**33. Implementing the applications using TCP file transfer in java/C.**

**Client**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <winsock2.h>**

**#define PORT 8080**

**#define BUFFER\_SIZE 1024**

**#define OUTPUT\_FILE "received.txt"**

**int main() {**

**WSADATA wsa;**

**SOCKET sock;**

**struct sockaddr\_in server\_addr;**

**FILE \*file;**

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

**// Initialize Winsock**

**WSAStartup(MAKEWORD(2, 2), &wsa);**

**// Create socket**

**sock = socket(AF\_INET, SOCK\_STREAM, 0);**

**if (sock == INVALID\_SOCKET) {**

**perror("Socket creation failed");**

**WSACleanup();**

**exit(EXIT\_FAILURE);**

**}**

**// Server details**

**server\_addr.sin\_family = AF\_INET;**

**server\_addr.sin\_port = htons(PORT);**

**server\_addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");**

**// Connect to server**

**if (connect(sock, (struct sockaddr \*)&server\_addr, sizeof(server\_addr)) == SOCKET\_ERROR) {**

**perror("Connection failed");**

**closesocket(sock);**

**WSACleanup();**

**exit(EXIT\_FAILURE);**

**}**

**printf("Connected to server. Receiving file...\n");**

**// Open file to save data**

**file = fopen(OUTPUT\_FILE, "w");**

**if (file == NULL) {**

**perror("Failed to create file");**

**closesocket(sock);**

**WSACleanup();**

**exit(EXIT\_FAILURE);**

**}**

**// Receive data**

**int bytes\_received;**

**while ((bytes\_received = recv(sock, buffer, BUFFER\_SIZE, 0)) > 0) {**

**fwrite(buffer, 1, bytes\_received, file);**

**}**

**printf("File received successfully!\n");**

**// Cleanup**

**fclose(file);**

**closesocket(sock);**

**WSACleanup();**

**return 0;**

**}**

**Server:**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <winsock2.h>**

**#include <ws2tcpip.h>**

**#pragma comment(lib, "ws2\_32.lib") // Link with Winsock library**

**#define PORT 8080**

**#define BUFFER\_SIZE 1024**

**int main() {**

**WSADATA wsa;**

**SOCKET server\_fd, new\_socket;**

**struct sockaddr\_in address;**

**int addrlen = sizeof(address);**

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

**// Initialize Winsock**

**if (WSAStartup(MAKEWORD(2, 2), &wsa) != 0) {**

**printf("WSAStartup failed. Error Code: %d\n", WSAGetLastError());**

**return 1;**

**}**

**// Create socket**

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

**printf("Socket creation failed. Error Code: %d\n", WSAGetLastError());**

**return 1;**

**}**

**// Configure server address**

**address.sin\_family = AF\_INET;**

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

**address.sin\_port = htons(PORT);**

**// Bind socket**

**if (bind(server\_fd, (struct sockaddr\*)&address, sizeof(address)) == SOCKET\_ERROR) {**

**printf("Bind failed. Error Code: %d\n", WSAGetLastError());**

**return 1;**

**}**

**// Listen for client**

**if (listen(server\_fd, 3) == SOCKET\_ERROR) {**

**printf("Listen failed. Error Code: %d\n", WSAGetLastError());**

**return 1;**

**}**

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

**// Accept client connection**

**if ((new\_socket = accept(server\_fd, (struct sockaddr\*)&address, &addrlen)) == INVALID\_SOCKET) {**

**printf("Accept failed. Error Code: %d\n", WSAGetLastError());**

**return 1;**

**}**

**printf("Client connected.\n");**

**while (1) {**

**memset(buffer, 0, BUFFER\_SIZE);**

**int valread = recv(new\_socket, buffer, BUFFER\_SIZE, 0);**

**if (valread <= 0) {**

**printf("Client disconnected.\n");**

**break;**

**}**

**printf("Client: %s", buffer);**

**printf("Server: ");**

**fgets(buffer, BUFFER\_SIZE, stdin);**

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

**}**

**closesocket(new\_socket);**

**closesocket(server\_fd);**

**WSACleanup();**

**return 0;**

**}**

**A screenshot of a computer screen

AI-generated content may be incorrect.**