

CS305 Programming Assignment 1 Report

Author: 陈贲 12212231

Lab: Friday 3-4 王晴老师

Basic Part

官方Testbench测试结果

```
0:1:[tmux] - "BensMacintosh.local" 1
>>> Actual: {'resp': "b'+OK message 1 deleted'"}
TEST>>> Step: Reset the deletion
>>> Expect: {'resp': "b'+OK'"}
>>> Actual: {'resp': "b'+OK'"}
TEST>>> Step: Cmd Noop
>>> Expect: {'resp': "b'+OK'"}
>>> Actual: {'resp': "b'+OK'"}
TEST>>> Step: quit
Teardown scenario: Send an email to another user in the same domain and test RETR, DELE, RSET and NOOP
Setup scenario: Send an email to another user in the different domain and test LIST
TEST>>> Step: Send email
TEST>>> Step: wait
TEST>>> Step: Login the account in different domain
TEST>>> Step: List the email
>>> Expect: [{'mesg_num': 1}]
>>> Actual: [{'mesg_num': 1, 'octets': 160}]
TEST>>> Step: quit
Teardown scenario: Send an email to another user in the different domain and test LIST
Setup scenario: Send an email to another user non-exist in different domain
TEST>>> Step: Send email
TEST>>> Step: wait
TEST>>> Step: Login the account in different domain
TEST>>> Step: List the email
>>> Expect: [{'mesg_num': 1}]
>>> Actual: [{'mesg_num': 1, 'octets': 166}]
TEST>>> Step: quit
Teardown scenario: Send an email to another user non-exist in different domain
Setup scenario: Send an email to another user from a non-existing email address
Teardown scenario: Send an email to another user from a non-existing email address

***** TEST SUMMARY *****
StudentID: 12212231
Score: 90/90
PASSED: 6          PARTIALLY PASSED: 1

***** TEST DETAILS *****
[PASSED] POP3 USER, PASS and QUIT:
Credit: 15/15
Message: None
[PASSED] Send an email to another user in the different domain and test LIST:
Credit: 5/5
Message: None
[PASSED] Send an email to another user in the different domain and test STAT:
Credit: 5/5
Message: None
[PASSED] Send an email to another user in the same domain and test RETR, DELE, RSET and NOOP:
Credit: 5/5
Message: None
[PASSED] Send an email to another user in the different domain and test LIST:
Credit: 50/50
Message: None
[PASSED] Send an email to another user non-exist in different domain:
Credit: 5/5
Message: None
[PARTIALLY PASSED] Send an email to another user from a non-existing email address:
Credit: 5/5
Message: Raised error in step Send email: SMTPSenderRefused -> 5 pts

0:1.1 10.24.86.56 116.6.234.162 1 [tmux]:* 2 Python 3 Python .../PA1 for CS305 2023 Fall SMTP Server/src Fri 2023-10-27 01:39
```

SMTP 抓包结果（基于 4.yml 测试）

[illegible]

POP 抓包结果 (基于 4.yml 测试)

Loopback: lo0

pop

No.	Time	Source	Destination	Protocol	Length	Info
43	6.017503	127.0.0.1	127.0.0.1	POP	96	S: +OK 12212231's dummy POP3 server ready
45	6.017594	127.0.0.1	127.0.0.1	POP	87	C: USER usr2@mail.sustech.edu.cn
47	6.017642	127.0.0.1	127.0.0.1	POP	72	S: +OK user found
49	6.017692	127.0.0.1	127.0.0.1	POP	72	C: PASS password2
51	6.017734	127.0.0.1	127.0.0.1	POP	89	S: +OK user successfully logged on
53	6.017821	127.0.0.1	127.0.0.1	POP	64	C: RETR 1
55	6.018858	127.0.0.1	127.0.0.1	POP	61	S: +OK
57	6.018895	127.0.0.1	127.0.0.1	POP/I	232	subject: Test Subject2, from: usr1@mail.sustech.edu.cn, (text/plain)
59	6.018973	127.0.0.1	127.0.0.1	POP	64	C: DELE 1
61	6.019020	127.0.0.1	127.0.0.1	POP	79	S: +OK message 1 deleted
63	6.019105	127.0.0.1	127.0.0.1	POP	62	C: RSET
65	6.019153	127.0.0.1	127.0.0.1	POP	61	S: +OK
67	6.019230	127.0.0.1	127.0.0.1	POP	62	C: NOOP
69	6.019273	127.0.0.1	127.0.0.1	POP	61	S: +OK
71	6.019339	127.0.0.1	127.0.0.1	POP	62	C: QUIT
73	6.019381	127.0.0.1	127.0.0.1	POP	102	S: +OK 12212231's dummy POP3 server signing off

> Frame 57: 232 bytes on wire (1856 bits), 232 bytes captured (1856 bits) on interface lo0, id 0

> Null/Loopback

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> Transmission Control Protocol, Src Port: 3110, Dst Port: 54344, Seq: 95, Ack: 56, Len: 176

> Post Office Protocol

> Internet Message Format

> Content-Type: text/plain; charset="utf-8"

MIME-Version: 1.0

Content-Transfer-Encoding: base64

Subject: Test Subject2

From: usr1@mail.sustech.edu.cn, 1 item

- > Item: usr1@mail.sustech.edu.cn/r/n

> Line-based text data: text/plain (1 lines)

Test Body2

0000 54 65 73 74 20 42 6f 64 79 32 Test Bod y2

Text item (text), 10 bytes

Frame (232 bytes)

base64 (10 bytes)

Packets: 78 · Displayed: 16 (20.5%) · Dropped: 0 (0.0%)

Profile: Default

Bonus Part 1 Email Exchange

Partner: 陈明志 12211414

His IP: 10.27.95.59

My IP: 10.27.96.208

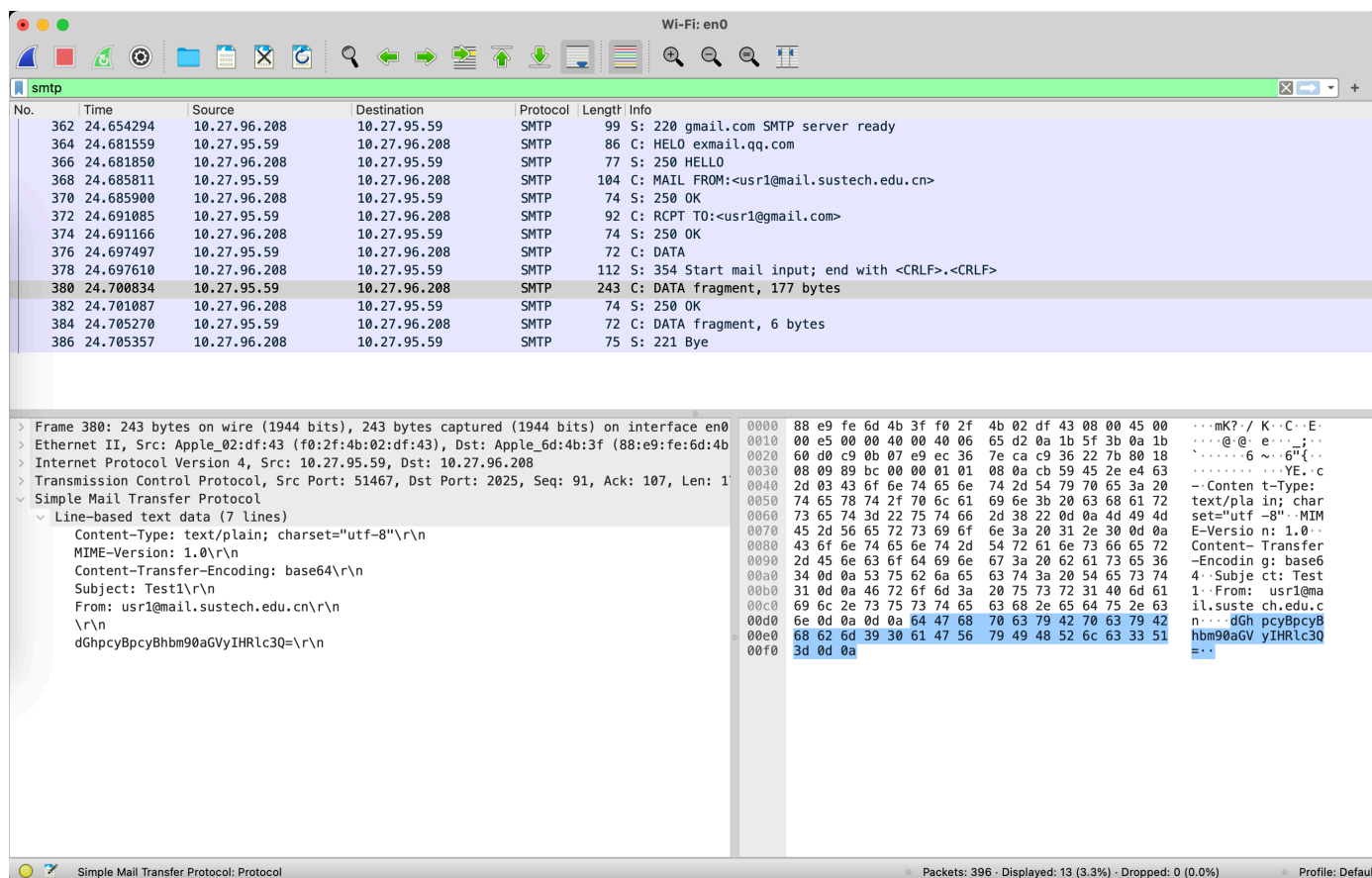
对方运行 exmail.qq.com, 并登陆 usr1@mail.sustech.edu.cn 我方运行 gmail.com, 并登陆
usr1@gmail.com

Scenerio 1: 对方给我方发送邮件

对方命令运行结果

```
计算机网络/PA1 for CS305 2023 Fall SMTP Server/src 1m 3s
> python3 agent.py -e usr1@mail.sustech.edu.cn -p password1
[smtp|pop|exit]>>> smtp
To: usr1@gmail.com
To:
Subject: Test
Content: This is a Test
[smtp|pop|exit]>>> smtp
To: usr1@gmail.com
To:
Subject: Test1
Content: this is another test
[smtp|pop|exit]>>> █
```

我方抓包结果



我方服务器收到邮件

```

→ src git:(main) x python3 agent.py -e usr1@gmail.com -p password1
[smtp|pop|exit]>>> pop
b"+OK 12212231's dummy POP3 server ready"
b"+OK user found"
b"+OK user successfully logged on"
[pop]>>> list
[b'1 160']
[pop]>>> list
[b'1 160', b'2 169']
[pop]>>> retr 1
b'Content-Type: text/plain; charset="utf-8"'
b'MIME-Version: 1.0'
b'Content-Transfer-Encoding: base64'
b'Subject: Test'
b'From: usr1@mail.sustech.edu.cn'
b''
b'VGhpcyBpcyBhIFRlc3Q='
[pop]>>> retr 2
b'Content-Type: text/plain; charset="utf-8"'
b'MIME-Version: 1.0'
b'Content-Transfer-Encoding: base64'
b'Subject: Test1'
b'From: usr1@mail.sustech.edu.cn'
b''
b'dGhpcyBpcyBhbm90aGVyIHRlc3Q='
[pop]>>>
3:2.1 ython 2 Python:* 3 zsh .../PA1 for CS305 2023 Fall SMTP Server/src Thu < 2023-10-26 < 16:11

```

Scenario 2: 我方给对方发送邮件

我方命令运行结果

```

→ src git:(main) x python3 agent.py -e usr1@gmail.com -p password1
[smtp|pop|exit]>>> smtp
To: usr1@mail.sustech.edu.cn
To:
Subject: Test
Content:
[smtp|pop|exit]>>> smtp
To: usr1@mail.sustech.edu.cn
To:
Subject: Test
Content: This is a test email
[smtp|pop|exit]>>>

```

我方抓包结果（本地回环）

Capturing from Loopback: lo0

Current filter: smtp

No.	Time	Source	Destination	Protocol	Length	Info
19	12.622033	127.0.0.1	127.0.0.1	SMTP	64	S: 250 OK
21	12.622938	127.0.0.1	127.0.0.1	SMTP	92	C: rcpt TO: <usr1@mail.sustech.edu.cn>
23	12.622980	127.0.0.1	127.0.0.1	SMTP	64	S: 250 OK
25	12.623014	127.0.0.1	127.0.0.1	SMTP	62	C: data
27	12.623051	127.0.0.1	127.0.0.1	SMTP	102	S: 354 Start mail input; end with <CRLF>.<CRLF>
29	12.623155	127.0.0.1	127.0.0.1	SMTP/...	195	subject: Test, from: usr1@gmail.c, (text/plain)
31	12.711222	127.0.0.1	127.0.0.1	SMTP	64	S: 250 OK
33	12.711324	127.0.0.1	127.0.0.1	SMTP	62	C: quit
35	12.711368	127.0.0.1	127.0.0.1	SMTP	65	S: 221 Bye
65	29.212511	127.0.0.1	127.0.0.1	SMTP	89	S: 220 gmail.com SMTP server ready
120	46.622421	127.0.0.1	127.0.0.1	SMTP	89	C: ehlo Chanbens-MacBook-Pro.local
122	46.622499	127.0.0.1	127.0.0.1	SMTP	67	S: 250 HELLO
124	46.622616	127.0.0.1	127.0.0.1	SMTP	84	C: mail FROM: <usr1@gmail.com>
126	46.622765	127.0.0.1	127.0.0.1	SMTP	64	S: 250 OK
128	46.622861	127.0.0.1	127.0.0.1	SMTP	92	C: rcpt TO: <usr1@mail.sustech.edu.cn>
130	46.622922	127.0.0.1	127.0.0.1	SMTP	64	S: 250 OK
132	46.622973	127.0.0.1	127.0.0.1	SMTP	62	C: data
134	46.623006	127.0.0.1	127.0.0.1	SMTP	102	S: 354 Start mail input; end with <CRLF>.<CRLF>
136	46.623074	127.0.0.1	127.0.0.1	SMTP/...	225	subject: Test, from: usr1@gmail.com, (text/plain)
146	46.824598	127.0.0.1	127.0.0.1	SMTP	64	S: 250 OK
148	46.824661	127.0.0.1	127.0.0.1	SMTP	62	C: quit
150	46.824694	127.0.0.1	127.0.0.1	SMTP	65	S: 221 Bye

> Frame 136: 225 bytes on wire (1800 bits), 225 bytes captured (1800 bits) on interface lo0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 49935, Dst Port: 2025, Seq: 104, Ack: 107, Len: 225
> Simple Mail Transfer Protocol
C: .
> [DATA fragment (166 bytes): #136(166)]
> Internet Message Format
> Content-Type: text/plain; charset="utf-8"
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Subject: Test
> From: usr1@gmail.com, 1 item
Item: usr1@gmail.com\r\nAddress: usr1@gmail.com\r\n> Line-based text data: text/plain (1 lines)
This is a test email

0000 02 00 00 00 45 00 00 dd 00 00 40 00 40 06 00 00 ...E...@...
0010 7f 00 00 01 7f 00 00 01 c3 0f 07 e9 0e 9a d4 23 ...#
0020 b5 c4 5e a2 80 18 18 ea fe d1 00 00 01 01 08 0a ...^...
0030 58 94 fb bf 01 c9 52 39 43 6f 6e 74 65 6e 74 2d X...R9 Content-
0040 54 79 70 65 3a 20 74 65 78 74 2f 70 6c 61 69 6e Type: te xt/plain
0050 3b 20 63 68 61 72 73 65 74 3d 22 75 74 66 2d 38 ; charse t="utf-8
0060 22 0d 0a 4d 49 4d 45 2d 56 65 72 73 69 6f 6e 3a "...MIME- Version:
0070 20 31 2e 30 0d 0a 43 6f 6e 74 65 6e 74 2d 54 72 1.0..Co ntent-Tr
0080 61 6e 73 66 65 72 2d 45 6e 63 6f 64 69 6e 67 3a ansfer-E ncoding:
0090 20 62 61 73 65 36 34 0d 0a 53 75 62 6a 65 63 74 base64..Subject
00a0 3a 20 54 65 73 74 0d 0a 46 72 6f 6d 3a 20 75 73 : Test.. From: us
00b0 72 31 40 67 6d 61 69 6c 2e 63 6f 6d 0d 0a 0d 0a r1@gmail .com...
00c0 56 47 68 70 63 79 42 70 63 79 42 68 49 48 52 6c VghpcyBp cyBhIHRl
00d0 63 33 51 67 5a 57 31 68 61 57 77 3d 0d 0a 2e 0d c3QgZW1h aWw=...
00e0 0a

Frame (225 bytes) Reassembled SMTP (166 bytes) base64 (20 bytes)

Loopback: lo0: <live capture in progress>

Packets: 481 · Displayed: 26 (5.4%) Profile: Default

对方agent接收结果

```
计算机网络/PA1 for CS305 2023 Fall SMTP Server/src 1m 3s
> python3 agent.py -e usr1@mail.sustech.edu.cn -p password1
[smtp|pop|exit]>>> smtp
To: usr1@gmail.com
To:
Subject: Test
Content: This is a Test
[smtp|pop|exit]>>> smtp
To: usr1@gmail.com
To:
Subject: Test1
Content: this is another test
[smtp|pop|exit]>>> pop
b'+OK POP3 server ready'
b'+OK'
b'+OK user successfully logged on'
[pop]>>> retr 1
b'Content-Type: text/plain; charset="utf-8"'
b'MIME-Version: 1.0'
b'Content-Transfer-Encoding: base64'
b'Subject: Test'
b'From: usr1@gmail.com'
[pop]>>> retr 2
b'Content-Type: text/plain; charset="utf-8"'
b'MIME-Version: 1.0'
b'Content-Transfer-Encoding: base64'
b'Subject: Test'
b'From: usr1@gmail.com'
b''
b'VGhpcyBpcyBhIHRlc3QgZW1haWw='
[pop]>>>
```

对方抓包结果

smtp or pop					
No.	Time	Source	Destination	Protocol	Length Info
122	29.146892	10.27.95.59	10.27.96.208	SMTP	93 S: 220 localhost SMTP Server
124	29.151471	10.27.96.208	10.27.95.59	SMTP	82 C: ehlo gmail.com
126	29.151787	10.27.95.59	10.27.96.208	SMTP	74 S: 250 OK
128	29.157082	10.27.96.208	10.27.95.59	SMTP	94 C: mail FROM:<usr1@gmail.com>
130	29.157290	10.27.95.59	10.27.96.208	SMTP	74 S: 250 OK
132	29.161787	10.27.96.208	10.27.95.59	SMTP	102 C: rcpt TO:<usr1@mail.sustech.edu.cn>
134	29.162292	10.27.95.59	10.27.96.208	SMTP	75 S: 250 OK~
136	29.166112	10.27.96.208	10.27.95.59	SMTP	72 C: data
138	29.166703	10.27.95.59	10.27.96.208	SMTP	103 S: 354 End data with <CR><LF>.<CR><LF>
140	29.172340	10.27.96.208	10.27.95.59	SMTP	202 C: DATA fragment, 136 bytes
142	29.172639	10.27.95.59	10.27.96.208	SMTP	88 S: 250 Message accepted
144	29.179222	10.27.96.208	10.27.95.59	SMTP	72 C: DATA fragment, 6 bytes
146	29.179440	10.27.95.59	10.27.96.208	SMTP	75 S: 221 Bye
663	63.245889	10.27.95.59	10.27.96.208	SMTP	93 S: 220 localhost SMTP Server
665	63.289940	10.27.96.208	10.27.95.59	SMTP	82 C: ehlo gmail.com
669	63.291215	10.27.95.59	10.27.96.208	SMTP	74 S: 250 OK
671	63.295208	10.27.96.208	10.27.95.59	SMTP	94 C: mail FROM:<usr1@gmail.com>
673	63.295410	10.27.95.59	10.27.96.208	SMTP	74 S: 250 OK
675	63.300575	10.27.96.208	10.27.95.59	SMTP	102 C: rcpt TO:<usr1@mail.sustech.edu.cn>
677	63.300943	10.27.95.59	10.27.96.208	SMTP	75 S: 250 OK~
679	63.306219	10.27.96.208	10.27.95.59	SMTP	72 C: data
681	63.306381	10.27.95.59	10.27.96.208	SMTP	103 S: 354 End data with <CR><LF>.<CR><LF>
684	63.310559	10.27.96.208	10.27.95.59	SMTP	232 C: DATA fragment, 166 bytes
686	63.311077	10.27.95.59	10.27.96.208	SMTP	88 S: 250 Message accepted
689	63.315869	10.27.96.208	10.27.95.59	SMTP	72 C: DATA fragment, 6 bytes
691	63.316034	10.27.95.59	10.27.96.208	SMTP	75 S: 221 Bye

> Frame 122: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface en0, id 0

> Ethernet II, Src: Apple_02:df:43 (f0:2f:4b:02:df:43), Dst: Apple_6d:4b:3f (08:e9:fe:6d:4b:3f)

> Internet Protocol Version 4, Src: 10.27.95.59, Dst: 10.27.96.208

> Transmission Control Protocol, Src Port: 1025, Dst Port: 49923, Seq: 1, Ack: 1, Len: 27

> Simple Mail Transfer Protocol

> Response: 220 localhost SMTP Server\r\n

0000 88 e9 fe 6d 4b 3f f0 2f 4b 02 df 43 08 00 45 00 ...mK?-/ K--C--E-

0010 00 4f 00 00 40 00 40 06 66 68 0a 1b 5f 3b 0a 1b `0...@: fh...;..

0020 60 d0 04 01 c3 03 53 0d ac 88 b8 34 62 7c 80 18 `.....S: ...4b]..

0030 08 0a a9 a0 00 00 01 01 08 0a ce 52 a5 56 d9 2b-R-V+

0040 20 98 32 32 30 20 6c 6f 63 61 6c 68 6f 73 74 20 ..220 lo calhost

0050 53 4d 54 50 20 53 65 72 76 65 72 0d 0a SMTP Ser ver...

Bonus Part 2 Error Handling

Error Part 1: POP Server

设计思路：POP Server主要有三种状态，登陆->处理->退出，且都为单向状态转移。故只需少许状态变量即可表示当前状态。

Error 1 登陆失败

USER 命令首先检查用户是否存在，若存在，保存用户名并返回 +OK；若不存在返回 -ERR。

PASS 命令首先检查是否执行过 **USER** 命令，若未执行，报错并后续不执行。接着检查密码是否正确，若错误，返回 -ERR。

```

if cmd == 'USER':
    if ACCOUNTS.__contains__(arg):
        conn.sendall(b'+OK user found\r\n')
        account = arg
    else:
        conn.sendall(b'-ERR invalid user\r\n')
elif cmd == 'PASS':
    if account == '':
        conn.sendall(b'-ERR please specific user\r\n')
    elif ACCOUNTS[account] != arg:
        conn.sendall(b'-ERR wrong password\r\n')
    else:
        login = True
        mail_count = len(MAILBOXES[account])
        conn.sendall(b'+OK user successfully logged on\r\n')

```

Error 2 LIST失败

`LIST x` 首先检查是否登陆，若未登陆后续不执行并返回 `-ERR`。然后检查参数是否合法，如在id范围内并且不是已删除邮件，若不合法返回 `-ERR`。

```

elif cmd == 'LIST':
    try:
        if not login:
            raise
        id = int(arg) - 1
        if id >= mail_count or id < 0:
            raise
        if id in dele_mail:
            raise
        size = 0
        if mail_count != 0:
            mail = MAILBOXES[account][id]
            size = len(mail.as_string())
            conn.sendall(f'+OK {id + 1} {size}\r\n'.encode())
    except:
        conn.sendall(b'-ERR\r\n')

```

`LISTR` 命令只检查是否已登陆，未登陆返回 `-ERR` 并退出。

```
elif cmd == 'LIST':
    if not login:
        conn.sendall(b'-ERR log in first\r\n')
        continue
    conn.sendall(f'+OK {mail_count} messages\r\n'.encode())
    for i in range(len(MAILBOXES[account])):
        if i not in dele_mail:
            mail = MAILBOXES[account][i]
            size = len(mail.as_string())
            conn.sendall(f'{i + 1} {size}\r\n'.encode())
    conn.sendall(b'.\r\n')
```

Error 3 RETR失败

未登陆和id不合法都会返回 `-ERR` 并退出


```

elif cmd == 'RETR':
    try:
        if not login:
            conn.sendall(b'-ERR log in first\r\n')
            continue
        id = int(arg) - 1
        if id >= mail_count or id < 0:
            raise
        if id in dele mail:
            raise
        mail_message = MAILBOXES[account][id]
        mail = mail_message.as_string().strip()
        mail = mail.replace('\n', '\r\n')
        conn.sendall('+OK\r\n'.encode())
        conn.sendall((mail + '\r\n').encode())
        conn.sendall(b'.\r\n')
    except:
        conn.sendall(b'-ERR invalid mail id\r\n')

```

Error 4 DELE失败

未登陆和id不合法都会返回 -ERR 并退出

```

elif cmd == 'DELE':
    try:
        if not login:
            conn.sendall(b'-ERR log in first\r\n')
            continue
        id = int(arg) - 1
        if id >= mail_count or id < 0:
            raise
        dele_mail.add(id)
        conn.sendall(f'+OK message {id + 1} deleted\r\n'.encode())
    except:
        conn.sendall(b'-ERR\r\n')

```

Error 5 STAT失败

未登陆时会返回 `-ERR` 并退出

```
elif cmd == 'STAT':
    if login:
        mail_total = 0
        byts_total = 0
        for i in range(len(MAILBOXES[account])):
            if i not in dele_mail:
                mail_total += 1
                byts_total += len(MAILBOXES[account][i].as_string())
        conn.sendall(f'+OK {mail_total} {byts_total}\r\n'.encode())
    else:
        conn.sendall(b'-ERR log in first\r\n')
```

Error 6 RSET失败

未登陆时返回 `-ERR` 并退出

```
elif cmd == 'RSET':
    if login:
        dele_mail.clear()
        conn.sendall(b'+OK\r\n')
    else:
        conn.sendall(b'-ERR log in first\r\n')
else:
    conn.sendall(b'-ERR illegal command\r\n')
```

Error 7 命令不合法

当命令未实现或不符合标准时，会返回 `-ERR`

```
else:
    conn.sendall(b'-ERR illegal command\r\n')
```

Error Part 2: SMTP Server

SMTP Server的状态多达五个，但是转换关系并不复杂：登陆->邮件来源->收件人->接收邮件->退出。

Error 1 登陆失败

首先验证是否为连接对象是否为邮件Server，具体表现为参数在server配置里。如不是server，只允许本地用户访问，即检查hostname后缀是否为localhost或arpa。

```
if cmd[0] == 'HELO' or cmd[0] == 'EHLO':
    server = cmd[1] in _config['server']
    if not server:
        prefix = cmd[1].split('.')[0]
        if prefix != 'arpa' and prefix != 'local':
            conn.sendall(b'551 User not local\r\n')
            continue
    auth = True
    conn.sendall(b'250 HELLO\r\n')
```

Error 2 邮件来源失败

首先验证是否已认证，即HELO命令。接着验证MAIL FROM命令是否执行过，防止多次执行。如果不是server发送的邮件，检查发送方是否合法，防止伪造发件人。

```
elif cmd[0] == 'MAIL':
    if auth:
        if mail_from:
            conn.sendall(b'503 Bad sequence of command\r\n')
            break
        fr = cmd[1][6:-1]
        if not server and fr not in ACCOUNTS.keys():
            conn.sendall(b'553 User not exists\r\n')
            continue
        sender = fr
        mail_from = True
        conn.sendall(b'250 OK\r\n')
    else:
        conn.sendall(b'503 Bad sequence of command\r\n')
```

Error 3 收件人失败

首先检查邮件是否已发出过，是否已执行MAIL FROM，如已发出或未执行MAIL FROM，返回错误。

```
elif cmd[0] == 'RCPT':
    if not mail_from or sent:
        conn.sendall(b'503 Bad sequence of command\r\n')
    else:
        mail_to = cmd[1][4:-1]
        domain = email_domain[mail_to.split('@')[-1]]
        if domain in rcpt_to.keys():
            rcpt_to[domain].append(mail_to)
        else:
            rcpt_to[domain] = []
            rcpt_to[domain].append(mail_to)
        conn.sendall(b'250 OK\r\n')
```

Error 4 接收邮件失败

如果前面流程未完成，返回错误并停止接收邮件。

```
if cmd == 'DATA':
    if len(rcpt_to) > 0 and not sent and auth and mail_from:
        conn.sendall(b'354 Start mail input; end with <CRLF>.<CRLF>\r\n')
        msg = conn.recv(1024).strip().decode()
        mail = message_from_string(msg.strip('.'))
        if server:
            for receiver in rcpt_to[args.name]:
                add_email(receiver, mail)
        else:
            for domain in rcpt_to:
                to = rcpt_to[domain]
                if domain == args.name:
                    for receiver in to:
                        add_email(receiver, mail)
                else:
                    SMTPClient(sender, to, mail)
            conn.sendall(b'250 OK\r\n')
            sent = True
    else:
        conn.sendall(b'503 Bad sequence of command\r\n')
```

Error 5 命令不合法

命令未实现或不合标准，返回错误状态码。

```
else:
```

```
    conn.sendall(b'501 Syntax error\r\n')
```