

**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 nëSisteme Të Shpërndara

**Grupi:** III-B

**Punoi:**  **Pranoi**:

Piro Gjikdhima MSc.Megi Tartari

**Ushtrimi 1**

Implementoni ne gjuhen C nje program qe realizon komunikimin e klientëve dhe serverit me anë të socket ku klienit dërgon uri e nje 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.

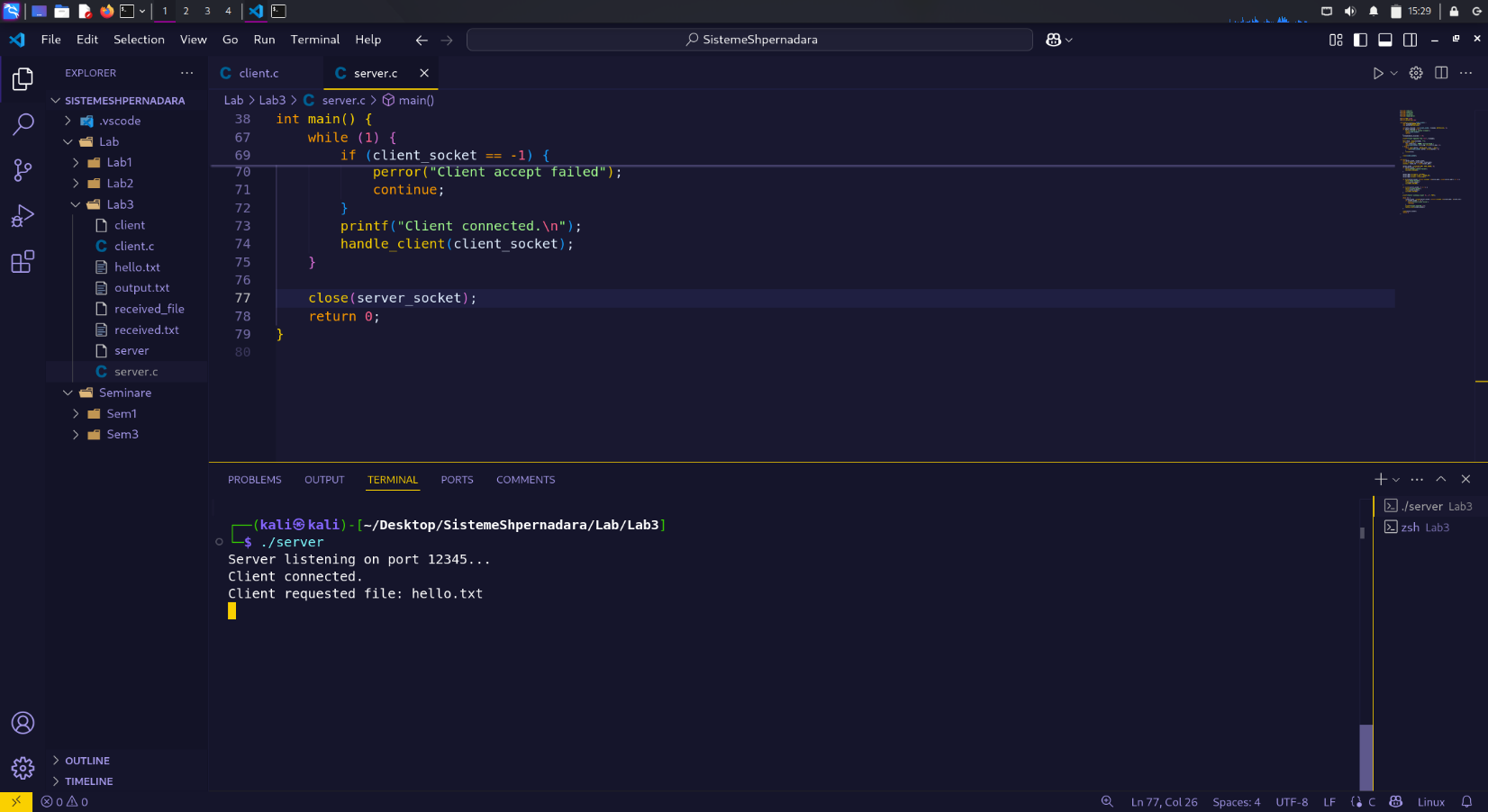
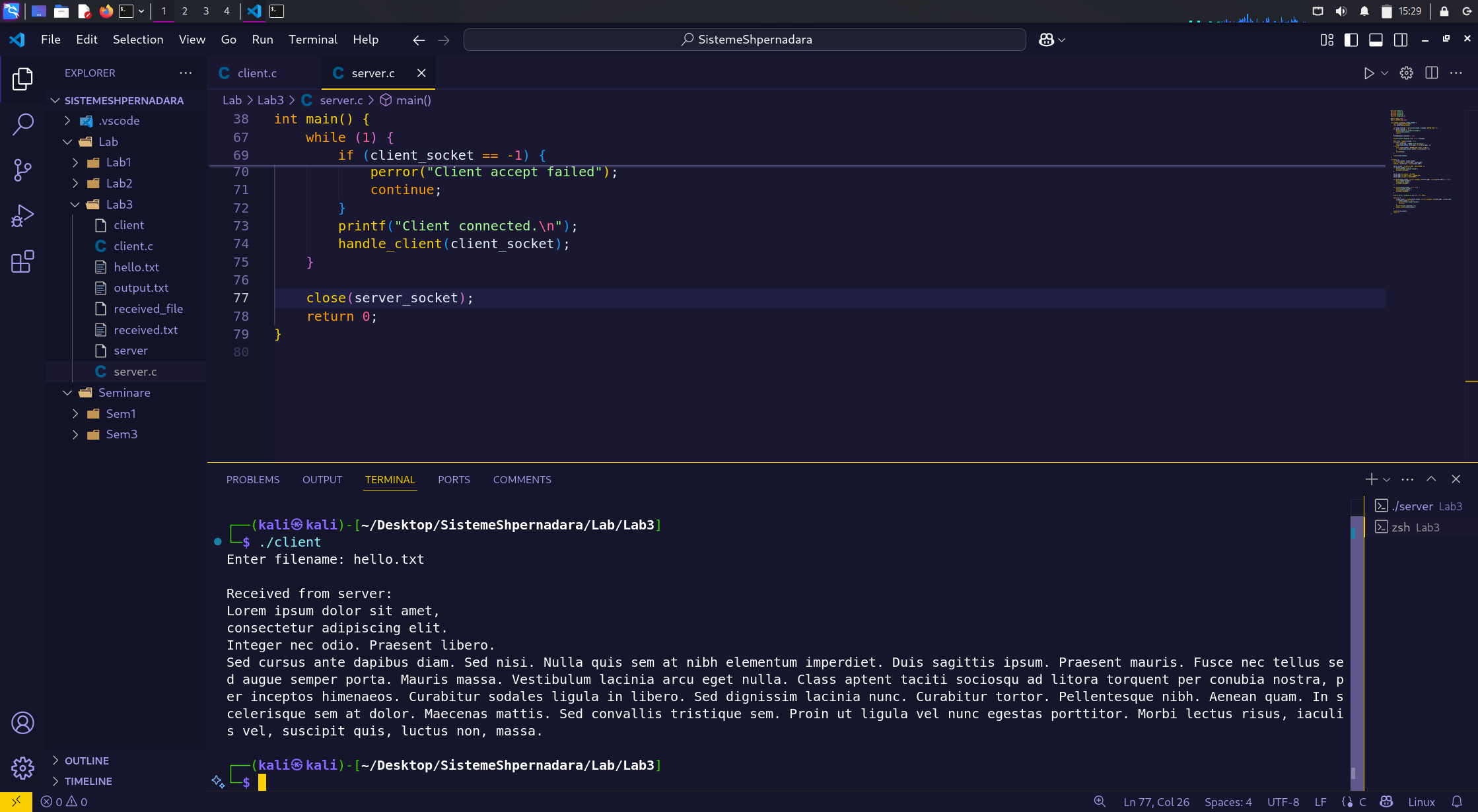
**Klienti (client.c)**

* **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 pranuara 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**.

A screenshot of a computer

AI-generated content may be incorrect.