

Caleb Naeger and Landon Spitzer

Professor Ahmad

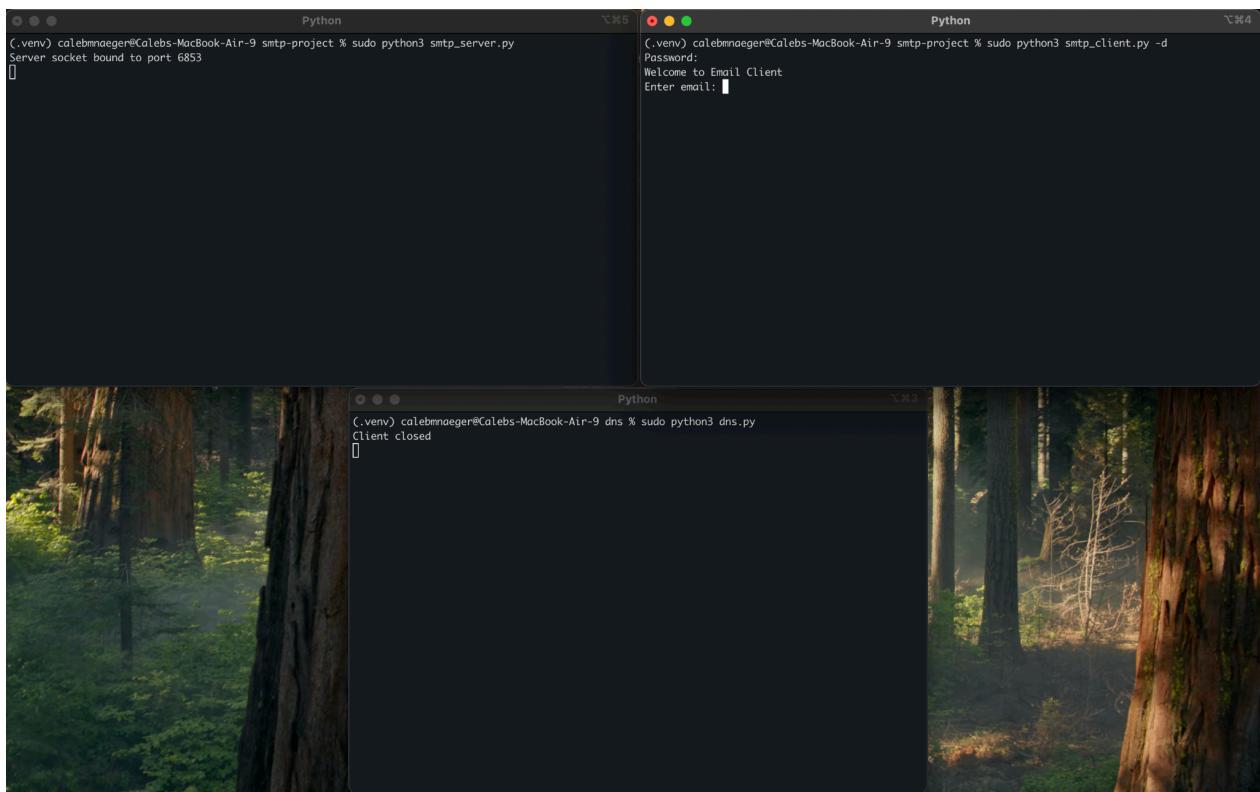
Foundations of Computer Networks

18 April 2025

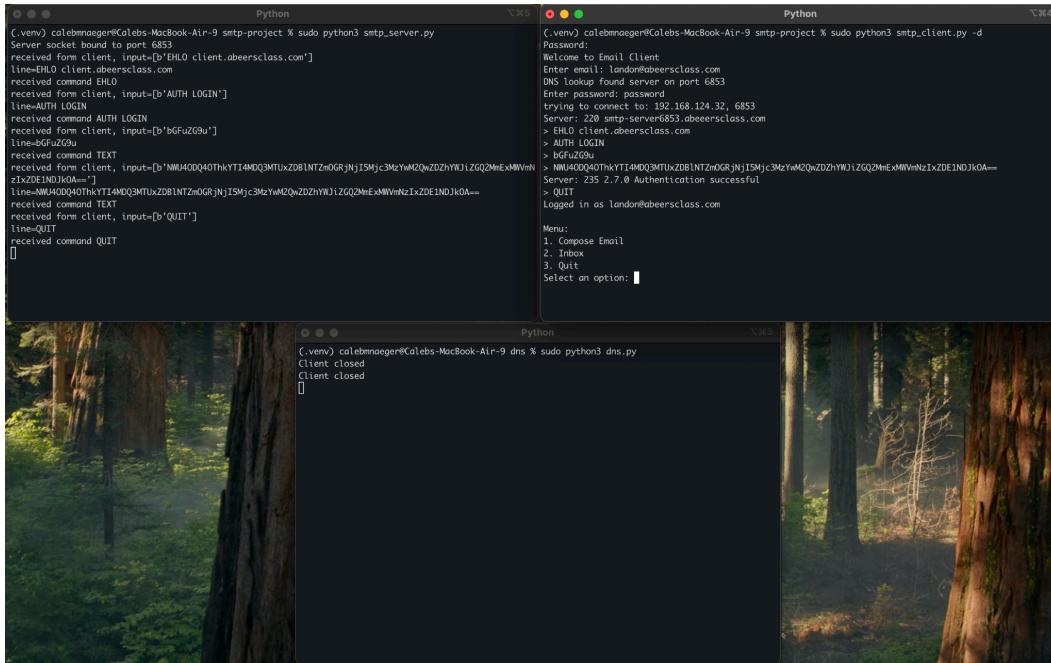
## SMTP/POP3 Project Report

<https://github.com/lbs9440/smtp-project>

Example of the initial setup of the SMTP Server (top-left), Client (top-right), and DNS server (bottom). The SMTP server is currently listening for a connection, while the Client terminal prompts the user for their login credentials. On startup the SMTP server alerted the DNS of its new IP and port. All of the following is with the DEBUG flag turned on (when this flag is off, the user is given a cleaner, more user-friendly, interface).



Once the server validates the client's credentials, the client is prompted with a menu to either compose an email, view their inbox, or quit the session. The user must select an option 1, 2, or 3.

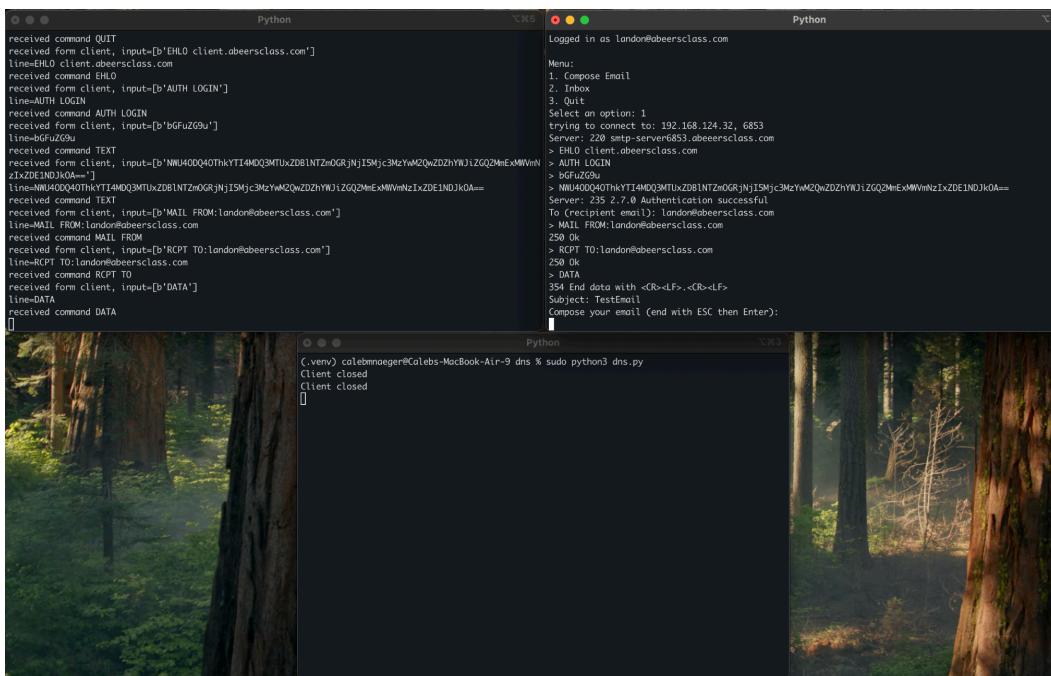


```
(venv) calebmaeager@Calebs-MacBook-Air-9 smtp-project % sudo python3 smtp_server.py
Server socket bound to port 6853
received form client, input=[b'EHLO client.abeersclass.com']
line=EHLO client.abeersclass.com
received command EHLO
received form client, input=[b'AUTH LOGIN']
line=AUTH LOGIN
received command AUTH LOGIN
received form client, input=[b'bgFuZG9u']
line=bgFuZG9u
received command TEXT
received form client, input=[b'NMU400040THkYT14MDQ3MTUzD81NT2m0GRjNjISMjc3MzYwM2QwZDZhYUjZGQ2MnExMWVm
zIxZDE1NDjKOa==']
line=NWU400040THkYT14MDQ3MTUzD81NT2m0GRjNjISMjc3MzYwM2QwZDZhYUjZGQ2MnExMWVm
zIxZDE1NDjKOa==']
received command TEXT
received form client, input=[b'QUIT']
line=QUIT
received command QUIT
[]
```

```
(venv) calebmaeager@Calebs-MacBook-Air-9 smtp-project % sudo python3 smtp_client.py -d
Password:
Name: london@abeersclass.com
Enter email: london@abeersclass.com
DNS lookup found server on port 6853
Enter password: password
trying to connect to: 192.168.124.32, 6853
Server: 220 smtp-server6853.abeersclass.com
> EHLO client.abeersclass.com
> AUTH LOGIN
> bgFuZG9u
> NMU400040THkYT14MDQ3MTUzD81NT2m0GRjNjISMjc3MzYwM2QwZDZhYUjZGQ2MnExMWVm
zIxZDE1NDjKOa=='
Server: 235 2.7.0 Authentication successful
> QUIT
Logged in as london@abeersclass.com
```

```
Menu:
1. Compose Email
2. Inbox
3. Quit
Select an option: 1
```

Here the user selected option 1, in order to compose an email. The user is then re-authenticated as per SMTP protocol (the user does not have to re-enter their credentials), then is prompted to enter an email address to send to, the Subject of their email, and the contents of the email to send



```
Logged in as london@abeersclass.com
Menu:
1. Compose Email
2. Inbox
3. Quit
Select an option: 1
trying to connect to: 192.168.124.32, 6853
Server: 220 smtp-server6853.abeersclass.com
> EHLO client.abeersclass.com
> AUTH LOGIN
> bgFuZG9u
> NMU400040THkYT14MDQ3MTUzD81NT2m0GRjNjISMjc3MzYwM2QwZDZhYUjZGQ2MnExMWVm
zIxZDE1NDjKOa=='
Server: 235 2.7.0 Authentication successful
To (Recipient email): london@abeersclass.com
> MAIL FROM:london@abeersclass.com
250 Ok
> RCPT TO:london@abeersclass.com
250 Ok
> DATA
354 End data with <R><LF>,<CR><LF>
Subject: TestEmail
Compose your email (end with ESC then Enter):
```

```
(venv) calebmaeager@Calebs-MacBook-Air-9 dns % sudo python3 dns.py
Client closed
Client closed
[]
```

Once the email is sent, the menu is displayed again to the user through the client, and the server handles the mail forwarding.

The screenshot shows two terminal windows side-by-side. The left window is a Python script running on a client machine, and the right window is a Python script running on a server machine. The client window shows the process of sending an email to 'london@obeersclass.com'. It includes code for receiving input from the client, handling the message body, and sending it via SMTP. The server window shows the process of receiving the email and then sending it via POP3. It includes code for receiving the message, adding it to a maildrop, and then retrieving it via POP3. Both windows show the final output of the email being sent and the menu being displayed again.

```

Python
received form client, input=[b'Subject: TestEmail', b'', b'This is a test email. \nAnd another line\n\nFinally,\nDone writing.\n']  
line  
line<Subject: TestEmail  
line  
line<This is a test email.  
And another line  
finally,  
done writing  
line  
received command TEXT  
Added line to message: b'Subject: TestEmail'  
received command TEXT  
Added line to message: b'  
received command TEXT  
Added line to message: b'This is a test email. \nAnd another line\n\nFinally,\nDone writing.'  
received command TEXT  
Added line to message: b'  
To domain = obeersclass.com  
Updating Emails  
received form client, input=[b'QUIT']  
line<QUIT  
received command QUIT
Python
> NW400QH0THKTYI4MDQ3MTUxZD81NTZn0GRjNjSmjc3MzYmZ2wZDZhWJiZQ2NmExMVmNzIxZDE1NDJkOA==  
Server: 2.7.0 Authentication successful  
To: london@obeersclass.com  
250 Ok  
> RCPT TO:london@obeersclass.com  
250 Ok  
> DATA  
350 End data with <R><LF><R><LF>  
Subject: TestEmail  
Compose your email (end with ESC then Enter):  
This is a test email.  
And another line  
Finally,  
done writing  
250 Ok: queued  
> QUIT  
Menu:  
1. Compose Email  
2. Inbox  
3. Quit  
Select on option: 1
Python
(&venv) calebmaeager@Calebs-MacBook-Air-9 dns % sudo python3 dns.py  
Client closed  
Client closed

```

Following this, option 2 is selected to view the user's inbox. The amount of messages along with the size of the total inbox is given, as well as the first 10 messages waiting for the user. The user can go through pages of 10 summaries including the From address and the subject of each email.

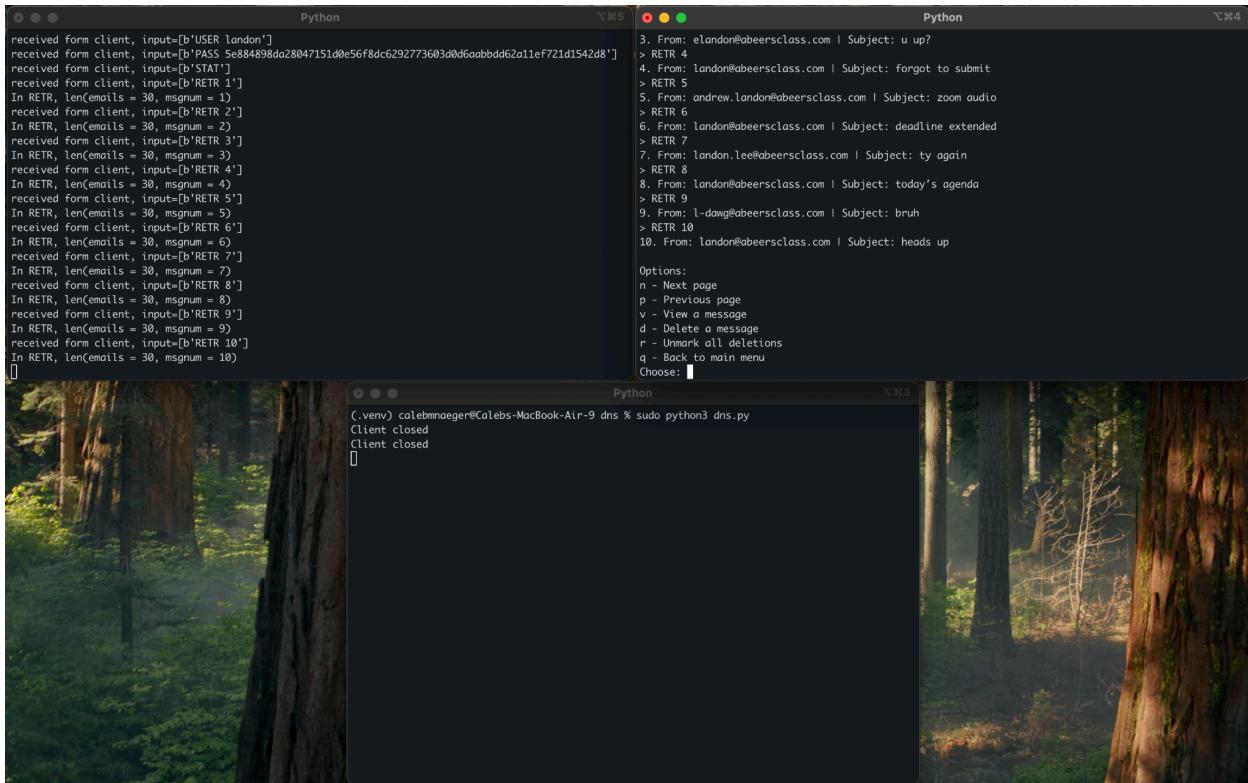
The screenshot shows two terminal windows side-by-side. The left window is a Python script running on a client machine, and the right window is a Python script running on a server machine. The client window shows the user selecting option 2 from the menu to view their inbox. It displays the total number of messages (30), the size of the inbox (2032 octets), and then lists the first 10 messages in the inbox, including the From address and Subject. The server window shows the process of retrieving messages via POP3. It includes code for retrieving messages from the maildrop and displaying them to the user. Both windows show the final output of the inbox being viewed and the menu being displayed again.

```

Python
received Form Client, input=[b'USER london']  
received Form Client, input=[b'PASS 5e884898da28047151d0e56f8dc6292773603d0d6adbd62a11ef721d1542d8']  
received Form Client, input=[b'STAT']  
received Form Client, input=[b'RETR 1']  
In RETR, len(emails) = 30, msgnum = 1  
received form client, input=[b'RETR 2']  
In RETR, len(emails) = 30, msgnum = 2  
received form client, input=[b'RETR 3']  
In RETR, len(emails) = 30, msgnum = 3  
received form client, input=[b'RETR 4']  
In RETR, len(emails) = 30, msgnum = 4  
received form client, input=[b'RETR 5']  
In RETR, len(emails) = 30, msgnum = 5  
received form client, input=[b'RETR 6']  
In RETR, len(emails) = 30, msgnum = 6  
received form client, input=[b'RETR 7']  
In RETR, len(emails) = 30, msgnum = 7  
received form client, input=[b'RETR 8']  
In RETR, len(emails) = 30, msgnum = 8  
received form client, input=[b'RETR 9']  
In RETR, len(emails) = 30, msgnum = 9  
received form client, input=[b'RETR 10']  
In RETR, len(emails) = 30, msgnum = 10
Python
Menu:  
1. Compose Email  
2. Inbox  
3. Quit  
Select on option: 2  
Server: +OK pop3-server#110 obeersclass.com POP3 server ready  
> USER london  
Server: +OK london  
> PASS 5e884898da28047151d0e56f8dc6292773603d0d6adbd62a11ef721d1542d8  
Server: +OK london's maildrop has 30 messages (2032 octets)  
> STAT  
Server: +OK 30 2032  
Showing messages 1 to 10 of 30:  
> RETR 1  
1. From: l.jameson@obeersclass.com | Subject: project files  
> RETR 2  
2. From: london@obeersclass.com | Subject: dinner?  
> RETR 3  
3. From: elandon@obeersclass.com | Subject: u up?  
> RETR 4  
4. From: london@obeersclass.com | Subject: forgot to submit  
> RETR 5
Python
(&venv) calebmaeager@Calebs-MacBook-Air-9 dns % sudo python3 dns.py  
Client closed  
Client closed

```

This image shows the rest of those 10 emails retrieved in the client terminal. As well as the options in which they can go to the next or a previous page (if one exists), view one of the messages, delete a message (marks a message for deletion once the POP3 update phase begins), unmark those emails from deletion, or head back to the main menu. The user is prompted to choose one of those options.



```

Python                                         Python
received form client, input=[b'USER landon']
received form client, input=[b'PASS 5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8']
received form client, input=[b'STAT']
received form client, input=[b'RETR 1']
In RETR, len(emails = 30, msgnum = 1)
received form client, input=[b'RETR 2']
In RETR, len(emails = 30, msgnum = 2)
received form client, input=[b'RETR 3']
In RETR, len(emails = 30, msgnum = 3)
received form client, input=[b'RETR 4']
In RETR, len(emails = 30, msgnum = 4)
received form client, input=[b'RETR 5']
In RETR, len(emails = 30, msgnum = 5)
received form client, input=[b'RETR 6']
In RETR, len(emails = 30, msgnum = 6)
received form client, input=[b'RETR 7']
In RETR, len(emails = 30, msgnum = 7)
received form client, input=[b'RETR 8']
In RETR, len(emails = 30, msgnum = 8)
received form client, input=[b'RETR 9']
In RETR, len(emails = 30, msgnum = 9)
received form client, input=[b'RETR 10']
In RETR, len(emails = 30, msgnum = 10)
[]

3. From: elandon@abeersclass.com | Subject: u up?
> RETR 4
4. From: landon@abeersclass.com | Subject: forgot to submit
> RETR 5
5. From: andrew.landon@abeersclass.com | Subject: zoom audio
> RETR 6
6. From: landon@abeersclass.com | Subject: deadline extended
> RETR 7
7. From: landon.lee@abeersclass.com | Subject: ty again
> RETR 8
8. From: landon@abeersclass.com | Subject: today's agenda
> RETR 9
9. From: l-dawg@abeersclass.com | Subject: bruh
> RETR 10
10. From: landon@abeersclass.com | Subject: heads up

Options:
n - Next page
p - Previous page
v - View a message
d - Delete a message
r - Unmark all deletions
q - Back to main menu
Choose: 

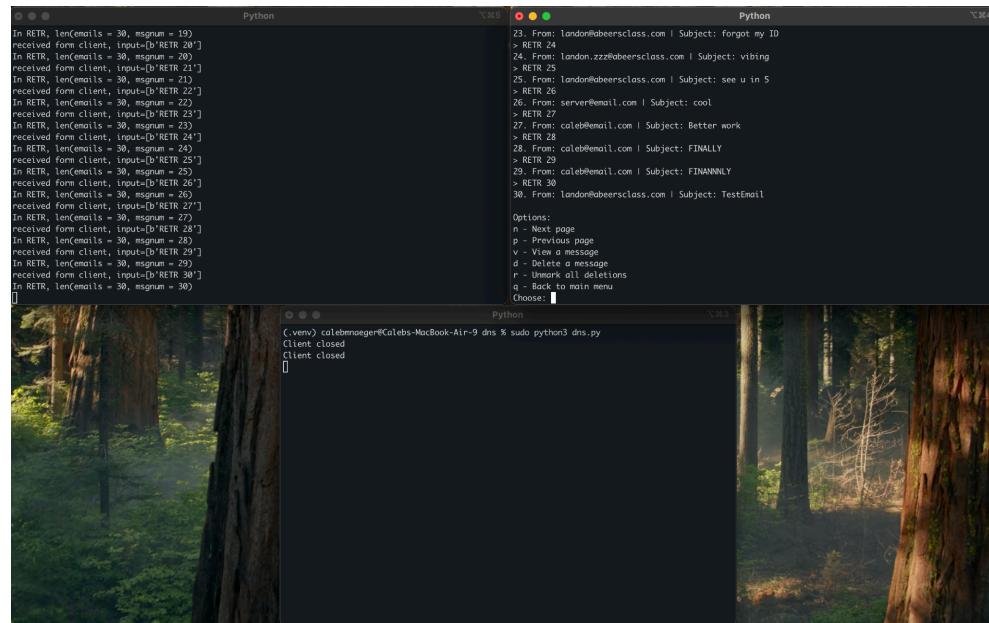
```

(venv) calebmnaege@Calebs-MacBook-Air-9 dns % sudo python3 dns.py  
Client closed  
Client closed

Here the next page option has been selected twice in order to view the 21st through 30th email.

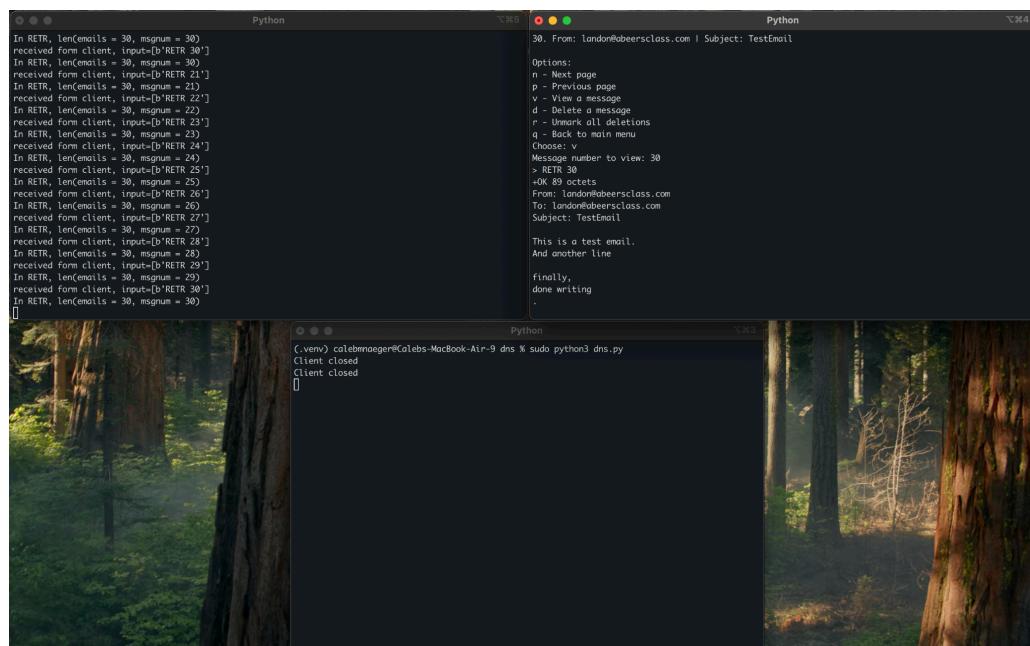
Each email is once again retrieved from the server to display the From address and the subject.

Once again the menu options are given, and the user is prompted for one of the choices.



```
In RETR, len(emails) = 30, msgnum = 19
received form client, input=b'RETR 20'
In RETR, len(emails) = 30, msgnum = 20
received form client, input=b'RETR 21'
In RETR, len(emails) = 30, msgnum = 21
received form client, input=b'RETR 22'
In RETR, len(emails) = 30, msgnum = 22
received form client, input=b'RETR 23'
In RETR, len(emails) = 30, msgnum = 23
received form client, input=b'RETR 24'
In RETR, len(emails) = 30, msgnum = 24
received form client, input=b'RETR 25'
In RETR, len(emails) = 30, msgnum = 25
received form client, input=b'RETR 26'
In RETR, len(emails) = 30, msgnum = 26
received form client, input=b'RETR 27'
In RETR, len(emails) = 30, msgnum = 27
received form client, input=b'RETR 28'
In RETR, len(emails) = 30, msgnum = 28
received form client, input=b'RETR 29'
In RETR, len(emails) = 30, msgnum = 29
received form client, input=b'RETR 30'
In RETR, len(emails) = 30, msgnum = 30
23. From: landon@beersclass.com | Subject: forgot my ID
> RETR 24
24. From: landon.zzz@beersclass.com | Subject: vibing
> RETR 25
25. From: landon@beersclass.com | Subject: see u in 5
> RETR 26
26. From: server@email.com | Subject: cool
> RETR 27
27. From: caleb@email.com | Subject: Better work
> RETR 28
28. From: caleb@email.com | Subject: FINALLY
> RETR 29
29. From: caleb@email.com | Subject: FINALLY
30. From: landon@beersclass.com | Subject: TestEmail
Options:
n - Next page
p - Previous page
v - View a message
d - Delete a message
u - Unmark all deletions
q - Back to main menu
Choose: v
```

In the case that option ‘v’ is selected for viewing a message, the user is prompted to input the corresponding number of the email they want to view. Once this is complete, the from, to, subject, and email contents are all displayed.



```
In RETR, len(emails) = 30, msgnum = 30
received form client, input=b'RETR 30'
In RETR, len(emails) = 30, msgnum = 30
received form client, input=b'RETR 21'
In RETR, len(emails) = 30, msgnum = 21
received form client, input=b'RETR 22'
In RETR, len(emails) = 30, msgnum = 22
received form client, input=b'RETR 23'
In RETR, len(emails) = 30, msgnum = 23
received form client, input=b'RETR 24'
In RETR, len(emails) = 30, msgnum = 24
received form client, input=b'RETR 25'
In RETR, len(emails) = 30, msgnum = 25
received form client, input=b'RETR 26'
In RETR, len(emails) = 30, msgnum = 26
received form client, input=b'RETR 27'
In RETR, len(emails) = 30, msgnum = 27
received form client, input=b'RETR 28'
In RETR, len(emails) = 30, msgnum = 28
received form client, input=b'RETR 29'
In RETR, len(emails) = 30, msgnum = 29
received form client, input=b'RETR 30'
In RETR, len(emails) = 30, msgnum = 30
30. From: landon@beersclass.com | Subject: TestEmail
Options:
n - Next page
p - Previous page
v - View a message
d - Delete a message
u - Unmark all deletions
q - Back to main menu
Choose: v
Message number to view: 30
> RETR 30
-OK 89 octets
From: landon@beersclass.com
To: landon@beersclass.com
Subject: TestEmail
This is a test email.
And another line.
finally,
done writing.
.
This is a test email.
And another line.
finally,
done writing.
.
(venv) calebmaege@Calebs-MacBook-Air-9: ~ % sudo python3 dns.py
Client closed
Client closed
```

The example above demonstrates how to run everything locally on a single machine. However, you can also run multiple clients and servers across physically different machines, including configuring email servers to serve different domains. For instructions on how to set this up, please refer to our README.