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：1901103 班

学生学号：190110308

学生姓名：罗锋

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

1. 邮件发送客户端详细设计

（默认邮件发送都会附带有正文：message.txt）

参数或变量的初始化：

```
const char* end_msg = "\r\n.\r\n";
const char* host_name = "smtp.qq.com"; /
const unsigned short port = 25; // SMTP
const char* user = "******\r\n"; // TODO
const char* pass = "******\r\n"; // TODO
const char* from = "**********@qq.com";
char dest_ip[16]; // Mail server IP addr
int s_fd; // socket file descriptor
struct hostent *host;
struct in_addr **addr_list;
int i = 0;
int r_size;
const char* rn = "\r\n";
```

设计解释：

end_msg：根据 smtp 协议，消息的结束需要以“回车”+“.”+“回车”为标志

host_name：若是本地邮件服务器设置为 localhost，若是 qq 则 smtp.qq.com

port：smtp 协议一般端口号为 25

user：需要转换为 base64 编码，qq 邮箱为 qq 号@qq.com

pass：smtp 服务的认证码的 base64 编码

from：自己的 qq 邮箱号

rn：为了方便后续编程设置的回车字符串

把待发送的文件内容转换为 base64 编码并读取到各自的字符串中：

```
char* message_file = get_base64(msg);
char* attach_file;
if(att_path != NULL)
    attach_file = get_base64(att_path);
```

设计解释：

调用我自己封装的 get_base64，把 msg 正文文件的内容转换为 base64 编码，存在 message_file 中；

判断是否有附件需要发送，若有再作上述同样操作，存在 attach_file 中。

创建 socket 套接字，并和邮箱服务器建立 TCP 连接：

```
s_fd = socket(AF_INET, SOCK_STREAM, 0);
if(s_fd == -1){
    printf("init a socket interface error!\n");
    return;
}
// connect------------------------------------------------------------
struct sockaddr_in servaddr;
servaddr.sin_family = AF_INET;
servaddr.sin_port = swap16(port);
bzero(servaddr.sin_zero, 1);
servaddr.sin_addr.s_addr = inet_addr(dest_ip);

if(connect(s_fd, (struct sockaddr_in *)&servaddr, sizeof(struct sockaddr)) == -1){
    printf("connect error!\n");
    return;
}
```

设计解释：

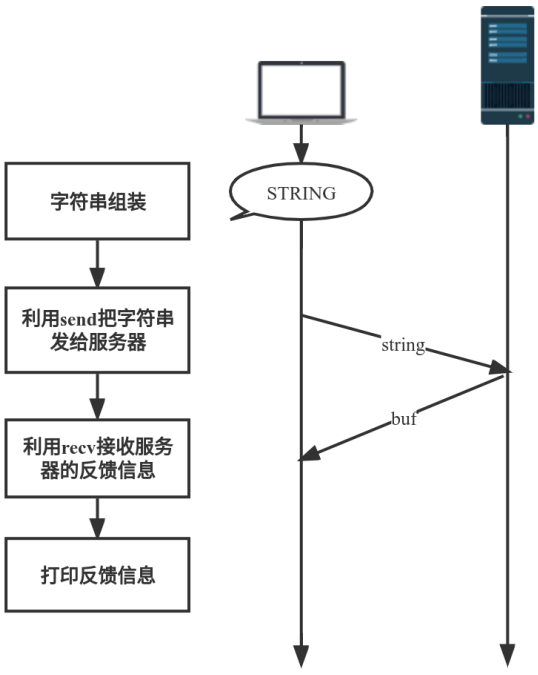
创建：

因为是 IPv4，所以用 AF_INET

连接：

端口号是 16 位存储，所以需要进行大小端转换，其他的照指导书一一填写调用

与邮件服务器互动：



所有命令的交互流程都是按照上图

1. 其中 EHLO、AUTH、user、pass、DATA、、Message ends、QUIT 的字符串并不需要组装，直接定义就可以：

（以 AUTH 代码为例）

```
const char* AUTH = "AUTH login\r\n";
send(s_fd, AUTH, strlen(AUTH), 0);
if ((r_size = recv(s_fd, buf, MAX_SIZE, 0)) == -1)
{
    perror("recv");
    exit(EXIT_FAILURE);
}
buf[r_size] = '\0'; // Do not forget the null terminator
printf("%s", buf);
```

2. MAIL FROM、RCPT TO 的组装方式相同：

（以 MAIL FROM 代码为例）

```
char* mail_from_begin = "MAIL FROM:<";
char* mail_from_end = ">\r\n";
char *MAIL_FROM = (char *) malloc(strlen(mail_from_begin)+strlen(from)+strlen(mail_from_end));
strcpy(MAIL_FROM,mail_from_begin);
strcat(MAIL_FROM,from);
strcat(MAIL_FROM,mail_from_end);
send(s_fd, MAIL_FROM, strlen(MAIL_FROM), 0);
if ((r_size = recv(s_fd, buf, MAX_SIZE, 0)) == -1)
{
    perror("recv");
    exit(EXIT_FAILURE);
}
buf[r_size] = '\0'; // Do not forget the null terminator
printf("%s", buf);
```

设计解释：

由于总体结构为：MAIL FROM:<****@qq.com>\r\n ，所以只有中间的邮箱号部分是变量，于是分为三部分字符串：begin、from、end，调用 malloc 分配空间，调用 strcpy、strcat 组装

3. data 部分的组装略微复杂一些，需要事先定义一个足够长度的字符串，按照邮件发送规则，一样一样地添加内容：

（代码截图）

```
// email head
// from
char email_head[MAX_SIZE+1] = "from:<";
strcat(email_head, from);
strcat(email_head, ">");strcat(email_head, rn);
// to
strcat(email_head, "to:<");
strcat(email_head, receiver);
strcat(email_head, ">");strcat(email_head, rn);
// subject
strcat(email_head, "subject:");
strcat(email_head, subject);
strcat(email_head, rn);
// MIME_VERSION
strcat(email_head, "MIME-Version:1.0");
strcat(email_head, rn);
// content_type
strcat(email_head, "Content-Type:multipart/mixed;boundary=thisisaboundary");
strcat(email_head, rn);
strcat(email_head, rn);
// sent text content
strcat(email_head, get_head("text/plain"));
strcat(email_head, ";name=");
strcat(email_head, msg);
strcat(email_head, rn);
strcat(email_head, "Content-Transfer-Encoding: base64");
strcat(email_head, rn);strcat(email_head, rn);
strcat(email_head, message_file);
strcat(email_head, rn);strcat(email_head, rn);
// sent attach
if(att_path != NULL)
{
    strcat(email_head, get_head("application/octet-stream"));
    strcat(email_head, ";name=");
    strcat(email_head, att_path);
    strcat(email_head, rn);
    strcat(email_head, "Content-Transfer-Encoding: base64");
    strcat(email_head, rn);strcat(email_head, rn);
    strcat(email_head, attach_file);
    strcat(email_head, rn);
    strcat(email_head, "--thisisaboundary--");
    strcat(email_head, rn);
}
```

设计解释：

strcat(email_head, rn)是指添加一个回车；

data 部分格式大致如下图：

```
from: <*********@qq.com>
to: <luo@hitsz-lab.com>
subject: TEST MAIL
MIME-Version: 1.0
Content-Type: multipart/mixed;boundary=thisisaboundary
Message-Id: <20220517092432.7C91B28011B@luo>
Date: Tue, 17 May 2022 17:24:32 +0800 (CST)

--thisisaboundary
Content-Type:text/plain;name=message.txt
Content-Transfer-Encoding: base64

Q29tcHV0ZXIgbmV0d29ya2luZyBpcyBzbyBtdWNoIGZ1biE=


--thisisaboundary
Content-Type:application/octet-stream;name=img.png
Content-Transfer-Encoding: base64

iVBORw0KGgoAAAANSUhEUgAAAOAAAABXCAYAAAAd4kksAAAABHNCSVQICAgIfAhkiAAAABl0
RVh0U29mdHdhcmUAZ25vbWUtc2NyZWVuc2hvdO8Dvz4AAA1TSURBVHic7Z17dBTVHce/d2b2
```

data 部分大致分为两部分，一部分是邮件头（最后是 content-type），一部分是内容（开始是 text content），两部分之间需要一个空行；

邮件头：因为我们需要发送不同格式的内容，所以 content-type 设置为 multipart/mixed；且设置分界线为“--thisisaboundary”，结束的界线为“--thisisaboundary--”（结束界线好像没有也可以）；

内容：至少需要声明类型和编码方式，注意判断是否有无附件

特色部分：

1. 设计了一个快速生成邮件内容头的函数：get_header()，作用是使得发送不同格式的内容时，不需要再重复地做同一繁琐的拼接操作

input: type (类似：text/plain, application/octet-stream)

output:

--thisisaboundary

Content-Type: type

具体设计为：1. 利用 malloc 给 header 字符串指针分配所需的空间（即 output 内容的所有字符串的长度）；2. 对字符串进行简单的拼接

代码：

```
char* get_head(char* type)
{
    char* header = malloc(strlen("--thisisaboundary")+strlen("\r\n")+strlen("Content-Type:")+strlen(type));
    strcpy(header, "--thisisaboundary");
    strcat(header, "\r\n");
    strcat(header, "Content-Type:");
    strcat(header, type);
    return header;
}
```

2. 封装了一个根据输入文件路径来返回该文件内容的 base64 编码的函数 get_base64()，作用是使得代码更加简洁了

input: file_path

output: base64 编码的文件内容

具体设计：1.开一个中间文件，用来储存文件内容的 base64 编码；2.调用 encode_file 方法；3.把中间文件的内容读取到字符串中并返回，必须用 malloc 给字符串分配好空间

代码：

```
char* get_base64(char* file_path)
{
    char* base64_content = malloc(MAX_SIZE);
    FILE *fd = fopen(file_path, "rb");
    FILE *temp = fopen("tempfile","w");
    fclose(temp);
    FILE *base64_fd = fopen("tempfile","rb+");
    fseek(base64_fd,0,SEEK_SET);        //移动流到文件头
    encode_file(fd, base64_fd);
    fseek(base64_fd,0,SEEK_SET);        //移动流到文件头
    fread(base64_content, 1, MAX_SIZE + 1 ,base64_fd);
    remove("tempfile");
    fclose(fd);
    fclose(base64_fd);
    return base64_content;
}
```

2. 邮件接收客户端详细设计

参数或变量的初始化：

```
const char* host_name = "pop.qq.com"; // T
const unsigned short port = 110; // POP3 s
const char* user = "********@qq.com\r\n";
const char* pass = "*********\r\n"; // TOD
char dest_ip[16];
int s_fd; // socket file descriptor
struct hostent *host;
struct in_addr **addr_list;
int i = 0;
int r_size;
```

设计解释：

host_name：若是本地邮件服务器设置为 localhost，若是 qq 则 pop.qq.com

port：smtp 协议一般端口号为 110

user：qq 邮箱为 qq 号@qq.com

pass：smtp/pop 服务的认证码

创建 socket 套接字，并和邮箱服务器建立 TCP 连接：

与邮件发送客户端没有区别

与邮箱服务器互动：

与邮件发送客户端的方式一致，只是发送的规则变了：

顺序为：user、pass、stat、list、retr、quit

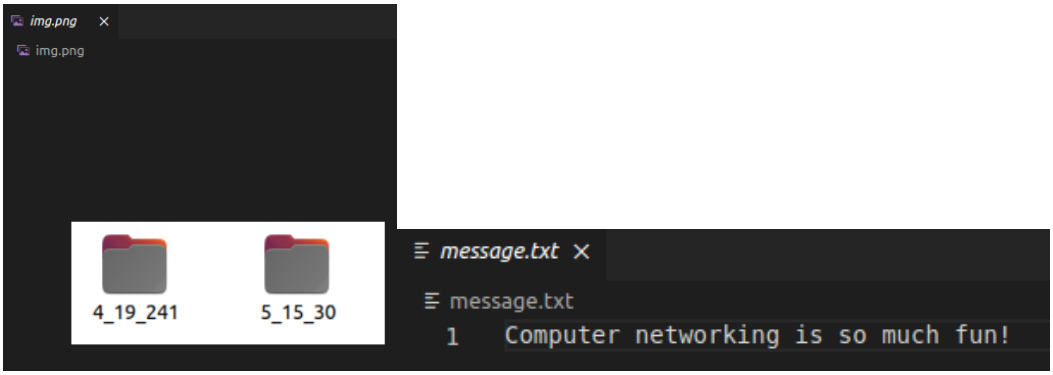
二、 实验结果截图及分析

1. 邮件发送客户端实验结果及分析

命令：./send 1160496587@qq.com -s "TEST MAIL" -m message.txt -a "img.png"

```
(base) luo@luo:~/HITSZ/Network/maillab-master$ ./send 1160496587@qq.com -s "TEST MAIL" -m message.txt -a "img.png"
220 newxmesmtplogicsvrsza5.qq.com XMail Esmtp QQ Mail Server.
250-newxmesmtplogicsvrsza5.qq.com
250-PIPELINING
250-SIZE 73400320
250-STARTTLS
250-AUTH LOGIN PLAIN XOAUTH XOAUTH2
250-AUTH=LOGIN
250-MAILCOMPRESS
250 8BITMIME
334 VXNlcm5hbWU6
334 UGFzc3dvcmQ6
235 Authentication successful
250 OK
250 OK
354 End data with <CR><LF>.<CR><LF>.
250 OK: queued as.
221 Bye.
(base) luo@luo:~/HITSZ/Network/maillab-master$
```

其中两个文件 message.txt 和 img.png 内容分别为：



终端打印信息：发送一切正常

QQ 邮箱收到的邮件：

2. 邮件接收客户端实验结果及分析

命令：./recv

```
(base) luo@luo:~/HITSZ/Network/maillab-master$ ./recv
+OK XMail POP3 Server v1.0 Service Ready(XMail v1.0)
+OK
+OK
+OK 5 31732
+OK
1 6246
2 6302
3 6352
4 6380
5 6452
.
+OK 6313
Received: from hitsz-lab.com ([223.73.111.181])
        by newxmesmtplogicsvrszc6.qq.com (NewEsmtp) with SMTP
        id ED71C2CE; Fri, 13 May 2022 21:59:23 +0800
X-QQ-mid: xmsmtpt1652450363t1anqm1qh
Message-ID: <tencent_9A2DF96017AB92F336C3CA1C942865246B08@qq.com>
X-QQ-XMAILINFO: NyTsQ4JOu2J2CH0tb8JUZli7ByX5WuCUqaUIdAMhxFhjBQhCaCwb8YhiZkQ/8a
         hkVmV5OwXeOnWC/F+iX2pmC8onPFISYu8bsGQsECxuv2z1ZFtxZkfIi8Nwj6jobfScnG8lsHUY7G
         f+xqh9RuzXgQGmNUSkY9rVLYwM2Nwlvc3K8b2P7nogea50f33Y0/9p4yFtO4SKFbT2m8Jk1nYOzj
         jQYBd13GBWsquEOQSBMkHvc5F9gAIuLIGQ9pHZL+8tsU/WUdv8uap01Ua1QIQ1IUqsZiA0c4HW6w
         CwiLSRYRWzSDvc/2Nmkx9yNOxpy6QuKmlUJLLxqRj9PkdMJxs18BUzH0UwhZU77BlCxe0bTDagWq
         8A4nOC5XV8bjm2iWX+9y/UXzoXeMFKg41Xz8Lhjz26/0OuMEgJdkP3v2SxeRtQdj+0ksf5B9Dymo
         huxWulmWkagGh0XvmkVqxd27nyKqkhlWPpIWcrl/scOqBnTvzUzHGwg+6AN08FysP1Unk1Dzaggy
         51fbFX4aVgWFVgEZi2AJgpxBwJ9kj7HrCugyMt8Z8oIXQ38VIFU2DMUE94bh7XFIW7VamE/U8R0l
         HP5mZbfNS6EwrzY3tMcjzh1gtnEd/ekOIPUaat8b73SdaKIulUPWrc1EJ/ffsWRLDeYB3g5o5gdo
         VyLV2HmvY2Z8kassVy9KujerRuHOmrIJPtrpzTRn9T5Ilka0A6JC1XGHbyH5BScCqb5RGyA75IOR
         CNETTi1GeHxO6APRMCNsBDzOiTSWd4Kcq4DMpjdJVzmP5gQ3Os7KOVWugDohHvWoX1SJ6+SRRVtE
         vIq7/nbs8bTLKy6Zv1zTCFaee39sgPsOy4bdBP4ihuNQ4l6LPzZ2vtsf1JmCLzaa7e/ijbeODAUw
         ==
from:<1160496587@qq.com>
to:<1160496587@qq.com>
subject:test1
MIME-Version:1.0
Content-Type:multipart/mixed;boundary=thisisaboundary

--thisisaboundary
Content-Type:text/plain

testmsg

--thisisaboundary
Content-Type:application/octet-stream;name=img.png
Content-Transfer-Encoding: base64
```

...

```
aHP0cXgbMfbhYdVNRESSal6cRJmu02jejwHyb55FY5y5NO3tMCUbX6ZL0110+pM7YpogTBbd
uGONQM7li7Dx1AMHp5YDX+GJi6fi08vexEu3jEvM2EnrhX6DiuCr+Ad+/sRnGHDlTJzfJ3p0
2//BPbj01m8x8YX3MO+C/LCv+iS1+whFSnVT+d2pWwgBQSZME9CLxqIsrxFr3ytHy8SxcAFo
+mgN1pk/wVVl+bYaWnTrllF69kAclx32hr8JOU4B91FFGDogo5tJAKHhxy/xbcV2fPXhMiyc
/zIqjrsXbzx0JjKj7Sp3YMnMZ7Dr1D/il0dX4csvqoLvCzfyhhYjX2m7jyAC2/Dv2a+ivng0
BvfV0LBlLRY8uBTe0TNx0bHdvN9md+r2foJFD32MXiePQGGmRO13a/DMn94Cxs3BBcdogD4B
N04fjecfvBE3D5mFK/K3YvEdL6H59Mfx6zH2ukuavVrTDh/ev/cMTFmRjqKR4zFx5rtYNu00
FMby9WRuwvrPPairux0T3gl73yjB3es2YpbNRLAtsgnVX7+NeXNnYVuND+7cITjpZw9j+axb
MLy7XdiNuml/PXaWv4gXnv4aOxsk0vsfh3EXzcWqP96AoToAOFB6+6tYemA6Zvzhcjx/IBsl
59+P1568DkV2+voDX5DLMJZis3jAMKmFMgNS9WKc1+F3fm0vvR+uXemzZd1MbLC+iUFdCuqv
RcXmHWjs7KynMJBTNBwDMw9zukNl3UxssL4JgceADGMhPAZkGAthAzKMhbABGcZC2IAMYyFs
QIaxEDYgw1gIG5BhLIQNyDAWwgZkGAthAzKMhbABGcZC2IAMYyFsQIaxkP8DgsyLDB4JgtIA
AAAASUVORK5CYII=

--thisisaboundary--


.
+OK Bye
(base) luo@luo:~/HITSZ/Network/maillab-master$
```

终端打印的信息显示：

总共有 5 封邮件；

第一封邮件的具体内容；

QQ 邮箱实际情况：

确实只有 5 封邮件；

第一封邮件的内容也与终端显示的对上了；

三、 实验中遇到的问题及解决方法

1. 发送客户端设计中，关于发送消息的组合拼接问题：该怎么方便的填写好邮件头、邮件内容等？

利用一个够大的字符串，每次要添加新内容就使用 strcat 函数进行拼接，这样子不仅可以使操作更方便，而且写出来的代码也更加通俗易懂、更有条理。

2. 如何对附件进行发送？

首先需要解决文件的编码问题，我们需要把附件转换成 base64 编码再发送。利用 C 语言文件操作、encode_file 方法，把文件转换成 base64 编码，然后编码后的内容读到字符串 attach_file 中；

然后在邮件头中的“Content-Type”选择以下种类：

Content-Type:multipart/mixed;boundary=thisisaboundary

正文文本之后，以 --thisisaboundary 为边界，定义类型为 application/octet-stream，编码模式为 base64，再把 attach_file 添加到这之后，最后正常发送即可。

3. 发送时报错段错误啥的

其实是我设置的 MAX_SIZE 太小，导致装不下附件，把 MAX_SIZE 调整地比附件大些就好了。

4. 连接不到本地邮件服务器、qq 邮箱服务器

如果是连接本地服务器，则把 hostname 设置为 localhost；

如果是发送到 qq 邮箱服务器，把 hostname 设置为 smtp.qq.com；

如果是接收 qq 邮箱服务器，把 hostname 设置为 pop.qq.com；

四、 实验收获和建议

收获：

总体来说，我觉得协议栈系列实验是最有成就感的，Socket 编程实验是最有趣的，配置验证实验是最简单的。

配置验证实验让我模拟了 VLAN、RIP、NAT 在现实中的网络配置，充分印证了这些协议的作用，对网络在实际上的一些部署有了更加清晰的认识；

协议栈系列实验自己跟着指导书的提示来编写代码，在经过花费相当时间的 coding 和 debugging 之后，看着测试一样一样通过，抓包的数据报也按照自己所学知识一样，认知与测试结果的契合让我非常有成就感；

Socket 编程实验与现实一直在使用的邮件系统结合，充分引起了我探索的兴趣，最后成功做出来的客户端能与自己的邮箱进行交互确实令人非常振奋，同时有趣；

最后，还补充了网线制作的小知识，收获一根自组装的网线。

建议：

指导书已经比较详细了，但希望邮件系统的指导书能整理的更加易懂一点点，我附件发送那块愣是看来看去没看明白到底怎么发。