin e skedarit nga klienti.
- **Leximi dhe dërgimi i skedarit:** Nëse skedari ekziston, ai lexohet dhe dërgohet bllok pas blloku te klienti me `send(client_socket, buffer, strlen(buffer), 0)`. Nëse nuk ekziston, serveri dërgon një mesazh gabimi.
- **Mbyllja e lidhjes:** Pas përfundimit të transmetimit, lidhja me klientin mbyllet me `close(client_socket)`.

Ky implementim siguron një metodë bazë për shkëmbimin e skedarëve në një rrjet duke përdorur socket-et në C.

```
(kali@kali) ~/Desktop/SistemeShpernadara/Lab/Lab3
$ ./client
Enter filename: hello.txt

Received from server:
Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Integer nec odio. Praesent libero.
Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Curabitur sodales ligula in libero. Sed dignissim lacinia nunc. Curabitur tortor. Pellentesque nibh. Aenean quam. In scelerisque sem at dolor. Maecenas mattis. Sed convallis tristique sem. Proin ut ligula vel nunc egestas porttitor. Morbi lectus risus, iaculis vel, suscipit quis, luctus non, massa.

(kali@kali) ~/Desktop/SistemeShpernadara/Lab/Lab3
$ ./server
Server listening on port 12345...
Client connected.
Client requested file: hello.txt
```

```
(kali㉿kali)-[~/Desktop/SistemeShpernadara/Lab/Lab3]  
$ ./client  
Enter filename: nofile.txt  
  
Received from server:  
ERROR: File not found.
```

```
(kali㉿kali)-[~/Desktop/SistemeShpernadara/Lab/Lab3]  
$ ./server  
Server listening on port 12345...  
Client connected.  
Client requested file: hello.txt  
Client connected.  
Client requested file: nofile.txt  
█
```

```
(kali㉿kali)-[~/Desktop/SistemeShpernadara/Lab/Lab3]  
$ ./client  
Connection failed: Connection refused
```