

## 第二章 Server 框架与消息处理

### 2.1 Server 框架

现在我们有网络模块，继续实现服务端 Server，上一章，我们的例子代码都是在 main.cpp 里，如果项目逻辑越来越大，那么 main.cpp 会很臃肿，这样堆砌代码是不好的，为了良好的设计架构，代码风格，我决定使用面向对象封装，把 Server 作为一个类，封装后，在 main 里只有几行代码即可，类似

```
int _tmain(int argc, _TCHAR* argv[])
{
    Server::GetInstance().Init();
    Server::GetInstance().Run();
}
```

Server.h

```
/*
 *
 * 主Server服务端类
 */
class Server
{
    PacketHandler m_PacketHandler; //消息处理

    SelectIOServer s; //网络引擎

    Server(void);

public:
    ~Server(void);

    static Server& GetInstance(){
        static Server s;
        return s;
    };

    bool Init();
}
```

```

void Run();

BOOL Send(    DWORD dwNetworkIndex ,    BYTE *pMsg ,    WORD
wSize ){return s.Send(dwNetworkIndex,pMsg,wSize);}
};

```

## 2.2 消息处理

服务端通讯肯定要定义一大堆消息和数据定义，如果数据消息只有几个，那么在回调接口里 if 判断几下处理就可以了，但是如果消息成百上千种，那么数据回调函数会很臃肿庞大，势必要用一个单独类来作为消息处理，我使用名称叫 PacketHandler 的类来专门处理数据消息，它提供一个 ParsePacket 的方法来负责解析处理数据包，这样在数据回调函数里只要一句代码即可，

PacketHandler.h

```

/*****
/*
    消息处理
*/
*****/

class PacketHandler
{
public:
    PacketHandler(void);
    ~PacketHandler(void);

    //处理 客户消息
    void ParsePacket(DWORD dwNetworkIndex , BYTE * pMsg, WORD w );

    //客户请求登陆
    void On_C2S_LOGIN(DWORD dwNetworkIndex, BYTE * pMsg, WORD w );
};

```

为了演示，我增加了第一个消息，登陆消息。我专门添加一个公用的 msgdef.h 头文件，放在项目文件夹的公共文件夹下。

msgdef.h 里面就只定义我们的消息 id 和消息结构体。

如下

```
#pragma once
```

```
#define C2S_LOGIN 0 //客户请求登陆  
#define S2C_LOGIN_Ret 1 //客户登陆结果
```

```
//客户请求登陆
```

```
struct MSG_Login  
{  
    int cmd; //命令 C2S_LOGIN  
    char name[20]; //用户名  
    char pwd[20]; //密码  
};
```

```
//客户登陆结果
```

```
struct MSG_Login_Ret  
{  
    int cmd; //命令 S2C_LOGIN_Ret  
    int errcode; //1=登陆成功 , 0=登陆失败  
    char pwd[20]; //密码  
};
```

消息 C2S 即表示 Client to Server，表示客户发送给服务端的。C2S S2C 这种定义前缀为了定义方便程序理解和开发。

ParsePacket 和 On\_C2S\_LOGIN 代码如下

```
//处理 客户消息
```

```
void PacketHandler::ParsePacket(DWORD idx , BYTE * pMsg, WORD wSize )  
{  
    int cmd;  
    memcpy(&cmd,pMsg,sizeof(cmd));  
  
    printf("PacketHandler::ParsePacket msg %d \n",cmd);  
  
    switch (cmd)  
    {  
        //登陆  
        case C2S_LOGIN : return On_C2S_LOGIN( idx,pMsg , wSize );  
  
        default:
```

```

        printf( "ParsePacket 未知消息id %d\n",cmd);
        break;
    }
}

//客户请求登陆
void PacketHandler::On_C2S_LOGIN(DWORD idx, BYTE * pMsg, WORD
wSize )
{
    printf( "On_C2S_LOGIN      %d\n",idx);

    MSG_Login s;
    memcpy((char*)&s, pMsg, wSize);

    //账号abc    密码123456
    if (strcmp(s.name,"abc")==0 && strcmp(s.pwd,"123456")==0 )
    {
        printf( "登陆成功      %s\n", s.name );
        MSG_Login_Ret ret;
        ret.cmd=S2C_LOGIN_Ret;
        ret.errcode=1;//登陆成功

        Server::GetInstance().Send(idx,(BYTE*)&ret,sizeof(MSG_Login_Ret));
        return ;
    }else{
        printf( "登陆失败      %s\n", s.name );
        MSG_Login_Ret ret;
        ret.cmd=S2C_LOGIN_Ret;
        ret.errcode=0; //登陆失败

        Server::GetInstance().Send(idx,(BYTE*)&ret,sizeof(MSG_Login_Ret));
        return ;
    }
}
}

```

客户端也添加一个 PacketHandler 类 处理 Server 传递过来的消息。  
 运行 Server 再运行 Client，运行结果如下，Client 登陆成功。  
 如此，我们消息处理的工作完成了，大家可以以此扩展，添加自己的消息定义以及消息处理。

```
"f:\code\doc\source\第二章\1.2\netserver\Debug\netserver.exe"
PacketHandler::ParsePacket  msg 0
On_C2S_LOGIN
登陆成功      abc
有[1]个客户连接
有[1]个客户连接
有[1]个客户连接

"f:\code\doc\source\第二章\1.2\netclient\Debug\netclient.exe"
CreateNetClient ok!  连接ok
PacketHandler::ParsePacket  msg 1
登陆成功!
```

2016.1.3            作者：司马威  
作者博客        <http://blog.csdn.net/smwhtjay>

编程交流 q 群: 316641007