



哈爾濱工業大學 (深圳)  
HARBIN INSTITUTE OF TECHNOLOGY

# 实验报告

开课学期: 2022 春季  
课程名称: 计算机网络  
实验名称: 邮件客户端的设计与实现  
学生班级: 1 班  
学生学号: 190110115  
学生姓名: 谢俊安  
评阅教师:  
报告成绩:

实验与创新实践教育中心制

2022 年 3 月

# 一、 实验详细设计

(注意不要完全照搬实验指导书上的内容，请根据你自己的设计方案来填写  
图文并茂地描述实验实现的所有功能和详细的设计方案及实验过程中的特色部分。)

## 1. 邮件发送客户端详细设计

- a) 获得域名的 IP (smtp.qq.com)
- b) 创建一个 socket, 建立一个 TCP 链接

```
// TODO: Create a socket, return the file descrip
struct sockaddr_in* servaddr;
servaddr->sin_family = AF_INET;
servaddr->sin_port = swap16(port);
bzero(servaddr->sin_zero, 8);
servaddr->sin_addr.s_addr = inet_addr(dest_ip);

s_fd = socket(AF_INET, SOCK_STREAM, 0);
connect(s_fd, servaddr, strlen(servaddr));
```

- c) 服务器发送欢迎消息
- d) 客户端发送 EHLO 命令表明身份, 服务器列出它支持的命令。

```
// TODO: Print server response to EHLO command
int recv1 = recv(s_fd, buf, MAX_SIZE, 0);
if (recv1 == -1)
    printf('Not received from server.');
```

```
else
    printf("%d", recv1);
```

- e) 客户端输入命令: AUTH login
- f) 服务器和客户端进行 username 的确认
- g) 服务器和客户端进行 username 对应的 password 的确认

```

// TODO: Authentication. Server response should be printed out.
const char* login = 'AUTH LOGIN\r\n';
send(s_fd, login, strlen(login), 0);
int rcv2 = recv(s_fd, buf, MAX_SIZE, 0);
printf("login: %d", rcv2);

const char* username = encode_str(user);
send(s_fd, username, strlen(username), 0);
int rcv3 = recv(s_fd, username, strlen(username), 0);
printf("user: %d", rcv3);
free(user);
const char* password = encode_str(pass);
send(s_fd, password, strlen(password), 0);
int rcv4 = recv(s_fd, buf, MAX_SIZE, 0);
printf("password: %d", rcv4);
free(pass);

```

- h) 客户端指定邮件的发送人和收件人

```

// TODO: Send MAIL FROM command and print server response
send(s_fd, from, strlen(from), 0);
int rcv5 = recv(s_fd, buf, MAX_SIZE, 0);
printf("mail from: %d", rcv5);

// TODO: Send RCPT TO command and print server response
send(s_fd, receiver, strlen(receiver), 0);
int rcv6 = recv(s_fd, buf, MAX_SIZE, 0);
printf("rcpt to: %d", rcv6);

```

- i) 服务器和客户端进行邮件内容的确认

```

// TODO: Send DATA command and print server response
const char* data = 'DATA\r\n';
send(s_fd, data, strlen(data), 0);
int rcv7 = recv(s_fd, buf, MAX_SIZE, 0);
printf("data: %d", rcv7);

// TODO: Send message data
strcat(msg, 'From: ');      strcat(msg, from);      strcat(msg, '\r\n');
strcat(msg, 'To: ');        strcat(msg, receiver);  strcat(msg, '\r\n');
strcat(msg, 'Subject: ');   strcat(msg, subject);   strcat(msg, '\r\n');
strcat(msg, 'Content: ');   strcat(msg, 'msg');      strcat(msg, '\r\n');
strcat(msg, 'Attach: ');    strcat(msg, att_path);   strcat(msg, '\r\n');

send(s_fd, msg, strlen(msg), 0);

```

- j) 客户端输入 “.” 表示邮件内容输入完毕，服务器提示成功。

```

// TODO: Message ends with a single period
send(s_fd, end_msg, strlen(end_msg), 0);
int rcv8 = recv(s_fd, buf, MAX_SIZE, 0);
printf("mail: %d", rcv8);

```

- k) 客户端输入 QUIT 命令断开与邮件服务器的连接，服务器提示连接中断。

```
// TODO: Send QUIT command and print server response
const char* QCommand = 'QUIT\r\n';
send(s_fd, QCommand, strlen(QCommand), 0);
int recv9 = recv(s_fd, buf, MAX_SIZE, 0);
printf("quit: %d", recv9);

close(s_fd);
```

## 2. 邮件接收客户端详细设计

- a) 获得域名的 IP (pop.163.com)  
b) 创建一个 socket，建立一个 TCP 链接

```
// TODO: Create a socket, return the file descrip
struct sockaddr_in* servaddr;
servaddr->sin_family = AF_INET;
servaddr->sin_port = swap16(port);
bzero(servaddr->sin_zero, 8);
servaddr->sin_addr.s_addr = inet_addr(dest_ip);

s_fd = socket(AF_INET, SOCK_STREAM, 0);
connect(s_fd, servaddr, strlen(servaddr));
```

- c) 服务器发送欢迎消息  
d) 客户端输入用户名和密码进行认证

```
// TODO: Send user and password and print server response
send(s_fd, user, strlen(user), 0);
int recv1 = recv(s_fd, buf, MAX_SIZE, 0);
printf("user: %d", recv1);
send(s_fd, pass, strlen(pass), 0);
int recv2 = recv(s_fd, buf, MAX_SIZE, 0);
printf("password: %d", recv2);
```

- e) 认证成功后，客户端可以输入一系列命令获取信息  
f) 客户端发送 QUIT 命令，结束会话。

```
// TODO: Send QUIT command and print server response
const char* QCommand = 'QUIT\r\n';
send(s_fd, QCommand, strlen(QCommand), 0);
int recv9 = recv(s_fd, QCommand, strlen(QCommand), 0);
printf("quit: %d", recv9);

close(s_fd);
```

## 二、 实验结果截图及分析

(对你自己实验的测试结果进行评价)

1. 邮件发送客户端实验结果及分析
2. 邮件接收客户端实验结果及分析

## 三、 实验中遇到的问题及解决方法

(包括设计过程中的错误及测试过程中遇到的问题)

1. 对 Socket 编程的不熟悉  
基本可以通过实验指导书入门了，但是初入门的同学直接上手编写一个邮件系统还是有些陌生和彷徨。
2. 对于初始变量设定的不理解

```
const char* host_name = "pop.163.com";
const unsigned short port = 110;
const char* user = "";
const char* pass = "";
char dest_ip[16];
int s_fd;
struct hostent *host;
struct in_addr **addr_list;
int i = 0;
int r_size;
```

多读指导书+结合理论课+网上搜寻资料可以解决。

## 四、 实验收获和建议

(关于本学期计算机网络实验的三种类型：配置验证实验、协议栈系列实验、Socket 编程实验，请给出您对于这三种类型实验的收获与体会，给出评论以及改进的建议。)

1. **配置设置的实验：**是最直观的能够感受到整个网络协议栈究竟是怎样运行的一种方式，他以可视化的方式让人一目了然的理解了网络运行中的一切，还是挺有帮助的。
2. **协议栈系列实验：**
  - a) 是从最底层让我们了解了整个协议栈的系统究竟是怎样的组成的，每一步每一个协议的实验都让我们从最开始做起。全部做下来的话收获还是挺多的，能够真正的去理解了为什么每一个报头是这样设计的，以及每一种运输方式设计的好处是什么。
  - b) 可以说协议栈实验对于加深对计算机网络的理解是最有力的一个实验。设计的挺好的，除了就是调试的时候可能还是有困难。 并且对于测试工具，用于测试的范围还不够广和覆盖，导致写完一个实验其实实际上可能是没有完成那个协议的，但是也能够做过测试。这种弊端就在写下一个协议的时候体现出来了。导致 debug 的难度非常的大。
3. **Socket 编程实验：**感觉放的篇幅太小了，还是不能够完全的去理解。入门就去做了一个邮件系统有点跨度太大了，比较难接受和实现。而且还是放在期末考的这一段时间就更加容易让人分心了。希望改进的话，能不能够把它再分成几个小的实验，然后一步一步的加深类似于协议栈的实验一样。