



中国石油大学(华东)  
CHINA UNIVERSITY OF PETROLEUM

## 互联网邮件收发程序设计

学生姓名：张世琛

学 号：1804030401

专业班级：计科 1802

学 院：计算机科学与技术学院

实验成绩	
教师评阅	

2020 年 11 月 20 日

## 1 实验目的

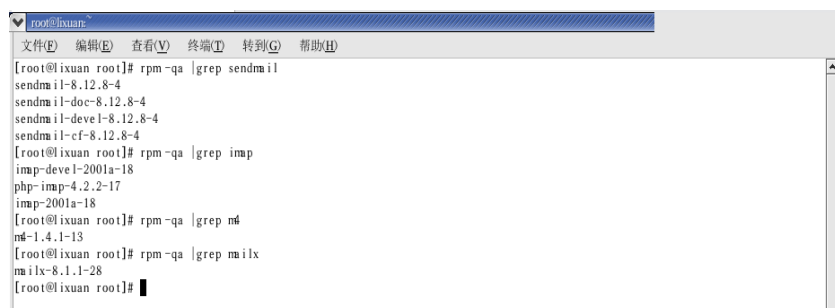
1. 熟练掌握通过 Linux Socket 编程实现 SMTP、POP3 命令字的交互。
2. 开发出收发邮件的程序。
3. 设计邮件收发程序的流程图。

## 2 环境要求

1. 硬件要求:  
PC 机;
2. 软件环境:  
VMware 虚拟机;Linux 操作系统;MAIL 服务器环境;vi 编辑器或 gedit 文本编辑器;gcc 编译环境;

## 3 实验内容

### 3.1 查看 MAIL 服务器的安装状态, 当前系统已安装 sendmail, pop3, imap, m4, mailx。



```
root@ixuan:~  
[root@ixuan root]# rpm -qa |grep sendmail  
sendmail-8.12.8-4  
sendmail-doc-8.12.8-4  
sendmail-devel-8.12.8-4  
sendmail-cf-8.12.8-4  
[root@ixuan root]# rpm -qa |grep imap  
imap-devel-2001a-18  
php-imap-4.2.2-17  
imap-2001a-18  
[root@ixuan root]# rpm -qa |grep m4  
m4-1.4.1-13  
[root@ixuan root]# rpm -qa |grep mailx  
mailx-8.1.1-28  
[root@ixuan root]#
```

图 1: 修改 ipop3 的文件

### 3.2 在 Linux 环境下配置启动 MAIL 服务器, 测试运行。

1. 设置虚拟机的网络连接, 如图所示

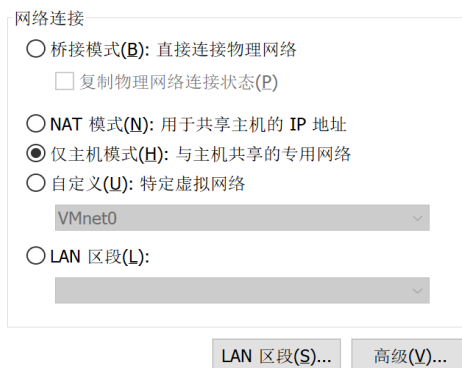


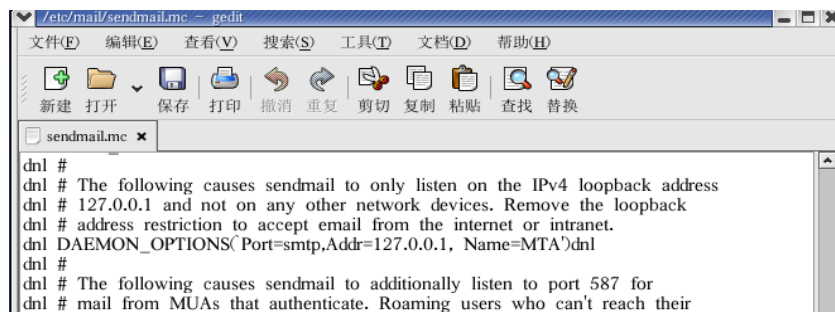
图 2: 虚拟机网络连接设置

## 2. 设置本地网络连接，如图所示



图 3: 设置本地网络连接

## 3. 修改 sendmail.mc 的文件



#### 4. 修改 ipop3 的文件

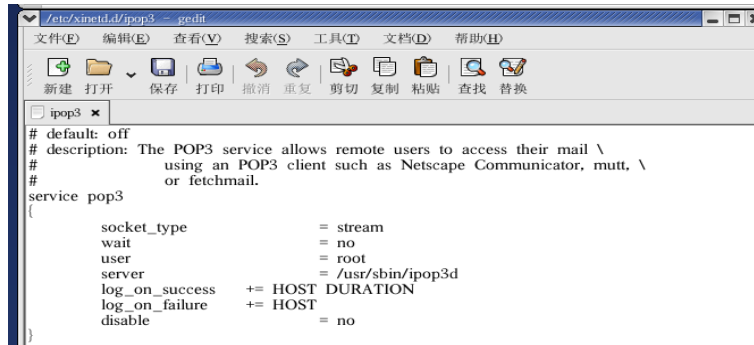


图 5: 修改 ipop3 的文件

#### 5. 输入 `m4 /etc/mail/sendmail.mc>/etc/mail/sendmail.cf`、`/etc/rc.d/init.d/sendmail restart`、`/etc/rc.d/init.d/senfmil restart`、`/etc/rc.d/init.d/xinetd restart` 命令

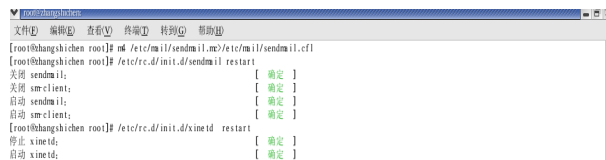


图 6: 修改 ipop3 的文件

#### 6. 检查 pop3 和 smtp 状态



图 7: 检查 pop3 和 smtp 状态

#### 7. 在 root 用户下使用 mail 服务器发送邮件给 `zhangshichen@zhangshichen.com` 和 `zhangshichen1@zhangshichen.com` 发送邮件，其中 `zhangshichen1@zhangshichen.com` 为抄送，如下图所示：

```
root@zhangshichen~  
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)  
[root@zhangshichen root]# mail  
Mail version 8.1 6/6/93. Type ? for help.  
"/var/spool/mail/root": 4 messages 4 new  
>N 1 Disk.Usage.Monitor.- Mon Nov 21 20:01 18/769 "Low disk space warnin"  
N 2 Disk.Usage.Monitor.- Mon Nov 21 21:01 18/769 "Low disk space warnin"  
N 3 root@localhost.local Mon Nov 21 21:01 19/841 "Anacron job 'cron.dai'"  
N 4 Disk.Usage.Monitor.- Thu Nov 24 16:01 18/769 "Low disk space warnin"  
&q  
Hit 4 messages in /var/spool/mail/root  
[root@zhangshichen root]# mail zhangshichen@zhangshichen.com  
Subject: Amber  
1234567890  
.  
Cc: zhangshichen1@zhangshichen.com  
[root@zhangshichen root]#
```

图 8: root 用户给 `zhangshichen@zhangshichen.com` 和 `zhangshichen1@zhangshichen.com` 发送邮件

## 8. zhangshichen@zhangshichen.com 收到 root 发来的邮件

```
zhangshichen@zhangshichen~  
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)  
[zhangshichen@zhangshichen zhangshichen]$ mail  
Mail version 8.1 6/6/93. Type ? for help.  
"/var/spool/mail/zhangshichen": 1 message 1 new  
>N 1 root@zhangshichen.co Tue Nov 17 20:02 17/615 "Amber"  
&l  
Message 1:  
From root@zhangshichen.com Tue Nov 17 20:02:01 2020  
Date: Tue, 17 Nov 2020 20:01:52 +0800  
From: root <root@zhangshichen.com>  
To: zhangshichen@zhangshichen.com  
Subject: Amber  
Cc: zhangshichen1@zhangshichen.com  
1234567890  
.
```

图 9: `zhangshichen@zhangshichen.com` 收到 root 发来的邮件

## 3.3 使用文本编辑器编辑 SMTP 邮件发送源程序以及 POP3 接收邮件源程序,并使用 GCC 编译两个源程序分别生成可执行程序。

### 3.3.1 程序代码

```
1 smtp代码  
2 #include<stdio.h>  
3 #include<stdlib.h>  
4 #include<errno.h>  
5 #include<string.h>  
6 #include<netdb.h>  
7 #include<sys/types.h>  
8 #include<netinet/in.h>  
9 #include<sys/socket.h>  
10 #include<stdbool.h>  
11 #include<linux/tcp.h>  
12  
13 #define POP3SERVPORT 25  
14 #define MAXDATASIZE 4096  
15 #define TRUE 1  
16  
17 main(int argc, char *argv[]) {  
18     int sockfd;  
19     struct hostent *host;
```

```

20 struct sockaddr_in serv_addr;
21 char *POPMessage[] = {
22     "helo\r\n",
23     "mailfrom:<root@zhangshichen.com>\r\n",
24     "rcptto:<zhangshichen@zhangshichen.com>\r\n",
25     "data\r\n",
26     "This is a test\r\n.\r\n",
27     "QUIT\r\n",
28     NULL
29 };
30 int iLength = 0;
31 int iMsg = 0;
32 char buf[MAXDATASIZE];
33 if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
34     perror("socketerror");
35     exit(1);
36 }
37 serv_addr.sin_family = AF_INET;
38 serv_addr.sin_port = htons(POP3SERVPORT);
39 serv_addr.sin_addr.s_addr = inet_addr("192.168.137.100");
40 bzero(&(serv_addr.sin_zero), 8); /*置指针变量(&(serv_addr.sin_zero)中前8个变量的值为零*/
41 if (connect(sockfd, (struct sockaddr *) &serv_addr, sizeof(struct sockaddr)) == -1) {
42     perror("connecterror");
43     exit(1);
44 }
45 iLength = recv(sockfd, buf, sizeof(buf), 0);
46 buf[iLength] = '\0';
47 printf("received:%s\n", buf); //依次发送SMTP命令, 发送邮件
48 do {
49     bool bNodelay = TRUE;
50     /*bool型变量只有两个值: false和true, 是0和1的区别*/
51     setsockopt(sockfd, IPPROTO_TCP, TCP_NODELAY, (const char *) &bNodelay, sizeof(bool))
52         ;
53     send(sockfd, POPMessage[iMsg], strlen(POPMessage[iMsg]), 0);
54     printf("havesent: %s", POPMessage[iMsg]);
55     iLength = recv(sockfd, buf, sizeof(buf), 0);
56     buf[iLength] = '\0';
57     iMsg++;
58     printf("received:%s,%d\n", buf, iMsg);
59 } while (POPMessage[iMsg]);
60 close(sockfd);
61 }

```

```

1 pop3代码
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <errno.h>
5 #include <string.h>
6 #include <netdb.h>

```

```

7  #include <sys/types.h>
8  #include <netinet/in.h>
9  #include <sys/socket.h>
10 #include <stdbool.h>
11 #include <linux/tcp.h>
12
13 #define POP3SERVPORT 110
14 #define MAXDATASIZE 1000
15 #define TRUE 1
16
17 main(int argc, char *argv[]) {
18     int sockfd;
19     struct hostent *host;
20     struct sockaddr_in serv_addr;
21     char *POPMessage[] = {
22         "USER test\r\n",
23         "PASS 123456\r\n",
24         "STAT\r\n",
25         "LIST\r\n",
26         "RETR 1\r\n",
27         "QUIT\r\n",
28         NULL};
29     int iLength;
30     int iMsg = 0;
31     char buf[MAXDATASIZE];
32     if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
33         perror("socket error");
34         exit(1);
35     }
36     serv_addr.sin_family = AF_INET;
37     serv_addr.sin_port = htons(POP3SERVPORT);
38     serv_addr.sin_addr.s_addr = inet_addr("192.168.137.100");
39     bzero(&(serv_addr.sin_zero), 8);
40     if (connect(sockfd, (struct sockaddr *) &serv_addr, sizeof(struct sockaddr)) == -1) {
41         perror("connect error");
42         exit(1);
43     }
44     iLength = recv(sockfd, buf, sizeof(buf), 0);
45     buf[iLength] = '\0';
46     printf("received:%s\n", buf);
47     do {
48         send(sockfd, POPMessage[iMsg], strlen(POPMessage[iMsg]), 0);
49         printf("have sent:%s", POPMessage[iMsg]);
50         iLength = recv(sockfd, buf, sizeof(buf), 0);
51         buf[iLength] = '\0';
52         printf("received: %s,%d\n", buf, iMsg);
53         if (iMsg == 5)
54             iLength = recv(sockfd, buf, sizeof(buf), 0);
55         buf[iLength] = '\0';

```

```

56     printf("%s\n", buf);
57     iMsg++;
58 } while (POPMessage[iMsg]);
59 close(sockfd);
60 }

```

### 3.3.2 使用文本编辑器编辑 SMTP 邮件发送源程序以及 POP3 接收邮件源程序

```

main(int argc, char *argv[]) {
    int sockfd;
    struct hostent *host;
    struct sockaddr_in serv_addr;
    char *POPMessage[] = {
        "helo\n",
        "mailfrom:<root@zhangshichen.com>\n\n",
        "rcptto:<test@zhangshichen.com>\n\n",
        "data\n",
        "This is a test\n\n.\n",
        "QUIT\n",
        NULL
    };
    int iLength = 0;
    int iMsg = 0;
    char buf[MAXDATASIZE];
    if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("socketerror");
        exit(1);
    }
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(POP3SERVPORT);
    serv_addr.sin_addr.s_addr = inet_addr("192.168.137.100");
    bzero(&(serv_addr.sin_zero), 8);/*置指针变量(&(serv_addr.sin_zero)中前8个变量的值为零*/if (
        connect(sockfd, (struct sockaddr *) &serv_addr, sizeof(struct sockaddr)) == -1) {
        perror("connecterror");
        exit(1);
    }
    iLength = recv(sockfd, buf, sizeof(buf), 0);
    buf[iLength] = '\0';
    printf("received:%s\n", buf);//依次发送SMTP命令，发送邮件
    do {
        bool bNodelay = TRUE;
        /*bool型变量只有两个值：false和true，是0和1的区别*/
        setsockopt(sockfd, IPPROTO_TCP, TCP_NODELAY, (const char *) &bNodelay, sizeof(bool));
        send(sockfd, POPMessage[iMsg], strlen(POPMessage[iMsg]), 0);
        printf("havesent: %s", POPMessage[iMsg]);
        iLength = recv(sockfd, buf, sizeof(buf), 0);
        buf[iLength] = '\0';
        iMsg++;
        printf("received:%s,%d\n", buf, iMsg);
    } while (POPMessage[iMsg]);
    close(sockfd);
}

```

行 1, 列 1 插入

图 10: 编写 smtp 程序



```

main(int argc, char *argv[]) {
    int sockfd;
    struct hostent *host;
    struct sockaddr_in serv_addr;
    char *POPMessage[] = {
        "USER test\r\n",
        "PASS 123456\r\n",
        "STAT\r\n",
        "LIST\r\n",
        "RETR 1\r\n",
        "QUIT\r\n",
        NULL};

    int iLength;
    int iMsg = 0;
    char buf[MAXDATASIZE];
    if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("socket error");
        exit(1);
    }

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(POP3SERVPORT);
    serv_addr.sin_addr.s_addr = inet_addr("192.168.137.100");
    bzero(&(serv_addr.sin_zero), 8);
    if (connect(sockfd, (struct sockaddr *) &serv_addr, sizeof(struct sockaddr)) == -1) {
        perror("connect error");
        exit(1);
    }

    iLength = recv(sockfd, buf, sizeof(buf), 0);
    buf[iLength] = '\0';
    printf("received:%s\n", buf);
    do {
        send(sockfd, POPMessage[iMsg], strlen(POPMessage[iMsg]), 0);
        printf("have sent:%s", POPMessage[iMsg]);
        iLength = recv(sockfd, buf, sizeof(buf), 0);
        buf[iLength] = '\0';
        printf("received: %s,%d\n", buf, iMsg);
        if (iMsg == 5)
            iLength = recv(sockfd, buf, sizeof(buf), 0);
        buf[iLength] = '\0';
        printf("%s\n", buf);
        iMsg++;
    } while (POPMessage[iMsg]);
    close(sockfd);
}

```

图 11: 编写 pop3 程序

### 3.3.3 并使用 GCC 编译两个源程序分别生成可执行程序, 并运行。

```

root@zhangshichen ~
[root@zhangshichen root]# gcc smtp.c -o smtp
[root@zhangshichen root]# ./smtp
received:220 zhangshichen.com ESMTP Sendmail 8.12.8/8.12.8; Tue, 17 Nov 2020 22:
49:16 +0800
have sent: helo
received:501 5.0.0 helo requires domain address
,1
have sent: mailfrom:root@zhangshichen.com
received:500 5.5.1 Command unrecognized: "mailfrom:root@zhangshichen.com"
,2
have sent: reptto:zhangshichen@zhangshichen.com
received:500 5.5.1 Command unrecognized: "reptto:zhangshichen@zhangshichen.com"
,3
have sent: data
received:503 5.0.0 Need MAIL command
,4
have sent: This is a test
received:500 5.5.1 Command unrecognized: "This is a test"
500 5.5.1 Command unrecognized: "."
,5
have sent: QUIT
received:221 2.0.0 zhangshichen.com closing connection
,6

```

图 12: 运行 smtp 程序结果

```
root@1808010204:~
文件(E)  编辑(E)  查看(V)  终端(T)  转到(G)  帮助(H)

[root@1808010204 root]# gedit pop3.c
[root@1808010204 root]# gcc pop3.c -o pop3
[root@1808010204 root]# ./pop3
received:+OK POP3 1808010204.com v2001.78rh server ready

have sent:USER test1
received: +OK User name accepted, password please
,0
+OK User name accepted, password please

have sent:PASS 123456
received: +OK Mailbox open, 2 messages
,1
+OK Mailbox open, 2 messages

have sent:STAT
received: +OK 2 1160
,2
+OK 2 1160

have sent:LIST
received: +OK Mailbox scan listing follows
1 577
2 583
.
,3
+OK Mailbox scan listing follows
1 577
2 583
.

have sent:RETR 1
received: +OK 577 octets
Return-Path: <root@1808010204.com>
Received: from 1808010204.com (localhost.localdomain [127.0.0.1])
        by 1808010204.com (8.12.8/8.12.8) with ESMTp id 0AJBhDxR004199
        for <test1@1808010204.com>: Thu, 19 Nov 2020 19:43:13 +0800
Received: (from root@localhost)
        by 1808010204.com (8.12.8/8.12.8/Submit) id 0AJBhDAc004197
        for test1@1808010204.com: Thu, 19 Nov 2020 19:43:13 +0800
Date: Thu, 19 Nov 2020 19:43:13 +0800
From: root <root@1808010204.com>
Message-Id: <202011191143.0AJBhDAc004197@1808010204.com>
To: test1@1808010204.com
Subject: 123
Status: O

12
21
.
,4
+OK 577 octets
Return-Path: <root@1808010204.com>
Received: from 1808010204.com (localhost.localdomain [127.0.0.1])
        by 1808010204.com (8.12.8/8.12.8) with ESMTp id 0AJBhDxR004199
        for <test1@1808010204.com>: Thu, 19 Nov 2020 19:43:13 +0800
Received: (from root@localhost)
        by 1808010204.com (8.12.8/8.12.8/Submit) id 0AJBhDAc004197
        for test1@1808010204.com: Thu, 19 Nov 2020 19:43:13 +0800
Date: Thu, 19 Nov 2020 19:43:13 +0800
From: root <root@1808010204.com>
Message-Id: <202011191143.0AJBhDAc004197@1808010204.com>
To: test1@1808010204.com
Subject: 123
Status: O

12
21
.

have sent:QUIT
received: +OK Sayonara
,5

[root@1808010204 root]#
```

图 13: 运行 pop3 程序结果

## 4 实验分析

### 4.1 SMTP 通信的三个阶段的过程

- (1) 连接建立，在发送主机的 SMTP 客户和接受主机的 SMTP 服务器之间建立，不使用中间的邮件服务器。
- (2) 邮件传送。
- (3) 连接释放：邮件发送完毕后，SMTP 释放 TCP 连接。

## 4.2 在电子邮件中，为什么需要使用 POP 和 SMTP 这两个协议？

SMTP 是用来发送电子邮件的协议，POP 是用来读取邮件的协议。

## 4.3 IMAP 与 POP 有什么区别？

POP3 协议允许电子邮件客户端下载服务器上的邮件，但是在客户端的操作（如移动邮件、标记已读等），不会反馈到服务器上。

IMAP 提供 webmail 与电子邮件客户端之间的双向通信，客户端的操作都会反馈到服务器上，对邮件进行的操作，服务器上的邮件也会做相应的动作。IMAP 整体上为用户带来更为便捷和可靠的体验。POP 更易丢失邮件或多次下载相同的邮件，但 IMAP 通过邮件客户端与 webmail 之间的双向同步功能很好地避免了这些问题。

## 4.4 使用 SMTP、POP 命令如何连接服务器？

SMTP 命令：telnet 邮件服务器名 25

POP 命令：telnet 邮件服务器名 110

## 4.5 发收邮件程序流程图

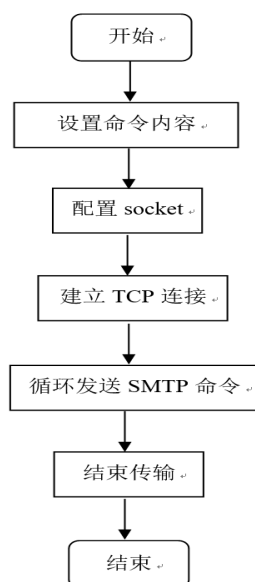


图 14: 发送邮件程序流程图

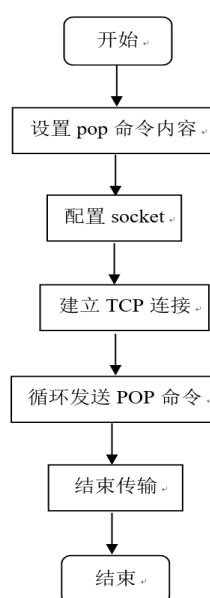


图 15: 接收邮件程序流程图

## 5 实验总结

通过本次实验，熟悉了 linux 操作系统，掌握了 linux 环境下配置 MAIL 服务器的方法，学会了使用 SMTP 和 POP 命令发送和接受邮件，了解了 Socket 编程，通过程序编写和实际操作更加理解了电子邮件收发的工作原理和通信过程。