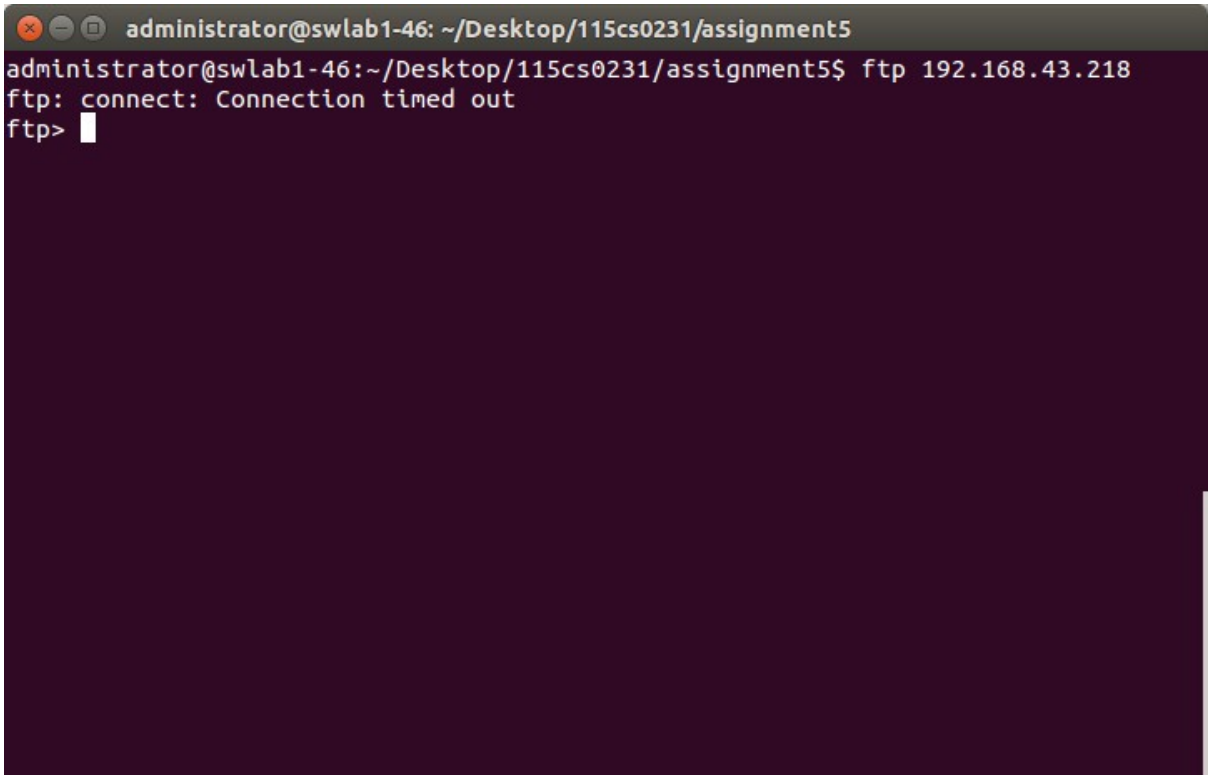
Thursday, March 28, 2019

[22] Write a program to demonstrate the loading of a local file to a remote FTP?

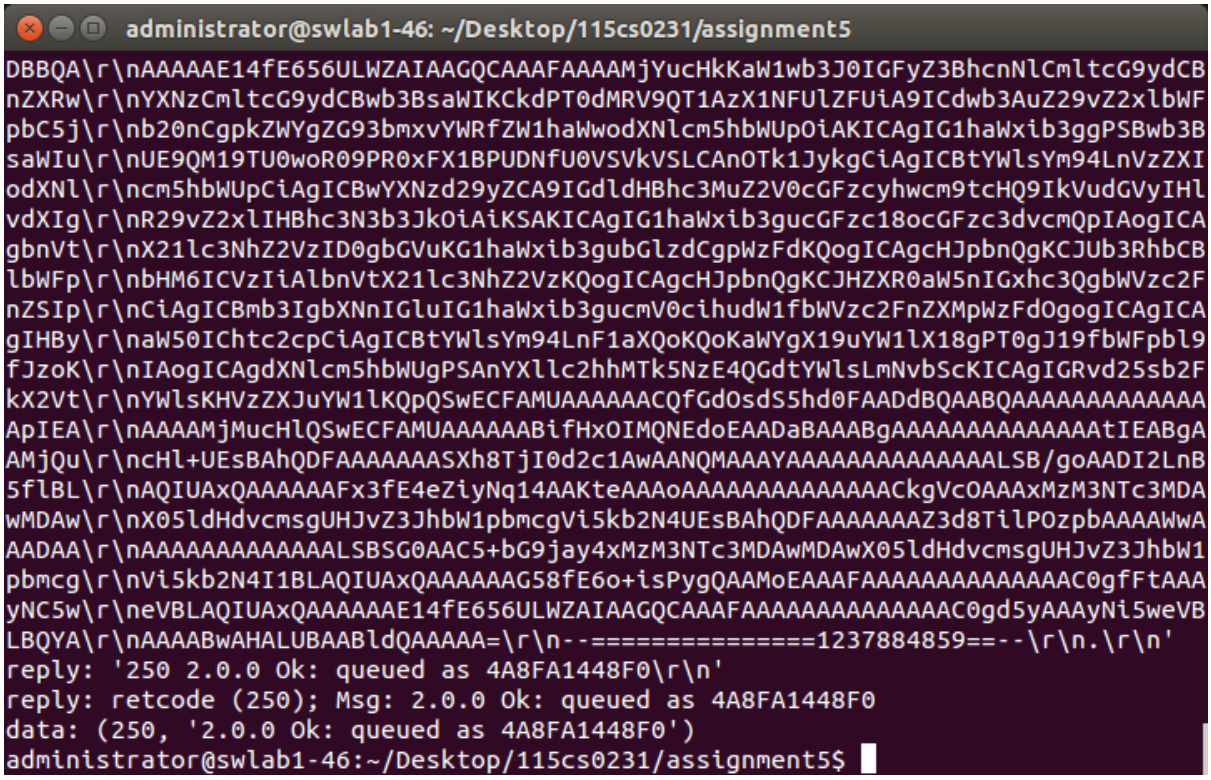
```
import os
import argparse
import ftplib
import getpass

LOCAL_FTP_SERVER = 'localhost'
LOCAL_FILE = 'file1.txt'

def ftp_upload(ftp_server, file_name):
    print ("Connecting to FTP server: %s" %ftp_server)
    ftp = ftplib.FTP(ftp_server)
    ext = os.path.splitext(file_name)[1]
    if ext in (".txt", ".htm", ".html"):
        ftp.storlines("STOR " + file_name, open(file_name))
    else:
        ftp.storbinary("STOR " + file_name, open(file_name, "rb"), 1024)
    print ("Uploaded file: %s" %file_name)


if __name__ == '__main__':
    ftp_server='192.168.43.218'
    ftp_upload(ftp_server,'file1.txt')
```



[23] Write a program that compress your current working directory and then email as a message. You can send the email message via an external Gmail SMTP host, or you can use a local email server to do this.

```
import os
import argparse
import smtplib
import zipfile
import tempfile
from email import encoders
from email.mime.base import MIMEBase
from email.mime.multipart import MIMEMultipart


sender='115cs0231@gmail.com'
recipient='115cs0221@gmail.com'


zf = tempfile.TemporaryFile(prefix='mail', suffix='.zip')
zip = zipfile.ZipFile(zf, 'w')
#print ("Zipping current dir: %s" %os.getcwd())
for file_name in os.listdir(os.getcwd()):
        zip.write(file_name)
zip.close()
zf.seek(0)

print ("Creating email message...")
email_msg = MIMEMultipart()
email_msg['Subject'] = 'File from path %s' %os.getcwd()
email_msg['To'] = ', '.join(recipient)
email_msg['From'] = sender
email_msg.preamble = 'Testing email from Python.\n'
msg = MIMEBase('application', 'zip')
msg.set_payload(zf.read())
encoders.encode_base64(msg)
msg.add_header('Content-Disposition', 'attachment',
          filename=os.getcwd()[-1] + '.zip')
email_msg.attach(msg)
email_msg = email_msg.as_string()

print ("Sending email message...")
```

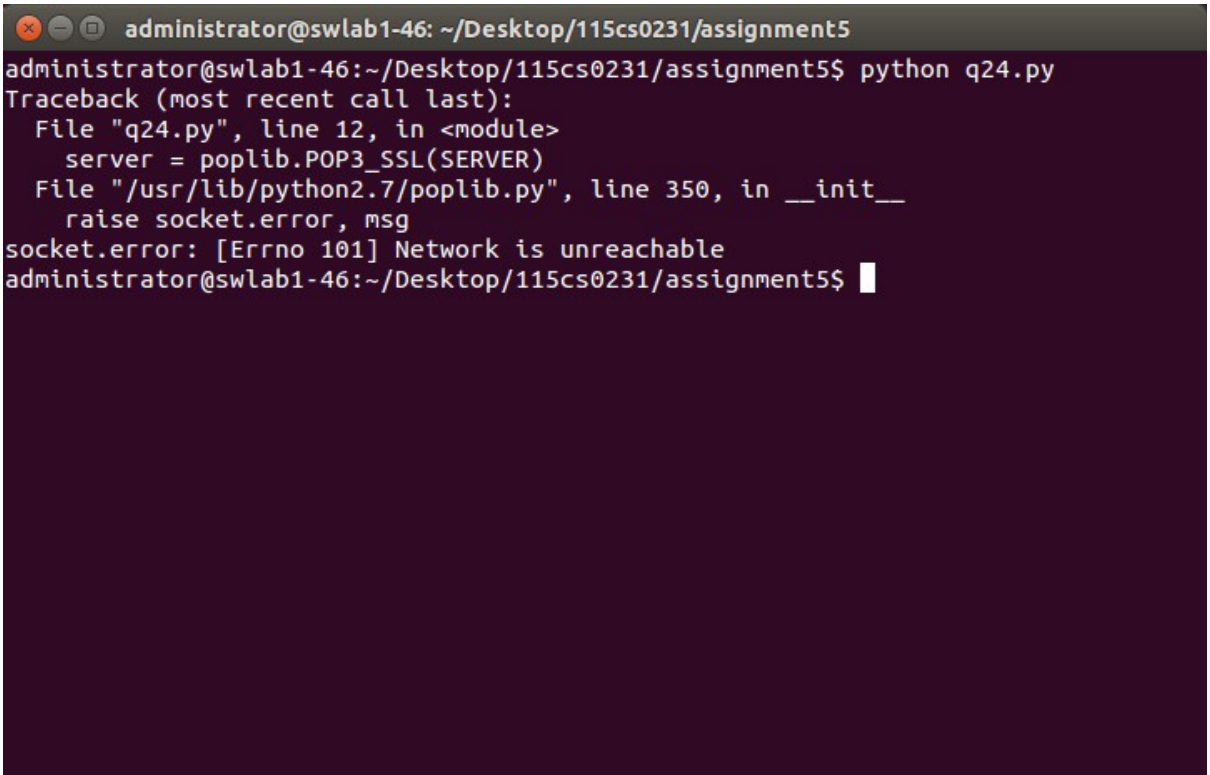
```
try:
        smtp = smtplib.SMTP('localhost')
        smtp.set_debuglevel(1)
        smtp.sendmail(sender, recipient, email_msg)
except Exception as e:
        print ("Error: %s" %str(e))
finally:
        smtp.close()
```



[24] Write a program that connects to your Google email account and read the email message. Display one or multiple email message on user request.

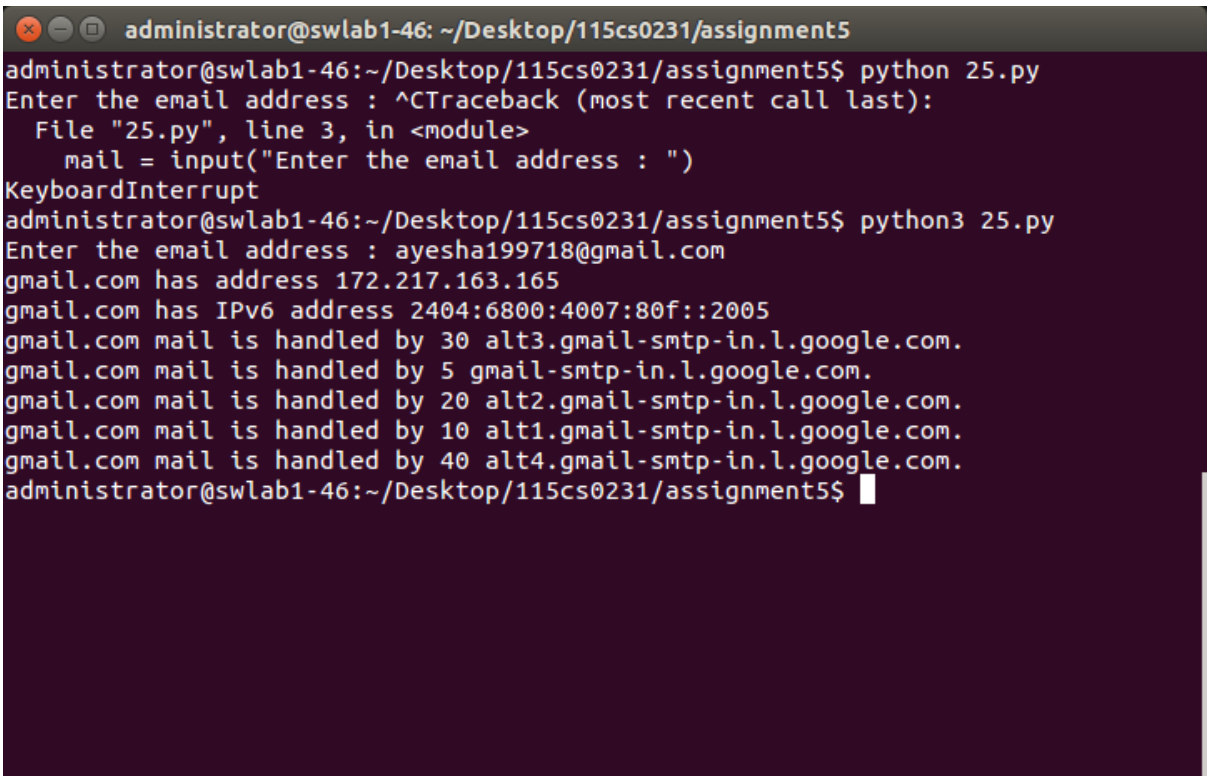
```
import poplib
import string, random
```

```
import StringIO, rfc822
import logging

SERVER = "pop.gmail.com"
USER  = "ayesha.patra2"
PASSWORD = "password"

# connect to server
logging.debug('connecting to ' + SERVER)
server = poplib.POP3_SSL(SERVER)
#server = poplib.POP3(SERVER)

# login
logging.debug('logging in')
server.user(USER)
server.pass_(PASSWORD)

# list items on server
logging.debug('listing emails')
resp, items, octets = server.list()

# download the first message in the list
id, size = string.split(items[0])
resp, text, octets = server.retr(id)

# convert list to Message object
text = string.join(text, "\n")
file = StringIO.StringIO(text)
message = rfc822.Message(file)

# output message
print(message['From']),
print(message['Subject']),
print(message['Date']),
#print(message.fp.read())
```



[25] Write a program that finds detail of the mail server from an email address.

```
import os

mail = input("Enter the email address : ")
mail = mail.split('@')
if len(mail) == 1:
	mail = mail[0]
else:
	mail = mail[1]

os.system("host "+mail)
```



[26] Write a program that connects to Google mail server with POP3 to fetch an email message from an email account.

```
import argparse
import getpass
import poplib

GOOGLE_POP3_SERVER = 'pop.googlemail.com'

def download_email(username):
    mailbox = poplib.POP3_SSL(GOOGLE_POP3_SERVER, '995')
    mailbox.user(username)
    password = getpass.getpass(prompt="Enter your Google password: ")
    mailbox.pass_(password)
    num_messages = len(mailbox.list()[1])
    print ("Total emails: %s" %num_messages)
    print ("Getting last message")
    for msg in mailbox.retr(num_messages)[1]:
        print (msg)
    mailbox.quit()

if __name__ == '__main__':

    username = 'ayesha199718@gmail.com'
    download_email(username)
```