
Subject: Learn2IRC!

Posted by [reborn](#) on Wed, 17 Feb 2010 12:55:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

I've been trying to connect to an IRC server programitically instead of using an IRC client like mIRC.

I'm not a big fan of IRC, namely because I dont understand it very much, but I've made progress (albeit slowly)...

The following code is a thread that I was trying to make connect to an IRC server...

It gets to "//Gets to here fine with no problems....

" with no issues, however, as you can see, I am truely confused by the protocol and ping/pong especially.

I'm not sure when to expect the ping/pong, so I was just checking for it all the time, I will obviously make the continous ping/pong requests seperate, this thread was just to connect as a proof of concept to build on really...

I'm pretty sure after seeing the server ping me like this: "PING : randomjunk" that I need to respond with "PONG : randomjunk", which is why the pong looks so weird, I am tokenising the ping request to get that randomjunk part and slap it on the end of my pong response...

All the code after, like trying to join channels/set name etc etc make the irc server send me a message about not being registered, until the ping request times out. I am assuming that to be registered, I have to properly respond to ping, and this is where all my issues lay...

Here, is my code, if anyone can help, then I would appreciate it (or just has a working example of connecting, or some detailed information about how the steps the irc is looking for me to do to connect properly). Don't be too mean about the code, it really was just a test to connect...

```
DWORD WINAPI MainThread( LPVOID lpParam ){
```

```
char buf1[1200];  
char nick[] = "rebot";  
char text1[4096];  
int n;
```

```
WORD wsver=MAKEWORD(2, 0);
```

```
int nret=WSAStartup(wsver, &wsaData);  
if(nret != 0){  
printf("Startup failed, error code: %d\n",WSAGetLastError());  
WSACleanup();  
return false;
```

```

}

printf("Init success\n");
SOCKET kSock=socket(AF_INET, SOCK_STREAM, 0);

if(kSock == INVALID_SOCKET){
printf("Socket init failed");
return false;
}

printf("Socket initialized\n");

sockaddr_in sin;

sin.sin_port=htons(6668);

sin.sin_addr.s_addr=inet_addr("85.25.143.169");

sin.sin_family=AF_INET;

if(connect(kSock,(sockaddr*)&sin, sizeof(sin)) == SOCKET_ERROR){
printf("Startup failed, error code: %d\n",WSAGetLastError());
WSACleanup();
return false;
}
printf("Connection successful!\n\n");

//Gets to here fine with no problems....

n = recv(kSock, buf1, 1200, 0);
if ( n > 0 ) {
printf(">>Server: %s\n",buf1);
if (strstr(buf1,"PING")){
printf("I got a ping, cool!\n");
char* myStringPtr = buf1;
myStringPtr+=6;
char * pch;
pch = strtok (myStringPtr," =\n");
std::vector<std::string> str_Vector;
while (pch != NULL){
std::string strData = pch;
str_Vector.push_back(strData);
pch = strtok (NULL, " =\n");
}
sprintf(text1,"PONG :%s\r\n",str_Vector.at(0).c_str());
send(kSock, text1, sizeof(text1), 0);
}
}

```

```

    printf(">>Client: %s\n",text1);
}
}
else {
    printf(">>Server: No Data\n");
}

```

```

sprintf(text1, "NICK rebot\r\n");
send(kSock, text1, strlen(text1), 0);
printf(">>Client: %s\n",text1);

```

```

n = recv(kSock, buf1, 1200, 0);
if ( n > 0 ) {
    printf(">>Server: %s\n",buf1);
    if (strstr(buf1,"PING")){
        printf("I got a ping, cool!\n");
        char* myStringPtr = buf1;
        myStringPtr+=6;
        char * pch;
        pch = strtok (myStringPtr, " =\n");
        std::vector<std::string> str_Vector;
        while (pch != NULL){
            std::string strData = pch;
            str_Vector.push_back(strData);
            pch = strtok (NULL, " =\n");
        }
        sprintf(text1,"PONG :%s\r\n",str_Vector.at(0).c_str());
        send(kSock, text1, sizeof(text1), 0);
        printf(">>Client: %s\n",text1);
    }
}
else {
    printf(">>Server: No Data\n");
}

```

```

/*
// ping and pong continuous, will be moved to own thread
while (1) {
n = recv(kSock, buf1, 1200, 0);
if ( n > 0 ) {
    printf(">>Server: %s\n",buf1);
    if (strstr(buf1,"PING")){
        printf("I got a ping, cool!\n");
        char* myStringPtr = buf1;

```

```
myStringPtr+=6;
char * pch;
pch = strtok (myStringPtr," =\n");
std::vector<std::string> str_Vector;
while (pch != NULL){
std::string strData = pch;
str_Vector.push_back(strData);
pch = strtok (NULL, " =\n");
}
sprintf(text1,"PONG :%s\n",str_Vector.at(0).c_str());
send(kSock, text1, sizeof(text1), 0);
printf(">>Client: %s\n",text1);
}
}
else {
printf(">>Server: No Data\n");
}*/
return 1;
}
```