

简易代理服务器的设计

摘要

代理服务器是介于浏览器和Web服务器之间的一台服务器，它的功能是代理网络用户去取得网络信息。它的工作原理是，接收客户机的数据连接请求，并将请求传给Web服务器；再从Web服务器接收返回的数据，最后把数据传给客户端显示。本论文讲述的是关于一个简单的代理服务器的实现，在实现中采用客户机/服务器结构模型开发应用程序，基于一般的访问浏览速度慢、效率低、安全功能不强等原因，将以简单、合理、有效为原则；为了方便分析和设计，系统设计中设置不同的状态量用于标识客户机、代理服务器及服务器间的联结状态。在保证系统基本功能的同时，使用多线程机制使得代理系统有更好的延续性。此代理系统具有简单，易用，网络化等优点，能够完成一般的HTTP访问代理服务。

关键词: 客户机/服务器结构；多线程；套接字；代理服务器

The Design of Simple Proxy Server

Abstract

The proxy server is a server which situates between the browser and a Web server. It's principle of work is, it receives the connection request from the client, and hands down the requested to the Web server, then receives data which is returned from the Web server., finally hands it down the data to the client. In the realization, it adopts Client/Server structural model to develop application. Because of browse speed slow, low efficient and not safe enough, it takes simple, reasonable and effective as the principle. In the system design, it sets different states to identify client, proxy server and the connective state between servers. When the system's basic function are guaranteed, it use the multithreading to enable the proxy server to have a better continuous feature. This proxy system has the merit such as simple, easy to use and so on, and it can complete the proxy of general HTTP visit.

Key words: Client / Server architecture; Multithreading ; Socket; Proxy

Server

1	引言.....	1
1.1	课题背景	1
1.2	本课题研究的意义	1
1.3	本课题的研究方法	1
2	理论基础.....	2
2.1	代理服务器	2
2.2	目前的代理服务技术	3
2.3	SOCKET 面向连接的编程模型.....	4
2.4	WINSOCK 库	5
2.4.1	winsock 的寻址方式.....	5
2.4.2	字节顺序	6
2.5	系统基本套接字调用	6
2.5.1	创建和关闭套接字—socket()和closesocket()	6
2.5.2	指定本地地址—bind()	6
2.5.3	设置监听状态—listen()	6
2.5.4	建立套接字连接—connect()和accept()	6
2.5.5	收发数据—send()和recv()	7
3	设计方案.....	7
3.1	基本函数设计	7
3.2	多线程流程	8
4	服务器的实现.....	8
4.1	环境创建	8
4.2	功能实现	9
4.2.1	数据变量定义	9
4.2.2	启动代理服务器	10
4.2.3	请求处理过程	11
4.2.4	请求响应	13
4.2.5	关闭服务器	15
4.2.6	控制主函数	15

5 测试结果 15

结 论 17

参考文献 18

致 谢..... 错误！未定义书签。

声 明..... 错误！未定义书签。

商业源代码，全套计算机毕业设计免费下载<http://www.ibaidusoft.com>

更多全套设计联系QQ: 1042897696

最新设计大全 <http://www.bylw168.com/sf/20100915/3539.html>

1引言

1.1课题背景

普通的因特网访问是一种典型的客户机与服务器结构,而代理服务器将运行于客户机与服务器之间,它作为Internet/Intranet上常用的一种服务器,通常配置在Intranet连接Internet的出口处,主要实现代理传输服务。可以这样认为,代理是双向的。即对于内部网上的用户来说,代理服务器可看作是一个外部网的代理;对于外部网络来说,代理服务器可以看作一个要访问外部网的客户。正是由于代理服务器的这种控制方式,可以使用它提高客户访问外网的效率、节省网络带宽,增强网络安全性以及减少网络投资等。代理服务器从提出到现在,不断的经历着内容更进和技术的革新,各种代理服务器产品更是层出不穷。不难看出,代理服务器在我们信息时代的生活中扮演着越来越重要的作用。

1.2本课题研究的意义

随着Internet与Intranet的飞速发展,作为连接Internet与Intranet的桥梁,代理服务器在实际应用中发挥着极其重要的作用。它可用于多个目的,最基本的功能是连接,此外还包括安全性、缓存、内容过滤、访问控制管理等功能。在代理服务器的众多功能中,安全性是一个突出且敏感的功能。绝大多数企业、部门在使用代理服务器的时候,都会考虑这个问题,把它作为选购代理服务器产品的重要依据。目前市场上流行的代理服务器,像Microsoft Proxy Server、Netscape Proxy Server、Win Gate等国外的产品,功能和性能等方面都还不错,我们正好可以通过借鉴它们产品的优点,开发一个具有自主知识产权的代理服务器产品。而且从保证安全性的角度出发,我们也很有必要开发一个自己的代理服务器。除了上面的因素外,通过一个简单的代理程序的开发,我们能从实现过程中学习到网络通信和网络编程的基础知识,加深理解和掌握我们所使用的开发语言。另外,我们也能从中学习到正确的程序开发流程,积累程序开发经验,为以后更深课题的研究打好基础。

1.3本课题的研究方法

此代理服务器系统将是建立在Windows NT平台上的网络应用程序设计。由于需要服务器为其他许多称为客户的主机提供服务,而客户主机又可以随时打开和关闭,在选择网络应用程序体系结构时就采用支持这些特点的客户机/服务器结构。还将运用Windows下网络编程的标准接口 WinSock,因为它允许两个或多个应用程序在相同机器上,或者是通过网络相互交流,它是真正协议无关的接口。为了便于直接使用Windows提供的网络编程接口,我们使用Microsoft Visual C++ 6.0作为开发工具,利用 MFC类库中提供的两个 Socket类 CAsyncSocket和 Csocket。使用这两个Socket类,我们可以运用面向对象的方法来进行Socket

编程，而且它们还分别在较低和较高层次上封装了 Windows Sockets API，为程序员开发Socket程序提供了便利。

2理论基础

2.1代理服务器

代理服务器的英文全称是Proxy Server，其功能就是代理网络用户去取得网络信息。形象的说：它是网络信息的中转站。很多人不知不觉中就在用代理服务器共享上网，比如sygate, wingate, isa, ccproxy, NT系统自带的网络共享等，它们可以提供企业级的文件缓存，复制和地址过滤等服务。在一般情况下，我们使用网络浏览器直接去连接其他Internet站点取得网络信息时，须送出 Request信号来得到回答，然后对方再把信息以bit方式传送回来。代理服务器是介于浏览器和Web服务器之间的一台服务器，有了它之后，浏览器不是直接到Web服务器去取回网页而是向代理服务器发出请求，Request信号会先送到代理服务器，由代理服务器来取回浏览器所需要的信息并传送给你的浏览器。而且，大部分代理服务器都具有缓冲的功能，就好像一个大的Cache，它有很大的存储空间，它不断将新取得数据储存到它本机的存储器上，如果浏览器所请求的数据在它本机的存储器上已经存在而且是最新的，那么它就不重新从Web服务器取数据，而直接将存储器上的数据传送给用户的浏览器，这样就能显著提高浏览速度和效率。更重要的是：Proxy Server（代理服务器）是Internet 链路级网关所提供的一种重要的安全功能，它的工作主要在开放系统互联(OSI)模型的对话层。主要的功能有：

1、 连接Internet与Intranet充当防火墙：因为所有内部网的用户通过代理服务器访问外界时，只映射为一个IP地址，所以外界不能直接访问到内部网；同时可以设置IP地址过滤，限制内部网对外部的访问权限；另外，两个没有互联的内部网，也可以通过第三方的代理服务器进行互联来交换信息。

2、 节省IP开销：前面所讲，所有用户对外只占用一个IP，所以不必租用过多的IP地址，降低网络的维护成本。这样，局域局内没有与外网相连的众多机器就可以通过内网的一台代理服务器连接到外网，大大减少费用。当然也有它不利的一面，如许多网络黑客通过这种方法隐藏自己的真实IP地址，而逃过监视。

3、 提高访问速度：本身带宽较小，通过带宽较大的proxy与目标主机连接。而且通常代理服务器都设置一个较大的硬盘缓冲区，当有外界的信息通过时，同时也将其保存到缓冲区中，当其他用户再访问相同的信息时，则直接由缓冲区中取出信息，传给用户，从而达到提高访问速度的目的。

4、 防止攻击：隐藏自己的真实地址信息，还可隐藏自己的IP,防止被黑客攻击。通过分析指定IP地址，可以查询到网络用户的目前所在地。例如，大家在一些论坛上看到，论坛中明确标出了发帖用户目前所在地，这就是根据论

坛会员登录时的IP地址解析的。还有平日里我们最为常用的显IP版QQ，在“发送消息”窗口中，可以查看对方的IP及解析出的地理位置。而当我们使用相应协议的代理服务器后，就可以达到隐藏自己当前所在地地址的目的了。

5、突破IP访问限制：使用它可以访问一些有IP禁止访问的服务器，因为封锁只禁止了你和目标服务器的连接，但并没有禁止你与代理服务器的连接以及代理服务器与目标服务器的连接。

代理服务器有许多种，大体来说有HTTP, FTP, SOCKS代理三种，其中又分透明代理和不透明代理。其中透明代理一般是网关，是硬件，所以不讨论透明代理。当机器通过代理服务器上网时。通讯是分两次的，先是机器和代理服务器通讯，再是代理服务器和目的地址通讯。机器和代理服务器通讯时，目的IP是代理服务器的IP。代理服务器和目的地址通讯时，源IP是代理服务器的IP，当然外部的数据也是一样的。在内网中，出现的IP数据，全是内网和代理服务器的IP。因此，从IP包头是看不出任何与外面通讯的信息的。只有从数据中才能看到。

2.2目前的代理服务技术

代理服务技术是在一台PC机上安装一套代理软件，主要用于用户对Internet资源的访问。

ICS 即 Internet 连接共享(Internet Connection Sharing)的英文简称，是Windows系统针对家庭网络或小型的Intranet网络提供的一种Internet连接共享服务。它实际上相当于一种网络地址转换器，所谓网络地址转换器就是当数据包向前传递的过程中，可以转换数据包中的IP地址和TCP/UCP端口等地址信息。有了网络地址转换器，家庭网络或小型的办公网络中的电脑就可以使用私有地址，并且通过网络地址转换器将私有地址转换成ISP分配的单一的公用IP地址从而实现Internet的连接。ICS方式也称之为Internet转换连接。例如有软件：Wingate，Winproxy等。

NAT即网络地址转换(Network Address Translator)，从广义上讲，ICS也是使用了一种NAT技术，不过我们这里讨论的NAT是指将运行Windows 2000 Server的计算机作为IP路由器，通过它在局域网和Internet主机间转发数据包从而实现Internet的共享。NAT方式也称之为Internet的路由连接。网络地址转换NAT通过将专用内部地址转换为公共外部地址，对外隐藏了内部管理的IP地址。这样，通过在内部使用非注册的IP地址，并将它们转换为一小部分外部注册的IP地址，从而减少了IP地址注册的费用。同时，这也隐藏了内部网络结构，从而降低了内部网络受到攻击的风险。例如有软件：WinRoute，Sygate。

2.3 Socket面向连接的编程模型

代理服务器既然是双向的客户机/服务器模型，它工作时通常有一个公共的

规则需要遵守，通信双方需要共同遵守这个规则才能保证通信的有效进行，如 需要量传输大量语音信息时须采用一种无须建立连接的传输方式，在这种方式下，数据可以以较快的速度传输。而当需要确保数据准确无误地到达时，则应 采用面向连接的传输方式。代理服务器就要求采用这种方式，在这种模型下， 当服务器程序的套接字创建并初始化完毕时，它先进入休眠状态，直到有客户 机向该服务器程序提出连接请求。这时，服务器程序被“唤醒”并开始响应客 户机提出的连接请求。这时双方协商数据由谁来接收和由谁来发送，在数据传 输完毕时，双方再分别关闭连接，释放因创建套接字而占用的资源，面向连接 的编程模型示意图如下：

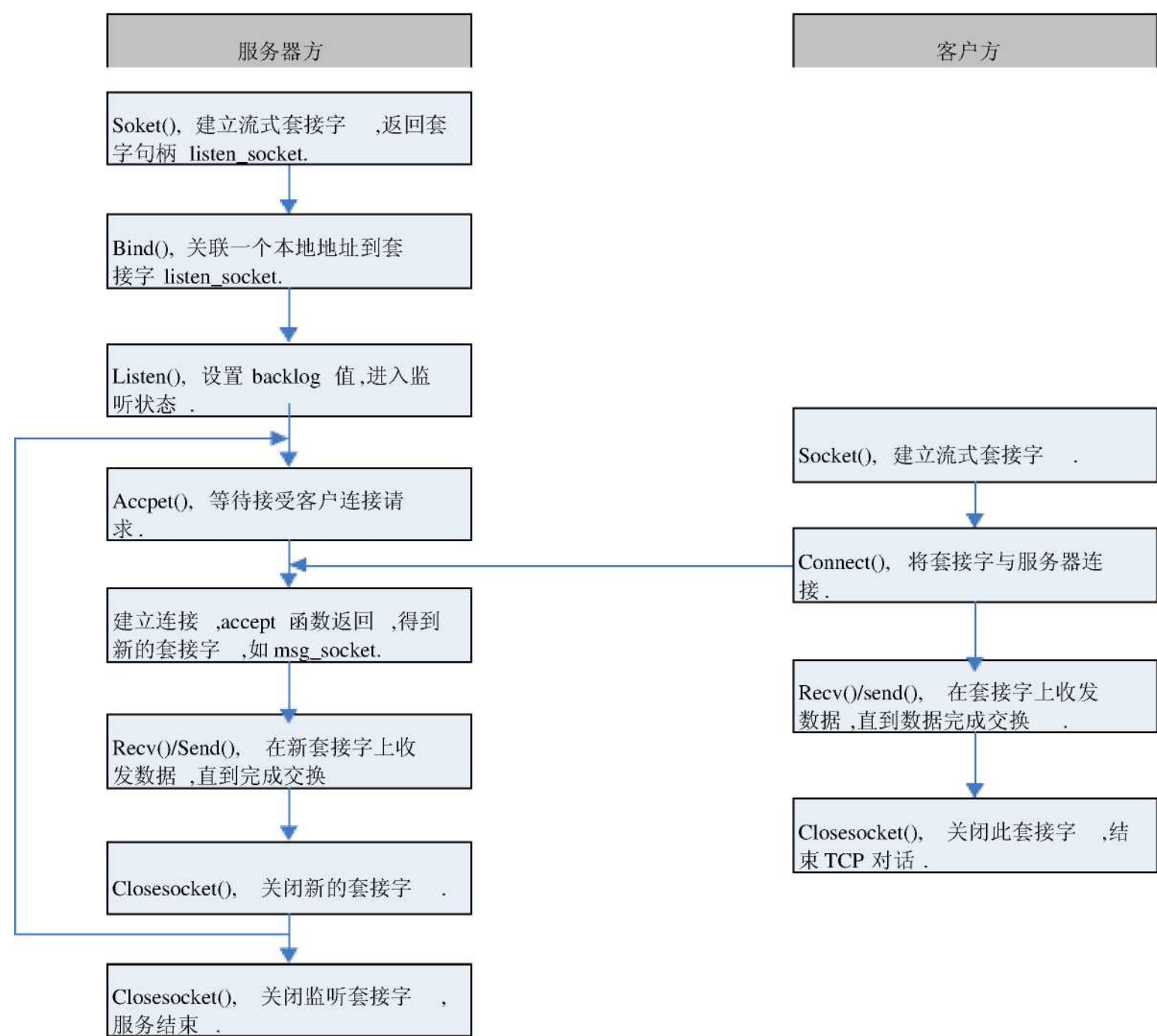


图1面向连接的编程模型

上面的示意图向我们显示了面向连接的服务器程序和客户程序的创建与结束过程。首先服务器端创建监听套接字，并为它关联一个本地地址(指定IP地址和端口号)，然后进入监听状态准备接收客户端的连接请求。为了接受客户端的连接请求，服务器必须调用`accept`函数。

客户端创建套接字后即可调用`connect`函数去试图连接服务器监听套接

字。当服务器端的accept函数返回后，connect函数也返回。此时客户端使用 socket函数创建的套接字，服务器端使用accept函数创建的套接字，双方就可以通信了。

当数据完成交换后，客户程序和服务器程序都会分别关闭创建的套接字句柄以完成双方的对话，至此程序服务结束。

2.4 Winsock 库

Winsock是Windows下网络编程的标准接口，它允许两个或多个应用程序在相同机器上，或者是通过网络相互交流。Winsock库的装入时是对Winsock DLL 版本的进行选择，每个Winsock应用程序都必须加载与之相应的版本的Winsock DLL。如果没有加载，将返回SOCKET_ERROR，加载的函数是WSAStartup，其定义如下：

```
int WSAStartup(WORD wVersionRequested, LPWSADATA lpWSADATA);
```

它的第一个参数就是要加载的库版本，第二个参数是用来返回DLL库的详细信息。

2.4.1 winsock的寻址方式

因为Winsock要兼容多个协议，所以必须使用通用的寻址方式。TCP/IP使用地址和端口号来指定一个地址，但是其它协议也许采用不同的形式。如果 Winsock强迫使用特定的寻址方式，添加其他协议就不大可能了。在Winsock 中，应用程序员通过SOCKADDR_IN结构来指定IP地址和端口号。定义如下：

```
Struct sockaddr_in(  
    short sin_family;  
    u_short sin_port;  
    struct in_addr sin_addr;  
    char sin_zero[8];  
);
```

应用程序可以使用inet_addr函数将一个小数点分隔的十进制IP地址字符串转化成由32位二进制数表示的IP地址。inet_ntoa函数将一个网络字节顺序的32位IP地址转化字符串。注意：inet_addr返回的32位二进制数是用网络顺序存储的。

2.4.2 字节顺序

字节顺序是长度跨越多个字节的数据被存储的顺序，分为小尾顺序和大尾顺序，TCP/IP统一规定使用大尾顺序方式传输数据，即网络字节顺序，它的字节顺序是最重要的字节首先存储。Winsock提供了一些函数来处理本地机器的字节顺序和网络字节顺序的转换，htons将主机字节顺序转化到TCP/IP网络字节顺序，ntohs将网络字节顺序转化到主机字节顺序，此外还有htonl和ntohl。这API数据都是平台无关的。使用它们可以保证程序正确运行在所有机器上。

2.5 系统基本套接字调用

2.5.1 创建和关闭套接字—`socket()`和`closesocket()`

应用程序在使用套接字前，首先必须拥有一个套接字，系统调用`socket()`向应用程序提供创建套接字的手段，如果成功将返回套接字句柄，如果创建失败，返回`INVALID_SOCKET`(即-1)。当不使用此套接字时，应该用`closesocket`函数关闭套接字，如果没有错误发生，函数返回0，否则返回`SOCKET_ERROR`。

2.5.2 指定本地地址—`bind()`

当一个套接字用`socket()`创建后，存在一个名字空间(地址族)，但它没有被命名。`bind()`将套接字地址(包括本地主机地址和本地端口地址)与所创建的套接字号联系起来，即将名字赋予套接字，以便能够有效地标识套接字。它用在没有建立连接的套接字上，如果没有错误发生，`bind()`返回0。否则返回`SOCKET_ERROR`。当然，客户端程序也可以在不显示绑定地址和端口号的情况下发送数据或者连接。这时系统会默认地为套接字绑定一个本地端口值。注意：地址在建立套接字通信过程中起着重要作用，程序使用中通常靠填充`sockaddr_in`结构来绑定套接字到本地地址。绑定到套接字上的本地名称包括主机地址、协议号和端口号3部分。

2.5.3 设置监听状态—`listen()`

`Listen`函数设置套接字进入监听状态。为了接受连接，首先使用`socket`函数创建套接字，然后使用`bind`函数将它绑定到本地地址，再用`listen`函数为到达的连接指定`backlog`，最后使用`accept`函数接受请求的连接。`Listen`函数仅应用在支持连接的套接字上，如`SOCK_STREAM`类型。函数执行成功后，套接字将进入被动模式，到来的连接会通知要排队并等候接受处理。在同一时间处理多个连接请求的服务器通常使用`listen`函数，如果一个连接请求到达并且排队也满，客户端将接收错误。

2.5.4 建立套接字连接—`connect()`和`accept()`

这两个系统调用用于完成一个完整相关的建立，用于客户机与网络中的服务器建立连接时，用`connect()`这个调用连接将请求发到侦听方。之后，服务端就会调用`accept`，而在调用`accept()`的参数前应该先调用过`listen()`，`Accept`函数定义如下：

```
SOCKET accept(SOCKET soc, struct sockaddr * addr, int * addrlen);
```

2.5.5 收发数据—`send0`和`recv()`

对流套接字来说，一般使用`send`和`recv`函数来收发数据。

`Send`函数在一个连接的套接字上发送缓冲区内的数据，返回发送数据的实

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/148131102133006050>