基于 C/S 模式的成像测井仪调试台架通信接口设计

杨喜峰[1,2],鞠晓东[1],吴文河[1]

1. 中国石油大学(北京)油气资源与探测国家重点实验室,北京 102249
2. 中国石油大学(华东)理学院,山东青岛 266580

摘要 根据微电阻率扫描测井仪调试台架功能需求,给出一种基于 C/S 模式的调试台架通信接口程序设计方案。通过对 B/S、C/S 通信模式、流式和数据报式套接字的特点对比,确定基于流式套接字 C/S 模式的接口通信方式。为解决简单流式套接字的阻塞问题,前端机引入多线程技术实现通信过程独立。为提高接口的扩展性和可移植性,上位机程序采用 Winsock 控件实现通信功能。接口的测试证明其具有很好的稳定性和时效性,并且该接口设计结构简单、性能优秀、可移植性好,可以直接应用于同类调试台架和类似结构的测试系统。

关键词 成像测井仪;调试台架;C/S 模式;Winsock 控件

中图分类号 TP331.1 **文献标识码** A **doi** 10.3981/j.issn.1000-7857.2012.36.005

Design of the Communication Software for Logging Tool Test-bench Based on C/S Model

YANG Xifeng[1,2], JU Xiaodong[1], WU Wenhe[1]

1. *State Key Laboratory of Petroleum Resources and Prospecting, China University of Petroleum, Beijing 102249, China*
2. *College of Science, China University of Petroleum, Qingdao 266580, Shandong Province, China*

Abstract According to the demand of the micro-resistivity scanning tool test-bench, this paper proposes a design scheme of the test-bench communication interface program based on the C/S mode. From a comparison of the characteristics between the B/S and C/S communication modes and the flow and data report type sockets, the paper chooses the flow socket C/S mode as the mode of the communication interface. In order to solve the blocking problem of simple flow sockets, in the front-end computer, the multithreading technology is introduced to realize the independent communication process. In order to improve the interface scalability and portability, in the PC program, the Winsock control is used to realize the communication function. It is shown by performance tests that the communication interface has a good stability and speed. The interface enjoys features of simple structure, excellent performance and good portability, and can be directly applied in the test-bench and the testing system of similar structures.

Keywords micro-resistivity scanning logging instrument; test-bench; C/S model; Winsock control

0 引言

成像测井技术是为解决非均质性储层问题而形成的新的测井技术。通过沿井眼纵向、径向采集大量地层信息,并利用相应图像处理技术获得井壁或井眼周围的图像。这些精确、直观的图像在复杂地质和油气藏勘探中发挥着重要作用[1]。成像测井仪器结构复杂,高新技术集中,在其研制和维修过程中需要专用测试装备支持,而目前国内缺少成像测井仪器专用测试系统。为满足测井仪器研发和使用过程中对该类测试装备的迫切需求,鞠晓东等[2]与 CPL 合作开展了相应成像测井仪调试台架研制工作,其中包括微电阻扫描成像测井仪、阵列感应成像测井仪和测井仪数据遥传 3 个调试台架。上述调试台架采用由前端机和上位机构成的主从式结构,前端机为系统测试核心,上位机为系统处理核心和测控界面,两者采用以太网作为通信介质。前端机与上位机间高

收稿日期:2012-10-08;修回日期:2012-10-25
基金项目:国家高技术研究发展计划(863 计划)项目(2006AA060702)
作者简介:杨喜峰,博士研究生,研究方向为测试仪器研制开发,电子信箱:yangxf@upc.edu.cn

速、稳定的通信接口是整个调试台架的基础。本文根据调试台架结构,提出基于C/S(客户机/服务器)模式的调试台架通信接口方案,并结合微电阻率扫描成像测井仪调试台架,给出了通信接口的具体实现方法。

1　微电阻率扫描测井仪调试台架结构

微电阻率扫描测井仪调试台架是一套针对MCI井下仪器而设计的专用测试系统,结构如图1所示。前端机由主控核心板和相应的检测接口板组成。主控接口板由基于ARM的嵌入式系统组成,是整个前端机的测控核心,一方面实现对前端机的管理和测试过程的控制,另一方面完成与上位的通信。上位机采用基于x86的通用计算机,对前端机的测试数据进行后续处理,同时为调试台架提供测控界面[2]。

通信接口软件包含前端机通信程序和上位通信程序两部分,承担着上位机与前端机间测控命令下传和测试数据上传的任务。为保证调试台架高效、稳定运行,要求通信接口软件具有较高的安全性和数据吞吐,而且通信不影响前端机的其他工作过程。

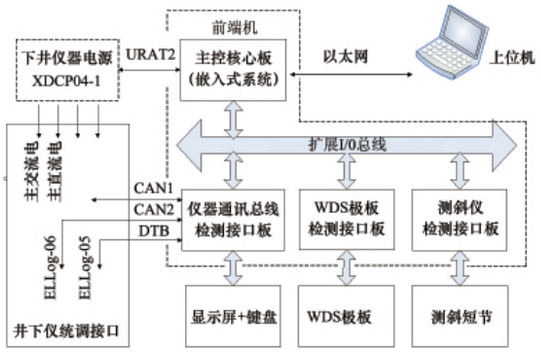

图1　微电阻率扫描测井仪调试台架结构

Fig. 1　Structure diagram of test-bench of micro-resistivity scanning logging tool

2　调试台架通信接口设计

2.1　通信模式选择

根据微电阻率扫描测井仪调试台架主从式结构特点,通信接口有C/S和B/S(浏览器/服务器)两种通信模式可供选择。其中,C/S模式是基于分布式处理的两层体系结构,其优点是交互性强、数据传输安全、响应速度快和数据处理能力强。B/S模式是随网络技术的发展,在C/S模式基础上形成的3层式C/S结构,该模式在服务器端提供Web Server,客户端引入Web浏览器,具体功能通过动态网页实现。用户能够通过浏览器访问到测试网页,实现测试控制。对于B/S通信模式,客户端只需具有Web浏览器,就可以实现客户端软件"零"维护的目的[3],但该模式的大部分工作由服务器端承担,容易造成服务器负担过重。根据调试台架的结构特点和通信性能要求,本调试台架采用C/S通信模式。前端机作为服务器端,上位机作为客户端。

2.2　Socket通信机制

Socket(套接字)是为了方便开发网络应用程序而提出的网络通信接口规范,最初由美国加州Berkley大学提出并应用于Unix系统,现在已经推广到Windows、Linux等多款操作系统。基于TCP/IP协议的网络有3种类型的套接字:流式套接字(SOCK_STREAM)、数据报套接字(SOCK_DGRAM)和原始套接字(SOCK_RAW)。流式套接字提供了一个面向连接、可靠的数据传输服务;数据报套接字提供了一个无连接,不保证无错的数据服务;原始式套接字提供了对低层协议的直接访问,常用于检验新的协议实现或访问现有服务中配置的新设备。由于调试台架对数据通信的可靠性要求很高,因此,采用流式套接字。

图2所示为流式套接字工作流程:服务器首先建立一个套接字,然后将该套接字与服务器端网络地址绑定,接着对套接字端口进行监听。客户端建立一个套接字,然后向服务器发出连接请求,服务器响应该请求,并建立套接字连接。连接建立后,客户端和服务器间就能够发送和接收数据,完成数据通信。最后,服务器和客户端关闭各自的套接字。

从图2所示的流式套接字工作流程可以发现,服务器程序在如"等待连接请求"的时候,将处于阻塞状态,即程序一直处于等待连接请求状态,不进行其他处理,这种阻塞现象将影响服务器的工作过程[4]。

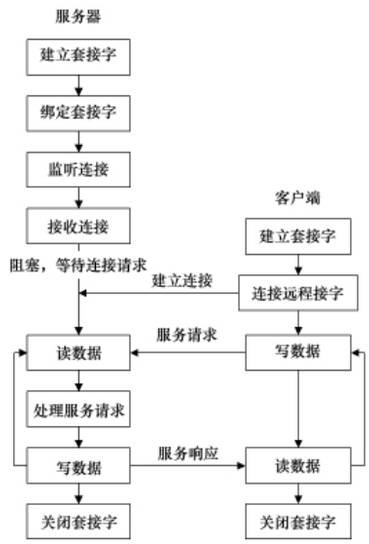

图2　流式套接字应用流程

Fig. 2　The flow chart of flow socket application

3　前端机网络通信程序实现

前端机核心板采用基于S3C2440的嵌入式系统,嵌入式操作系统选择Linux 2.6。微电阻率扫描测井仪调试台架将前端机作为通信服务器,为避免上述阻塞状况发生,在前端机程序中引入多线程方法,通过独立的线程来完成通信。前端机通信程序结构如图3所示,主程序首先对系统进行初始化,接着创建Socket服务器,等待用户连接请求。当有用户连接

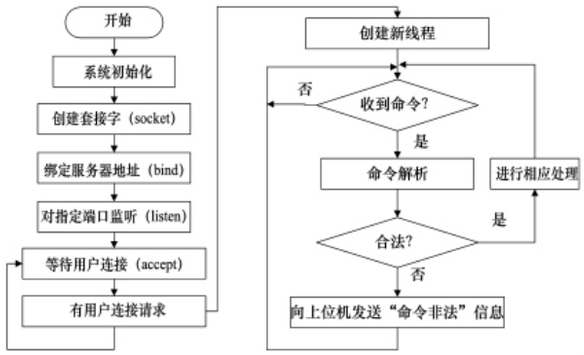

图 3　前端机通信程序流程

Fig. 3　Flow chart of communications program for front-end computer

请求到来时,建立通信线程。

Linux Socket 编程过程如下。

(1) 使用 socket()创建套接字。

int sockfd=socket(int domain, int type, int protocol),参数 domain 为通信发生的区域，一般取 AF_INET 值;type 为套接字的类型，一般取 SOCK_STREAM 和 SOCK_DGRAM 其中之一;protocol 为套接字使用的协议,0 为 TCP,1 为 UDP。

(2) 使用 bind()绑定套接字。

int bind(int sockfd, struct sockaddr *my_addr, int addrlen),参数 sockfd 为套接字描述符;my_addr 为一个含有 IP 地址和端口号等绑定相关信息的结构体;addrlen 为 sockaddr 结构的大小。在调用 bind 函数之前,需要用如下方法对 my_addr 和 addrlen 进行赋值。

my_addr.sin_family=AF_INET;

my_addr.sin_addr.s_addr=INADDR_ANY;

my_addr.sin_port=htons(port);

addrlen=sizeof(my_addr);

(3) 使用 listen()监听连接。

int listen(int sockfd, int backlog),参数 sockfd 为套接字描述符,backlog 为请求队列中允许的最大请求数。

(4) 使用 accept()连接端口的服务请求。

int accept(int sockfd, void *addr, int *addrlen),参数 sockfd 为套接字描述符,addr 为指向数据结构 sockaddr_in 的指针,addrlen 为 sockaddr_in 的大小。

(5) 使用 send()和 recv()完成数据传输。

int send(int sockfd, const void *msg, int len, int flags);

int recv(int sockfd, void *buf, int len, int flags);

参数 sockfd 为套接字描述符,msg 为指向待发数据块的指针,buf 为指向接收缓冲区的指针,flags 常设为 0。

(6) 使用 close()关闭连接。

int close(int sockfd);

参数 sockfd 为套接字描述符。

多线程开发过程如下。

(1) 使用 pthread_create()创建子线程。

int pthread_create(pthread_t *thread, const pthread_attr_t *attr, void *(*start_routine)(void *), void *arg);

参数 thread 为指向线程标识符的指针，参数 attr 用来设置线程属性,第 3 个参数为线程运行函数的起始地址,参数 arg 为运行函数的参数。

(2) 使用 pthread_join()等待线程结束。

int pthread_join(pthread_t thread, void **value_ptr);

参数 thread 为被等待的线程标识符，第 2 个参数为返回等待线程的指针。

(3) 使用 pthread_exit()退出线程。

void pthread_exit(void *value_ptr);

4　上位机网络通信程序实现

基于 x86 架构的上位机采用 Windows XP 操作系统。根据性能要求,调试台架采用 VC++ 6.0 作为上位机软件开发平台。在 VC++环境中,开发基于 Socket 的网络通信程序有 3 种方法:(1) 利用 Windows Socket API 函数;(2) 使用 MFC 类库中 Winsock 类。在 MFC 中有 CAsyncSocket 和 CSocket 两个类,CAsynSocket 对 Windows Socket API 进行了封装,CSocket 是 CAsyncSocket 的派生类。通过这两个类提供的接口可以方便对套接字通信进行处理。(3) 利用 Winsock 控件[5]。相对前 2 种开发方法，第 3 种方法对 Socket 底层技术进行了高级封装,使通信程序开发更简单,因此,本文选择 Winsock 控件完成上位机网络通信程序开发。

4.1　Winsock 控件

Winsock 控件是微软开发的一种用于网络通信的 ActiveX 控件,它提供了基于 Socket 的网络通信所需的接口,可以在 Access、VB、VC 和 VFP 等开发环境中使用[6]。Winsock 控件提供的属性、方法和事件是程序访问的接口。

Winsock 控件常用属性:

(1) LocalPort 本地计算机的端口号;

(2) RemotePort 远程计算机通信端口;

(3) RemoteHost 远程计算机名或 IP 地址;

(4) state 控件的当前状态;

(5) Protocol 通信协议。

Winsock 控件常用方法:

(1) Connect 发出连接请求;

(2) Close 关闭 TCP 连接;

(3) GetData 获取远程计算机传来的数据;

(4) SendData 向远程计算机发送数据。

Winsock 控件常用事件:

(1) ConnectionRequest 当远程计算机请求连接时发生;

(2) DataArrival 数据到达时发生。

利用 Winsock 控件创建基于 C/S 客户端的过程如下。

首先对 Winsock 控件的相应属性进行赋值（包括前端机名或 IP 地址、前端机通信端口)。然后调用 Connect 方法,向前端机发出连接请求，根据 Winsock 控件状态判断是否连接

成功。如果连接成功,可以通过 Winsock 控件的 SendData 方法向前端机发送数据。如果有 DataArrival 事件发生,表明前端机有数据返回,在 DataArrival 事件内通过调用 GetData 方法来得到数据。

4.2　上位机通信程序实现

在 VC++6.0 中建立基于 SDI 的应用程序工程 MCIClient,选择支持 ActiveX 技术,View 继承 CFormView 类。

(1) 将 Winsock 控件加入工程。通过菜单 Project→Add to Project →Component and Controls →Registered ActiveX Controls→Microsoft WinSock Control Version 6.0(sp6),可以将 Winsock 控件加入工程。然后将 Winsock 控件加入 MCIClientView 界面,并定义 m_Winsock 变量。

(2) Winsock 控件初始化。在 MCIClientView 的构造函数中加入如下代码:

```
m_Winsock.SetProtocol(0); //TCP 协议
m_Winsock.SetLocolPort(4000); //本地通信端口
m_Winsock.SetRemoteHost ("192.168.1.25"); //前端机 IP 地址
m_Winsock.SetRemotePort(4000); //前端机通信端口
…
```

(3) 连接前端机。在“连接”按钮的单击响应函数 OnConnectBtn()中加入如下代码:

```
CString strRemoteHost="192.168.1.25";
VARIANT vtRemoteHost, vtRemotePort;
vtRemoteHost.vt=VT_BSTR;
vtRemotePort.vt=VT_I2;
vtRemoteHost.bstrVal= strRemoteHost.AllocSysString();
vtRemotePort.lval=4000;
m_Winsock.Connect(vtRemoteHost, vtRemotePort);
…
```

(4) 向前端机发送数据。在“发送”按钮的单击响应函数 OnSendBtn()中加入如下代码:

```
CString strSendData="Hello";
VARTANT vtSendData;
vtSendData.vt=VT_BSTR;
vtSendData.bstrVal=strSendData. AllocSysString();
m_Winsock.SendData(vtSendData);
…
```

(5) 接收来自前端机的数据。在 m_Winsock 的数据到达事件响应函数 OnDataArrivalWinsock1()中加入如下代码:

```
CString strRecvData="Hi";
VARIANT vtRecvData, vtLen, vtType;
vtRecvData.vt=VT_BSTR;
vtType.vt=VT_BSTR;
vtLen.vt=VT_I2;
vtRecvData.bstrVal=strRecvData.AllocSysString();
vtType.bstrVal=strRecvData.AllocSysString();
vtLen.lVal=bytesTotal;
m_Winsock.GetData(&vtRecvData, vtType, vtLen);
…
```

(6) 程序退出。连接自动关闭,在 MCIClient 的 DestroyWindow()加入下面代码:

```
m_Winsock.Close();
```

5　接口性能测试

通信接口性能直接影响调试台架工作的稳定性和效能。本文对数据传输误码率与传输速度 2 个方面进行测试,从而考查通信接口的性能。测试方法:首先通过上位机与前端机互传系统时间的方法,获得少量数据(系统时间)传输时间和两系统时间差的信息。然后由上位机向前端机发送 1kB 数据,其中包含发送时间;前端机记录接收完成的时刻,并将数据与该时刻一起返回上位机;上位机记录数据接收完成的时刻。通过上述测试时刻值,可以计算出 1kB 的传输时间。通过 CRC 差错校验,可以获知通信过程中是否出现误码。

通过 1000 次以上的通信测试,获得 1kB 的平均传输时间为 3854μs,折算成平均传输速度为 4.06Mb/s,误码率为 0。

6　结论

本文给出了基于 C/S 模式的微电阻率扫描测井仪调试台架通信接口的设计方案,在作为服务器的前端机通信程序中引入了多线程技术,在作为客户机的上位机引入 Winsock 控件。通信测试证明,该接口通信稳定、速度较快,完全满足微电阻率扫描测井仪调试台架对通信接口的性能要求,同时为同类测试系统通信接口开发提供一个可行的方案。

参考文献(References)

[1] 吴鹏程, 陈一健, 杨琳, 等. 成像测井技术研究现状及应用[J]. 天然气勘探与开发, 2007, 30(2): 36–40.
Wu Pengcheng, Chen Yijian, Yang Lin, *et al. Natural Gas Exploration and Development*, 2007, 30(2): 36–40.

[2] 鞠晓东, 成向阳, 卢俊强. 基于嵌入式架构的测井仪器调试台架系统设计[J]. 测井技术, 2009, 33(3): 270–274.
Ju Xiaodong, Cheng Xiangyang, Lu Junqiang. *Well Logging Technology*, 2009, 33(3): 270–274.

[3] 刘媛, 张伟, 王知学. 基于 B/S 和 C/S 架构的嵌入式远程监控系统[J]. 仪表技术与传感器, 2008(10): 39–41.
Liu Yuan, Zhang Wei, Wang Zhixue. *Instrument Technique and Sensor*, 2008(10): 39–41.

[4] 王远洋, 周渊平, 郭焕丽. Linux 下基于 socket 多线程并发通信实现[J]. 微计算机信息, 2009, 25(5): 70–72.
Wang Yuanyang, Zhou Yuanping, Guo Huanli. *Microcomputer Information*, 2009, 25(5): 70–72.

[5] 李林静, 叶冬芬. 运用 Winsock 构建基于 C/S 模式的网络通信[J]. 计算机工程与科学, 2009, 31(2): 20–23
Li Linjing, Ye Dongfen. *Compter Engineering & Science*, 2009, 31(2): 20–23.

[6] 高灵霞. 在 VC 环境下利用 Winsock 实现网络通信[J]. 电脑知识与技术, 2012, 8(20): 4819–4862.
Gao Lingxia. *Computer Knowledge and Technology*, 2012, 8(20): 4819–4862.

(责任编辑　刘志远)