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。

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

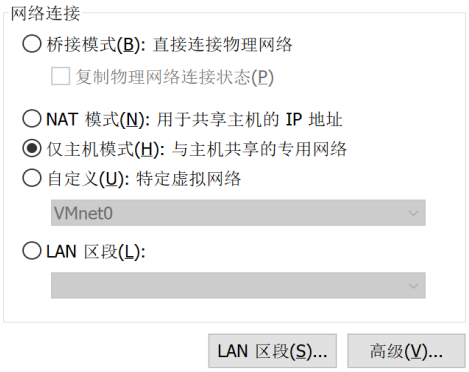

图 2: 虚拟机网络连接设置

2. 设置本地网络连接，如图所示

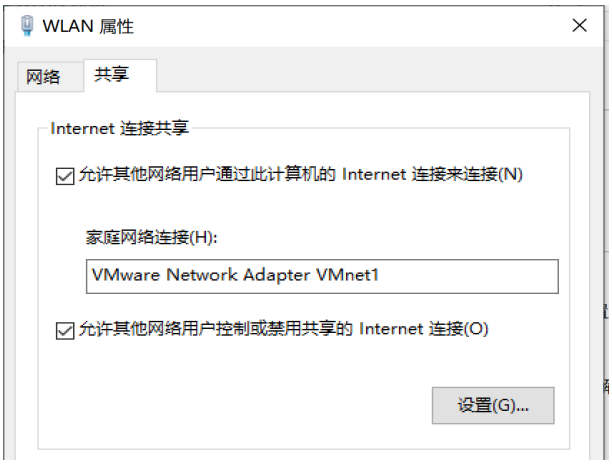

图 3: 设置本地网络连接

3. 修改 sendmail.mc 的文件

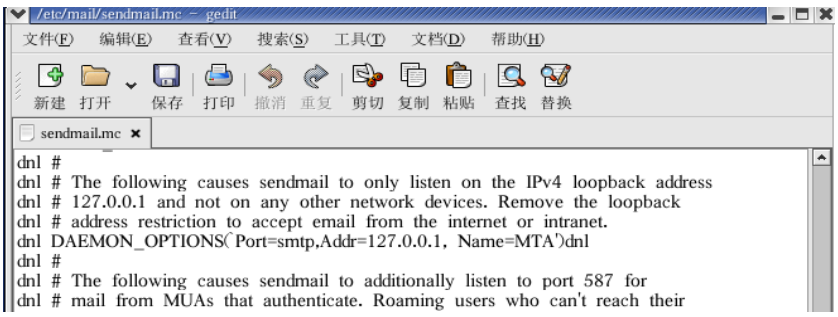

图 4: 修改 sendmail.mc 的文件

4. 修改 ipop3 的文件

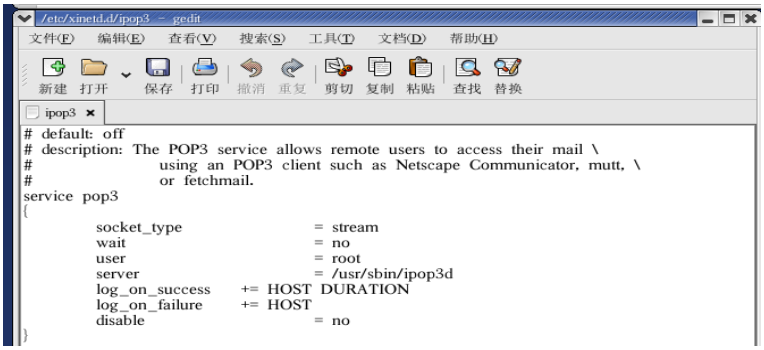

图 5: 修改 ipop3 的文件

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

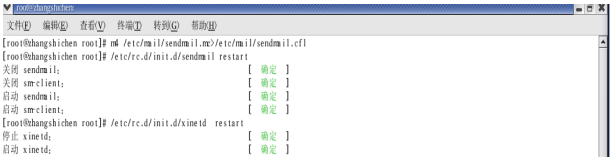

图 6: 修改 ipop3 的文件

6. 检查 pop3 和 smtp 状态

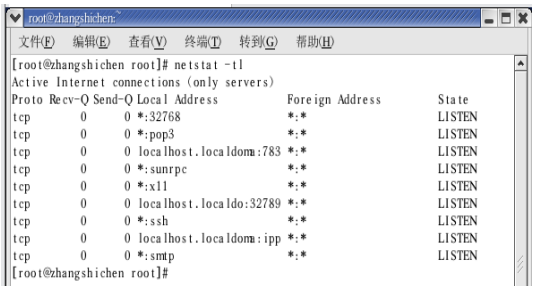

图 7: 检查 pop3 和 smtp 状态

7. 在 root 用户下使用 mail 服务器发送邮件给 *zhangshichen*@*zhangshichen.com* 和 *zhangshichen*1@*zhangshichen.com* 发送邮件，
其中 *zhangshichen*1@*zhangshichen.com* 为抄送，如下图所示：

图 8: root 用户给 *zhangshichen*@*zhangshichen.com* 和 *zhangshichen*1@*zhangshichen.com* 发送邮件

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

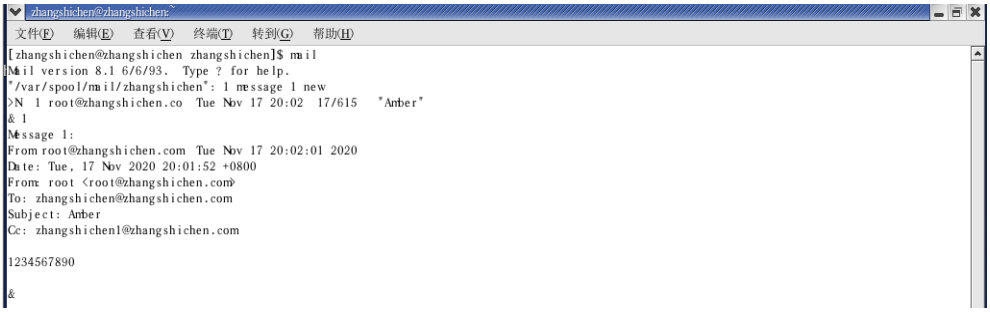

图 9: *zhangshichen*@*zhangshichen.com* 收到 root 发来的邮件

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

```
smtp代码
#include<stdio.h>
#include<stdlib.h>
#include<errno.h>
#include<string.h>
#include<netdb.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<sys/socket.h>
#include<stdbool.h>
#include<linux/tcp.h>

#define POP3SERVPORT 25
#define MAXDATASIZE 4096
#define TRUE 1

main(int argc, char *argv[]) {
   int sockfd;
   struct hostent *host;
```

```
    struct sockaddr_in serv_addr;
    char *POPMessage[] = {
          "helo\r\n",
          "mailfrom:<root@zhangshichen.com>\r\n",
          "rcptto:<zhangshichen@zhangshichen.com>\r\n",
          "data\r\n",
          "Thisis a test\r\n.\r\n",
          "QUIT\r\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SMPT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);
}
```

```
pop3代码
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
```

```c
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <stdbool.h>
#include <linux/tcp.h>

#define POP3SERVPORT 110
#define MAXDATASIZE 1000
#define TRUE 1

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';
```

```
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\r\n",
            "mailfrom:<root@zhangshichen.com>\r\n",
            "rcptto:<test@zhangshichen.com>\r\n",
            "data\r\n",
            "Thisis a test\r\n.\r\n",
            "QUIT\r\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SMPT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。

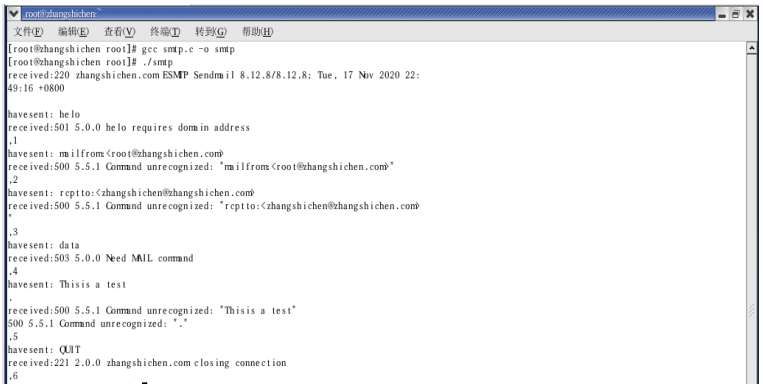

图 12: 运行 smtp 程序结果

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

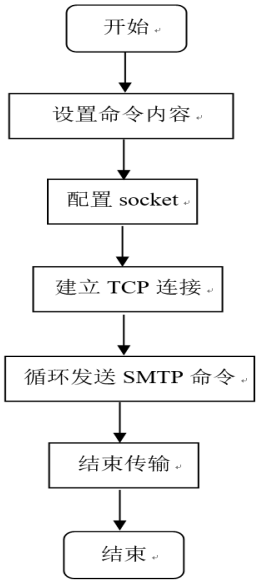

图 14: 发送邮件程序流程图

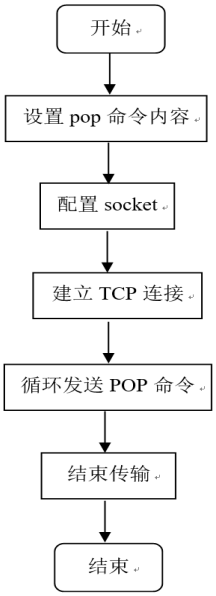

图 15: 接收邮件程序流程图

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。