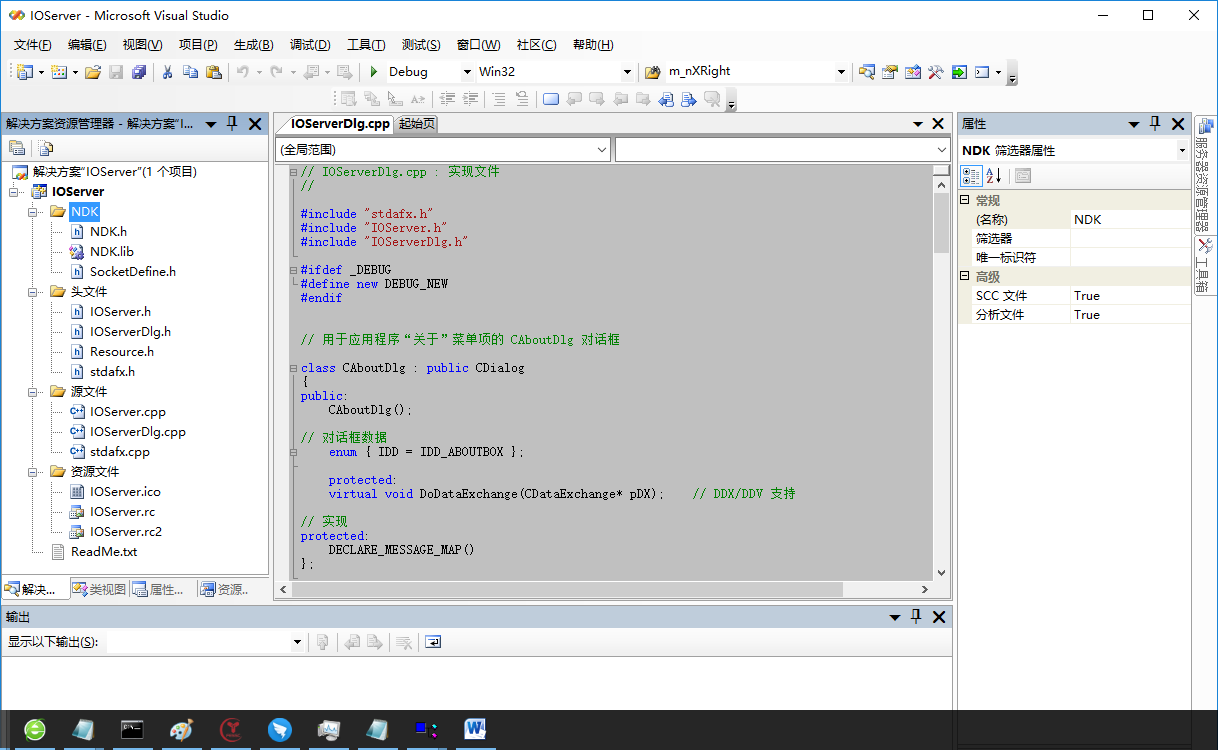
1. 引入库



1. 第三方动态库拷贝到运行目录
2. 引用头文件和socket类

#include "ndk\ndk.h"

CNDKSocket \*m\_pNDKSocket;

1. 创建对象，并且初始化通信模型

m\_pNDKSocket=GetNDKSocketPtr(SOCKET\_TYPE\_IO\_OVERLAP\_SERVER);

if(NULL == m\_pNDKSocket)

{

CString msg;

msg.Format("启动TCP Server错误");

AfxMessageBox(msg);

exit(0);

}

int nRet=0;

nRet=m\_pNDKSocket->Initial(RecvFunc, (long\*)this, m\_nPort, "0.0.0.0");//127.0.0.1

if(NDK\_SUCC != nRet)

{

CString msg;

msg.Format("启动TCP Server错误, Error=%d",nRet);

AfxMessageBox(msg);

exit(0);

}

1. 加入SOCKET头文件

#include <afxsock.h> // MFC 套接字扩展

1. 编写回调函数

int WINAPI RecvFunc(char\* pData, unsigned long ulLen, long\* lpParent, NDK\_SOCKET\* pClient)

{

CIOServerDlg \*pNewServerUIView = (CIOServerDlg\*)lpParent;

switch(pClient->ulOperator)

{

case 1: // 连接

{

if(NULL != pNewServerUIView && NULL!=pClient)

{

int nSendBufLen=1\*1024\*1024;

if (0!=setsockopt(pClient->sSocket, SOL\_SOCKET, SO\_SNDBUF, (char \*)&nSendBufLen ,sizeof(nSendBufLen) ) )

{

/\*CString strMsg;

strMsg.Format("setsockopt error=%d",GetLastError());

pNewServerUIView->m\_LogTrace.WriteLogError(strMsg);\*/

}

CString strMsg;

strMsg.Format("TCP Connect: sSocket=%d,ulIp=%ld,ulPort=%ld,ulLoginTime=%ld,ulOperator=%ld,ucIp=%s\n",

pClient->sSocket,

pClient->ulIp,

pClient->ulPort,//对方Port

pClient->ulLoginTime,

pClient->ulOperator,

pClient->ucIp//对方IP

);

//pNewServerUIView->m\_LogTrace.WriteLogError(strMsg);

setSocket.insert(pClient->sSocket);

}

}

break;

case 2: // 断开连接

{

if(NULL != pNewServerUIView && NULL!=pClient)

{

CString strMsg;

strMsg.Format("TCP Disconnect: sSocket=%d,ulIp=%ld,ulPort=%ld,ulLoginTime=%ld,ulOperator=%ld,ucIp=%s\n",

pClient->sSocket,

pClient->ulIp,

pClient->ulPort,//对方Port

pClient->ulLoginTime,

pClient->ulOperator,

pClient->ucIp//对方IP

);

//pNewServerUIView->m\_LogTrace.WriteLogError(strMsg);

setSocket.erase(pClient->sSocket);

}

}

break;

case 3: // 接收到数据

{

}

break;

}

return 0;

}

1. 编译通过。
2. 简单程序就做好了。