

嵌入式系统下的邮件服务器的设计

袁志祥, 秦 锋, 王小林

(安徽工业大学 计算机学院, 安徽 马鞍山 243002)

摘 要: 目前 Linux 作为一种源码开放、安全可靠的系统平台, 已成为嵌入式系统的首选操作系统。文中主要介绍嵌入式系统 Linux 下邮件收发系统的设计与开发。该邮件系统按照 SMTP 和 POP3 协议的规范, 实现邮件系统的收发功能, 包括邮件服务器和邮件客户端两部分。邮件服务器实现邮件的保存和传送功能。客户端可以执行邮件编辑、发送邮件和接收邮件等操作。该系统的设计坚持以方便用户为原则, 它的逻辑清晰、界面简单友好, 便于使用。

关键词: Linux; 邮件系统; 邮局协议(第3版); 简单邮件传输协议

中图分类号: TP393.098

文献标识码: A

文章编号: 1005-3751(2006)01-0154-04

The Design of Mail Server in Embedded System of Linux

YUAN Zhi-xiang, QIN Feng, WANG Xiao-lin

(School of Computer, Anhui University of Technology, Maanshan 243002, China)

Abstract: Linux is used as a kind of source code opened and safe system platform, it had used to the OS of embedded system firstly. This paper is a mail system that was developed with C language under the Linux environment. According to the protocol of SMTP and the protocol of POP3, this mail system realized the functions of mail system. It includes the mail server and the mail client. The mail server can save and send the mails; the client can edit mails, send mails and receive mails. The design of this system persisted that it brought convenience to the users. Its clear logic and the friendly interface made the users use it very easily.

Key words: Linux; mail system; POP3; SMTP

0 引言

Linux 操作系统自 1991 年诞生以来, 因其卓越的性能、开放的源代码、高效的安全性已成为 Windows 操作系统的强有力竞争者, 在服务器领域的市场占有率已接近 Windows 产品, 同时在个人计算机上越来越普及。目前网络家电、手机、PDA 以及智能化仪器仪表等网络设备同样需要操作系统的支持, 如何将 Linux 系统引入嵌入式系统中是当前操作系统和嵌入式系统研究的一个重点领域^[1,2]。目前许多 Linux 系统都带有诸如 sendmail 邮件系统, 但是都不适用嵌入式系统, 因为其硬件要求较高。因此文中的主要工作是在 Linux 环境下利用 C 语言开发一个邮件收发系统, 主要是利用 SMTP 和 POP3 协议实现邮件服务器和邮件客户端软件。文中主要从 3 个方面, 即邮件系统的理论知识、系统分析、系统设计展开。其中邮件系统的理论知识都有 RFC 文档规范^[3-6], 故不再详细阐述。

1 相关理论知识

1.1 Socket 通信流程

Internet 网络的核心是 TCP/IP。在对 Internet 的网络应用系统的开发中, Linux 系统提供了套接字(Socket)机制。Socket 是 TCP/IP 网络协议的应用程序接口(API), 通过 Socket 访问低层的网络协议, 它可以访问多种通信协议, 实现人们所需的网络功能。在嵌入式系统中, 主要是利用将 TCP/IP 的协议栈放入网络设备的 BOOT ROM 中, 或者直接放入 Linux 的内核中, 无论哪种方法, 对于嵌入式系统的网络应用软件的开发思路是一致的。

对 Socket 编程要熟悉 TCP/IP 网络协议的通信机制, 如图 1 所示。关于其具体的 Socket 网络编程的常用函数和流程可参考文献[7,8]。

1.2 电子邮件的有关协议

为了规范电子邮件的发展, 出现了一系列相应协议, 它们由一组 RFC (Request For Comment) 文档进行定义, 电子邮件的核心结构在 RFC822^[5]和 RFC1123^[6]这两个文档中定义。

上述 RFC 文档是对电子邮件格式和结构的定义, 另外还有一系列对电子邮件的传输、访问等进行规范, 最主要的有:

(1) 简单邮件传输协议^[3] (Simple Mail Transfer Proto-

收稿日期: 2005-04-23

基金项目: 安徽省教育厅青年教师资助计划(2004j9128); 安徽省教育厅自然科学基金(2004kj062, 2005kj070, 2005kj071)

作者简介: 袁志祥(1973—), 男, 安徽当涂人, 讲师, 主要研究方向为 petri 网理论、计算机软件理论和网络应用技术。

ool,SMTP),在 RFC821 中定义。

(2) 邮局协议^[4](Post Office Protocol,目前是第 3 版,称为 POP3),RFC1939 是其最新的定义文档。

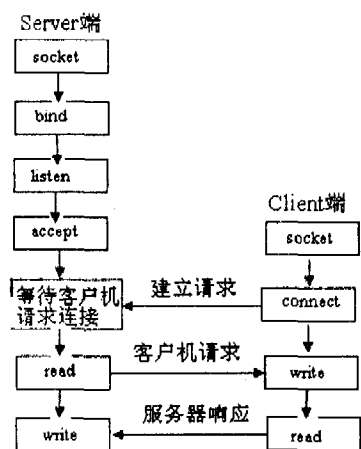


图 1 TCP 服务器/客户机通信模型

2 系统描述和分析

本邮件程序由两大部分组成:邮件服务器程序和邮件客户端程序。邮件服务器程序负责监听来自客户端的连接请求和接收客户端发送的邮件信息。它维护一个包含邮件用户信息的文件和一些包含用户邮件内容的文件,通过对这些文件的操作对邮件进行存取。邮件客户端程序则负责建立和维护与服务器的连接,它又包括邮件接收程序和邮件发送程序。邮件发送程序向服务器发送本客户端的邮件信息,以便邮件服务器可以将邮件传送到邮件接收程序;邮件接收程序负责接收和显示从服务器发送来的邮件消息,当然这里的邮件是别的邮件用户发送到邮件服务器的,它们控制着服务器的每一步操作。

本系统主要实现 Linux 操作系统环境下的邮件收发,所以整个系统是利用 Linux 环境下的 C 语言编程实现,其中客户端界面是利用 GTK+ 库^[9]编写的。

本邮件系统收发系统的作用是将邮件从一台主机经过服务器发送到另一台主机。它主要分为两个部分:邮件服务器和邮件客户端。邮件服务器端监听客户端的连接请求和接收客户端的发送和接收请求,然后根据客户端的请求执行相应的操作。邮件客户端负责向服务器发送邮件还有接收和显示服务器发送来的邮件,它控制着邮件服务器的各个操作。它们的基本模型如图 2 所示。

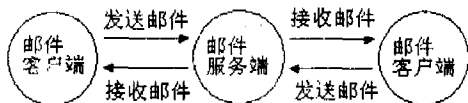


图 2 邮件客户端和服务端模型

本邮件收发系统的工作流程如下:

1) 服务器端建立一个套接字(Socket),然后监听客户端的连接请求。

2) 客户端建立一个 Socket,然后向邮件服务器发送连接请求。

3) 服务器接收请求,并建立一个子进程,用于接收客户端发送的数据。

4) 客户端向服务器端发送信息,控制邮件服务器进行各种操作,这些信息是以命令的形式发送的,通过这些命令将邮件发送到邮件服务器或从服务器接收邮件。

5) 邮件服务器对客户端的命令进行响应,如客户端是发送邮件则将发送的邮件保存到文件中;如果客户端是接收邮件,则将用户文件发送到客户端。

6) 客户端显示接收的邮件。

3 邮件收发系统设计

邮件收发系统的作用是把信件从一个地方传送到另外一个地方。它由信件、把信件从一个地方传送到另外一个地方的协议和用户执行各种相关的接口组成。从前面的基础知识中知道,从处理的角度看一个邮件系统分为 3 个部分:MTA(邮件传送代理)、MDA(邮件投递代理)、MUA(邮件用户代理)。其中可以将其简化成两种接口:一种接口是帮助用户完成邮件的编写,既 MUA(邮件用户代理),另外它还负责显示用户回收的邮件;另一种接口是负责邮件在网络上的传输。

因为邮件是在网络上传输的,而目前互联网络上基本上都是使用 TCP/IP 标准,所以嵌入式系统的在这类网络上传送邮件,通常用 SMTP(简单邮件传输协议)。SMTP 协议会在 SMTP 服务器和发信人的计算机之间建立连接,以使邮件从发信人客户机发送到邮件服务器。在邮件服务器接收到邮件后,收信人计算机必须能够收到邮件才可以,这时又将用到 POP3 协议(邮局传输协议),它使邮件服务器上的信件传送到目的计算机。本邮件收发系统中邮件的收发机制如图 3 所示。

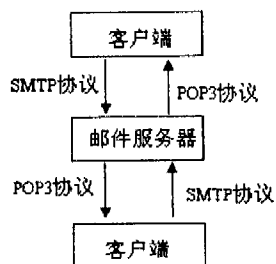


图 3 邮件收发机制

另外还要涉及到的一个问题就是邮件由客户端发送到邮件服务器上以后,需要对邮件进行存储,当接收邮件客户端接收邮件时必须将邮件从存储的位置取出,以发送到客户端。由于本系统是在嵌入式系统下实现基本邮件收发功能的邮件收发系统,所以在本系统中采用文件系统来代替数据库,利用文件系统来存储用户名、密码等用户信息和用户的邮件,邮件服务器程序对这些文件进行写入、读取、删除等操作,以获得对邮件进行存取的功能。

综上所述,本邮件收发系统应包括两大模块:客户端模块和服务器模块。邮件客户端模块包括配置服务器、接收邮件、发送邮件、编辑邮件和连接这几个子模块。邮件

服务端包括接收子模块、发送子模块、连接子模块。

图 4 为系统的程序模块图:

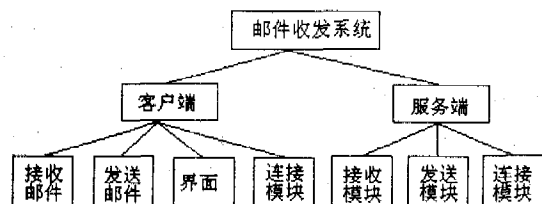


图 4 程序模块图

以上是对本邮件收发系统较为概括的分析,基本介绍了本邮件收发系统的总体设计,以下将介绍本邮件收发系统各模块的设计。这里仅仅列出邮件服务器连接模块设计的核心代码,需要更详尽的代码和资料请与作者联系。

3.1 服务器模块设计

在整个邮件收发系统中,最重要的部分就是服务器部分,它负责接收邮件客户端发出的命令请求,然后做出相应的处理,并负责保存用户发出的邮件。它可以分为 SMTP 服务器和 POP3 服务器,SMTP 服务器用于接收和处理用户发送邮件的相应操作,POP3 服务器用于接收和处理用户接收邮件的相应操作。其流程如图 5 所示。

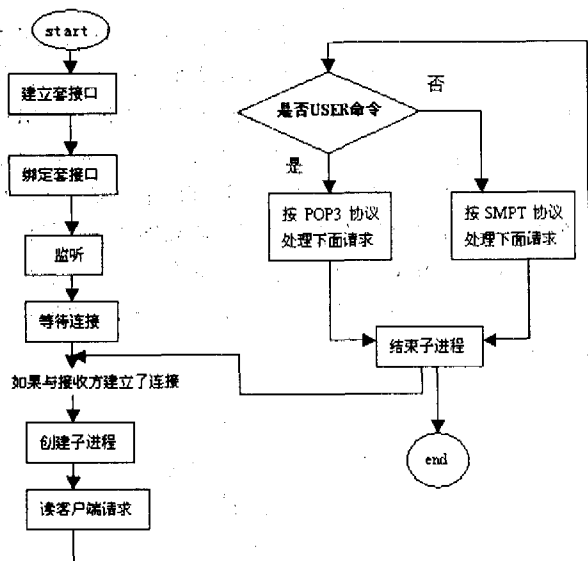


图 5 服务器端流程图

3.2 连接模块设计代码解释

1) 创建套接字(Socket)。

本设计中套接字的创建都是基于 Linux 下的 C 语言网络编程实现的,调用其中的 Socket 函数来创建 Socket,此函数包含一些参数,输入函数的不同会产生不同的结果。本设计中一般用如下格式:

```
sockfd = socket(AF_INET, SOCK_STREAM, 0)
```

其中 AF_INET 参数表示是一个 IPv4 协议,而 SOCK_STREAM 参数表示要创建的是一个流式套接字,此函数返回一个整数值,在此程序中将它赋给变量 sockfd (就相当于一个文件描述符,如果创建不成功则返回值是 -1)。

2) 绑定地址和端口。

为了使其它进程能够使用所建立的套接字,必须给套接字一个名字。对于 AF_INET 套接字,它必须与一个端口联系起来。本程序中对套接字的绑定如下:

```
bind(sockfd, (struct sockaddr *)&serv_addr,
      sizeof(struct sockaddr))
```

其中, sockfd 就是刚建立的套接字函数的返回值, serv_addr 是一种套接字地址数据结构,可以给它赋值,以确定要绑定的端口。本程序中对它的赋值为:

```
serv_addr.sin_family = AF_INET;
serv_addr.sin_port = htons(POP3SERVPORT);
serv_addr.sin_addr.s_addr = INADDR_ANY;
```

POP3SERVPORT 是程序中定义的一个宏,它表示要绑定的端口,本系统中它定义为 1980。INADDR_ANY 表示绑定的地址为任意地址,也可指定为一个 IP 地址。

3) 等待连接。

一般的,服务器程序要在一个循环中处理连接,对于每一个连接,它建立一个进程处理连接。本程序中也不例外,也是在程序中设置一个循环,然后在其中等待着客户端的连接,与客户端连接后,将建立一个子进程以处理与客户端的连接。本设计中的实现过程请见以下代码:

```
while(1){
    sin_size = sizeof(struct sockaddr_in);
    if((client_fd = accept(sockfd, (struct sockaddr *)&client_addr, &sin_size)) == -1){
        perror("accept error!");
        exit(1);
    }
    if((pid = fork()) == 0){
        close(sockfd);
        .....(处理连接)
        close(client_fd);
        exit(0);
    }
    close(client_fd);
}
```

4) 关闭套接字。

每一个套接口在用过后,都应该关闭,否则浪费资源。

关闭用 close() 函数。

```
close(sockfd);
```

Sockfd 为建立套接字的返回值。

4 结论与展望

本项目是在嵌入式系统 Linux 下实现的邮件收发系统,主要利用 SMTP 和 POP3 协议实现邮件服务器和邮件客户端的应用软件的设计。邮件客户端又分为发送部分和接收部分,用户可以将编辑好的邮件通过邮件发送部分发送到邮件服务器,也可以从邮件服务器接收自己的邮件。目前该系统已在嵌入式系统上实验演示,其运行界面

如图 6 所示。

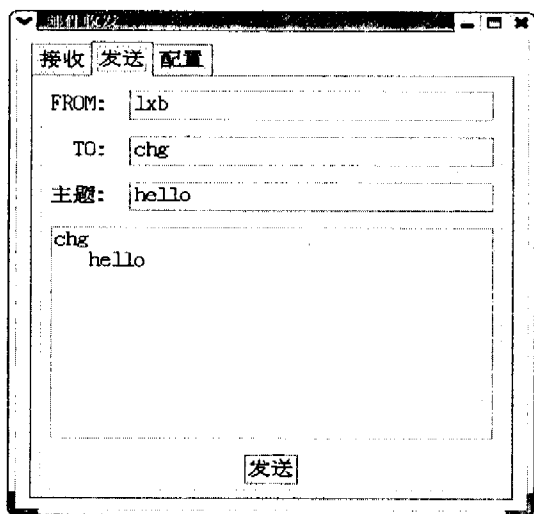


图 6 运行界面

基于 Linux 的邮件服务器其实有很多而且功能都非常强大,比如 Sendmail、Qmail 等,但是为了将其移植到嵌入式系统中需要做许多工作,其重点内容是研究邮件服务器的核心内容,如何在嵌入式系统中具体应用,而不是简单的二次开发。本系统的特点是基于 Linux 操作系统,兼具邮件服务器端和邮件客户端,另外邮件客户端还提供一个图形界面,方便了用户的操作。

本系统虽然在功能上实现了邮件的收发,但还存在着

(上接第 134 页)

等。通过利用元数据信息和分类信息将大大提高信息资源查找效率。

(3)信息检索实现。通过 Web 服务器端软件,为用户提供树型分类目录以及信息查询的界面。采用语义分析相匹配的方法实现概念检索,当输入自然语言进行检索时,分词处理是关键,主要是获取分词结果中的对象的语义,在查询请求和该对象在语义内容上的匹配和比较的过程中进行判断,检索过程将对待检索的对象进行语义分析,并与该对象所属分类的属性进行比较,得出该对象的判断结果并返回给用户。基于 Ontology 可以将同义词扩展检索、语义蕴涵、外延扩展检索、语义相关扩展检索等技术与信息检索结合起来,提供交互式的智能化信息检索服务,而信息检索的结果(如网页)可以作为知识检索的背景内容。

4 结束语

信息检索的分布化和网络化、开放性和集成性要求越来越高,使得要采用相应的信息组织方式和信息检索技术来检索和整合不同来源和结构的信息。信息检索在互联网信息日益增长的情况下起着非常重要的作用。文中研究了信息组织和信息检索技术,并针对部分存在的缺点,

一些缺点,诸如系统功能还不够完善,安全问题的解决、媒体文件的传输以及适用于嵌入式系统的后台数据库的研究,有待进一步完善。笔者将继续对本系统进行修改和完善,以实现在网络家电和智能化仪器仪表上的应用。

参考文献:

- [1] 刘文峰,李程远,李善平. 嵌入式 Linux 操作系统的研究[J]. 浙江大学学报(工学版), 2004, 38(4): 447-452.
- [2] 陈闯中. Linux 在嵌入式操作系统中的应用[J]. 同济大学学报(自然科学版), 2001, 29(5): 564-566.
- [3] Postel J B. RFC821[DB/OL]. <http://www.ietf.org>, 1982-08.
- [4] Myers J. RFC1939[DB/OL]. <http://www.ietf.org>, 1996-05.
- [5] Crocker D H. RFC822[DB/OL]. <http://www.ietf.org>, 1982-08.
- [6] Braden R. RFC1123[DB/OL]. <http://www.ietf.org>, 1989-10.
- [7] 于明俭,陈向阳,方 汉. Linux 程序设计权威指南[M]. 北京:机械工业出版社, 2001.
- [8] 周良源. UNIX 平台下的 C 编程指南[M]. 北京:电子工业出版社, 2000.
- [9] Wright P. GTK+ /GNOME 程序设计[M]. 钟 鸣,石永平译. 北京:机械工业出版社, 2002.

进行了改进。

参考文献:

- [1] 张 帆. 信息存储与检索[M]. 北京:高等教育出版社, 2003.
- [2] Studer R, Benjamins V R, Fensel D. Knowledge Engineering, Principles and Methods[J]. Data and Knowledge Engineering, 1998, 25(1-2): 161-197.
- [3] Lassila O, Swick R. Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation[EB/OL]. <http://www.w3.org/TR/REC-rdf-syntax>, 1999-02-22.
- [4] Park J, Hunting S. XML Topic Maps: creating and using Topic Maps for the web[M]. New York: Prentice, 2003.
- [5] Patel - Schneider P F, Hayes P. OWL[EB/OL]. <http://www.w3.org/TR/2003/CR-owl-semantics-20030818/>, 2003.
- [6] Guarino N, Welty C. Towards a methodology for ontology-based model engineering. In Proceedings of the ECOOP - 2000 Workshop on Model Engineering[EB/OL]. <http://www.ladseb.pd.cnr.it/infor/ontology/Papers/OntologyPapers.html>, 2000.