基于 VC++Socket 的多线程局域网聊天应用的设计与实现

周恕冉 白俊鸽
四川大学锦城学院计算机与软件学院 四川 成都 610000

【摘 要】在计算机日益发展的今天，由互联网而诞生的应用层出不穷，比如在局域网和互联网技术上的衍生。在更加高效、便捷等要求的催生下，许多的企业内部开始了局域网的通信应用的产生，特别有些企业为了自身安全的考虑，希望员工仅仅通过局域网联系。因此为了基于企业内部的要求，合理的方式是创建局域网的内部通信软件，也就是聊天系统。本文建立的局域网系统是在 socket 的基础上实现的。且本系统采用了 C/S 架构，引用了 VC++ 编程语言，利用面向对象的设计思想，从而实现局域网内的中通信目的。

【关键词】局域网；聊天软件

1 引言

本系统基于 C/S 模式，采用 VC++ 编程语言，利用面向对象的设计思想，在 Windows 平台上，将客户端和服务端融合在同一个程序之内，使用多线程实现不同的并行任务，从而实现局域网内的通信 [1]。本文主要通过系统开发的使用工具、功能需求分析、系统设计、系统实现、系统测试等方面的介绍，来介绍如何构建局域网聊天室。

2 开发技术的选择

现在计算机网络的通信采用的是 TCP/IP 协议，无论是我们在局域网还是公网上，都得服从于 TCP/IP 协议。在我们熟知的 TCP/IP 协议族中包括运输层、网络层、链路层，言简意赅地说也就是说我们要实现通信则必须用到 TCP/IP 协议来实现目的。而在 windows 开发环境下比较合适我们的局域网聊天室的便是采用 socket 进行编程了。在 windows 中提供的 socket 套接字程序对消息的收发，便可以通过 socket 来实现。也就是说我们要实现程序间的通信，就可以通过 socket 套接字来进行实现。

由于 socket 编程中，采用 C/S 模式，则服务器一方必然要去监听，等待客户机一方的连接。在等待的过程中，服务器端由于等待客户机的响应，从而未获得充分的运行条件，则会出现死锁，所以这里我们必须要采用多线程技术来解决这个死锁的问题，从而在设定端口进行监听时程序不会一直停在那里；除此之外，由于是多人聊天室，所以服务器不能够仅仅只和一个客户机交互，而且一旦服务器和客户机建立连接以后客户机和服务器便开始进行通信，之前的服务器程序中在监听端口过程中，已经监听到了客户机，则此时如果不再创建线程，则在通信过程中将无法继续对端口进行监听，则不能再有新的客户机加入，则程序便会变成 1 对 1 的模式，此时服务器和客户机都只能一对一的交互，所以，需要多线程技术来对目标端口持续进行监听，以便其它客户机连接时可以正常相应。此外服务器和每个客户机之间还需要创建一个线程，这样客户机才能分别和服务器进行交互，彼此间互不干扰。

3 主要功能的描述

本程序将客户机和服务器的功能放在了一个程序中，当用户启动程序以后可以自行选择是作为服务器还是客户机。服务器用户启动程序以后，会在网络设计区会进行选择，如果是服务器则需要填入想要监听的端口号，点击按钮开启服务器，则服务器启动成功，服务器会在自己的 IP 地址和服务器方选择的端口号等待 client 的连接；当客户机用户启动该程序选择网络设计区的的客户机区，则需要填入连接的目标服务器的地址和希望连接该地址服务器的哪个端口号，然后会对所填写地址的服务器和这个服务器的目标端口号的进行连接。当服务器和客户机连接成功以后，客户机和服务器之间就可以互相发送信息了。当客户机向服务器发送消息时，服务器方收到信息后除了会在服务器的消息栏显示以外，还会转发给当前所有和本服务器的这个端口已经建立了连接的客户机，并且会在这些客户机的消息栏中显示。若是服务器发送信息，则当前所有的 client 都会收到信息并显示在客户机的消息栏中，这样以来就可以形成一个多人的局域网聊天室了。

4 界面设计

程序有聊天区部分，聊天区由消息记录、发送框两部分构成，消息记录选择 MFC 中 Edit Control，并将其命名为 EditControl_MessageRecord, 另外发送框也采用 MFC 中的 Edit Control 空间，并将其命名为 EditControl_MessageSend。此外还要引入 MFC 中两个 Button 按钮的空间，分别命名为，Button_Send 和 Button_Over, 分别用作发送消息和退出。

程序另一部分则是属于网络设计区，网络设计区又分为上下两部分，上面一部分如果是 Client 那么填写这部分。这部分引入 MFC 中的一个 IP Address Control 控件和一个 Edit Control 控件以及由一个连接的 Button 控件构成构成，如果用户像作为客户端启动程序，那么在 client 部分需要填写目标服务器的 IP 地址和端口号，如果 IP 地址和端口号正确则进行连接。

网络设计区的下面一部分是如果用户想作为 Server 启动程序那么需要输入这部分然后启动程序，这部分由 MFC 中的一个 Edit Control 和一个 Button 组成，Edit Control 是为了 server 在启动程序时填写 server 想要监听的端口号，而 Button 则是为了作为 server 启动程序的按钮，当 server 完成填写需要监听的端口号，然后点击开始监听按钮，则用户完成了作为 server 的程序的启动。

5 功能实现

5.1 服务器和用户端的连接

Server 和 client 在 windows 编程环境下的连接，要用到 socket 编程，这里我将 sokcet 中 server 和 client 分别封装到两个函数之中，不管是 server 还是 client 依次调用封装函数，当用户登录程序在 server 和 client 对应需要填写的位置填写了相应内容以后，按下启动按钮 server 和 client 各自调用各自需要的函数，便可以运行各自对应的封装函数，如此便了实现 client 和 server 的连接。

Server 对应封装函数所包含的需要调用 socket 相应函数顺序：WSAStartup、socket、bind、listen、accept、recv、send、accept、closesocket、WSACleanup.

Client 对应封装函数所包含的需要调用 socket 相应函数的顺序：WSAStartup、socket、connect、recv、send 、closesocket、WSACleanup.

开发过程是在 MFC 中进行的，所以这里需要注意、在一开始 MFC 创建的初始化的函数中，需要加入 WSAStartup。进行了初始化以后，再经过之后的步骤这样才能够实现 client 和 server 的之间的连接。

5.2 开启多个客户端

由于当服务器开始以后，我们需要一直对某个选择的端口进行监听，这样才能达到连接一个 client 后下一个 client 仍能和服务器连接，需要在服务器创建的时候，就要创建一个线程，如： CreateThread(NULL, 0, ListenThread, this, 0, NULL);

而这里的 ListrnThread 就是需要将 server 和 client 的连接过程中 server 所对应的函数，这样当用户按下创建服务器按钮以后，server 将能一直对目标端口进行监听。

在监听的封装函数中，由于每一个 client 的出现都将和 server 建立连接，但每个 client 和 server 之间的交流是通过彼此连接实现的，所以需要在 ListenThread 中的 while 循环里在引入一个线程，而这个线程的作用是 server 对 client 的操作做出的回应，如 CreateThread(NULL, 0, ClientThreadProc, &(pChatRoom->m_ClientArray.GetAt(idx)), CREATE_SUSPENDED, NULL)； 这个线程中，server 将接收 client 的消息并通过封装的 show（）函数将内容输出到消息框中，show（）函数所包含的是 MFC 中将字符串在控件 Edit Control 的操作，除此之外这个函数还会对 client 的一些行为做出反应，如：client 退出的时候，在消息栏上会显示 client 退出了聊天室。

而 client 也要创建相应的线程，client 所需要创建的线程包含 client 的连接和 client 的行为，这些都放在了这一个线程之中。由于每一个 client 和 server 连接以后是除一方断开，否则要保持连接，这样才能达到通信的效果，否则 client 和 server 连接成功后 client 发送一句信息，server 收到以后，就没有后续过程了。这里每一个 client 点击连接服务器 button 以后，则会创建线程 CreateThread(NULL, 0, ConnectThread, this, 0, NULL); 而 client 的连接和操作封装在 ConnectThread（）中，这个函数中包含了和服务器建立连接的过程，以及建立连接以后一个 client 接收服务器信息并调用 show（）函数的过程。这样便实现了 client 和 server 的简单交互。

除了上述的用 socket 来实现 server 和 client 连接的建立，多线程来满足多个 client 和 server 的交互以外，还需要引入 accept。因为即使我们采用了多线程操作，其最直接实现的其实是为我们解决了服务器可以监听多个客户端、服务器可以操作多个客户端、服务器可以连接多个客户端等，但实际上操作过程中 server 仍然会一直去等待，服务器会一直在等有没有 client 来和服务器进行连接，这样来说就大大降低了程序效率，也降低了用户的体验感。因此我们可以采用 select 选择，使这个问题能够获得很好的解决，通俗点说当程序中采用了了 select 选择以后，服务器就不会出现一直等待的情况了而是在每一次在运行的时候仅仅去查看一下，去看看有没有 client 来向服务器发起连接请求，如果有则 select 就会出现响应这样 server 就知道有客户端来连接了，然

后就去执行连接的这个过程，如果这个部分仍然没有响应那么服务器就去做其它的事，当有响应出现以后我再来处理这个响应。而这个有没有响应程序中是如何进行相应的判断呢，Select 中包含了许多标志，当 select 出现响应的时候即当一切准备就绪的时候 selecr 会修改相应的标志，当程序发现标志出现了变化以后，则程序就可开始运行相应的部分。到此为止，我们就能够构造出了一个局域网的聊天室。此外在这个程序中我封装了一个 SOCKET_Select（ ）函数，这个函数的作用其实就是在每次进入线程中 while 循环之前便使用这个 select 进行一次 if 的判断，如果 select 选择出现了相应的响应则进入 while 循环执行接下来的步骤，如果没有响应则程序不进行下面的操作。

6 测试及结论

采用黑盒测试的方式，对本局域网聊天室进行了功能性测试。

启动程序以后，首先输入聊天区部分的监听端口号，点击“开启服务器按钮”；然后再启动一个程序，输入右上角部分服务器 IP 和端口号以后，点击“连接服务器”，此时消息栏显示“连接成功！”，该功能正常。再启动一个程序，输入 IP 和端口号以后，点击“连接服务器”，服务器消息栏显示，“客户端加入”，该功能正常。

启动的客户端再消息输入栏中输入消息，然后点击“发送”按钮，服务器收到客户端发送的消息，另一个客户端也收到了该消息，该功能正常。

服务器在消息栏中，输入消息，点击“发送”按钮，两个客户端的消息栏都受到了服务器发送的消息，该功能正常。

客户端点击“退出”按钮，服务器消息栏显示“客户端退出”，该功能正常。

服务器点击“退出”按钮，两个客户端消息栏显示“服务器退出”，该功能正常。

总结：该局域网聊天室基础功能均能够正常使用。

7 结束语

本文介绍了局域网聊天室的设计的基本构造和实现的原理，该局域网聊天室有功能简单，界面简洁的特点。但相较于目前大多数局域网聊天软件还有许多不足，由于开发成本低，代码实现简单，所以非常适用于初学者进行相应的练习。

【参考文献】

[1] 雷文礼，任新成，张栋，高瑛 . 基于 DirectShow 的网络视频点播系统的设计与实现 [J]. 现代电子技术 ,2015,38(07):31–33+38.