#include <iostream>

#include <cstring>

#include <unistd.h>

#include <signal.h>

#include <arpa/inet.h>

#define PORT 8080

int server\_fd, new\_socket;

struct sockaddr\_in address;

socklen\_t addrlen = sizeof(address);

void signalHandler(int signum) {

std::cout << "Signal " << signum << " received, shutting down server...\n";

if (new\_socket > 0) {

close(new\_socket);

}

close(server\_fd);

exit(signum);

}

int main() {

signal(SIGINT, signalHandler);

signal(SIGTERM, signalHandler);

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

perror("Socket creation error");

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");

exit(EXIT\_FAILURE);

}

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

perror("Listen");

exit(EXIT\_FAILURE);

}

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

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

perror("Accept");

exit(EXIT\_FAILURE);

}

char buffer[1024] = {0};

while (true) {

ssize\_t valread = read(new\_socket, buffer, 1024);

if (valread > 0) {

std::cout << "Received: " << buffer << std::endl;

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

std::memset(buffer, 0, sizeof(buffer));

}

}

    return 0;

}