ZMODEM协议在文件传输中的应用与实现

孙金秋[1],怀红旗[2]
(1.南京航空航天大学,江苏 南京 210016;
2.东南大学,江苏 南京 210096)

摘 要:在PC与PC之间,PC与嵌入式系统之间利用非网络方式进行文件传输时,ZMODEM协议是人们的首选。分析了ZMODEM协议的工作机制,并具体实现了两台PC机之间基于ZMODEM协议的文件传输。同时提供给用户接口函数以便用户编写自己的应用程序。所实现的文件传输具有可靠性高、传输速度快、断点续传等优点,从而说明了在文件传输时,ZMODEM协议是一个不错的选择。

关键词:ZMODEM协议;文件传输;可靠性;传输速度;断点续传
中图分类号:TP393.093 **文献标识码**:A **文章编号**:1673-629X(2008)05-0241-04

Realization of File Transmission Based on ZMODEM Protocol

SUN Jin-qiu[1], HUAI Hong-qi[2]
(1.Nanjing University of Aeronautics and Astronautics,Nanjing 210016,China;
2.Southeast University, Nanjing 210096,China)

Abstract:Because the ZMODEM protocol has merits of easing use, reliability, integrity and robustness, mostly it has retransmission for error correction, so it is usually used when transfer files between computer and other equipments. Introduces the mechanism of ZMODEM protocol in detail. In addition ,sending and receiving programs are realized based on the computer platform. Finally, gives some interfaces which can be porting easily.

Key words:ZMODEM protocol;file transmission;reliability;transmission speed;retransmission for error correction

0 引 言

在计算机和网络技术迅速发展的今天,人们每天通过网络从万维网和FTP站点传输文件的数量是巨大的,网络也成为文件传输的主要方式。但网络并不是每时每刻都存在,这时人们就会想到用串口、并口、USB口等通信方式直接进行文件传输。若要使文件传输速度快、可靠性高,文件传输协议的选择就尤为重要[1~6]。

目前常用的文件传输协议有:XMODEM,YMODEM,ZMODEM等等。其中XMODEM,YMODEM相对比较简单,较容易实现。但它们存在一些局限性,如:可靠性不高、传输速度低等。ZMODEM是在XMODEM,YMODEM之后出现的。它虽然是一个比较复杂的协议,然而它优于其它协议。ZMODEM协议中所有的数据都受CRC保护;数据以连续的包流方式发送而无需等待确认;记录每个数据包的位置。所以它具有可靠性高、传输速度快、断点续传等优点,因而成为现在非常流行的协议选择。

下面将主要介绍ZMODEM协议在文件传输中的具体应用与实现[1,7]。

1 ZMODEM协议的工作原理

ZMODEM协议中规定,发送器和接收器之间所传递的信息都包含在包中,即帧中。ZMODEM帧中有两个组件:报头和数据子包。每个帧以报头开始,它标识帧的类型,并带有至少4个字节的信息。16位CRC二进制报头的格式为:ZPAD ZDLE A frame-type ZF3 ZF2 ZF1 ZF0 CRC-1 CRC-2。ZPAD ZDLE表示帧的开始,A表示报头的数据是二进制格式,frame-type表示帧类型,ZF3 ZF2 ZF1 ZF0是4个字节的信息,当帧类型不同时,它的含义也不同,CRC-1 CRC-2为校验码。数据子包中包含的是原始的数据块,每个数据子包可以包含至多1024个字节的数据,其后跟随一个CRC值用以校验。数据子包可以有选择地跟随一个报头后面,而一个报头之后也可以连接无数个数据

收稿日期:2007-08-31
作者简介:孙金秋(1979-),硕士,研究方向为电源和嵌入式系统。

子包,这种机制保证了数据的传输速度[7]。

1.1 ZMODEM 协议中的帧类型及作用

ZMODEM 协议中有 18 种帧类型,下面将介绍主要用到的帧类型:

ZRQINIT=0:当 ZMODEM 发送端启动时,发送此帧。请求接收端发送它的 ZRINIT 帧。开始文件的传输。

ZRINIT=1:接收端发送此帧,表明它已准备好从发送端接收文件。

ZFIFLE=4:此帧用于初始化实际的文件传输,它包含一个报头,后面跟随一个数据子包。数据子含有文件的信息,如文件名称、长度、日期等等。

ZRPOS=9:接收端在任何时刻都可以发送这个帧。4 个字节的偏移量填充到报头信息的 4 字节,偏移量是接收器请求从文件的某个位置开始发送数据的请求。此帧可实现断点续传的功能。

ZDATA=10:这个帧的 4 个报头字节包含跟随数据的偏移量,其后可跟随任意个数据子包。数据子包含文件内容。

ZEOF=11:指明所有文件数据都已发送,4 个报头字节包含了 EOF 的偏移量。

ZFIN=8:当发送端再没有文件传送时发送这个帧。接收端在退出之前以它自己的 ZFIN 帧应答。

1.2 ZMODEM 协议的工作过程

发送端和接收端要传输文件,首先必须建立连接,双方通过一些传输选项的选择和协商指定好文件传输所必须遵循的规范。然后发送端开始进行文件传输,其中 ZFIFLE 帧中含有文件处理的选项,而 ZDATA 帧将具体用来传送文件数据。在文件传输过程中,对于特定的要求发送端和接收端会进行一些信息的请求与响应。最后当所有文件都传输完毕,会话将被终止。

按照结构化的方法对系统进行设计,ZMODEM 协议的工作过程可以细分为以下几个部分:

1) Start Sending(发送端建立连接);

2) Start Receiving(接收端建立连接);

3) Transmission of Sender(发送端传输文件过程);

4) Transmission of Receiver(接收端接收文件过程);

5) Management of ZDATA frame(接收端 ZDATA 帧的处理过程);

6) Management of ZEOF frame(对 ZEOF 帧的处理过程);

7)End Sending(发送端终止发送);

8)End Receiving(接收端终止接收)。

对于每一部分均可以用流程图阐释其工作过程。

因篇幅的原因,下面只给出 1)Start Sending 和 3) Transmission of Sender 两部分的详细流程图,分别为图 1 和图 2。

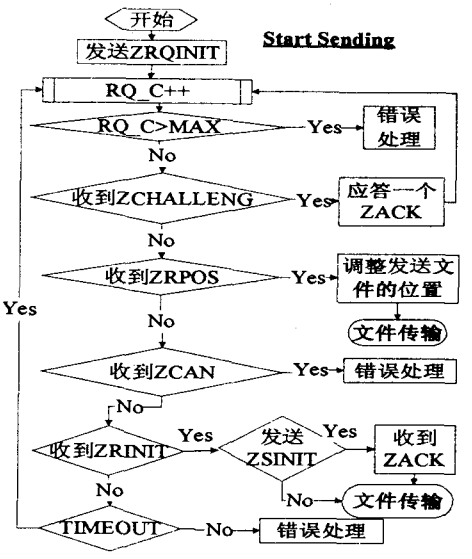

图 1 Start Sending 流程图

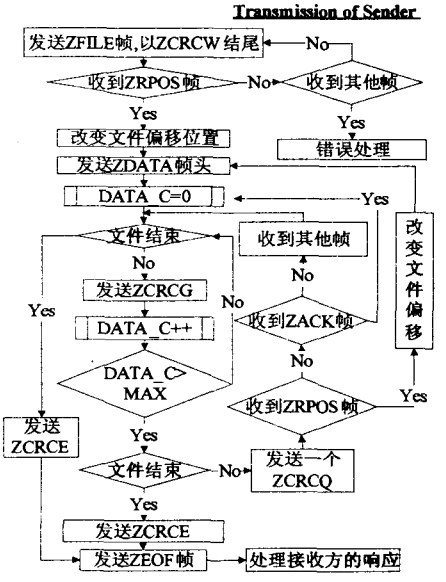

图 2 Transmission of Sender 流程图

图 1 说明了发送端建立连接的过程,发送端首先发送一个测试文本 rz\r,然后再发送 ZRQINIT 帧,之后若收到接收端反馈回来的帧为 ZRINIT,就进入发送端文件传输过程,若反馈回来的帧为别的类型,则做相应的处理再进入发送端文件传输过程。图 2 说明了发送端传输文件的详细过程,发送端先发送 ZFIFLE 以初始化待传输的文件的信息,然后再发送 ZDATA 帧,其后的数据子包含有要传输的文件内容,而数据子包的个数则是由待传输文件的大小决定的。同时需注意的是,若在发送 ZDATA 帧之前,或之中,发送端检测到接收端反馈回的 ZRPOS 帧,则发送端需调整文件

指针,从接收端要求的位置重新发送文件。

2 基于ZMODEM协议传输文件具体实现

2.1 系统的整体框架

为了此系统可以用于不同的场合,如PC与PC之间的文件传输,PC与嵌入式系统之间的文件传输,嵌入式系统与嵌入式系统之间的传输等等,本系统采用了分层设计的实现方法。系统的整体框架见图3。

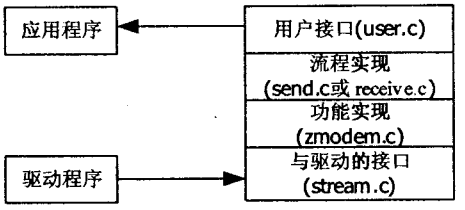

图3　系统的整体框架

在图3所示的六层结构中,最底层是驱动程序。当在不同的发送端和接收端传输文件时,或选择不同的通信方式时,只需要重新编此驱动程序即可。六层结构的第二层(由下而上)主要通过与驱动程序相接而实现字符的发送和接收,以及字符的编码和解码。六层结构的第三层为功能实现层,主要实现ZMODEM协议帧中各个包头及数据子包的发送和接收。六层结构的第四层为流程实现层,主要功能是:建立连接,开始发送;完成发送端和接收端的初始化;完成单个文件的发送和接收,从而将所有的文件都传输完毕;结束文件传输等等。六层结构的第五层为用户接口层,主要将代码封装,只留给用户接口函数。六层结构的第六层为应用程序层,用户可根据系统提供的接口函数编写自己的应用程序。

2.2 系统的具体实现

下面将介绍基于ZMODEM协议的文件传输的具体实现。PC与PC之间,嵌入式系统与PC之间都可能需要文件传输,此系统选择的是在两台PC机之间实现文件传输。环境为μC/OS-Ⅱ嵌入式系统,编程语言为C。在并口、串口、USB口、红外接口这几种通信方式中,串口通信因其在数字信号传输过程中抗干扰能力、传输距离、传输线等方面的优势,而得到广泛的应用。所以本系统采用的是串口通信的方式[8]。

因篇幅原因,只给出第一层驱动程序和第五层用户接口两部分的具体实现。

2.2.1 串口通信方式时驱动程序的接口函数

(1) PCOMM_HANDLE Comm_Open(INT8U port, COMM_SETUP setup, INT16U timeout);

功能:设置通讯参数,并获得串口独占使用权。

参数:port;串口名(COM1,COM2);setup串口通讯参数;Timeout获得权限的最大等待时间,0为永远等待。

返回值:串口句柄,如果为NULL时表明没有获得对该串口的使用权。

(2) BOOLEAN Comm_Close(PCOMM_HANDLE port_handle);

功能:释放串口的使用权。

参数:要释放的串口的句柄。

返回值:释放成功标志,(TRUE为成功,FALSE为失败)。

(3) BOOLEAN Comm_Send(PCOMM_HANDLE port_handle, INT8U * pSource, INT8U size, INT16U timeout);

功能:从串口发送size个字节在pSource中的数据。

参数:port_handle串口句柄;pSource发送缓冲区首地址;size要发送数据字节数;timeout每发送一个字节的最大等待时间,0为永远等待。

返回值:发送成功标志(TRUE为成功,FALSE为失败)。

(4) BOOLEAN Comm_Receive(PCOMM_HANDLE port_handle, INT8U * pTarget, INT8U size, INT16U timeout);

功能:从串口接收size个字节,并存放到pTarget中。

参数:port_handle串口句柄;pTarget接收缓冲区首地址;size要接收数据字节数;timeout每接收一个字节的最大等待时间,0为永远等待。

返回值:接收成功标志(TRUE为成功,FALSE为失败)。

2.2.2 采用ZMODEM协议进行文件传输的用户接口函数

(1) BOOLEAN Init(INT16U * err)

功能:系统初始化;参数:err错误参数;返回值:TRUE成功 FALSE失败。

(2) BOOLEAN SendFiles(INT8U * file[4], RS232 com, Cmd_Options op, INT16U * err)

功能:使用ZMODEM协议发送文件;

参数:file[]文件名;com串口参数;op命令行参数;err错误参数。

返回值:TRUE成功,FALSE失败。

```
typedef struct{
    INT8U port;
    COMM_SETUP setup;
    }RS232, * pRS232
typedef struct{
    INT32U baud;
    INT8U bits;
```

```
    INT8U stops;
    INT8U parity;
    INT8U mode;
    }COMM_SETUP, * PCOMM_SETUP;
typedef struct{
    INT8U management;
    BOOLEAN asc;
    BOOLEAN bin;
    BOOLEAN esc;
    BOOLEAN crc16;
    }Cmd_Option, * pCmd_Option;
```

其中:management = 0 不覆盖文件;management = 1 无条件覆盖文件;management = 2 追加在目标文件后;management = 4 源文件新则覆盖目标文件。

(3) BOOLEAN ReceiveFiles(RS232 com, Cmd_Options op, INT16U * err)

功能:使用 ZMODEM 协议接收文件。

参数:com 串口参数;op 命令行参数;err 错误参数。

返回值:TRUE 成功,ALSE 失败。

3 结 语

实现了在 μC/OS-Ⅱ环境下,两台 PC 机之间基于 ZMODEM 协议的文件传输,其传输速度快、可靠性高、断点能够续传。从而说明 ZMODEM 协议在文件传输时是一个不错的选择。采用模块化设计方法,使 ZMODEM 协议化繁为简,更便于理解与掌握。采用分层实现方法,使代码简洁、清晰,且可以只修改驱动程序而使此系统用于不同场合的文件传输。而在本系统中与操作系统与硬件相关的代码大部分放在 PC.C 文件中,便于此系统以后的移植。

参考文献:

[1] Forsberg C. The ZMODEM Inter Application File Transfer Protocol[M]. [s.l.]:Omen Technology Inc,1988.

[2] 岳晓庆,张其善. 串口扩口技术在嵌入式系统中的实现[J]. 电子测量技术,2006,29(2):45-46.

[3] 许春冬,陈良军. 嵌入式数字视频监控系统中串口通信的设计与实现[J]. 电子科技,2005(11):61-63.

[4] 刘燕军,蒋存波,陈占海. 嵌入式系统与 IPC 的一种串口通讯协议及其实现[J]. 桂林工学院学报,2006,26(4):579-582.

[5] 赵世湖,周 辉. 数字成像嵌入式 DSP 系统与 PC 间的串行通信[J]. 影像技术,2007(2):26-28.

[6] 贾瑞玉,赖大荣. 车流量测量仪串口通信的设计与实现[J]. 计算机技术与发展,2006,16(10):199-201.

[7] Nelson M. 串行通信开发指南[M]. 潇湘工作室译. 北京:中国水利水电出版社,2001.

[8] 王荣良. 微机原理与接口技术[M]. 北京:高等教育出版社,2005.

(上接第 240 页)

发往主机 2000:1:1:1:217:a4ff:fecd:bfc3 的数据报通过分类 3 转发(分类 3 的速 1Mbit)

```
# ip route add 2000:1:1:1:217:a4ff:fecd:bfc3
dev eth0 via 2000:1:1:1:210:5cff:fef9:52d8 realm 3
```

经过以上步骤,就能够对通过网卡 eth0 的流量进行控制和管理,同时还可以用命令对包括现有队列、分类和过滤器的状况进行监视,看看控制是否达到预期的效果,可用的命令有:

* tc qdisc ls dev:显示队列的状况
* tc class ls dev:显示分类的状况
* tc - s filter ls dev:显示过滤器的状况

4 结 语

文章针对区分服务流分类和如何利用 Linux 的流量控制策略,根据队列、分类、过滤器和路由来设计并实现对不同数据流的区分和带宽保证,提高了传统网络的服务质量。为了让 DiffServ 机制更完善,还有很多工作需继续进行,如在流量巨大的网络中,如何寻找到一种最合理的排队机制、最有效的分配资源,是需要进一步努力的,同时也需要更大量的实践和仿真。

参考文献:

[1] Deering S, Hinden R. Internet Protocol, Version 6(IPv6) Specification[S/OL]. RFC 2460. 1998-12. http://www.faqs.org/rfcs/rfc2460.html.

[2] Jacobson V, Nichols K, Poduri K. An Expedited Forwarding PHB[S/OL]. IETF RFC 2598. 1999-06. http://www.ietf.org/rfc/rfc2598.txt.

[3] Blake S, Black D, Carlson M, et al. An Architecture for Differentiated Services[S/OL]. RFC 2475. 1998. http://www.ietf.org/rfc/rfc2475.txt.

[4] 吴建平,盛立杰,林 闯,等. 区分服务及其几个热点问题的研究[J]. 计算机学报,2004(4):419-433.

[5] Conta A, Carpenter B. The IPv6 flow label and use of IPv6 flow labels with DiffServ[R/OL]. 2001. http://playground.sun.com/pub/ipng/html/presentations/aug2001/IPv6-flow-label-07.PDF.

[6] 李蔚泽. Red Hat Linux 9 系统管理[M]. 北京:清华大学出版社, 2004.

[7] Almesberge W. Linux Traffic Control [EB/OL]. 2001. http://feela.network.cz/ltc.html.

[8] Hubert B, Netherlabs B V. Linux Advanced Routing & Traffic Control[EB/OL]. 2003. http://lartc.org/lartc.pdf.