

[1/2OT] a buggy TCP client program

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2007-12/msg02729.html

- *From:* RoS <Ros@xxxxxxxxxx>
 - *Date:* Tue, 18 Dec 2007 09:20:53 +0100
-

in windows xp
the problem is that if i have a tcp sever istalled in the port 1000 in
my computer, and "cli.exe" is the C program "client program" below and
type

```
cli 127.1.1.1 1000
```

connect of client fail with "WSAECONNREFUSED 10061"
don't understand why

there is someone expert in this there?
what is the answer?

then there is someone can point out some C program for the use api
"ReadFile" with an handle return by api "socket"?

thank you

```
#include <WINSOCK2.H>  
#include <stdio.h>  
#include <windows.h>  
#include <winbase.h>  
#include <wincon.h>
```

```
#define P printf  
#define R return  
#define F for  
#define G goto  
#define uns unsigned
```

```
int sockSta=0;
```

```
int openSock_m(void)  
{  
WORD wVersionRequested;  
WSADATA wsaData;
```

[1/2OT] a buggy TCP client program

```
int err;

if(sockSta==1) R 1;
wVersionRequested = MAKEWORD( 1, 1 );

err = WSASStartup( wVersionRequested, &wsaData );
if ( err != 0 )
{P("Tell the user that we couldn't find a usable"
"WinSock DLL.");
return 0;
}

/* Confirm that the WinSock DLL supports 1.1 */
/* Note that if the DLL supports versions greater */
/* than 1.1 in addition to 1.1, it will still return */
/* 1.1 in wVersion since that is the version we */
/* requested. */

if ( LOBYTE( wsaData.wVersion ) != 1 ||
HIBYTE( wsaData.wVersion ) != 1 )
{
P( "Tell the user that we couldn't find a usable"
"WinSock DLL., LB=%u HB=%u\n", (unsigned)LOBYTE(
wsaData.wVersion ) ,
(unsigned) HIBYTE( wsaData.wVersion ) );
WSACleanup( );
return 0;
}
sockSta=1;
return 1;
}

/* a[0] c=1, a[1] c=2, a[2] c=3*/
int main(int c, char** a)
{int i, clilen, m;
char *v;
uns porta;
SOCKET r, sc;
HANDLE hf;
char buf[4048];
uns long nu;

struct sockaddr_in cli_addr, srv_addr;
if(c!=3||a[1]==0||a[2]==0)
{10;;
P("Questo programma si connette a un server\n");
P("Uso: > thisprogram indirizzo NumeroPorta\n");
P("Con numero porta <=%u\n", 0xFFFF);
R 0;
}
}
```

[1/2OT] a buggy TCP client program

```
if( sscanf(a[1], "%u", &porta)!=1 ) G 10;
if(porta>0xFFFF) G 10;
if(openSock_m()==0)
{P("Errore di apertura socket\n"); R 0;}

if( (r=socket(AF_INET, SOCK_STREAM, 0))!=INVALID_SOCKET)
{F(i=0, v=(char*)&srv_addr; i<sizeof(struct sockaddr_in); ++i)
v[i]=0;
srv_addr.sin_family = AF_INET;
if( (nu=inet_addr(a[1]))==INADDR_NONE)
{P("Errore di inet_addr\n"); G 112;}
srv_addr.sin_addr.s_addr = nu;
srv_addr.sin_port=htons(porta);
m=0;
c0;;
if(connect(r, (sockaddr*) &srv_addr, sizeof(srv_addr)))
{P("Errore di connect\n");
P("Numero errore=%d [WSAECONNREFUSED 10061]\n",
WSAGetLastError());
if(m==0){++m; G c0;}
G 112;
}
if( fgets(buf, 2048, stdin) == 0)
{P("Errore di fgets\n"); G 11;}
nu=strlen(buf);
nu=sendto(sc, buf, nu+1, 0, 0, 0);
P("Bytes mandati %u bytes\n", nu);
P("Mandato: [%s]\n", buf);
}
else {P("Errore di apertura socket\n"); G 12;}
11;;
if( shutdown (r,SD_BOTH)) P("Errore di shutdown\n");
112;;
if( closesocket ( r ) )
P("Errore di closesocket\n");
12;;
if(WSACleanup( )==0) P("Tutto ok socket chiuso\n");
else P("Socket non chiuso\n");

R 0;
}
```