Server

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <arpa/inet.h>

#define PORT 65432

#define BUFFER\_SIZE 1024

void toggle\_case(char \*text) {

for (int i = 0; text[i]; i++) {

if (text[i] >= 'a' && text[i] <= 'z') {

text[i] -= 32;

} else if (text[i] >= 'A' && text[i] <= 'Z') {

text[i] += 32;

}

}

}

int main() {

int server\_fd, new\_socket;

struct sockaddr\_in address;

int opt = 1;

int addrlen = sizeof(address);

char buffer[BUFFER\_SIZE] = {0};

if ((server\_fd = socket(AF\_INET, SOCK\_STREAM, 0)) == 0) {

perror("Socket failed");

exit(EXIT\_FAILURE);

}

if (setsockopt(server\_fd, SOL\_SOCKET, SO\_REUSEADDR | SO\_REUSEPORT, &opt, sizeof(opt))) {

perror("Setsockopt failed");

close(server\_fd);

exit(EXIT\_FAILURE);

}

address.sin\_family = AF\_INET;

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

address.sin\_port = htons(PORT);

if (bind(server\_fd, (struct sockaddr \*)&address, sizeof(address)) < 0) {

perror("Bind failed");

close(server\_fd);

exit(EXIT\_FAILURE);

}

if (listen(server\_fd, 3) < 0) {

perror("Listen failed");

close(server\_fd);

exit(EXIT\_FAILURE);

}

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

while (1) {

if ((new\_socket = accept(server\_fd, (struct sockaddr \*)&address, (socklen\_t\*)&addrlen)) < 0) {

perror("Accept failed");

close(server\_fd);

exit(EXIT\_FAILURE);

}

int valread = read(new\_socket, buffer, BUFFER\_SIZE);

if (valread > 0) {

printf("Received data: %s\n", buffer);

toggle\_case(buffer);

send(new\_socket, buffer, strlen(buffer), 0);

}

close(new\_socket);

}

close(server\_fd);

return 0;

}

Client

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <arpa/inet.h>

#define PORT 65432

#define BUFFER\_SIZE 1024

int main() {

int sock = 0;

struct sockaddr\_in serv\_addr;

char buffer[BUFFER\_SIZE] = {0};

char text[BUFFER\_SIZE];

printf("Enter text to toggle case: ");

fgets(text, BUFFER\_SIZE, stdin);

text[strcspn(text, "\n")] = 0; // Remove trailing newline

if ((sock = socket(AF\_INET, SOCK\_STREAM, 0)) < 0) {

perror("Socket creation error");

exit(EXIT\_FAILURE);

}

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/ Address not supported");

exit(EXIT\_FAILURE);

}

if (connect(sock, (struct sockaddr \*)&serv\_addr, sizeof(serv\_addr)) < 0) {

perror("Connection Failed");

exit(EXIT\_FAILURE);

}

send(sock, text, strlen(text), 0);

int valread = read(sock, buffer, BUFFER\_SIZE);

if (valread > 0) {

printf("Received toggled data: %s\n", buffer);

}

close(sock);

return 0;

}