// project1\_datacomp\_cont.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#define \_WINSOCK\_DEPRECATED\_NO\_WARNINGS

#include <stdio.h>

#include <winsock2.h>

#include <windows.h>

#include <iostream>

#include <cstdlib>

#include <stdlib.h>

#include <string.h>

#include <string>

#include <ws2tcpip.h>

#pragma comment (lib, "ws2\_32.lib") // load ws2\_32.dll

DWORD WINAPI Protocol(LPVOID IPParameter);

using namespace std;

int main()

{

cout << "Server Connected!" << endl;

// initialize winsock

WSADATA wsaData;

WSAStartup(MAKEWORD(2, 2), &wsaData);

// create a socket

SOCKET servSock = socket(PF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

// bind the socket to IP\_ADDRESS:PORT

sockaddr\_in sockAddr;

memset(&sockAddr, 0, sizeof(sockAddr)); // fill every byte with 0s

sockAddr.sin\_family = PF\_INET; // use IPv4 address

sockAddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1"); // specify the IP address

sockAddr.sin\_port = htons(1234); // choosing port

bind(servSock, (SOCKADDR\*)&sockAddr, sizeof(SOCKADDR));

// start listening

listen(servSock, 20);

// connect to the client

SOCKADDR\_IN clntAddr;

int nSize = sizeof(SOCKADDR);

int s = 0;

char host[NI\_MAXHOST];

char service[NI\_MAXSERV];

ZeroMemory(host, NI\_MAXHOST);

ZeroMemory(service, NI\_MAXSERV);

while (true)

{

SOCKET clntSock = accept(servSock, (SOCKADDR\*)&clntAddr, &nSize);

char szIpAddress[16]; // name of the online host

strcpy\_s(szIpAddress, sizeof(szIpAddress), inet\_ntoa(clntAddr.sin\_addr));

cout << "Local IP address: " << szIpAddress << "." << endl;

if (getnameinfo((SOCKADDR\*)&sockAddr, sizeof(sockAddr), host, NI\_MAXHOST, service, NI\_MAXSERV, NI\_NUMERICSERV) == 0)

{

cout << host << " connected via port " << service << " " << NI\_NUMERICSERV << endl;

}

s = (int)ntohs((sockAddr.sin\_port));

cout << "Client is using " << inet\_ntoa(sockAddr.sin\_addr) << s << " IPA" << endl;

HANDLE hThread\_1 = CreateThread(NULL, 0, Protocol, (LPVOID)&clntSock, 0, NULL);

CloseHandle(hThread\_1);

}

closesocket(servSock); // close the socket

// clean up winsock

WSACleanup();

cout << "Server ended.\n" << endl;

system("pause");

return 0;

}

DWORD WINAPI Protocol(LPVOID IPParameter)

{

SOCKET clntSock = \*(SOCKET\*)IPParameter;

// data returned by the server from the client

char szBuffer[MAXBYTE] = { 0 };

recv(clntSock, szBuffer, MAXBYTE, NULL);

cout << "Client's Message: " << szBuffer << endl; // output the message from the client

// send data to the client

char \*str = szBuffer;

send(clntSock, str, strlen(str) + sizeof(char), NULL);

//closesocket(clntsock) // close the socket

return 0;

}