#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

#include <unistd.h>

#include <sys/socket.h>

#include <netinet/in.h>

#define PORT 8080

#define MAX\_BUFFER\_SIZE 1024

int main() {

int server\_fd, new\_socket, valread;

struct sockaddr\_in address;

int addrlen = sizeof(address);

char buffer[MAX\_BUFFER\_SIZE] = {0};

// Creating socket file descriptor

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

perror("Socket creation failed");

exit(EXIT\_FAILURE);

}

address.sin\_family = AF\_INET;

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

address.sin\_port = htons(PORT);

// Binding the socket to the specified port

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

perror("Bind failed");

exit(EXIT\_FAILURE);

}

// Start listening for incoming connections

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

perror("Listen failed");

exit(EXIT\_FAILURE);

}

// Accept incoming connection

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

perror("Accept failed");

exit(EXIT\_FAILURE);

}

// Read data from the client

valread = read(new\_socket, buffer, MAX\_BUFFER\_SIZE);

printf("Received request from client: %s\n", buffer);

// Get current date and time

time\_t current\_time;

time(&current\_time);

char \*date\_time = ctime(&current\_time);

// Send date and time to the client

send(new\_socket, date\_time, strlen(date\_time), 0);

printf("Sent date and time to the client\n");

// Close the socket

close(new\_socket);

close(server\_fd);

return 0;

}