

CNT Assignment 3A

Name: Kuber Kishore

Class: TY CSAI - A

Roll Number: 57

Batch: 3

Assignment: Write the client server programs using TCP Berkeley socket primitives for wired /wireless network for following

- a. to say Hello to Each other
- b. File transfer

Code

server.cpp

```
#include <iostream>
#include <cstring>
#include <unistd.h>
#include <arpa/inet.h>

int main() {
    int server_fd, new_socket;
    sockaddr_in address;
    int addrlen = sizeof(address);

    server_fd = socket(AF_INET, SOCK_STREAM, 0);
    if (server_fd == -1) {
        perror("Socket failed");
        return 1;
    }

    address.sin_family = AF_INET;
    address.sin_addr.s_addr = INADDR_ANY;
    address.sin_port = htons(8080);

    if (bind(server_fd, (sockaddr*)&address, sizeof(address)) < 0) {
        perror("Bind failed");
        return 1;
    }

    if (listen(server_fd, 1) < 0) {
        perror("Listen failed");
        return 1;
    }

    std::cout << "Server listening on port 8080..." << std::endl;
```

```

new_socket = accept(server_fd, (sockaddr*)&address, (socklen_t*)&addrlen);
if (new_socket < 0) {
    perror("Accept failed");
    return 1;
}

char buffer[1024] = {0};
read(new_socket, buffer, 1024);
std::cout << "Received: " << buffer << std::endl;

const char* reply = "Hello from server!";
send(new_socket, reply, strlen(reply), 0);

close(new_socket);
close(server_fd);

return 0;
}

```

client.cpp

```

#include <iostream>
#include <cstring>
#include <unistd.h>
#include <arpa/inet.h>

int main() {
    int sock = socket(AF_INET, SOCK_STREAM, 0);
    if (sock < 0) {
        perror("Socket creation error");
        return 1;
    }

    sockaddr_in serv_addr;
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(8080);

    if (inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr) <= 0) {
        perror("Invalid address");
        return 1;
    }

    if (connect(sock, (sockaddr*)&serv_addr, sizeof(serv_addr)) < 0) {
        perror("Connection failed");
        return 1;
    }

    const char* msg = "Hello from client!";
    send(sock, msg, strlen(msg), 0);
}

```

```
    char buffer[1024] = {0};
    read(sock, buffer, 1024);
    std::cout << "Server replied: " << buffer << std::endl;

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

file_server.cpp

```
#include <iostream>
#include <fstream>
#include <cstring>
#include <unistd.h>
#include <arpa/inet.h>

#define PORT 8080
#define BUFFER_SIZE 1024

int main() {
    int server_fd, new_socket;
    sockaddr_in address;
    int addrlen = sizeof(address);

    server_fd = socket(AF_INET, SOCK_STREAM, 0);
    if (server_fd == -1) {
        perror("Socket failed");
        return 1;
    }

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

    if (bind(server_fd, (sockaddr*)&address, sizeof(address)) < 0) {
        perror("Bind failed");
        return 1;
    }

    if (listen(server_fd, 1) < 0) {
        perror("Listen failed");
        return 1;
    }

    std::cout << "Server listening on port " << PORT << "..." << std::endl;

    new_socket = accept(server_fd, (sockaddr*)&address, (socklen_t*)&addrlen);
    if (new_socket < 0) {
```

```

        perror("Accept failed");
        return 1;
    }

    std::ifstream file("send.txt", std::ios::binary);
    if (!file) {
        perror("File open failed");
        return 1;
    }

    char buffer[BUFFER_SIZE];
    while (!file.eof()) {
        file.read(buffer, BUFFER_SIZE);
        std::streamsize bytesRead = file.gcount();

        if (bytesRead > 0) {
            // Display file contents to terminal
            std::cout.write(buffer, bytesRead);
            std::cout.flush();

            // Send data to client
            if (send(new_socket, buffer, bytesRead, 0) == -1) {
                perror("Send failed");
                break;
            }
        }
    }

    std::cout << "\n\nFile sent successfully." << std::endl;

    file.close();
    close(new_socket);
    close(server_fd);

    return 0;
}

```

file_client.cpp

```

#include <iostream>
#include <fstream>
#include <cstring>
#include <unistd.h>
#include <arpa/inet.h>

#define PORT 8080
#define BUFFER_SIZE 1024

int main() {

```

```
int sock = socket(AF_INET, SOCK_STREAM, 0);
if (sock < 0) {
    perror("Socket creation error");
    return 1;
}

sockaddr_in serv_addr;
serv_addr.sin_family = AF_INET;
serv_addr.sin_port = htons(PORT);

if (inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr) <= 0) {
    perror("Invalid address");
    return 1;
}

if (connect(sock, (sockaddr*)&serv_addr, sizeof(serv_addr)) < 0) {
    perror("Connection failed");
    return 1;
}

std::ofstream file("received.txt", std::ios::binary);
if (!file) {
    perror("File create failed");
    return 1;
}

char buffer[BUFFER_SIZE];
ssize_t bytesReceived;
while ((bytesReceived = read(sock, buffer, BUFFER_SIZE)) > 0) {
    // Save to file
    file.write(buffer, bytesReceived);

    // Display to terminal
    std::cout.write(buffer, bytesReceived);
    std::cout.flush();
}

std::cout << "\n\nFile received successfully." << std::endl;

file.close();
close(sock);

return 0;
}
```

Output

Server terminal

```
root@Kuber:/mnt/d/VIT TY/Sem V/CNT/Lab/Lab 4# g++ server.cpp -o server
root@Kuber:/mnt/d/VIT TY/Sem V/CNT/Lab/Lab 4# ./server
Server listening on port 8080...
Received: Hello from client!
```

Client terminal

```
root@Kuber:/mnt/d/VIT TY/Sem V/CNT/Lab/Lab 4# g++ client.cpp -o client
root@Kuber:/mnt/d/VIT TY/Sem V/CNT/Lab/Lab 4# ./client
Server replied: Hello from server!
```

File_Server terminal

```
root@Kuber:/mnt/d/VIT TY/Sem V/CNT/Lab/Lab 4# g++ file_server.cpp -o file_server
root@Kuber:/mnt/d/VIT TY/Sem V/CNT/Lab/Lab 4# ./file_server
Server listening on port 8080...
```

Hello from send.txt!

File sent successfully.

File_Client terminal

```
root@Kuber:/mnt/d/VIT TY/Sem V/CNT/Lab/Lab 4# g++ file_client.cpp -o file_client
root@Kuber:/mnt/d/VIT TY/Sem V/CNT/Lab/Lab 4# ./file_client
Hello from send.txt!
File received successfully.
```