// BlockingServer.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <stdio.h>

#include <stdlib.h>

#include <winsock2.h>

SOCKET clients[64];

SOCKADDR\_IN clientAddrs[64];

int length = 0;

BOOL bIsFinished;

DWORD WINAPI SenderThread(LPVOID lpParam)

{

char buf[1024];

int len;

while (1)

{

fflush(stdin);

fgets(buf,1024,stdin);

for (int i=0;i<64;i++)

{

if (clients[i] != -1)

send(clients[i],buf,strlen(buf),0);

}

}

return 0;

}

DWORD WINAPI ReceiverThread(LPVOID lpParam)

{

char buf[1024];

int len = 0;

int i = (int)lpParam;

while (!bIsFinished)

{

len = recv(clients[i],buf,1024,0);

if (len <= 0)

{

clients[i] = -1;

break;

}

buf[len] = '\0';

printf("From %s %d: %s",inet\_ntoa(clientAddrs[i].sin\_addr),ntohs(clientAddrs[i].sin\_port),buf);

if (buf[len-1] == '\n') buf[len-1] = '\0';

if (strcmp(buf,".") == 0)

{

clients[i] = -1;

break;

}

}

return 0;

}

int main(int argc, char\* argv[])

{

int i, len;

WSADATA wsaData;

WORD wVersion = MAKEWORD(2,2);

int ret = WSAStartup(wVersion,&wsaData);

SOCKET server;

SOCKADDR\_IN serverAddr;

server = socket(AF\_INET,SOCK\_STREAM,IPPROTO\_TCP);

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

serverAddr.sin\_port = htons(8888);

ret = bind(server,(sockaddr\*)&serverAddr,sizeof(serverAddr));

if (ret == SOCKET\_ERROR)

{

printf("Error %d while binding!\n",WSAGetLastError());

return 0;

}

ret = listen(server,10);

for (i=0; i<64;i++) clients[i] = -1;

while(1)

{

for (i=0;i<64;i++)

if (clients[i] == -1) break;

if (i == 64)

{

printf("Too many connections\n");

sleep(1000);

continue;

}

len = sizeof(SOCKADDR\_IN);

clients[i] = accept(server,(sockaddr\*)&clientAddrs[i],&len);

CreateThread(0,0,ReceiverThread,(LPVOID)i,0,0);

printf("There is a connection from %s %d\n",inet\_ntoa(clientAddrs[i].sin\_addr),ntohs(clientAddrs[i].sin\_port));

length++;

if (length == 1)

CreateThread(0,0,SenderThread,0,0,0);

}

return 0;

}

// BlockingClient.cpp :

//

#include "stdafx.h"

#include <stdio.h>

#include <winsock2.h>

SOCKET client;

BOOL bIsFinished = false;

DWORD WINAPI ReceiverThread(LPVOID lpParam)

{

char buf[1024];

int len = 0;

while (!bIsFinished)

{

len = recv(client,buf,1024,0);

if (len <= 0) break;

buf[len] = '\0';

printf("Server: %s",buf);

if (buf[len-1] == '\n') buf[len-1] = '\0';

if (strcmp(buf,".") == 0) break;

}

return 0;

}

int main(int argc, char\* argv[])

{

WSADATA wsaData;

WORD wVersion = MAKEWORD(2,2);

int ret = WSAStartup(wVersion,&wsaData);

char buf[1024];

int len = 0;

SOCKADDR\_IN serverAddr;

client = socket(AF\_INET,SOCK\_STREAM,IPPROTO\_TCP);

if (client == SOCKET\_ERROR)

{

printf("Error %d while creating socket!\n",WSAGetLastError());

return 0;

}

serverAddr.sin\_family = AF\_INET;

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

serverAddr.sin\_port = htons(8888);

ret = connect(client,(sockaddr\*)&serverAddr,sizeof(serverAddr));

if (ret == SOCKET\_ERROR)

{

}

CreateThread(0,0,ReceiverThread,0,0,0);

while (1)

{

fflush(stdin);

fgets(buf,1024,stdin);

len = send(client,buf,strlen(buf),0);

buf[len-1] = '\0';

if (strcmp(buf,".") == 0)

{

bIsFinished = true;

break;

}

}

return 0;

}

// SelectServer.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <stdio.h>

#include <winsock2.h>

int main(int argc, char\* argv[])

{

WSADATA wsaData;

WORD wVersion = MAKEWORD(2,2);

int ret = WSAStartup(wVersion,&wsaData);

SOCKET server,clients[64],tmp;

SOCKADDR\_IN serverAddr,clientAddrs[64],tmpAddr;

int clientAddrLen = sizeof(SOCKADDR\_IN);

int i, count = 0;

for (i=0;i<64;i++) clients[i]= 0;

server = socket(AF\_INET,SOCK\_STREAM,IPPROTO\_TCP);

if (server == SOCKET\_ERROR)

{

printf("Error: %d\n",WSAGetLastError());

return 0;

}

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

serverAddr.sin\_port = htons(8888);

ret = bind(server,(sockaddr\*)&serverAddr,sizeof(serverAddr));

if (ret == SOCKET\_ERROR)

{

printf("Error %d while binding!\n",WSAGetLastError());

closesocket(server);

return 0;

}

ret = listen(server,10);

fd\_set readfds;

char buf[1024];

int len;

while (1)

{

FD\_ZERO(&readfds);

FD\_SET(server,&readfds);

for (i=0;i<64;i++)

if (clients[i]>0) FD\_SET(clients[i],&readfds);

ret = select(0,&readfds,0,0,0);

if (ret == SOCKET\_ERROR)

{

printf("Error %d!\n", WSAGetLastError());

break;

}

if (FD\_ISSET(server,&readfds))

{

for (i=0;i<64;i++)

if (clients[i] == 0) break;

if (i==64)

{

printf("Too crowd!\n");

tmp = accept(server,(sockaddr\*)&tmpAddr,&clientAddrLen);

shutdown(tmp,SD\_BOTH);

closesocket(tmp);

continue;

}

clients[i] = accept(server,(sockaddr\*)&clientAddrs[i],&clientAddrLen);

printf("There is a connection from %s %d\n",inet\_ntoa(clientAddrs[i].sin\_addr),ntohs(clientAddrs[i].sin\_port));

count++;

}

for (i=0;i<64;i++)

if (clients[i] > 0)

if (FD\_ISSET(clients[i],&readfds))

{

len = recv(clients[i],buf,1024,0);

if (buf[len-1] == '\n') buf[len-1] = '\0';

else buf[len] = 0;

if (strcmp(buf,".") == 0)

{

printf("Client %s %d stop connecting\n",inet\_ntoa(clientAddrs[i].sin\_addr),ntohs(clientAddrs[i].sin\_port));

clients[i] = 0;

count--;

if (count == 0) exit(0);

continue;

}

else

printf("Client %s %d: %s\n",inet\_ntoa(clientAddrs[i].sin\_addr),ntohs(clientAddrs[i].sin\_port),buf);

}

}

return 0;

}

// TCPServer2.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <stdio.h>

#include <winsock2.h>

int main(int argc, char\* argv[])

{

WSADATA wsaData;

WORD wVersion = MAKEWORD(2,2);

int ret = 0;

SOCKET server, client;

SOCKADDR\_IN serverAddr, clientAddr;

int clientAddrLen = sizeof(clientAddr);

FILE \*fHello, \*fText;

char buf[1024];

int len;

ret = WSAStartup(wVersion,&wsaData);

if (argc != 4)

{

printf("Too much or too low parameters!\n");

exit(1);

}

server = socket(AF\_INET,SOCK\_STREAM,IPPROTO\_TCP);

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_port = htons(atoi(argv[1]));

serverAddr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

ret = bind(server,(sockaddr\*)&serverAddr,sizeof(serverAddr));

if (ret == SOCKET\_ERROR)

{

printf("Error: %d, when bind, close!",WSAGetLastError());

goto END;

}

ret = listen(server,10);

printf("Listening...\n");

if (ret == SOCKET\_ERROR)

{

printf("Error: %d, when listening, close!",WSAGetLastError());

goto END;

}

client = accept(server,(sockaddr\*)&clientAddr,&clientAddrLen);

if (client == SOCKET\_ERROR)

{

printf("Error while accepting client!\n");

closesocket(client);

goto END;

}

printf("There is a connection from %s %d!\n",inet\_ntoa(clientAddr.sin\_addr),ntohs(clientAddr.sin\_port));

fopen\_s(&fHello,argv[2],"r");

if (fHello == NULL)

{

printf("Error when opening hello file or file not found. Exit now!");

closesocket(client);

goto END;

}

fopen\_s(&fText,argv[3],"w");

if (fText == NULL)

{

printf("Error when creating text file or file not found. Exit now!\n");

closesocket(client);

fclose(fHello);

fclose(fText);

}

fgets(buf,1024,fHello);

len = send(client,buf,strlen(buf),0);

while (1)

{

len = recv(client,buf,1024,0);

if (len >= 0)

{

buf[len] = '\0';

printf("Client: %s",buf);

fprintf(fText,"%s",buf);

if (buf[len-1] == '\n') buf[len-1] = '\0';

else

{

printf("\n");

fprintf(fText,"\n");

}

if (strcmp(buf,".") == 0)

{

send(client,"bye",3,0);

closesocket(client);

fclose(fHello);

fclose(fText);

break;

}

}

}

END:

closesocket(server);

WSACleanup();

return 0;

}

// TCPClient2.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <winsock2.h>

#include <stdio.h>

int main(int argc, char\* argv[])

{

WSAData wsaData;

WORD wVersion = MAKEWORD(2,2);

int ret = 0;

SOCKET client;

SOCKADDR\_IN serverAddr;

char buf[1024];

int len;

ret = WSAStartup(wVersion,&wsaData);

if (ret == SOCKET\_ERROR)

{

printf("Error: %d, when WSAStartup, exit now!\n",WSAGetLastError());

getchar();

exit(1);

}

if (argc != 3)

{

printf("Too low or too much parameters! Please try again!\n");

exit(1);

}

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_addr.s\_addr = inet\_addr(argv[1]);

serverAddr.sin\_port = htons(atoi(argv[2]));

client = socket(AF\_INET,SOCK\_STREAM,IPPROTO\_TCP);

ret = connect(client,(sockaddr\*)&serverAddr, sizeof(serverAddr));

if (ret == SOCKET\_ERROR)

{

printf("Error: %d, when connect, exit now!\n",WSAGetLastError());

goto END;

}

while (1)

{

printf("Client: ");

fflush(stdin);

gets(buf);

len = send(client,buf,strlen(buf),0);

if (len == SOCKET\_ERROR)

{

printf("Got an error, break!\n");

break;

}

if(buf[len-1] == '\n') buf[len-1] = '\0';

if (strcmp(buf,".") == 0)

{

printf("End of transmission, exit now!\n");

break;

}

}

END:

closesocket(client);

WSACleanup();

return 0;

}

// UDPServer1.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <winsock2.h>

#include <stdio.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

WSADATA wsaData;

WORD wVersion = MAKEWORD(2,2);

SOCKET receiver;

SOCKADDR\_IN addr, source;

char buf[1024];

int len = sizeof(source);

int ret, datalen;

ret = WSAStartup(wVersion,&wsaData);

// Creating a socket

receiver = socket(AF\_INET,SOCK\_DGRAM,IPPROTO\_UDP);

if (receiver == SOCKET\_ERROR)

{

printf("Error: %d\n",WSAGetLastError());

getchar();

WSACleanup();

return 0;

}

// Initialise address and port

addr.sin\_family = AF\_INET;

addr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

addr.sin\_port = htons(8888);

// Bind socket

ret = bind(receiver,(sockaddr\*)&addr,sizeof(SOCKADDR\_IN));

if (ret == SOCKET\_ERROR)

{

printf("Error: %d\n",WSAGetLastError);

goto END;

}

// Waiting for datagram

while (1)

{

// Receiving data

datalen = recvfrom(receiver,buf,1024,0,(sockaddr\*)&source,&len);

// Checking length

if (datalen > 0)

{

buf[datalen] = '\0';

printf("Data from %s %d: %s",inet\_ntoa(source.sin\_addr),htons(source.sin\_port),buf);

if (buf[datalen-1] == '\n') buf[datalen-1] = '\0';

else printf("\n");

if (strcmp(buf,".") == 0) break;

}

}

END:

getchar();

closesocket(receiver);

WSACleanup();

return 0;

}

// UDPClient1.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <winsock2.h>

#include <stdio.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

WSADATA wsaData;

WORD wVersion = MAKEWORD(2,2);

int ret = 0;

SOCKET sender;

SOCKADDR\_IN receiverAddr;

char buf[1024];

int len;

ret = WSAStartup(wVersion,&wsaData);

// create a socket

sender = socket(AF\_INET,SOCK\_DGRAM,IPPROTO\_UDP);

if (sender == SOCKET\_ERROR)

{

printf("Error: %d\n",WSAGetLastError());

getchar();

WSACleanup();

return 0;

}

// fill the destination address

receiverAddr.sin\_family = AF\_INET;

receiverAddr.sin\_port = htons(8888);

receiverAddr.sin\_addr.s\_addr = inet\_addr("10.2.0.1");

while (1)

{

printf("Client: ");

fflush(stdin);

gets(buf);

len = sendto(sender,buf,strlen(buf),0,(SOCKADDR\*)&receiverAddr,sizeof(receiverAddr));

if (strcmp(buf,".") == 0) break;

}

closesocket(sender);

WSACleanup();

return 0;

}