

# Computer Network Laboratory

*Programming with FTP, CGI and Email*

Jiawei Chang

Dept. of Computer Science and Information Engineering  
National Taichung University of Science and Technology

# Outline

1. 27\_list\_files\_on\_ftp\_server
2. 28\_upload\_file\_to\_ftp\_server
3. 29\_cgi\_server
4. 30\_verify\_mail
5. 31\_secure\_mail\_client (寫信)
6. 32\_pop3\_mail\_client (收信)

**LIST\_FILES\_ON\_FTP\_SERVER**

# list files on ftp server

```
#!/usr/bin/env python
# This program is optimized for Python 2.7.12 and Python 3.5.2.
# It may run on any other version with/without modifications.

FTP_SERVER_URL = 'ftp.ed.ac.uk'

import ftplib
from ftplib import FTP

def test_ftp_connection(path, username, email):
    #Open ftp connection
    ftp = ftplib.FTP(path, username, email)

    #List the files in the /pub directory
    ftp.cwd("/pub")
    print ("File list at %s:" %path)
    files = ftp.dir()
    print (files)

    ftp.quit()

if __name__ == '__main__':
    test_ftp_connection(path=FTP_SERVER_URL, username='anonymous',
                        email='nobody@nourl.com',
                        )
```

# list files on ftp server

```
File list at ftp.ed.ac.uk:
drwxr-xr-x  4 1005      bin          4096 Oct 11  1999 EMWAC
drwxr-xr-x 13 dns      netserve     4096 May  5  2010 EdLAN
drwxrwxr-x  4 6267      6268          4096 Oct 11  1999 ITFMP
drwxr-xr-x  4 1407      bin          4096 Oct 11  1999 JIPS
drwxr-xr-x  2 root      bin          4096 Oct 11  1999 Mac
drwxr-xr-x  2 10420     7525          4096 Oct  7  2003 PaedTBI
drwxr-xr-x  2 jaw       bin          4096 Oct 11  1999 Printing
drwxr-xr-x  3 root      bin          4096 Oct 11  1999 Student_Societies
drwxr-xr-x  6 root      bin          4096 Feb 19  2014 Unix
drwxr-xr-x  2 root      bin          4096 Mar  5  16:21 Unix1
drwxr-xr-x  2 1109      bin          4096 Oct 19  2016 Unix2
-rwxr--r--  1 root      root        180271184 Aug 15  2018 VMware-converter-en-6.2.0-8466193.exe
drwxr-xr-x  2 2022      bin          4096 Oct 11  1999 X.400
drwxr-xr-x  2 20076     bin          4096 Feb 17  2000 atoz
drwxr-xr-x  2 1403      bin          4096 Aug  9  2001 bill
drwxr-xr-x  2 4414      bin          4096 Oct 11  1999 cartonet
-rwxr--r--  1 root      root        818416 Aug 15  2018 convsa_62_guide.pdf
drwxr-xr-x  2 1115      bin          4096 Oct 11  1999 courses
drwxr-xr-x  2 10498     bin          4096 Oct 11  1999 esit04
drwxr-xr-x  2 6314      bin          4096 Oct 11  1999 flp
drwxr-xr-x  2 1400      bin          4096 Nov 19  1999 george
drwxr-xr-x  3 309643    root        4096 Sep 10  2008 geos
drwxr-xr-x  2 1663      root        4096 Apr 23  2013 hssweb
drwxr-xr-x  2 6251      bin          4096 Oct 11  1999 ierm
drwxr-xr-x  2 2126      bin          4096 Nov 12  2004 jbm
drwxr-xr-x  2 1115      bin          4096 Oct 11  1999 kusch
drwxr-xr-x  2 root      bin          4096 Oct 11  1999 lrtt
drwxr-xr-x  7 scott     bin          4096 Nov  9  2015 mail
drwxr-xr-x  3 1407      bin          4096 Oct 11  1999 maps
drwxr-xr-x  2 2009      bin          4096 Oct 11  1999 mmaccess
drwxr-xr-x  2 2009      bin          4096 Oct 11  1999 mmsurvey
drwx--x-x-x  3 1943      bin          4096 Dec  1  2000 mww
drwxr-xr-x  2 root      bin          4096 Oct 11  1999 pbowers
drwxr-xr-x  5 7324      bin          4096 Oct 11  1999 rip99
drwxr-xr-x 11 2223      bin          4096 Sep 30  2011 soroti
drwxr-xr-x  2 root      bin          4096 Oct  6  2000 steve
drwxr-xr-x  2 2000      bin          4096 Oct 11  1999 ucsq
drwxr-xr-x  7 20099     bin          4096 Jul 28  2003 unixhelp
drwxr-xr-x  2 root      bin          4096 Oct 11  1999 utopia
drwxr-xr-x  2 2022      bin          4096 Oct 11  1999 whiteosi
None
```

**UPLOAD\_FILE\_TO\_FTP\_SERVER**

# upload file to ftp server

1. `sudo apt-get install vsftpd`
2. `sudo vi /etc/vsftpd.conf`
3. remove the `;"` before  
`"write_enable=YES"`
4. `/etc/init.d/vsftpd start`  
or `/etc/init.d/vsftpd restart`

# upload file to ftp server

```
#!/usr/bin/env python
# This program is optimized for Python 2.7.12 and Python 3.5.2.
# It may run on any other version with/without modifications.

import os
import argparse
import ftplib
import getpass

LOCAL_FTP_SERVER = 'localhost'
LOCAL_FILE = 'readme.txt'

def ftp_upload(ftp_server, username, password, file_name):
    print ("Connecting to FTP server: %s" %ftp_server)
    ftp = ftplib.FTP(ftp_server)
    print ("Login to FTP server: user=%s" %username)
    ftp.login(username, password)
    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__':
    parser = argparse.ArgumentParser(description='FTP Server Upload Example')
    parser.add_argument('--ftp-server', action="store", dest="ftp_server", default=LOCAL_FTP_SERVER)
    parser.add_argument('--file-name', action="store", dest="file_name", default=LOCAL_FILE)
    parser.add_argument('--username', action="store", dest="username", default=getpass.getuser())
    given_args = parser.parse_args()
    ftp_server, file_name, username = given_args.ftp_server, given_args.file_name, given_args.username
    password = getpass.getpass(prompt="Enter you FTP password: ")
    ftp_upload(ftp_server, username, password, file_name)
```



# upload file to ftp server

> python3 28\_upload\_file\_to\_ftp\_server.py --username=user

```
--username=jwchang  
Enter you FTP password:  
Connecting to FTP server: localhost  
Login to FTP server: user=jwchang  
Uploaded file: readme.txt
```

# CGI SERVER

# cgi server

```
#!/usr/bin/env python
# This program requires Python 3.5.2 or any later version
# It may run on any other version with/without modifications.
#
# Follow the comments inline to make it run on Python 2.7.x.

import os
import cgi
import argparse

import http.server
# Comment out the above line and uncomment the below for Python 2.7.x.
#import BaseHTTPServer

# Uncomment the below line for Python 2.7.x.
#import CGIHTTPServer

import cgi
cgi.enable() ## enable CGI error reporting

def web_server(port):

    server = http.server.HTTPServer
    # Comment out the above line and uncomment the below for Python 2.7.x.
    #server = BaseHTTPServer.HTTPServer

    handler = http.server.CGIHTTPRequestHandler #RequestsHandler
    # Comment out the above line and uncomment the below for Python 2.7.x.
    #handler = CGIHTTPServer.CGIHTTPRequestHandler #RequestsHandler

    server_address = ("", port)
    handler.cgi_directories = ["/cgi-bin", ]
    httpd = server(server_address, handler)
    print ("Starting web server with CGI support on port: %s ..." %port)
    httpd.serve_forever()

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='CGI Server Example')
    parser.add_argument('--port', action="store", dest="port", type=int, required=True)
    given_args = parser.parse_args()
    web_server(given_args.port)
```

# cgi server

## front-end

```
<html>
  <body>
    <form action="/cgi-bin/29_get_feedback.py" method="post">
      Name: <input type="text" name="Name"> <br />
      Comment: <input type="text" name="Comment" />
      <input type="submit" value="Submit" />
    </form>
  </body>
</html>
```

# cgi server

back-end

```
#!/usr/bin/env python
# This program is optimized for Python 2.7.12 and Python 3.5.2.
# It may run on any other version with/without modifications.

#!/usr/bin/python

# Import modules for CGI handling
import cgi
import cgitb

# Create instance of FieldStorage
form = cgi.FieldStorage()

# Get data from fields
name = form.getvalue('Name')
comment = form.getvalue('Comment')

print ("Content-type:text/html\r\n\r\n")
print("<html>")
print("<head>")
print("<title>CGI Program Example </title>")
print("</head>")
print("<body>")
print("<h2> %s sends a comment: %s</h2>" % (name, comment))
print("</body>")
print("</html>")
```

# cgi server

Name:

Comment:

**user1 sends a comment: ttt**

```
Starting web server with CGI support on port: 8800 ...
127.0.0.1 - - [28/Apr/2019 09:01:35] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Apr/2019 09:01:45] "GET /29_send_feedback.html HTTP/1.1" 200 -
127.0.0.1 - - [28/Apr/2019 09:01:50] "POST /cgi-bin/29_get_feedback.py HTTP/1.1" 200 -
127.0.0.1 - - [28/Apr/2019 09:06:42] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Apr/2019 09:06:46] "GET /29_send_feedback.html HTTP/1.1" 200 -
127.0.0.1 - - [28/Apr/2019 09:07:19] "POST /cgi-bin/29_get_feedback.py HTTP/1.1" 200 -
```

**VERIFY EMAIL**

# verify email

```
#!/usr/bin/env python
# This program is optimized for Python 2.7.12 and Python 3.5.2.
# It may run on any other version with/without modifications.

import re
import smtplib
import dns.resolver
import argparse

def mail_checker(fromAddress, toAddress):

    regex = '^([a-z0-9][a-z0-9._%+-]{0,63})@([a-z0-9-]+\.[a-z0-9-]+)*(\.[a-z]{2,})$'

    addressToVerify = str(toAddress)

    match = re.match(regex, addressToVerify)
    if match == None:
        print('Bad Syntax in the address to verify. Re-enter the correct value')
        raise ValueError('Bad Syntax')

    splitAddress = addressToVerify.split('@')
    domain = str(splitAddress[1])

    records = dns.resolver.query(domain, 'MX')
    mxRecord = records[0].exchange
    mxRecord = str(mxRecord)

    server = smtplib.SMTP()
    server.set_debuglevel(1)

    try:
        server.connect(mxRecord)
    except Exception as e:
        print("Mail Check Failed Due to Error: %s" %str(e))
        return

    server.helo(server.local_hostname)
    server.mail(fromAddress)
    code, message = server.rcpt(str(addressToVerify))
    server.quit()

    if code == 250:
        print('Successfully verified the email: %s', fromAddress)
    else:
        print('Failed to verify the email: %s', fromAddress)
```



# verify email

```
if __name__ == '__main__':  
    parser = argparse.ArgumentParser(description='Mail Server Example')  
    parser.add_argument('--fromAddress', action="store", dest="fromAddress", type=str, required=True)  
    parser.add_argument('--toAddress', action="store", dest="toAddress", type=str, required=True)  
    given_args = parser.parse_args()  
    mail_checker(given_args.fromAddress, given_args.toAddress)
```

# verify email

```
connect: ('mailgw.nutc.edu.tw.', 25)
connect: to ('mailgw.nutc.edu.tw.', 25) None
reply: b'220 Cellopoint E-mail Firewall v4.1.7 Build 0918 ready\r\n'
reply: retcode (220); Msg: b'Cellopoint E-mail Firewall v4.1.7 Build 0918 ready'
connect: b'Cellopoint E-mail Firewall v4.1.7 Build 0918 ready'
send: 'helo [127.0.1.1]\r\n'
reply: b'250 Cellopoint E-mail Firewall v4.1.7 Build 0918\r\n'
reply: retcode (250); Msg: b'Cellopoint E-mail Firewall v4.1.7 Build 0918'
send: 'mail FROM:<0@qqq11w888.com>\r\n'
reply: b'250 Sender <0@qqq11w888.com> Ok\r\n'
reply: retcode (250); Msg: b'Sender <0@qqq11w888.com> Ok'
send: 'rcpt TO:<jwchang@nutc.edu.tw>\r\n'
reply: b'250 Recipient <jwchang@nutc.edu.tw> Ok\r\n'
reply: retcode (250); Msg: b'Recipient <jwchang@nutc.edu.tw> Ok'
send: 'quit\r\n'
reply: b'221 Cellopoint E-mail Firewall v4.1.7 Build 0918 closing connection\r\n'
reply: retcode (221); Msg: b'Cellopoint E-mail Firewall v4.1.7 Build 0918 closing connection'
Successfully verified the email: %s_0@qqq11w888.com
```

**SECURE\_MAIL\_CLIENT**

# secure mail client

```
#!/usr/bin/env python
# This program is optimized for Python 2.7.12 and Python 3.5.2.
# It may run on any other version with/without modifications.

import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import argparse
import getpass

def mail_client(host, port, fromAddress, password, toAddress, subject, body):
    msg = MIMEMultipart()

    msg['From'] = fromAddress
    msg['To'] = toAddress
    msg['Subject'] = subject
    message = body
    msg.attach(MIMEText(message))

    mailserver = smtplib.SMTP(host,port)

    # Identify to the SMTP Gmail Client
    mailserver.ehlo()

    # Secure with TLS Encryption
    mailserver.starttls()

    # Reidentifying as an encrypted connection
    mailserver.ehlo()
    mailserver.login(fromAddress, password)

    mailserver.sendmail(fromAddress,toAddress,msg.as_string())
    print ("Email sent from:", fromAddress)

    mailserver.quit()
```

# secure mail client

```
if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Mail Server Example')
    parser.add_argument('--host', action="store", dest="host", type=str, required=True)
    parser.add_argument('--port', action="store", dest="port", type=int, required=True)
    parser.add_argument('--fromAddress', action="store", dest="fromAddress", type=str, required=True)
    parser.add_argument('--toAddress', action="store", dest="toAddress", type=str, required=True)
    parser.add_argument('--subject', action="store", dest="subject", type=str, required=True)
    parser.add_argument('--body', action="store", dest="body", type=str, required=True)
    password = getpass.getpass("Enter your Password:")
    given_args = parser.parse_args()
    mail_client(given_args.host, given_args.port, given_args.fromAddress, password, given_args.toAddress, given_args.subject, given_args.body)
```

降低安全性，使該程式取得權限發信

<https://www.google.com/settings/security/lesssecureapps>

```
>python3 31_secure_mail_client.py
--host='smtp.gmail.com'
--port=587
--fromAddress=test.nutc.chang@gmail.com
--toAddress=XXXXXX@gmail.com
--subject="Hi, Hello.."
--body="Good to see you all. Keep in touch. Take Care"
```

# POP3 MAIL CLIENT

# pop3 mail client

```
#!/usr/bin/env python
# This program is optimized for Python 2.7.12 and Python 3.5.2.
# It may run on any other version with/without modifications.

import getpass
import poplib
import argparse

def mail_client(host, port, user, password):
    Mailbox = poplib.POP3_SSL(host, port)
    Mailbox.user(user)
    Mailbox.pass_(password)
    numMessages = len(Mailbox.list()[1])
    print (Mailbox.retr(1)[1])
    Mailbox.quit()

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Mail Server Example')
    parser.add_argument('--host', action="store", dest="host", type=str, required=True)
    parser.add_argument('--port', action="store", dest="port", type=int, required=True)
    parser.add_argument('--user', action="store", dest="user", type=str, required=True)
    password = getpass.getpass("Enter your Password:")
    given_args = parser.parse_args()
    mail_client(given_args.host, given_args.port, given_args.user, password)
```





# 延伸閱讀

- Socket Programming in Python (Guide)
  - <https://realpython.com/python-sockets/#socket-api-overview>
- Python 网络编程
  - <http://www.runoob.com/python/python-socket.html>

Resource is available by  
<https://jiaweichang.github.io/biography/>

**THANKS**