

## 一、简介

电子邮件作为人们沟通交流的主要工具，在网络中有着广泛的应用。邮件系统的架构可分为邮件传输代理 MTA、邮件投递代理 MDA 和邮件用户代理 MUA。邮件用户代理是一个发信和收信的程序，负责将电子邮件发送到 SMTP 服务器或者从邮件服务器取回收到的邮件。常用的邮件用户代理有微软的 OUTLOOK、腾讯的 FOXMAIL 等，其可以从遵循 POP3 协议的邮件服务器中收取邮件。UDP、TCP/IP 等相关网络协议，以及应用程序网络协议的设计。

## 二、总体设计

### 1. 基本原理

SMTP 客户通过命令向 SMTP 服务器发送操作请求，而服务器则通过 3 位的数字对响应的请求作出响应。SMTP 规定了 14 条命令和 21 条应答信息，每条命令由 4 个字母组成，而每一种应答一般只有一行信息，又一个 3 位数字的代码开始。

POP3 采用 C/S 的工作方式。在接收邮件的用户 PC 机中的用户代理 UA 必须运行 POP3 客户程序，而在收件人所连接的 ISP 的邮件服务器中则运行 POP 的服务器程序。POP 服务器只有在用户鉴别信息（用户名+密码）后，才运行对方对邮箱进行读取。

### 2. 设计步骤

#### （1）发送邮件

①加载套接字，创建套接字库；

②得到有关于域名信息，链接到 QQ 邮箱服务器；

连接端口 25，端口 25 对应简单邮件传输服务器 SMTP。

③向服务器发送请求，用于建立客户端与服务器之间的连接；

不论是客户还是服务器应用程序都用 `recv()` 函数从 TCP 连接的另一端接收数据，`recv()` 函数返回其实际 copy 的值接受，如果 `recv()` 在 copy 时出错，那么它返回 `SOCKET_ERROR`；如果 `recv` 在等待协议接收数据时网络中断了，那么它返回 0。

④登录邮箱服务器：进行用户身份认证，由于使用 EHL0，需要发送 base64 加密的用户名和密码；

ehol 可以支持用户认证，客户端发送 ehlo 命令，用来把客户的域名通知服务器，在 TCP 连接建立阶段，发送方和接收方都是通过他们的 IP 地址来告诉对方的。

⑤客户发送 MAIL FORM 报文介绍报文的发送者；

MAIL FORM 报文包括发送人的邮件地址，可以给服务器在返回差错或报文时的返回邮件地址。

⑥客户发送 RCPT (收件人) 报文，包括收件人的邮件地址；

RCPT 命令的作用是：先弄清接收方系统是否已经准备好接受邮件的准备，然后才发送邮件这样做是为了避免浪费通信资源，不至于发送了很长的邮件之后才知道是因地址错误。

⑦客户发送 DATA 报文对报文的传送进行初始化；

DATA 命令表示要开始传送邮件的内容了。

⑧发送邮件内容；

⑨在报文传送成功后，客户就终止连接。

客户发送 QUIT 命令，服务器响应 221（服务关闭）或其他代码，在连接终止阶段，TCP 连接必须关闭。

## （2）查看邮件

①加载套接字，创建套接字库；

②得到有关于域名信息，链接到 qq 邮箱服务器；

POP 是邮件读取协议，端口 110 是为 POP3 服务开放的，使用电子邮件客户端程序的时候，会要求输入 POP3 服务器地址，默认情况下使用的就是 110 端口。

③向服务器发送请求，用于建立客户端与服务器之间的连接；

④向服务器发送账号和授权码登录到邮箱服务器；

⑤通过 stat 命令查看总共多少封信件，多少字节；

⑥通过 list 命令列出每封信件的字节数；

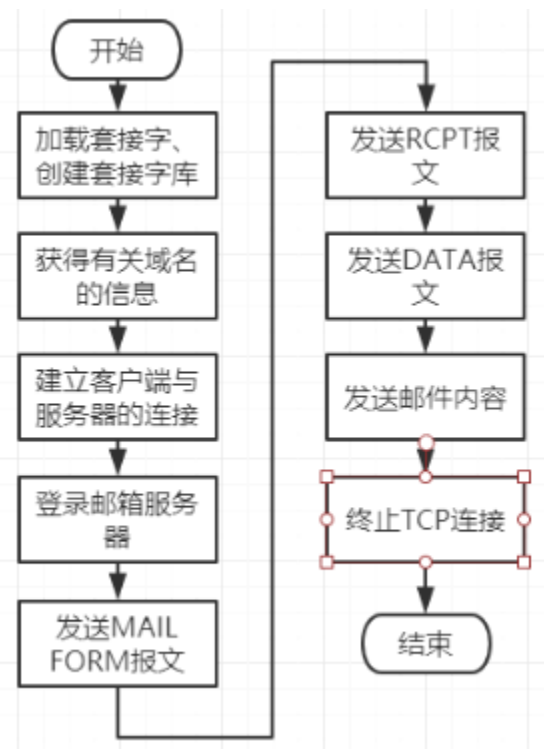
⑦通过 retr n 命令读取第 n 封信；

⑧客户终止连接。

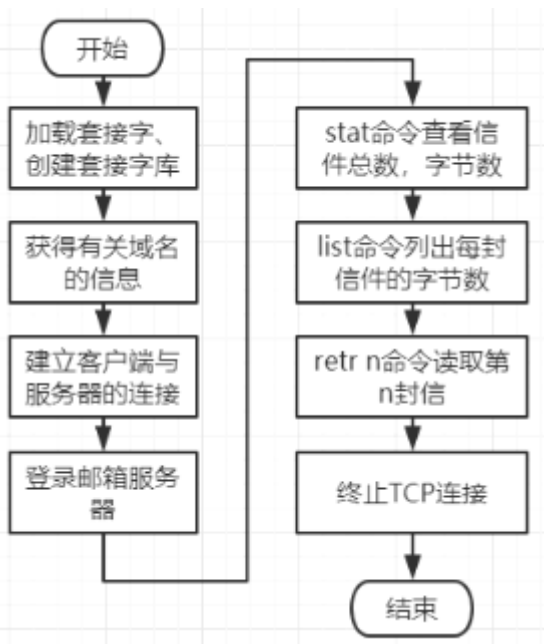
## 三、详细设计

### 1. 程序流程图

#### （1）发送邮件



## (2) 查看邮件



## 三、邮件客户端实现（发邮件）

首先对任意的二进制文件，可用 base64 编码。这种编码方法是先把二进制代码划分为一个个 24 位长的单元，然后把每一个 24 位单元划分为 4 个 6 位组。每一个 6 位组按以下方法转换成 ASCII 码。6 位的二进制代码共有 64 种不同的值，从 0 到 63。用 A 表示 0，用 B 表示 1...26 个大写字母排列完毕后，接下去再排 26 个小写字母，再后面是 10 个数字，最后用 '+' 表示 62，用 '/' 表示 63。再用两个连在一起的等号“==”和一个等号“=”分别表示最后一组的代码只有 8 位或 16 位。回车和换行符都忽略，它们可在任何地方插入。又由于邮件是以 base64 加密格式传送的，所以要定义数据格式如下：

```

struct Base64Date6
{
    unsigned int d4 : 6;
    unsigned int d3 : 6;
    unsigned int d2 : 6;
    unsigned int d1 : 6;
};

```

然后对数据进行加密传送，传送给 socket 套接字，并对数据进行编码

输入接收者的邮箱以及发送的标题和内容

```

int OpenSocket(struct sockaddr* addr); // 打开 socket 套接字
int main()
{
    char EmailTo[] = "17783445209@163.com"; // 接收者的邮箱
    char EmailContents[1000];
    char info[100];
    sprintf_s(info, "Subject: %s\r\n", "明天考试了");
    sprintf_s(EmailContents, "From: \163邮箱\19822920490@163.com\r\n" "To: \163邮箱\17783445209@163.com\r\n" "%s\r\n" "明天就考试了亲", info);
    SendMail(EmailTo, EmailContents);
    return 0;
}

```

然后要连接 163 的服务器，先获取 163 服务器的地址

```

void SendMail(char* email, char* body)
{
    int sockfd = { 0 };
    char buf[1500] = { 0 };
    char rbuf[1500] = { 0 }; // STMP 协议传送邮件的应答 (状态码)
    char login[128] = { 0 };
    char pass[128] = { 0 };
    WSADATA WSAData;
    struct sockaddr_in their_addr = { 0 };
    WSStartup(MAKEWORD(2, 2), &WSAData);
    memset(&their_addr, 0, sizeof(their_addr));
    their_addr.sin_family = AF_INET;
    their_addr.sin_port = htons(25);
    hostent* hptr = gethostbyname("smtp.163.com"); // 用的是 163 服务器
    memcpy(&their_addr.sin_addr.S_un.S_addr, hptr->h_addr_list[0], hptr->h_length);
    printf("IP of smtp.163.com is : %d:%d:%d:%d\n",
        their_addr.sin_addr.S_un.S_un_b.s_b1,
        their_addr.sin_addr.S_un.S_un_b.s_b2,
        their_addr.sin_addr.S_un.S_un_b.s_b3,
        their_addr.sin_addr.S_un.S_un_b.s_b4);
}

```

再连接

```

// cout << "look:" << sockfd << endl;
memset(rbuf, 0, 1500);
while (recv(sockfd, rbuf, 1500, 0) == 0)
{
    cout << "重新连接..." << endl;
    Sleep(5);
    sockfd = OpenSocket((struct sockaddr*)&their_addr);
    memset(rbuf, 0, 1500);
}

```

返回值为 220 表示连接成功。

然后输入发送者的邮箱名及邮箱密码还有收件者的邮箱名

然后发送邮件

```

// DATA 准备开始发送邮件内容
sprintf_s(buf, "data\r\n");
send(sockfd, buf, strlen(buf), 0);
cout << buf << endl;
memset(rbuf, 0, 1500);
recv(sockfd, rbuf, 1500, 0);
cout << rbuf << endl;

// 发送邮件内容, \r\n.\r\n内容结束标记
sprintf_s(buf, "%s\r\n.\r\n", body);
send(sockfd, buf, strlen(buf), 0);
memset(rbuf, 0, 1500);
recv(sockfd, rbuf, 1500, 0);
cout << rbuf << endl;

```

当然我们要打开 TCP socket 连接

```

int OpenSocket(struct sockaddr* addr)
{
    int sockfd = 0;
    sockfd = socket(PF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
    {
        cout << "Open sockfd(TCP) error!" << endl;
        exit(-1);
    }
    if (connect(sockfd, addr, sizeof(struct sockaddr)) < 0)
    {
        cout << "Connect sockfd(TCP) error!" << endl;
        exit(-1);
    }
    return sockfd;
}

```

运行程序。

```

Microsoft Visual Studio 调试控制台
334 dXNlcm5hbWU6
Base64 UserName: Mtk4MjI5MjA0OTBAMTYzLmNvbQ==
334 UGFzc3dvcmQ6
Base64 Password: U0ZNVkVVKWE9EVk1SRFh0Vg==
235 Authentication successful
mail from:<19822920490@163.com>
250 Mail OK
rcpt to:<17783445209@163.com>
250 Mail OK
data
354 End data with <CR><LF>.<CR><LF>
250 Mail OK queued as smtp11,DSCowAD3eAwMhdxgew2MAA--.9837S2 1625064721
221 Bye

C:\Users\14847\source\repos\Project7\Debug\Project7.exe (进程 19496) 已退出，代码为 0。
按任意键关闭此窗口。 . . .

```

显示登录成功且发送成功  
我们去往邮箱发现了邮件



当然我们也可以利用接收端查看邮箱下面会讲到。  
上述就利用 SMTP 和 TCP socket 完成的邮件的发送。

## 四、邮件客户端实现（收邮件）

```
int main()
{
    int sockfd = { 0 };
    char buf[1500] = { 0 };
    char rbuf[1500] = { 0 }; //POP协议传送邮件的应答
    char login[128] = { 0 };
    char pass[128] = { 0 };
    WSADATA WSAData;
    struct sockaddr_in their_addr = { 0 };
    WSStartup(MAKEWORD(2, 2), &WSAData);
    memset(&their_addr, 0, sizeof(their_addr));
    their_addr.sin_family = AF_INET;
    their_addr.sin_port = htons(110);
    hostent* hptr = gethostbyname("pop3.163.com"); // 用的是163服务器
    memcpy(&their_addr.sin_addr.S_un.S_addr, hptr->h_addr_list[0], hptr->h_length);
    printf("IP of pop3.163.com is : %d:%d:%d:%d\n",
        their_addr.sin_addr.S_un.S_un_b.s_b1,
        their_addr.sin_addr.S_un.S_un_b.s_b2,
        their_addr.sin_addr.S_un.S_un_b.s_b3,
        their_addr.sin_addr.S_un.S_un_b.s_b4);
    //OK, 163邮箱的IP地址已获取到, 下面就要开始进行连接了
    sockfd = OpenSocket((struct sockaddr*)&their_addr);
    cout << "look:" << sockfd << endl;
    memset(rbuf, 0, 1500);
    // ... (rest of the code is partially visible and cut off)
```

首先连接 163 服务器

```
// USER
memset(buf, 0, 1500);
sprintf_s(buf, "%s\r\n", "user 17783445209@163.com"); //输入你的邮箱账号
send(sockfd, buf, strlen(buf), 0);
cout << buf << endl;
memset(rbuf, 0, 1500);
recv(sockfd, rbuf, 1500, 0);
cout << rbuf << endl;

// PASSWORD
sprintf_s(buf, "%s\r\n", "pass RXRFQWJUCNMVBLQD"); //你的邮箱密码
send(sockfd, buf, strlen(buf), 0);
cout << buf << endl;
memset(rbuf, 0, 1500);
recv(sockfd, rbuf, 1500, 0);
cout << rbuf << endl;
//cout<<"通过"<<endl;
```

输入邮箱账号和密码

LIST 命令列出了邮箱的所有邮件

```

    }
    // 打开TCP Socket连接
int OpenSocket(struct sockaddr* addr)
{
    int sockfd = 0;
    sockfd = socket(PF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
    {
        cout << "Open sockfd(TCP) error!" << endl;
        exit(-1);
    }
    if (connect(sockfd, addr, sizeof(struct sockaddr)) < 0)
    {
        cout << "Connect sockfd(TCP) error!" << endl;
        exit(-1);
    }
    return sockfd;
}
```

当然也要打开 TCP socket 连接

运行程序

```

C:\Users\14847\source\repos\Project8\Debug\Project8.exe
IP of pop3.163.com is : 220:181:12:110
look:728
+OK Welcome to coremail Mail Pop3 Server (163coms[10774b260cc7a37d26d71b52404dc
user 17783445209@163.com

+OK core mail

pass RXRFQWJUCNMVBLQD

+OK 2 message(s) [18730 byte(s)]

list

+OK 2 18730
1 18069
2 661
.

请输入查询命令:
_
```

显示我们登录成功，然后我们默认一个 list 命令显示了邮箱中的邮件个数和信息  
我们输入 RETR 1 查看第一封邮件信息

```
请输入查询命令:
RETR 1
+OK 18069 octets

(Quit Receive: Received: from =?UTF-8?B?572R5piT6YKu5Lu25Lit5b+D?= (unknown [192.168.194.38])
by mfast9 (Coremail) with SMTP id uMCowABXBGqRg9xgmCB0Gg--.56820S2;
Wed, 30 Jun 2021 22:45:37 +0800 (CST)
From: =?UTF-8?B?572R5piT6YKu5Lu25Lit5b+D?= <mail@service.netease.com>
Sender: mail@service.netease.com
To: zhouyou5209@163.com
Message-ID: <808073103.22019345.1625064338236.JavaMail.mail@service.netease.com>
Subject: =?UTF-8?B?44CQ6YeN6KaB5o+Q6YaS44CR6L+Y5beu5LiA?=?
=?UTF-8?B?5q2177yB5byA5ZCv56e75Yqo5Yqe5YWs5LmL5peF?=?
MIME-Version: 1.0
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: quoted-printable
X-CM-TRANSID: uMCowABXBGqRg9xgmCB0Gg--.56820S2
X-Coremail-Antispam: 1Uf129KBjvdXoWruFy3ur1rKFWUCF45GFW3ZFb_yoW3uFbE9r
y5Wr9Igay3Cw4IqalvyF4IyFsrC34qg34DA34kW3srAry29wlj9rs3CFW5ZF1ftF40qan5
Wr13CalrJrnXZjkaLaAFLSUUU2u8b8apIn2vfkv8UJUUUU8Yxn0WfASr-VFAUDa7-sFnT
9fnUUvcSsGvfc2KfnxnjZKAaVFWWhVr0xZFpf9x0zRUUUUUUUUU=
X-Originating-IP: [192.168.194.38]
Date: Wed, 30 Jun 2021 22:45:37 +0800 (CST)

<table width=3D"750" border=3D"0" bgcolor=3D"#FFFFFF" cellspacing=3D"0" cel=
padding=3D"0" align=3D"center"
style=3D"width: 750px; margin: 0 auto; text-align: center; background-col=
or: #ffffff; margin: 0 auto;">
<tbody>
<tr>
<td>
<table width=3D"750" border=3D"0" cellspacing=3D"0" cellpadding=3D"=
0">
<tbody>
<tr>
<td style=3D"text-align: cente
```

附：POP 命令

事务状态命令	
STAT	状态通知
LIST [msg]	确认指定邮件大小（获取一览表）
RETR [msg]	取得邮件信息
DELE [msg]	删除服务器中保存的邮件（QUIT 命令执行时才真正删除）
RSET	撤销所有的 DELE 命令，通信结束
QUIT	执行 DELE 命令，终止通信
TOP msg n	只要邮件的前 n 行内容
UIDL [msg]	获得该邮件的唯一标识

以上我们就完成了邮件的接受查看端。

## 五、收获和遇到的问题

经过这次计网课设，对 SMTP 协议和 POP3 协议有了更加深刻的理解，在实践中碰到了许多问题，通过不断解决这些问题，使得自己在编程能力、理解能力、学习能力等各方面能力都得到了有效的锻炼。我用 SMTP 命令和 POP3 命令在命令行下反复发送和接收邮件，观察对不同命令，服务器的应答有什么不同。这时我也发现，对于一些服务器，不需要用户登录就可以发送邮件，但更多的服务器要求登录才能发送邮件。而登录的命令“auth login”却很少在教学资料上提到。最后，根据对两个协议的熟悉，以在命令行下的操作为参考流程，设



计出了发送邮件程序和邮件接收程序，并用程序设计的知识和网络知识对程序进行调试。虽然时间很紧迫，进行课设的同时还得上课，课余时间基本上都花在了课设上，经过努力，完成了所有的基本要求，感觉很开心。但还是有一些问题没有得到解决，希望能够在今后的学习中将这些问题解决掉。