Universitatea „Ştefan cel Mare” - Suceava
Facultatea de Inginerie Electrică şi Ştiinţa Calculatoarelor

Proiect la Rețele de Calculatoare din cadrul cursurilor de Licență Calculatoare

Conducător științific: as. drd. ing. Cătălin Beguni

Proiect

Simularea functionării unui Server FTP

Student: Maciuc Simon-Gabriel

Grupa: 3121A

Cuprins

1. **Introducere:**

1.1 Scop:

Proiectul constă in crearea unui server FTP prin intermediul limbajului Python.

1.2 Obiective:

-instalarea mediului de programare si a limbajului Python

-instalarea librăriilor necesare pentru server

-crearea server-ului si programarea

-verificarea gestiunii fișierelor prin intermediul sistemului de operare Windows, si a unui mediu de gestiune a fișerelor (ex. FileZilla)

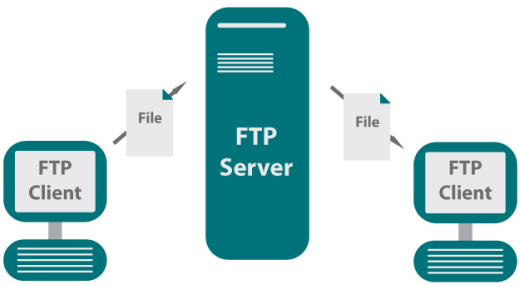
-prezentare proiect

1.3 De ce Python?

Python este un limbaj de programare dinamic, de nivel înalt, ce pune accent pe expresivitatea și înțelegerea ușoară a codului. Sintaxa sa permite implementări echivalente cu alte limbaje în mai puține linii de cod. Datorită acestui fapt, Python este foarte răspândit atât în programarea de aplicații. În cazul acestui proiect, Python dispune de librării care pot crea un server FTP, într-un mod eficient și prin cod relativ puțin.

1.4 Ce este un server FTP?

FTP este un acronim pentru "**F**ile **T**ransfer **P**rotocol" și este cea mai simplă metodă de a transfera fișiere de pe un computer pe altul prin intermediul Internetului.



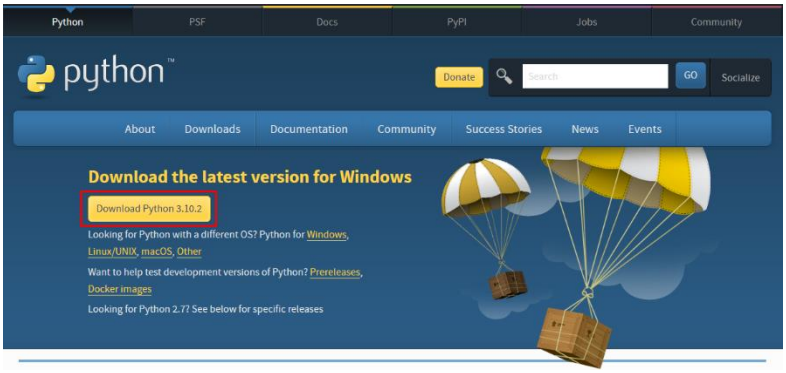
Așa cum sugerează si numele, **FTP** reprezintă protocolul de rețea standard folosit pentru a copia fișiere de pe un host pe altul (de pe un server sau calculator pe altul), folosind o rețea clasică TCP/IP (Transfer Control Protocol/Internet Protocol), cum este Internetul.

2. Instalarea componentelor necesare:

2.1 Instalare Python.

Se descarcă limbajul/mediul de programare Python de pe următorul link și se instalează pe calculator urmând pașii specificați de fișierul executabil.

https://www.python.org/downloads/

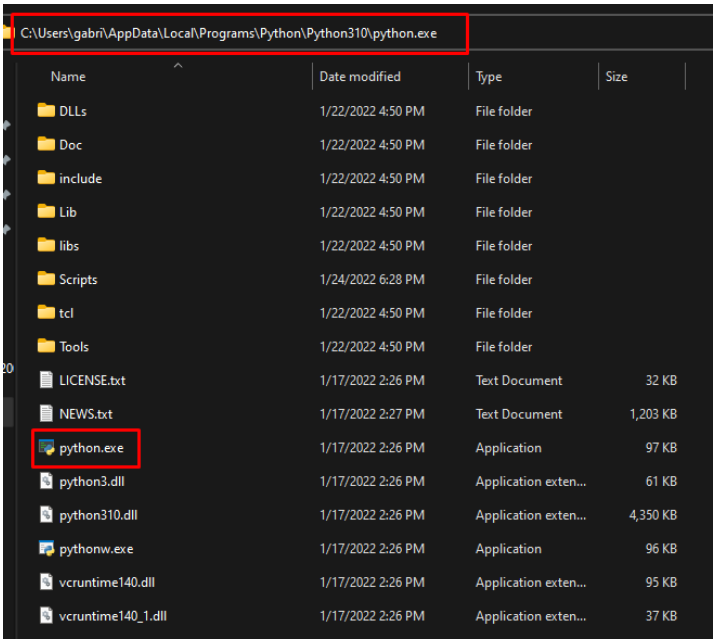


2.2 Instalare librărie pentru server.

Pentru a instala librăria pentru server FTP trebuie să scriem urmatoarea comandă in Command Prompt "*pip install pyftpdlib*".

Însă pentru a funcționa comanda trebuie specificată calea către fișierul executabil "python.exe". Cel mai ușor mod de a găsi calea este următorul:

- se caută in meniul de start "Python 3.10 (64-bit)"
- se deschide locația fișierului
- se deschide locația fișierului "Python 3.10 (64-bit)"
- se copiază calea fișierului "python.exe"



Se scrie in Command Prompt calea fișierului urmat de "–*m*" și de comanda de instalare.

```
-m mod : run library module as a script (terminates option list)
```

```
C:\Users\gabri>C:\Users\gabri\AppData\Local\Programs\Python\Python310\python.exe -m pip install pyftpdlib
Collecting pyftpdlib
  Using cached pyftpdlib-1.5.6.tar.gz (188 kB)
  Preparing metadata (setup.py) ... done
Using legacy 'setup.py install' for pyftpdlib, since package 'wheel' is not installed.
Installing collected packages: pyftpdlib
    Running setup.py install for pyftpdlib ... done
Successfully installed pyftpdlib-1.5.6
```

2.3 Instalare FileZilla. De ce FileZilla?

FileZilla este unul dintre cei mai populari clienți FTP. Scopul principal al Filezilla este să vă faciliteze încărcarea și descărcarea fișierelor de pe serverul dvs. de găzduire web. De asemenea, puteți edita fișierele și salva modificările fără a fi nevoie de descărcare și încărcare manuală. Se descarcă programul de pe link-ul următor:

https://filezillaproject.org/download.php?type=client

3. Scriere cod

3.1 Importarea claselor din librărie

```
# Clasa de autorizare „Dummy”, este potrivită pentru subclasare pentru
# a crea proprii "authorizers". Un „ authorizer” este o clasă care gestionează
# autentificări și permisiuni ale serverului FTP. Este folosit în interiorul
# clasei "pyftpdlib.handlers.FTPHandler" pentru verificarea parolei utilizatorului,
# obținerea directorului principal al utilizatorilor, verificarea permisiunilor
# utilizatorului atunci când are loc un eveniment de citire/scriere a sistemului
# de fișiere și schimbarea utilizatorului înainte de a accesa sistemul de fișiere.
from pyftpdlib.authorizers import DummyAuthorizer

# Această clasă implementează serverul FTP gestionând comenzile primite de la client
# pe canalul de control prin apelarea metodei corespunzătoare comenzii.
from pyftpdlib.handlers import FTPHandler

#Creează un socket care ascultă address-ul, trimițând cererile către handler.
from pyftpdlib.servers import FTPServer
```

3.2 Crearea variabilelor

```
# Portul pe care server-ul FTP il va folosi.
# Numarul trebuie sa fie mai mare decat 1023 deoarece
# porturile de la 0 la 1023 sunt rezervate, numite si "well-known ports".
PORT_FTP = 1234

# Numele utilizatorului care se poate loga pe server-ul FTP.
UTILIZATOR_FTP = "Gabri"

# Parola utilizatorului.
PAROLA_FTP = "parola"

# Directorul care va contine informatiile.
DIRECTOR_FTP = "C:/Users/gabri/OneDrive/Documents/RC_proiect/ftp_content"
```

3.3 Scrierea instrucțiunilor necesare

```
def main():
    # Creeaza o instantiere a unui "authorizer" inactiv pentru a gestiona
    # utilizatorii virtuali.
    authorizer = DummyAuthorizer()

    # Definim un nou utilizator care va avea anumite permisiuni date de variabila "perm".
    authorizer.add_user(UTILIZATOR_FTP, PAROLA_FTP, DIRECTOR_FTP, perm='elrafmw')

    # Creeaza o instantiere a unui "handler".
    handler = FTPHandler
    handler.authorizer = authorizer

    # Se instatiaza clasa server-ului FTP si va folosi adresa '127.0.0.1:1234
    address = ('127.0.0.1', PORT_FTP)
    server = FTPServer(address, handler)

    # Se creeaza o limitare de conexiuni.
    server.max_cons = 256
    server.max_cons_per_ip = 5

    # Face server-ul sa functioneze pana se va inchide aplicatia python.
    server.serve_forever()


if __name__ == '__main__':
    main()
```

3.4 Permisiunile clientului față de server

Variabila “perm” oferă permisiunile clientului care accesează server-ul, acestea fiind următoarele:

Permisiuni de citire:

- "e" = schimbă directorul (comanda CWD, CDUP)
- "l" = listă fișiere (comanda LIST, NLST, MLSD, MLST, SIZE)
- "r" = preluează fișierul de pe server (comanda RETR)

Permisiuni de scriere:

- "a" = adaugă date la un fișier existent (comanda APPE)
- "d" = ștergere fișier sau director (comanda DELE, RMD)
- "f" = redenumire fișier sau director (comanda RNFR, RNTO)
- "m" = creare director (comanda MKD)
- "w" = stocarea unui fișier pe server (comanda STOR, STOU)
- "M" = schimbare mod/permisiunea unui fișier (comanda SITE CHMOD)
- "T" = modificare “timp de modificare” a unui fișier (comanda SITE MFMT)

4. Simulare server

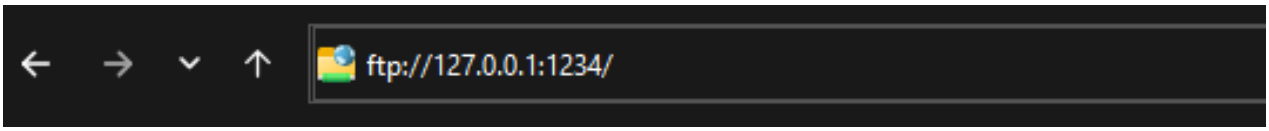
4.1 Pornire server

Se rulează programul scris mai sus, urmând să fie afișat următorul mesaj:

```
====== RESTART: C:\Users\gabri\OneDrive\Documents\RC_proiect\ftp_server.py =====
[I 2022-01-24 22:59:23] concurrency model: async
[I 2022-01-24 22:59:23] masquerade (NAT) address: None
[I 2022-01-24 22:59:23] passive ports: None
[I 2022-01-24 22:59:23] >>> starting FTP server on 127.0.0.1:1234, pid=8400 <<<
```

4.2 Accesare director

Se scrie in bara de căutare din “File Explorer” adresa următoare:

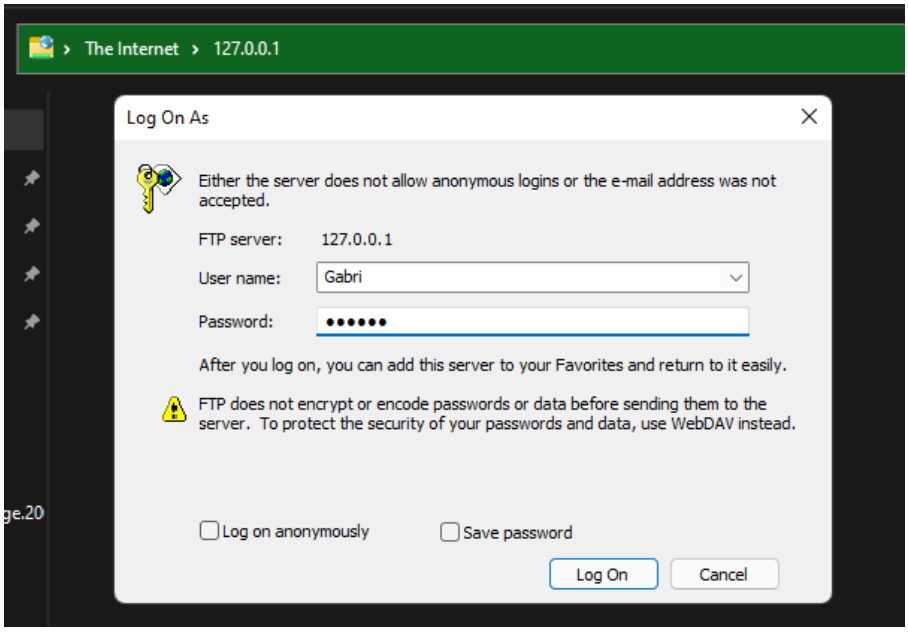


In mod normal am fi obișnuiți să scriem HTTP in loc de FTP, însă in cazul de față ne vom conecta la un server FTP, nu la un server World Wide Web (www).

4.3 Încercare logare

În momentul accesării server-ului, acesta va cere un nume de utilizator și o parolă.

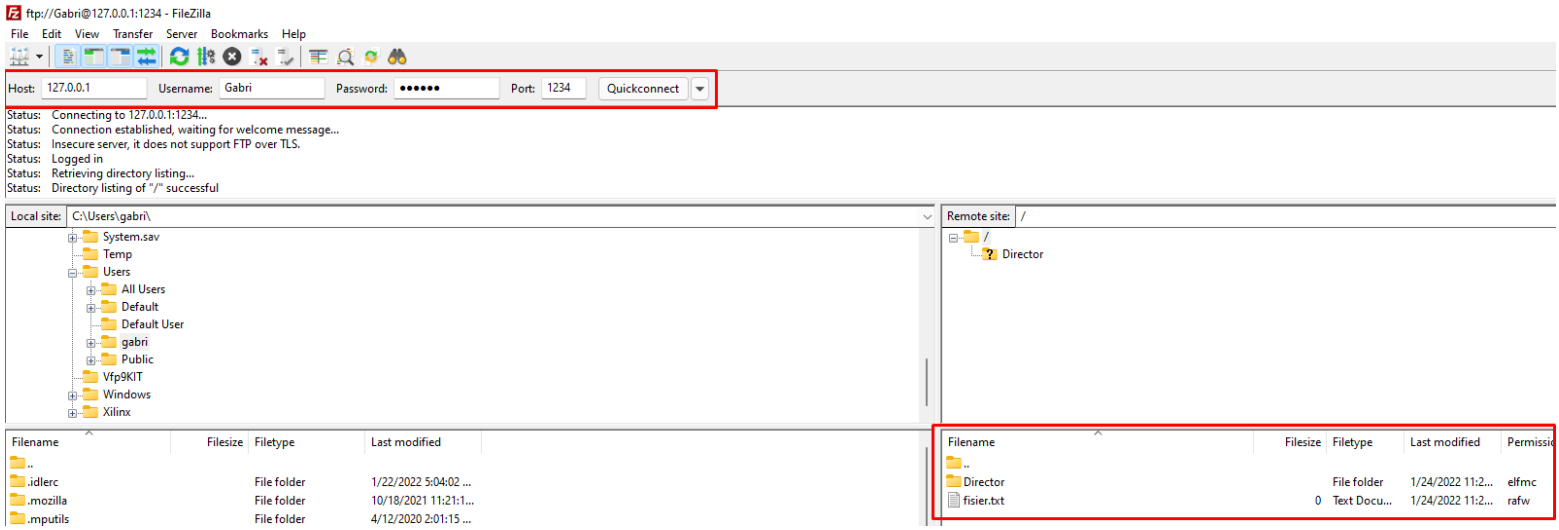
După logare, clientul va putea crea/modifica/salva fișiere din server-ul respectiv (în funcție de permisiunile oferite), aceste acțiuni fiind afișate în program.



4.4 Încercare conectare utilizând FileZilla

Se introduce adresa de conectare, datele utilizatorului, și port-ul de conectare.

Dacă datele sunt corecte, ar trebui sa apară următoarea fereastră.



4.5 Afișarea acțiunilor utilizatorului

```
====== RESTART: C:\Users\gabri\OneDrive\Documents\RC_proiect\ftp_server.py =====
[I 2022-01-25 15:21:52] concurrency model: async
[I 2022-01-25 15:21:52] masquerade (NAT) address: None
[I 2022-01-25 15:21:52] passive ports: None
[I 2022-01-25 15:21:52] >>> starting FTP server on 127.0.0.1:1234, pid=37140 <<<
```

-mesaj pornire server-

```
127.0.0.1:54782-[] FTP session opened (connect)
127.0.0.1:54782-[Gabri] USER 'Gabri' logged in.
```

-mesaj conectare utilizator-

```
RNFR C:\Users\gabri\OneDrive\Documents\RC_proiect\ftp_content\New Folder 350
RNTO C:\Users\gabri\OneDrive\Documents\RC_proiect\ftp_content\folder nou 250
```

-mesaj creare director nou și redenumire-

```
RMD C:\Users\gabri\OneDrive\Documents\RC_proiect\ftp_content\Director 550 'Not enough privileges.'
```

-încercare șterge director (fără permisiune)-

```
RETR C:\Users\gabri\OneDrive\Documents\RC_proiect\ftp_content\fisier.txt completed=1 bytes=0 seconds=0.01
```

-copiere fișier din director-ul server-

4.6 Verificare număr maxim de conectări

În programul scris, am specificat numărul maxim de conectări de pe un dispozitiv să fie 5. În imaginea de mai jos se observa faptul că nu se poate efectua a 6-a conectare.

5. Produs final

- Un mediu de programare, cu un limbaj de programare relativ simplu.
- Un program de gestiune a fișierelor, într-un mod comod pentru utilizator.
- Un cod Python, usor de înțeles pentru student.
- Un server FTP care conține informații.
- Conectarea utilizatorilor care au adresa, port-ul, și date de utilizator în program.
- Gestiunea fișierelor din server, cu anumite permisiuni date de program.
- Afișarea acțiunilor utilizatorilor conectați la server.

6. Bibliografie

1. https://pyftpdlib.readthedocs.io/en/latest/api.html
2. https://pypi.org/project/pyftpdlib/
3. http://purepython.eaudeweb.ro/wiki/Cursuri/Introducere.html
4. https://megahost.ro/blog/terminologie-in-web-hosting-ftp.html
5. https://realpython.com/what-is-pip/
6. https://www.ipswitch.com/blog/what-is-file-transfer-protocol-ftp
7. https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers
8. https://www.python.org/downloads/
9. https://filezilla-project.org/download.php?type=client

7. Anexă

```
from pyftpdlib.authorizers import DummyAuthorizer

from pyftpdlib.handlers import FTPHandler

from pyftpdlib.servers import FTPServer

PORT_FTP = 1234

UTILIZATOR_FTP = "Gabri"

PAROLA_FTP = "parola"

DIRECTOR_FTP = "C:/Users/gabri/OneDrive/Documents/RC_proiect/ftp_content"


def main():
    authorizer = DummyAuthorizer()

    authorizer.add_user(UTILIZATOR_FTP, PAROLA_FTP, DIRECTOR_FTP,
perm='elrafmw')

    handler = FTPHandler
    handler.authorizer = authorizer

    address = ('127.0.0.1', PORT_FTP)
    server = FTPServer(address, handler)

    server.max_cons = 256
    server.max_cons_per_ip = 5

    server.serve_forever()


if __name__ == '__main__':
    main()
```