

ARM 平台多进程微内核嵌入式操作系统 Ymodem 下载功能设计与实现

吴兆芝

(南京晓庄学院 数学与信息技术学院, 江苏 南京 211171)

摘要:文章阐述了 ARM 平台多进程微内核嵌入式操作系统 Ymodem 下载功能设计与实现的技术细节, 内容包括常用串口通信协议(Xmodem、Ymodem 及 Zmodem)简介, ARM 平台多进程微内核嵌入式操作系统简介, Ymodem 协议关键技术及下载功能编程要点. 最后介绍了使用 Ymodem 协议设计实现 Shell 命令 rx 的基本方法与应用实例. 实践证明, Ymodem 协议的功能与效率明显优于 Xmodem 协议. 同时与 Zmodem 协议相比, Ymodem 协议又具有程序结构简单、易于理解与实现的优势. 因此 Ymodem 协议非常适合于嵌入式操作系统下载功能的实现.

关键词: Ymodem 协议; 多进程; 微内核; 嵌入式操作系统; ARM

中图分类号: TP316 **文献标识码:** A **文章编号:** 1009-7902(2013)03-0079-06

随着嵌入式系统应用越来越普遍, 嵌入式操作系统逐渐成为业内研究重点, 如何设计实现适用于嵌入式系统的专用操作系统成为一种技术研究趋势.

鉴于操作系统研究与嵌入式系统课程教学^[3]的需要, 笔者参与设计了一个基于 ARM 平台的多进程微内核嵌入式操作系统 Mikenix. 该系统是在 Linux 平台使用 GNU 工具链^[4]交叉编译器开发而成, 使用 Ymodem 协议实现串口下载功能, 方便了操作系统的开发与应用. 实践证明 Ymodem 的效率及功能明显优于 Xmodem, 同时与 Zmodem 相比又具有程序简洁、易于实现的优势.

本文阐述了 ARM 平台多进程微内核嵌入式操作系统 Ymodem 下载功能设计与实现的技术细节, 内容包括常用串口通信协议(Xmodem、Ymodem 及 Zmodem)简介, ARM 平台多进程微内核嵌入式操作系统简介, Ymodem 协议关键技术及下载功能编程要点. 最后介绍了使用 Ymodem 协议设计实现 Shell 命令 rx 的基本方法与应用实例.

1 常用串口通信协议简介

众所周知, 著名的 Xmodem、Ymodem 及 Zmodem 是在使用调制解调器的个人计算机通信中应用最广泛的异步文件传输协议. 其中, Xmodem 协议结构最简单、使用最广泛, 但效率也最低; Ymodem 是 Xmodem 的改进版本, 增强了功能与效率, 可以传输文件名和文件长度, 以及批量文件; Zmodem 功能最强、效率最高, 但结构也最复杂.

1.1 Xmodem 协议

Xmodem^[5]是著名黑客、美国芝加哥人 Ward Christensen 在他的终端程序 modem.asm 中所开发的一个简单的文件传输协议. 很大程度上由于该协议结构简单、易于实现, Xmodem 协议在早期的 BBS 市场中分外流行.

与大多数文件传输协议一样, Xmodem 协议将原始数据分解为一系列“分组”(packet), 每一个分组与附

收稿日期: 2013-03-02

作者简介: 吴兆芝, 南京晓庄学院数学与信息技术学院高级实验师, 研究方向: 计算机应用技术.

加信息一道发送给接收端. 根据这些附加信息, 接收端能够确定所收到的分组正确与否.

最初的 Xmodem 协议使用 128 字节长的分组, 与早期的 CP/M 操作系统软盘块尺寸相同. 每个 Xmodem 数据帧由 3 字节前缀、128 字节分组及 1 字节后缀构成, 其中 3 字节前缀为:

- 1) SOH(Start of head), 帧起始标志, 其 ASCII 码值为 1;
 - 2) Blk#(Block number), 块序号, 1 - 255;
 - 3) Cmpl(Complementary of block number), 块序号反码(1 的补码), 255 - Blk#.
- 1 字节后缀为 Csum(CheckSum), 垂直累加校验和. 整个帧结构如图 1 所示.

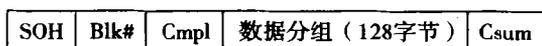


图 1 Xmodem 协议帧格式

校验和 csum 为 128 字节分组中的所有字节之和除以 256 的余数. 该值可以简单地通过舍弃数据和的高位而只留下最低 8 位二进制数, 或者以 8 位补码求和舍弃进位而求得. 这使得校验和限制在 8 位二进制数之内, 因此只需一个字节就可存放. 例如, 假设一个很小的数据分组, 只有两个非 0 字节, 其值分别为 130、131, 则其总和为 261, 按上述方法得到校验和为 5.

显然, 每个帧的长度为 132 字节, 其中包含 128 字节的有效数据, 因此其通道效率约为 97%.

文件传输的结束标志是在最后一个数据帧之后所发送的 EOT(End of Transmission) 字符, 其 ASCII 码值为 4. 该字符不包含在数据帧里, 而是作为单独一个字节独立发送的. 由于文件长度不是协议的组成部分所以不被发送, 如果文件长度不是 128 字节的整倍数的话, 其最后一个分组就会以 SUB(Substitute) 字符填充. Xmodem 协议建议使用 CTRL-Z(ASCII 码值为 26) 作替代字符. 协议标准允许任何字符作为填充替代字符而协议本身不会有任何变化, 因此一个实现方案如果改变了填充字符, 只需使用同一实现方案的客户端对新的填充字符作正确解读.

为了提高传输效率, 后来又出现了 Xmodem-1K, 以 1024 字节作为分组长度.

1.2 Ymodem 协议

Ymodem^[5] 协议是在 Xmodem-1K 的基础上修改升级而成, 可以实现文件的批量传输. 它是一个半双工(half-duplex) 协议, 因此不能同时双向收发控制信号, 这有助于减少缓冲区超限问题. Ymodem 协议的操作与 Xmodem 类似, 但它可以在传输文件内容之前用正规的 Xmodem 数据块(0 号块) 发送文件名、文件时间戳及文件长度.

最初的 Ymodem 协议选项之一是 Ymodem-1K, 块长度为 1024 字节. 另一个选项是 Ymodem-g, 可以看作是 Ymodem 的一个变体, 设计用来与支持错误控制的调制解调器一道工作. Ymodem 的 g 选项由接收端驱动, 通过发送一个字符“g”启动一个批量传输. 当发送端识别到“g”之后, 放弃等待接收端对每一个传输块的确认(ACK) 信号, 将后续块以最快的速度发送出去. 发送端期待一个“g”来启动一次文件传输, 而当文件传输结束之后, 期待对 EOT(End Of Transmission) 信号的确认, 即 ACK(Acknowledgment).

与其它类似协议不同, Ymodem-g 协议不提供软件错误的更正与恢复, 而是期待调制解调器提供等价的服务. 这种流协议将分组作为连续的数据流发送与接收, 直至被指令所终止. 所有数据块被持续发送, 在每一个块被发送之后不需等待接收方的确认. 如果一个数据块没有传输成功, 则整个操作被取消.

1.3 Zmodem 协议

Zmodem^[6] 是 1986 年由 Telenet 资助的一个项目中开发出来的一个文件传输协议, 目的是在 X.25 网络中改善文件传输性能. 与以往协议相比, 除了性能明显改善之外, Zmodem 协议还提供了由发送方启动的重传功能, 扩展的 32 位 CRC 校验功能, 以及控制字符引用(control character quoting). 该协议在 90 年代初被 BBS 系统广泛应用, 以取代早期的文件传输协议, 如 Xmodem 及 Ymodem 等.

Zmodem 协议的一个关键性的改进是引入了滑动窗口(sliding window) 机制以改善性能. 通常文件传输协议将一个文件分解成一系列分组, 然后将这些分组依次发送给接收端, 每次发送一个分组. 如果一个分组被正确接收, 则接收端向发送端发送一个确认信号(ACK), 发送端收到该信号后继续发送下一个分组. 然而, 电话系统所引起的小量的延迟, 称之为等待时间(latency), 却对这个过程带来影响. 即使接收方马上发送

了确认信号(ACK),由于电话线路的延迟也会导致发送方不能及时收到从而推迟后续分组的发送.当调制解调器速度增加时,这种影响会更加明显.

滑动窗口协议可以避免这种问题的产生,它允许发送方无需等待确认信号(ACK)而直接发送下一个分组.这种方法与TCP协议的滑动窗口机制类似.

Zmodem的自动启动功能通过允许发送方启动传输而简化了管理.以前,用户需要首先从发送端请求一个文件,设置为“等待”(Waiting)状态,然后返回到本地程序,调用启动文件传输的命令.使用自动启动功能,用户只需选择请求的文件,然后发送端将自动启动用户程序.

重启是Zmodem的另一个新增功能,当用户端超时或掉线时自动重启传输.

比较而言,Zmodem是三者中功能最强效率最高的一种串口通信文件传输协议,但其结构也最为复杂,不如前两种协议那样易于理解与实现.

2 ARM平台多进程微内核嵌入式操作系统简介

近年来,随着电子技术及计算机技术的不断发展与普及,嵌入式系统的应用领域越来越广泛.作为嵌入式系统核心基础的嵌入式操作系统也越来越受到人们的关注与重视.尽管多年来有诸多著名的嵌入式操作系统^[2]问世,如VxWorks、QNX、ECOS、uCOS、uCLinux、嵌入式Linux以及Window CE等,但由于嵌入式系统本身的固有特点,如品种多、规模小、硬件性能低、软件资源少等,如何设计满足特定环境及特定需求的嵌入式操作系统仍然是计算机领域的一个热点问题.与此同时,各高等院校也大量开设了嵌入式系统相关课程,其中非常重要的一门课程就是嵌入式操作系统.

为满足嵌入式操作系统研究及其相关课程教学的需要,笔者参与开发了一个基于ARM平台的多进程微内核嵌入式操作系统,命名为Mikenix.该操作系统(以下简称Mikenix)采用与MINIX^[1]相同的微内核体系结构,以消息传递(Message Passing)机制作为进程间通信(Inter-Process Communication)的主要途径,在ARM平台上实现了一般嵌入式操作系统所需的基本功能,包括多进程创建、多优先级队列进程调度、设备驱动、信号处理、文件系统、TTY及简易Shell等.

图2展示了Mikenix的体系结构.

由于嵌入式系统平台功能的局限性,系统开发工作只能在另外的开发主机上完成,然后下载到嵌入式系统主机上加载运行.本文描述的Ymodem下载功能的主要目的就是将在开发主机上编译链接后生成的可执行目标程序通过串口下载到嵌入式主机.这对于嵌入式操作系统的开发,尤其是Shell程序的设计与实现带来极大方便.

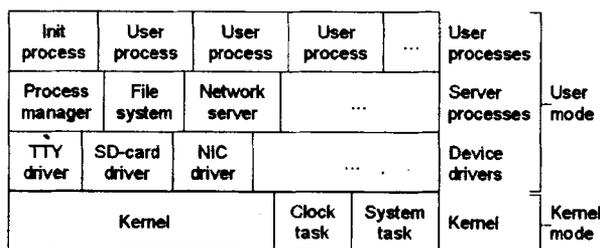


图2 多进程微内核嵌入式操作系统 Mikenix 体系结构

3 Ymodem 关键技术及下载功能编程要点

Ymodem协议是一种“发送<—>等待”协议,发送方每发送一个数据包以后,要等待接收方的确认.如果是确认(ACK)信号,则继续发送后续数据包;如果是出现错误(NAK)信号,则重发或者错误退出.

使用Ymodem协议进行文件传输的过程如下:

1) 由接收方启动传输.接收方发出一个大写字母C开启传输,然后进入等待(SOH)状态,如果没有回应,则重新开启或超时退出.

2) 发送方开始时处于等待过程中,等待接收方的启动信号“C”.收到“C”之后,发送方依次发送数据包开始信号(SOH)、数据包序号(Blk#)、数据包序号反码(Cmpl)、“文件名”、“空格”、“文件大小”、“填充字节(除去数据包序号外,补满128字节)”及CRC校验码(两个字节,高位在前,低位在后),然后进入等待“ACK”状态.

3) 接收方收到上述内容后,进行16位CRC校验.若校验成功,则发送确认信号(ACK).发送方接收到ACK后,又进入等待“文件传输开启”信号,即重新进入等待“C”的状态.

4) 至此接收方收到待传输文件的文件名,接收方又发出一个“C”信号,开始准备接收文件.进入等待“SOH 或 STX”状态.

5) 发送接收到“C”以后,发送数据包:(SOH 或 STX)、(序号 Blk#)、(反码 Cmpl)、(128 或 1024 字节数据)及(CRC 校验码),等待接收方的“ACK”.

6) 文件发送完以后毕,发送方发出一个“EOT”信号,接收方以“ACK”回应.至此,一个文件的传输结束.利用 Ymodem 协议,还可以重新启动传输,继续发送下一个文件,实现文件的批量发送.

SOH	Blk#	Cmpl	数据分组 (128字节)	CRC-16
STX	Blk#	Cmpl	数据分组 (1024字节)	CRC-16

图3 Ymodem 数据帧的两种格式(128 或 1024 字节)

Ymodem 协议的数据帧格式有两种,一种是以 SOH(Start of head)为前缀的 128 字节数据包,一种是以 STX(Start of Text)为前缀的 1024 字节数据包,如图 3 所示.

```

u16 crc16(char *buf, u32 count)
{
    u16 crc = 0;
    int i;
    while(count--) {
        crc = crc ^ *buf++ << 8;
        for(i=0; i<8; i++)
            if(crc & 0x8000) crc = crc << 1 ^ 0x1021;
            else crc = crc << 1;
    }
    return crc;
}

```

图4 CRC-16 校验的 C 代码

为确认文件传输内容的正确性,接收方要对接收到的数据包进行 16 位的 CRC(循环冗余)校验.根据 Ymodem 协议规范,其 CRC 的生成多项式为

$$X^{16} + X^{12} + X^5 + X^0$$

图 4 展示了实现 CRC-16 校验的 C 代码,其中 0x1021 代表上述生成多项式.

4 Shell 命令 rx 设计与实现

为实现 Ymodem 协议下载功能,专门设计了一个 Shell^[1] 命令程序 rx. 该命令可使用不同的命令参数,从而极大方便了下载操作.

1) 无参数. 此时只需在命令行方式下直接使用 rx 命令下载,rx 会根据 Ymodem 协议传输内容获取远程文件的文件名及文件长度,用该文件名在当前目录下保存下载文件.

2) 文件名作参数. 此时 rx 命令使用该文件名保存下载文件. 该文件名可以在当前目录,也可以在其它目录.

3) 目录作参数. 此时 rx 命令根据 Ymodem 协议传输内容获取远程文件的文件名及文件长度,用该文件名在参数指定的目录下保存下载文件.

图 5 展示了命令参数处理的关键代码.

在 Linux 平台使用 GNU 工具链交叉编译器^[4]将 Shell 命令程序编译链接形成 ARM 平台可执行程序后,需将其下载到 ARM 主机运行. 下载方法主要有两种. 一种是先将其拷贝到根文件系统盘中^[3],然后重新拷贝到存储介质(如 SD 卡);另一种方法就是在超级终端上使用 rx 命令通过串口下载. 图 6 展示了 rx 下载的一个实例.

```

if(argc == 2) {
    i = strlen(argv[1]); strcpy(filename, argv[1]);
    if(stat(filename, &sb)) {
        if(filename[i - 1] == '/') {
            printf("Error: Directory %s not exist\n",
filename);
            exit(1);
        }
    } else {
        if((sb.st_mode & S_IFDIR)) {
            dir = 1; fn += i;
            if(filename[i - 1] != '/')
                *fn++ = '/';
        }
    }
}
}
}

```

图5 Shell命令rx处理命令参数的关键代码



图6 Shell命令rx下载演示实例

4 结束语

如前所述,随着嵌入式系统应用越来越广泛,对于嵌入式操作系统的研究也越来越深入.本文阐述的ARM平台多进程微内核操作系统Ymodem下载功能的设计与实现,就是一个很有价值的研究实例.由于篇幅所限,省略了部分技术细节,如Shell命令的程序结构、交叉编译的实现等.

本文所述操作系统Mikenix及Ymodem下载功能使用的开发平台及工具为Linux操作系统及GNU工具链,既适用于实际的嵌入式系统开发工程,也适用于高校计算机及相关专业的嵌入式系统课程教学与实验.不断加强这方面研究,无疑具有重要的实际意义.

参考文献:

- [1] A. S. Tanenbaum. Operating Systems: Design and Implementation[M]. Third Edition, Prentice Hall, Inc., 2008.
- [2] T. Noergaard. Embedded Systems Architecture, A Comprehensive Guide for Engineers and Programmers[M]. Elsevier Inc, 2005.
- [3] B. Qu and Z. Z. Wu. Design and Implementation of Tiny Educational OS[J]. Lecture Notes in Electrical Engineering, Recent Advances in Computer Science and Information Engineering, 2012, 126: 437 - 442.
- [4] R. M. Stallman. Using the GNU Compiler Collection[EB/OL]. <http://www.gnuarm.com/pdf/gcc.pdf>, 2002 - 12 - 30.
- [5] C. Forsberg. XModem/YModem Protocol Reference[EB/OL]. <http://pauillac.inria.fr/~doligez/zmodem/ymodem.txt>, 1988 - 10 - 14.
- [6] C. Forsberg. The ZMODEM Inter Application File Transfer Protocol[EB/OL]. <http://pauillac.inria.fr/~doligez/zmodem/zmodem.txt>, 1988 - 10 - 14.

(责任编辑:王海军)

(下转第89页)

一步自主实现全部程序编码奠定了坚实基础。

参考文献:

- [1] S. Andrew, Tanenbaum, Computer Networks, Fourth Edition[M]. Prentice Hall, Inc. 2008.
- [2] S. William. Network Security Essentials: Applications and Standards, Third Edition[M]. Prentice Hall, Inc. 2007.
- [3] J. Yaghmour, J. Masters, G. Ben-Yossef, P. Gerum. Building Embedded Linux System[M]. O' Reilly Media, Inc. 2008.
- [4] B. Qu, Z. Z Wu. Design and Implementation of SSL Secure Proxy for ARM Platform[J]. Proceedings of 2011 International Conference on Internet Technology and Applications, 2011(1):374-377.
- [5] B. Qu, Z. Z Wu. Design and Implementation of Embedded Secure Web Server for ARM Platform[J]. Proceedings of 2011 International Conference on Electronic and Mechanical Engineering and Information Technology, 2011(1):359-362.

(责任编辑:王海军)

Research on Porting Embedded Secure Proxy and Secure Web Server

LI Ming-shang, QU Bo, ZHANG Xin-juan, LU Qing-juan

(School of Mathematics and Information Technology, Nanjing Xiaozhuang University, Nanjing 211171, Jiangsu)

Abstract: The paper describes the key techniques of porting embedded secure proxy and secure Web server based on the ARM platform, mainly including the porting method of openssl on the ARM platform, the method of porting, installing and running SSL proxy and secure Web server on ARM host. Two demo examples are provided finally to show the running effects of the secure proxy and secure Web server, respectively.

Key words: embedded system; secure proxy; secure Web server; porting; SSL

(上接第 83 页)

Design and Implementation of Ymodem Downloading for Multi-Process Micro-Kernel Embedded OS on ARM

WU Zhao-zhi

(School of Mathematics and Information Technology, Nanjing Xiaozhuang University, Nanjing 211171, Jiangsu)

Abstract: This paper describes the design and implementation of Ymodem downloading for multi-process micro-kernel embedded operating system on ARM in technical details, including an introduction of commonly-used communication protocols via serial port (Xmodem, Ymodem, and Zmodem), an overview of multi-process micro-kernel embedded operating system on ARM, and key techniques and programming essentials of Ymodem. The paper finally describes the way of designing shell command, named rx, as well as its demo with Ymodem. It has been proved that the performance and efficiency of Ymodem protocol is superior to that of Xmodem. On the other hand, compared with Zmodem, Ymodem protocol has such advantages as the program structure is simple and hence helpful for understanding and implementing. Therefore, Ymodem is very suitable for implementing the downloading function for embedded operating system.

Key words: Ymodem; multi-process; micro-kernel; embedded operating system; ARM

作者: [吴兆芝, WU Zhao-zhi](#)
作者单位: [南京晓庄学院数学与信息技术学院, 江苏南京, 211171](#)
刊名: [南京晓庄学院学报](#)
英文刊名: [Journal of Nanjing Xiaozhuang University](#)
年, 卷(期): 2013(3)

参考文献(6条)

1. [A. S. Tanenbaum](#) [Operating Systems: Design and Implementation](#) 2008
2. [T. Noergaard](#) [Embedded Systems Architecture, A Comprehensive Guide for Engineers and Programmem](#) 2005
3. [B. Qu; Z. Z. Wu](#) [Design and Implementation of Tiny Educational OS](#) 2012
4. [R. M. Stallman](#) [Using the GNU Compiler Collection](#) 2002
5. [C. Forsberg](#) [XModem/YModem Protocol Reference](#) 1988
6. [C. Forsberg](#) [The ZMODEM Inter Application File Transfer Protocol](#) 1988

本文链接: http://d.g.wanfangdata.com.cn/Periodical_njzxxyxb201303020.aspx