

基于 Telnet 协议的 Delphi 应用程序与 Linux 主机的通信

范喜全 赵洪利 (国防科工委指挥技术学院 101407)

摘要:本文叙述了在 Windows 95 操作系统与 Linux 操作系统互连的环境下,利用 Delphi 网络编程实现基于 Telnet 协议的 Delphi 应用程序与 Linux 主机之间的通信。重点讨论了 C 语言与 Delphi 编程语言在网络编程应用中的异同。

关键词:Telnet 协议 子协商 虚拟网络终端(NVT) Linux 服务器 Object Pascal 语言

一、Telnet 协议简介

Telnet 协议的目的是提供一种通用的、八比特位定向通信手段,其基本功能是通过网络提供远程登录或虚拟终端能力。Telnet 协议可以用来进行终端——终端的通信(连接)或进程——进程的通信(分布计算)。

Telnet 连接实际上是一种 TCP/IP 连接(Transmission Control Protocol),它通过在 TCP 传输数据流中嵌入 Telnet 控制信息来实现连接的,Telnet 通过 TCP 端口 23 提供服务。它的建立与实现基于三种思想:“网络虚拟终端(Network Visual Terminal)”的概念;子协商操作原理;对称终端和过程。

二、环境配置

Delphi 提供了一种全新的 Windows 应用程序开发方法,它速度快,具有强大且容易使用的可视化开发环境,具有领先的数据库技术和强大的网络编程能力。

使用 Delphi 开发工具(Object Pascal 语言)利用 Tel-

net 协议实现与 Linux 主机的通信,首先要建立环境,即 Linux 与 Windows 95 联网。Windows 95 环境下的 Delphi 应用程序将作为客户端,而 Linux 操作系统将作为服务器,它们之间的关系如下:

1. 安装 Linux



按操作指南安装 Linux 操作系统。Linux 是 Unix 操作系统的一种变种,它采用了微内核技术。

2. 配置网络

(1)启动 Linux,并键入“root”以管理员身份登录。运行 netconfig,根据屏幕提示配置网络的服务器名、域名、主机的 IP 地址、子网屏蔽及命名服务器 IP 地址。

(2)修改 rc.modules 文件。此文件是 Linux 系统的资源配置模块文件,由于 Linux 并没有自动配置网络的功能,所以必须进行手工配置。

进入 etc \ rc.d 目录,用 vi 命令编辑 rc.modules 文件。在 Network 族中,将适用自己的网卡处(如 ne IO=Oxnn)前面的屏蔽位“#”去掉,然后修改 IO 端口(如 IO=0x240)。键入 qw 命令保存修改,退出文件。此修改后的文件将被加入到 Linux 系统文件中,并在 Linux 启动时自动生效。重新启动 Linux,将会看到以下的网络配置提示信息:

```
Network NE2000 IO=0x240,11
```

(3)测试服务。将已物理连接的待联机的具有 Windows 95 操作系统的客户机添加 TCP/IP 协议,并配置好 IP 地址及标识。重新启动以使配置生效,在 Linux 操作系统下键入“ping + IP 地址”来测试网络的连接,将会看到请求连接建立的时间等信息,说明 Linux 与 95 已经通过 TCP/IP 协议连接成功,如要测试 Telnet 服务,只须在 Linux 服务器上执行 Telnet 命令即可。

三、网络编程(Delphi 应用程序与 Linux 主机通信)

目前,软件领域流行的网络编程的资料及代码范例,基本上都是基于 C 语言开发的。C 语言是一种半高级语言,对于指针、内存操作有极大的灵活性,而 Pascal 是一种高级语言,其内部具有许多保护机制,避免了许多影响操作系统正常运行的误操作。

由 C 向 Object Pascal(Delphi 编程语言)的代码转化并不是非常困难的事情,困难在于 C 语言结构与 Object Pascal 不同,类型、变量的表示也相差很大,这就需要熟悉 C 语言结构与 ObjectPascal 的联系与区别,特别是必要的数据结构的表示方法。下表列出作者在实现 Telnet 协议时,通过实际摸索得出的 C 语言与 Object Pascal 在数据结构与语法表示方面在网络编程中的差异:

	C 语言	OBJECT PASCAL 语言
套接字 s	SOCKET	TSOCKET
socket 初始化数据结构	WSAData	PWSAData
socket 地址	Socket-addr	Psockaddr
主机入口结构	hostent	Phostent
缓冲区指针定义	far char *	^string[n]
缓冲区指针使用	& 指针变量	指针变量(使用前需调用 NEW()进行初始化)

Delphi 网络编程支持 Winsock,其中的许多函数均可以直接调用,但要注意函数调用过程中的 Pascal 语言的参数传递风格,尤其要小心指针参数的正确使用(如表中所示)。

1. 建立连接

由于 Object Pascal 支持 Winsock,在程序定义阶段的 uses 部分加入 Winsock 的说明,这样就可以使用 Winsock 中的函数。

```
uses ..., Winsock ...
```

直接启动 WSAStartup()。将套接字 s 设为 Tsocket 类型,调用 socket() 创建套接字。初始化 socket 地址(Psockaddr 类型),设置 Linux 服务器的 IP 地址。

```
... } 变量定义 |
s : TSOCKET;
wca : PWSAData;
... } 主要 SOCKET 函数 |
Status := WSAStartup(MAKEWORD(1, 1), wca);
} 启动 |
...
s := socket(AF_INET, sock-type, 0); } 创建套接字 |
...
```

调用 connect 函数建立连接,其主机结构参数 dst-addr 由初始化的 Psockaddr 结构决定。此中要注意网络字节顺序及数目,如运用不当,很可能无法建立连接。

```
if (connect(s, dst-addr, sizeof(dst-addr)) = SOCKET-
ET-ERROR) then
begin
...
end
```

2. 协商通信

一旦连接成功,主机将发回请求信息,从而进入 Telnet 协议的子协商过程。子协商过程是客户与服务器之间以协议约定好的命令结构形式的数据流进行相互通信的过程。这种子协商过程的命令结构如图所示:

解释为命令(IAC)	命令代码	选项
------------	------	----

这些命令符都是八比特的 ASCII 字符,命令之间没有任何间隔符,这些 ASCII 字符可以用 ord() 和 chr() 函

数来进行字符与数字之间的转换。

建立接收缓冲区,调用 `recv()` 或 `recvfrom()` 函数等待服务器发送的信息。

```
lpBufferp : ^string[255];| 变量定义 |
...
len := recv(s, lpBufferp, length, 0);
...
```

通常,服务器将发送如下子协商命令:

```
IAC DO TERMINAL-TYPE (终端类型)
IAC DO TERMINAL-SPEED (终端速率)
IAC DO X-DISPLAY-LOCATION (显示方式)
IAC DO ENVIRONMENT (环境)
```

建立发送缓冲区,根据上述信息进行回应,调用 `chr()` 函数将命令代码转换为 ASCII 字符填入发送缓冲区,注意在“选项”中出现的非命令字符的字符串(如表示终端类型的字符串)要以 NVT ASCII String 形式出现。调用 `send()` 或 `sendto()` 函数发送至服务器,并等待服务器的回应信息。

```
...
length := send(s, lpBufferp, len, 0);
...
```

此后按照 Telnet 协议进一步进行子协商。例如,可以这样进行回应:

```
IAC DO TERMINAL-TYPE (终端类型)
IAC DONT TERMINAL-SPEED (终端速率)
IAC DONT X-DISPLAY-LOCATION (显示方式)
IAC DONT ENVIRONMENT (环境)
```

一旦子协商命令执行完毕,服务器将发送服务器本身的版本信息和登录信息,如上述 linux 与 95 连网系统将会得到:

```
IAC WILL ECHO 换行 回车 Linux 2.0.0 (Server.
user)(ttyo) 换行 回车
server login
```

此时将提示用户输入账号进行远程登录,如登录失败(如服务器没有要登录的帐户),则服务器返回空信息,

并继续等待用户进行登录,直到 60 秒后,服务器将发送:

```
Login times out after 60 seconds 换行 回车
```

并断开与客户已建立的虚电路连接,此时用户的发送与接收命令都将产生错误,这种错误可以通过调用 `WSAGetLastError()` 函数得到错误代码。

登录成功后,服务器将返回服务器版本信息、最近一次登录信息及用户有关信息等。此时,客户(Windows 95)与服务器(Linux)就可以使用服务器的 Telnet 服务进行数据传输,客户已经成为服务器的一个远程虚拟终端,客户机通过数据传输方式将等待执行的指令发送到服务器一端。服务器得到数据,将分离出的命令指令在后台执行,并将执行结果以数据的形式返回给客户机,如发送“ls”命令,将列出当前服务器的目录信息。

四、结束语

本文所讨论的基于 Telnet 协议的 Delphi 与 Linux 主机通信的这一软件,在实际的 LAN 中运行良好。开发这一软件的目的是探讨在 Delphi 下运用 Object Pascal 语言进行网络编程的基本方法,以及 Telnet 协议的实现流程。本文的着眼点在于为网络编程人员提供利用 Object Pascal 语言进行网络编程的一点经验,尤其是由 C 代码向 Object Pascal 转化中的一些注意事项,以免进入误区。

本软件只是有关 Delphi 的网络编程部分,实际上可以在 Delphi 开发环境下定制(或开发)一可视化控件,将本软件的核心代码包装成此控件的一些方法,就可以为 Delphi 增加一个可视化网络控件,用户只需将此控件拖入预定义的窗体中,就可以通过调用该控件的网络方法来实现可视化的网络编程。这部分工作作者正在进行。

参考文献

- [1] Unix 环境下的网络程序设计,孙义等编写,1994,学苑出版社
- [2] Windows Sockets 网络程序设计指南,蒋东兴编写,1995,清华大学出版社
- [3] TCP/IP 及相关协议, Uyles Black 著,1995,学苑出版社

(来稿时间:1997年9月)