

PA1 for CS305 2023 Fall: SMTP Server

12110644周思呈

Task 1&2: Basic Requirements

Screenshot of test script:

```
server.py .../src 1  server.py .../as1-benchmark-release 1, U  agent.py 1
repo > CS305-23F-Assignment-1 > as1-benchmark-release > server.py > ...
问题 3  输出 调试控制台 端口 终端
(base) zhousicheng@zhousichengdeMacBook-Pro as1-benchmark-release % python test.py
Setup scenario: POP3 USER, PASS and QUIT
TEST>>> Step: Login the account in different domain
TEST>>> Step: quit
Teardown scenario: POP3 USER, PASS and QUIT
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: [{"msg_num": 1}]
    >>> Actual: [{"msg_num": 1, "octets": 171}]
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 in the different domain and test STAT
TEST>>> Step: Send email
TEST>>> Step: wait
TEST>>> Step: Login my account
TEST>>> Step: Show the status
    >>> Expect: [{"count": 1}]
    >>> Actual: [{"count": 1, "size": 171}]
TEST>>> Step: quit
Teardown scenario: Send an email to another user in the different domain and test STAT
Setup scenario: Send an email to another user in the same domain and test RETR, DELE, RSET and NOOP
TEST>>> Step: Send email
TEST>>> Step: wait
TEST>>> Step: Login the account in different domain
TEST>>> Step: Retrieve the email
    >>> Expect: {'from': 'usr1@mail.sustech.edu.cn', 'subject': 'Test Subject2'}
    >>> Actual: {'from': 'usr1@mail.sustech.edu.cn', 'subject': 'Test Subject2'}
TEST>>> Step: Delete the email
    >>> Expect: {'resp': "b'OK'"}
    >>> Actual: {'resp': "b'OK' message 1 marked for deletion'"}
TEST>>> Step: Reset the deletion
    >>> Expect: {'resp': "b'OK'"}
    >>> Actual: {'resp': "b'OK' revoke all DELE commands'"}
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
```

```

    >>> Actual: [{'msg_num': 1, 'octets': 171}]
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: [{'msg_num': 1}]
    >>> Actual: [{'msg_num': 1, 'octets': 176}]
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: 12110644
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

```

Screenshot of Wireshark during the test procedure (too many packages, choose some to show):

No.	Time	Source	Destination	Protocol	Length	Info
41	0.546820	127.0.0.1	127.0.0.1	TLSv1...	573	Client Hello
42	0.546833	127.0.0.1	127.0.0.1	TCP	56	7890 -> 62239 [ACK] Seq=40 Ack=658 Win=407616 Len=0 TSval=1245586271 TSecr=2928
43	0.579332	127.0.0.1	127.0.0.1	TCP	68	62259 -> 54530 [SYN] Seq=0 Win=65535 Len=0 MSS=16344 WS=64 TSval=112980873 TSecr=254
44	0.579462	127.0.0.1	127.0.0.1	TCP	68	54530 -> 62259 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=16344 WS=64 TSval=254
45	0.579483	127.0.0.1	127.0.0.1	TCP	56	62259 -> 54530 [ACK] Seq=1 Ack=1 Win=408256 Len=0 TSval=112980873 TSecr=2547199
46	0.579501	127.0.0.1	127.0.0.1	TCP	56	[TCP Window Update] 54530 -> 62259 [ACK] Seq=1 Ack=1 Win=408256 Len=0 TSval=254
47	0.579621	127.0.0.1	127.0.0.1	TCP	56	62236 -> 62235 [FIN, ACK] Seq=1 Ack=1 Win=6376 Len=0 TSval=2369772241 TSecr=641
48	0.579651	127.0.0.1	127.0.0.1	TCP	56	62235 -> 62236 [ACK] Seq=1 Ack=2 Win=6378 Len=0 TSval=641962166 TSecr=236977224
49	0.579689	127.0.0.1	127.0.0.1	TCP	56	62235 -> 62236 [FIN, ACK] Seq=1 Ack=2 Win=6378 Len=0 TSval=641962166 TSecr=2369
50	0.579723	127.0.0.1	127.0.0.1	TCP	56	62236 -> 62235 [ACK] Seq=2 Ack=2 Win=6376 Len=0 TSval=2369772241 TSecr=64196216
51	0.597741	127.0.0.1	127.0.0.1	TLSv1...	288	Application Data
52	0.597763	127.0.0.1	127.0.0.1	TCP	56	61398 -> 7890 [ACK] Seq=985 Ack=233 Win=5843 Len=0 TSval=1205260716 TSecr=29919
53	0.598179	127.0.0.1	127.0.0.1	TLSv1...	573	Client Hello
54	0.598200	127.0.0.1	127.0.0.1	TCP	56	54530 -> 62259 [ACK] Seq=1 Ack=518 Win=407744 Len=0 TSval=2547199870 TSecr=1129
55	0.598401	127.0.0.1	127.0.0.1	TLSv1...	1086	Server Hello, Certificate, Server Hello Done
56	0.598433	127.0.0.1	127.0.0.1	TCP	56	62259 -> 54530 [ACK] Seq=518 Ack=1031 Win=407232 Len=0 TSval=112980892 TSecr=25
57	0.598984	127.0.0.1	127.0.0.1	TLSv1...	374	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
58	0.599004	127.0.0.1	127.0.0.1	TCP	56	54530 -> 62259 [ACK] Seq=1031 Ack=836 Win=407424 Len=0 TSval=2547199871 TSecr=1
59	0.602476	127.0.0.1	127.0.0.1	TLSv1...	288	Application Data
60	0.602501	127.0.0.1	127.0.0.1	TCP	56	61398 -> 7890 [ACK] Seq=985 Ack=465 Win=5839 Len=0 TSval=1205260721 TSecr=29919
61	0.602944	127.0.0.1	127.0.0.1	TLSv1...	107	Change Cipher Spec, Encrypted Handshake Message
62	0.603010	127.0.0.1	127.0.0.1	TCP	56	62259 -> 54530 [ACK] Seq=836 Ack=1082 Win=407168 Len=0 TSval=112980897 TSecr=25
63	0.603116	127.0.0.1	127.0.0.1	TCP	68	62261 -> 62260 [SYN] Seq=0 Win=65535 Len=0 MSS=16344 WS=64 TSval=3066604639 TSecr=1129808
64	0.603213	127.0.0.1	127.0.0.1	TLSv1...	176	Application Data
65	0.603362	127.0.0.1	127.0.0.1	TCP	68	62260 -> 62261 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=16344 WS=64 TSval=198
66	0.603377	127.0.0.1	127.0.0.1	TCP	56	54530 -> 62259 [ACK] Seq=1082 Ack=956 Win=407296 Len=0 TSval=2547199875 TSecr=1129808

Task 3: Advanced Functions

More Commands

LIST

Usage: LIST/LIST[x]

Description: Returns the total number of bytes for the specified message. If not specified, returns all separately.

Screenshot of code:

```
89     # list
90     elif command.lower().startswith('list'):
91         index = command.split(" ")[-1]
92         if index.lower()=='list': # index not specified, returns all separately
93             msgs = MAILBOXES[username]
94             conn.sendall(f'+OK {len(msgs) - len(DELETE_INDEXES[username])} messages\r\n'.encode('utf-8'))
95             for i, msg in enumerate(msgs):
96                 if i+1 in DELETE_INDEXES[username]: continue
97                 conn.sendall(f'{i+1} {len(msg.data)}\r\n'.encode('utf-8'))
98             conn.sendall(b'\r\n')
99             print('Response: +OK\n')
100        else:
101            index = int(index)
102            if index > len(MAILBOXES[username]) or index < 1 or index in DELETE_INDEXES[username]:
103                conn.sendall(b'=ERR Index out of range!\r\n')
104            else:
105                conn.sendall(f'+OK {index-1} {len(MAILBOXES[username][index-1].data)}\r\n'.encode('utf-8'))
106                print('Response: +OK\n')
```

Screenshot of test:

Server:

```

Received: rcpt T0:<usr1@mail.sustech.edu.cn>
Response: 250 OK

Received: data
Response: 354 Start mail input; end with <CRLF>.<CRLF>

data: b'Content-Type: text/plain; charset="utf-8"\r\nMIME-Version: 1.0\r\nContent-Transfer-Encoding: base64\r\nSubject: a\r\nFrom: usr1@mail.sustech.edu.cn\r\n\r\nYQ==\r\n.\r\n'
Response: 250 OK

Received: quit
Response: 221 Bye

Received: ehlo 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa
Response: 250 HELLO

Received: mail FROM:<usr1@mail.sustech.edu.cn>
usr1@mail.sustech.edu.cn
Response: 250 OK

Received: rcpt T0:<usr1@mail.sustech.edu.cn>
Response: 250 OK

Received: data
Response: 354 Start mail input; end with <CRLF>.<CRLF>

data: b'Content-Type: text/plain; charset="utf-8"\r\nMIME-Version: 1.0\r\nContent-Transfer-Encoding: base64\r\nSubject: a\r\nFrom: usr1@mail.sustech.edu.cn\r\n\r\nncw==\r\n.\r\n'
Response: 250 OK

Received: quit
Response: 221 Bye

Received: USER usr1@mail.sustech.edu.cn
Response: +OK

Received: PASS pass1
Response: +OK

Received: LIST
Response: +OK

Received: LIST 2
Response: +OK

```

Client:

```

(base) zhousicheng@zhousichengdeMacBook-Pro src % python agent.py -e usr1@mail.sustech.edu.cn -p pass1
[smtp|pop|exit]>>> smtp
To: usr1@mail.sustech.edu.cn
To:
Subject: a
Content: a
[smtp|pop|exit]>>> smtp
To: usr1@mail.sustech.edu.cn
To:
Subject: a
Content: s
[smtp|pop|exit]>>> pop
b'+OK POP3 Server Ready'
b'+OK'
b'+OK user successfully logged on'
[pop]>>> list
[b'1 152', b'2 152']
[pop]>>> list 2
b'+OK 1 152'
[pop]>#

```

UIDL

Usage: UIDL/UIDL [x]

Description: Retrieve unique ID of each email. Each mail will be encapsulated as an object and has a unique ID which is generated by hashing while initialization.

Screenshot of code:

```

49 class MAIL():
50     def __init__(self, data):
51         self.data = data
52         hash_object.update(data)
53         # 获取哈希值的十六进制表示
54         self.id = hash_object.hexdigest()
55
56
57         # uidl
58         elif command.lower() == 'uidl':
59             self.request.sendall(b"+OK Unique ID listing follows\r\n")
60             for i, mail in enumerate(MAILBOXES[username], 1):
61                 response = f"{i} {mail.id}\r\n"
62                 conn.sendall(response.encode('utf-8'))
63             conn.sendall(b".\r\n")
64             print('Response: +OK\n')
65         elif command.lower().startswith("uidl"):
66             index = int(command.split(" ")[-1])
67             if index > len(MAILBOXES[username]) or index < 1 or index in DELETE_INDEXES[username]:
68                 conn.sendall(b"-ERR Index out of range!\r\n")
69             else:
70                 conn.sendall(f'+OK {index} {MAILBOXES[username][index-1].id}\r\n'.encode('utf-8'))
71             print('Response: +OK\n')

```

Screenshots of test:

```

(base) zhousicheng@zhousichengdeMacBook-Pro src % python agent.py -e usr1@mail.sustech.edu.cn -p pas
s1
[smtp|pop|exit]>>> smtp
To: usr1@mail.sustech.edu.cn
To:
Subject: a
Content: a
[smtp|pop|exit]>>> smtp
To: usr1@mail.sustech.edu.cn
To:
Subject: s
Content: s
[smtp|pop|exit]>>> pop
b'+OK POP3 Server Ready'
b'+OK'
b'+OK user successfully logged on'
[pop]>>> list
[b'1 152', b'2 152']
[pop]>>> uidl
(b'+OK Unique ID listing follows', [b'1 5b2ebe4ff9205fe6c205393fad4b31ecc39a99f2012d5c8af6278ecea28d
411b', b'2 531626b4275b98e8e1af2d46342bd879bbfad625558b90f852d410c643668777'], 136)
[pop]>>> uidl 2
b'+OK 2 531626b4275b98e8e1af2d46342bd879bbfad625558b90f852d410c643668777'

```

TOP

Usage: TOP [msg] [n]

Description: Retrieve the first [n] lines in [msg].

Screenshot of code:

```
158 # top
159 elif command.lower().startswith("top"):
160     msg = int(command.split(" ")[1])
161     lines = int(command.split(" ")[2])
162     if msg > len(MAILBOXES[username]) or msg < 1 or msg in DELETE_INDEXES[username] or lines < 0:
163         conn.sendall(b'-ERR Index out of range!\r\n')
164     else:
165         conn.sendall(b'+OK\r\n')
166         # send it all
167         if lines > len(MAILBOXES[username][msg-1].data.split(b"\n")):
168             conn.sendall(MAILBOXES[username][msg-1].data)
169         # send top lines of the mail
170         else:
171             for line in MAILBOXES[username][msg-1].data.split(b"\n")[0:lines]:
172                 conn.sendall(line)
173             conn.sendall(b"\r\n.\r\n")
174     print('Response: +OK\r\n')
```

Screenshot of test:

Peer Mailing

My friend's agent send a mail to his gmail server, then his gmail server send this message to my exmail server.

Modified code (10.32.8.214 is my IP address):

```

def sendMail(sender, receptors, message):
    failed_rcpt=[]
    for rcpt in receptors:
        mx=rcpt.split('@')[1]
        mx=fdns_query(mx,'MX')
        PORT=int(fdns_query(mx,'P'))
        if PORT==SMTP_PORT:
            MAILBOXES[rcpt].append(_Mail(message))
            # print(len(MAILBOXES))
        else:
            soc=socket.create_connection(("10.32.8.214", PORT), socket._GLOBAL_DEFAULT_TIMEOUT)
            # soc=socket.create_connection("", PORT), socket._GLOBAL_DEFAULT_TIMEOUT)
            file=soc.makefile('rb')

```

Screenshot of my friend's cmd:

```

(Computer_Networks) qinyuzhou@QinYuzhoudeMacBook-Pro src % python agent.py -e usr@gmail.com -p pass
[smtp|pop|exit]>>> smtp
To: usr1@mail.sustech.edu.cn
To:
Subject: A
Content:
[smtp|pop|exit]>>> smtp
To: usr1@mail.sustech.edu.cn
To:
Subject: Love Sister
Content:
[smtp|pop|exit]>>>
Invalid command
[smtp|pop|exit]>>> smtp
To: usr1@mail.sustech.edu.cn
To:
Subject: A
Content: A
[smtp|pop|exit]>>> smtp

```

Screenshot of my cmd:

```

(base) zhousicheng@zhousichengdeMacBook-Pro src % python server.py -n exmail.qq.com
Received: MAIL FROM:<usr@gmail.com>
usr@gmail.com
Response: 250 OK

Received: RCPT TO:<usr1@mail.sustech.edu.cn>
Response: 250 OK

Received: DATA
Response: 354 Start mail input; end with <CRLF>.<CRLF>

data: b'Content-Type: text/plain; charset="utf-8"\r\nMIME-Version: 1.0\r\nContent-Transfer-Encoding: base64
\r\nSubject: Love Sister\r\nFrom: usr@gmail.com\r\n\r\n.\r\n'
Response: 250 OK

Received: QUIT
Response: 221 Bye

```

Screenshot of Wireshark:

Wireshark - peer_mailing.pcapng

tcp.stream eq 60

No.	Time	Source	Destination	Protocol	Length	Info
348	23.975366	10.32.40.94	10.32.8.214	TCP	78	52229 → 1025 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=3840278898 TSecr=0
349	23.975809	10.32.8.214	10.32.40.94	TCP	78	1025 → 52229 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=64 TSval=10726
350	23.985152	10.32.40.94	10.32.8.214	TCP	66	52229 → 1025 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=3840278916 TSecr=1072609
351	23.985251	10.32.8.214	10.32.40.94	TCP	66	[TCP Window Update] 1025 → 52229 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=1072609
352	23.986082	10.32.8.214	10.32.40.94	SMTP	89	S: 220 SMTP Server Ready
353	23.991470	10.32.40.94	10.32.8.214	TCP	66	52229 → 1025 [ACK] Seq=1 Ack=24 Win=131712 Len=0 TSval=3840278922 TSecr=107260
354	23.991473	10.32.40.94	10.32.8.214	SMTP	93	C: MAIL FROM:<usr@gmail.com>
355	23.991613	10.32.8.214	10.32.40.94	TCP	66	1025 → 52229 [ACK] Seq=24 Ack=28 Win=131712 Len=0 TSval=1072609624 TSecr=38402
356	23.992264	10.32.8.214	10.32.40.94	SMTP	97	S: 250 Sender <usr@gmail.com> OK
357	23.998940	10.32.40.94	10.32.8.214	TCP	66	52229 → 1025 [ACK] Seq=28 Ack=55 Win=131712 Len=0 TSval=3840278929 TSecr=10726
358	23.998942	10.32.40.94	10.32.8.214	SMTP	102	C: RCPT TO:<usr@mail.sustech.edu.cn>
359	23.999083	10.32.8.214	10.32.40.94	TCP	66	1025 → 52229 [ACK] Seq=55 Ack=64 Win=131648 Len=0 TSval=1072609631 TSecr=38402
360	23.999326	10.32.8.214	10.32.40.94	SMTP	111	S: 250 Recipient <usr@mail.sustech.edu.cn> OK
366	24.012782	10.32.40.94	10.32.8.214	TCP	66	52229 → 1025 [ACK] Seq=64 Ack=100 Win=131648 Len=0 TSval=3840278943 TSecr=10726
367	24.012782	10.32.40.94	10.32.8.214	SMTP	72	C: DATA
368	24.012843	10.32.8.214	10.32.40.94	TCP	66	1025 → 52229 [ACK] Seq=100 Ack=70 Win=131648 Len=0 TSval=1072609645 TSecr=38402
369	24.013014	10.32.8.214	10.32.40.94	SMTP	112	S: 354 Start mail input; end with <CRLF> <CRLF>
370	24.018564	10.32.40.94	10.32.8.214	TCP	66	52229 → 1025 [ACK] Seq=70 Ack=146 Win=131584 Len=0 TSval=3840278950 TSecr=10726
371	24.018565	10.32.40.94	10.32.8.214	SMTP/...	211	subject: Love Sister, from: usr@gmail.c, (text/plain)
372	24.018650	10.32.8.214	10.32.40.94	TCP	66	1025 → 52229 [ACK] Seq=146 Ack=215 Win=131520 Len=0 TSval=1072609651 TSecr=38402
373	24.018802	10.32.8.214	10.32.40.94	SMTP	101	S: 250 Message accepted for delivery
374	24.025249	10.32.40.94	10.32.8.214	TCP	66	52229 → 1025 [ACK] Seq=215 Ack=181 Win=131584 Len=0 TSval=3840278956 TSecr=1072609652
375	24.025251	10.32.40.94	10.32.8.214	SMTP	72	C: QUIT
376	24.025349	10.32.8.214	10.32.40.94	TCP	66	1025 → 52229 [ACK] Seq=181 Ack=221 Win=131520 Len=0 TSval=1072609657 TSecr=38402
377	24.025521	10.32.8.214	10.32.40.94	SMTP	75	S: 221 Bye
378	24.025580	10.32.8.214	10.32.40.94	TCP	66	1025 → 52229 [FIN, ACK] Seq=190 Ack=221 Win=131520 Len=0 TSval=1072609657 TSecr=38402

Total Length: 64
Identification: 0x0000 (0)
> 010. = Flags: 0x2, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 63
Protocol: TCP (6)
Header Checksum: 0xf644 [validation disabled]
[Header checksum status: Unverified]
Source Address: 10.32.40.94

Source Address (ip.src), 4 bytes

Packets: 770 · Displayed: 30 (3.9%) · Dropped: 0 (0.0%) · Profile: Default

Wireshark - Follow TCP Stream (tcp.stream eq 60) · peer_mailing.pcapng

tcp.stream eq 60

```

220 SMTP Server Ready
MAIL FROM:<usr@gmail.com>
250 Sender <usr@gmail.com> OK
RCPT TO:<usr1@mail.sustech.edu.cn>
250 Recipient <usr1@mail.sustech.edu.cn> OK
DATA
354 Start mail input; end with <CRLF>.<CRLF>
Content-Type: text/plain; charset="utf-8"
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Subject: Love Sister
From: usr@gmail.com

.
250 Message accepted for delivery
QUIT
221 Bye

```

Total Length: 52
Identification: 0x0000 (0)
> 010. = Flags: 0x2, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 64
Protocol: TCP (6)
Header Checksum: 0xf550 [validation disabled]
[Header checksum status: Unverified]
Source Address: 10.32.8.214

Source Address (ip.src), 4 bytes

Entire conversation (409 bytes) Show data as ASCII Stream 60

Find: Find Next

Help Filter Out This Stream Print Save as... Back Close

3.9% · Dropped: 0 (0.0%) · Profile: Default