

Praktikum Netzwerke

Socketprogrammierung mit Visual Studio

Erzeugen eines Client-Sockets und Aufbau einer Verbindung

Als erste Aufgabe programmieren Sie einen Client, der sich mittels TCP-Socket mit einem Server verbindet. Das folgende Programm dient als Muster:

```
// aufgabe1.cpp : TCP Socket: Verbindung eines Clients mit einem Server
//

#include "stdafx.h"
#include <iostream>
#include "socket.h"

using namespace std;

int main(int argc, char* argv[])
{
    Socket socket;
    unsigned short port = 10;
    bool ret;

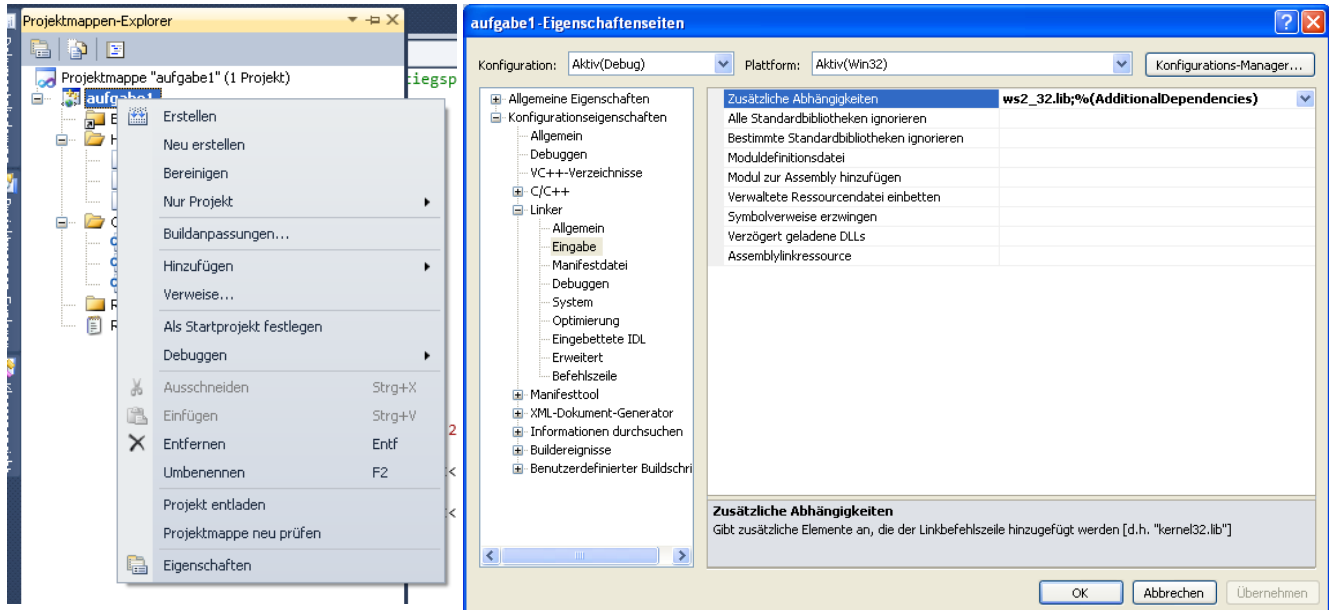
    socket.create();
    ret = socket.connect("127.0.0.1", port);
    if (ret == true)
        cout << "connect successful\n";
    else
        cout << "connect failed\n";

    return 0;
}
```

Das Programm enthält zwei „Bezeichner“, die Ihnen vermutlich neu sind:

- *cout*: Die Funktion *cout* ersetzt in der Sprache C++ die „eierlegende Wollmichsau“ *printf*. *cout* ist wesentlich flexibler als das alte *printf* der Sprache C.
Socket: Mit dieser Klasse werden die recht umständlichen Socketfunktionen gekapselt. Um diese Klasse verwenden zu können, müssen Sie zwei Dateien zu Ihrem Projekt hinzufügen. Beide Dateien stammen aus dem Buch C++ von J. Wolf und enthalten die Deklaration bzw. Definition einer einfachen Socket- Klasse:
 - `socket.h`
 - `socket.cpp`

Bevor Sie erfolgreich linken können, muss noch die Bibliothek **ws2_32.lib** unter *Projekteigenschaften->Linker->Eingabe->Zusätzliche Abhängigkeiten* hinzugefügt werden.



Aufgabe 1: Testen Sie das Beispielprogramm mit verschiedenen IP-Adressen und Port-Nummern (z.B: 127.0.0.1, Adresse des eigenen PCs, Port 770).

Aufgabe 2: Schreiben Sie ein Programm, das nacheinander alle Ports eines Servers anspricht und jeweils versucht, eine Verbindung herzustellen („Port Scan“). Nach jedem erfolgreichen Verbindungsversuch soll folgendes ausgegeben werden:

Verbindungsaufbau an Port xx erfolgreich

Achtung: verwenden Sie für Aufgabe ausschließlich die privaten 192.168.10.xxx Adressen, andernfalls könnte uns das LRZ dies sehr übel nehmen!!!

Die Datei socket.h enthält die Deklaration der Klasse Socket:

```
/* socket.h für MS-Windows */
#ifndef SOCKET_H_
#define SOCKET_H_
#include <string>
#include <winsock.h>
#include <io.h>
using namespace std;

// Max. Anzahl Verbindungen
const int MAXCONNECTIONS = 5;
// Max. Anzahl an Daten die auf einmal empfangen werden
const int MAXRECV = 1024;

// Die Klasse Socket
class Socket {
private:
    // Socketnummer (Socket-Deskriptor)
    int m_sock;
    // Struktur sockaddr_in
    sockaddr_in m_addr;

public:
    // Konstruktor
    Socket();
    // virtueller Destruktor
    virtual ~Socket();

    // Socket erstellen - TCP
    bool create();
    // Socket erstellen - UDP
    bool UDP_create();
    bool bind( const int port );
    bool listen() const;
    bool accept( Socket& ) const;
    bool connect( const string host, const int port );
    // Datenübertragung - TCP
    bool send( const string ) const;
    int recv( string& ) const;
    // Datenübertragung - UDP
    bool UDP_send( const string, const string,
                  const int port ) const;
    int UDP_recv( string& ) const;
    // Socket schließen
    bool close() const;
    // WSACleanup()
    void cleanup() const;
    bool is_valid() const { return m_sock != -1; }
};
#endif
```

Im Programmierpraktikum zu der Thematik Sockets werden wir uns ausschließlich mit diesen Funktionen befassen. Die relativ komplizierten Aufrufe, die in der Sprache C nötig wären, sind in den Funktionen dieser Klasse gekapselt.

Sie können dies leicht veranschaulichen, wenn Sie sich die Definitionen der einzelnen Klassenfunktionen ansehen.

Da im Praktikum ausschließlich mit der Konsole gearbeitet wird, ist es vorteilhaft, Parameter über die Kommandozeile eingeben zu können.

Das folgende Programm gibt die Parameter der Kommandozeile auf der Konsole aus:

```
#include <stdio.h>
#include <stdlib.h>

int main (int argc, char **argv)
{
    while (argc-- > 0)
        printf("%d: %s\n", argc, *argv++);
    return 0;
}
```