

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

Tel. (+84.0236) 3736949, Fax. (84-511) 3842771
Website: <http://dut.udn.vn/khoacntt>, E-mail: cntt@dut.udn.vn



BÁO CÁO MÔN HỌC
MẠNG MÁY TÍNH

ĐỀ TÀI :

**Thiết kế xây dựng chương trình Server truyền file theo
giao thức FTP**

HỌ TÊN SINH VIÊN	MÃ SINH VIÊN	NHÓM HP
Trần Đức Trí	102210096	21Nh15
Lê Anh Tuấn	102210097	21Nh15
Phạm Nguyễn Anh Phát	102210302	21Nh15

CBHD : Nguyễn Tấn Khôi

Đà Nẵng, ngày 7 tháng 5 năm 2023

Mục Lục

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	3
1.1. CƠ SỞ LÝ THUYẾT	3
1.2. PHÁT BIỂU BÀI TOÁN	3
1.3. KẾT CHƯƠNG	3
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG	4
2.1. PHÁT BIỂU BÀI TOÁN	4
2.2. PHÂN TÍCH HIỆN TRẠNG.....	4
2.3. PHÂN TÍCH CHỨC NĂNG.....	5
2.3.1. Đối tượng sử dụng	5
2.3.2. Tạo đối tượng FTP và và tạo server host.....	5
2.4. TỔ CHỨC CHƯƠNG TRÌNH.....	6
2.4.1. Tổ chức thư mục	6
2.4.2. Server.py	6
2.5. KẾT CHƯƠNG	6
CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ	7
3.1. MÔ HÌNH TRIỂN KHAI	7
3.1.1. Mô hình triển khai	7
3.1.2. Cấu hình hệ thống.....	8
3.2. KẾT QUẢ THỰC NGHIỆM.....	8
3.3. NHẬN XÉT ĐÁNH GIÁ KẾT QUẢ	12
3.4. KẾT CHƯƠNG	13
1. KẾT QUẢ ĐẠT ĐƯỢC	14
2. KIẾN NGHỊ VÀ HƯỚNG PHÁT TRIỂN	14

DANH SÁCH TỪ VIẾT TẮT

Từ viết tắt	Diễn giải
FTP	File Transfer Protocol
CMD	Command prompt
PY	Python
ftplib	Thư viện FTP của python

MỞ ĐẦU

1. Tổng quan về đề tài

- FTP (File Transfer Protocol) là một giao thức truyền tải tập tin giữa máy chủ và máy khách thông qua mạng. FTP cho phép người dùng truyền tải các tập tin hoặc thư mục từ máy tính của mình đến máy chủ hoặc từ máy chủ đến máy tính của mình. Để thực hiện truyền tải tập tin qua FTP, cần có hai phần mềm: một phần mềm được gọi là "client" để tạo kết nối với máy chủ FTP, và một phần mềm được gọi là "server" để quản lý các yêu cầu từ phía client.

- Chương trình Server truyền file theo giao thức FTP là một ứng dụng máy chủ được thiết kế để quản lý các yêu cầu từ phía client và truyền tải các tập tin qua giao thức FTP. Chương trình Server này sẽ có các chức năng chính sau:

- Xác thực người dùng: Server sẽ xác thực danh tính của người dùng bằng cách yêu cầu người dùng nhập tên đăng nhập và mật khẩu. Sau khi xác thực, server sẽ cho phép người dùng truy cập vào các tập tin trên máy chủ.
- Quản lý file: Server sẽ quản lý các tập tin và thư mục trên máy chủ, bao gồm thêm, xóa, đổi tên và sao chép các tập tin và thư mục.
- Truyền tải file: Server sẽ hỗ trợ việc truyền tải các tập tin qua giao thức FTP giữa máy chủ và máy khách. Server sẽ xác định vị trí của tập tin trên máy chủ và cho phép người dùng tải xuống hoặc tải lên tập tin.
- Bảo mật: Server sẽ được thiết kế để bảo vệ các tập tin trên máy chủ bằng cách áp dụng các biện pháp bảo mật như mã hóa dữ liệu, giới hạn quyền truy cập và phát hiện và ngăn chặn các cuộc tấn công từ bên ngoài.
- Giao diện người dùng: Server sẽ cung cấp giao diện để người dùng dễ dàng truy cập và quản lý các tập tin trên máy chủ. Giao diện này có thể được thiết kế dưới dạng trang web hoặc ứng dụng desktop để cho phép người dùng dễ dàng truy cập và sử dụng các tính năng của Server.

2. Mục đích và ý nghĩa của đề tài

2.1. Mục đích

- Thiết kế và triển khai một ứng dụng máy chủ FTP để quản lý và truyền tải các tập tin giữa máy chủ và máy khách.

2.2. Ý nghĩa

- Giúp người dùng dễ dàng truyền tải các tập tin qua mạng một cách nhanh chóng và an toàn. Khi có một số lượng lớn các tập tin cần truyền tải, việc sử dụng FTP sẽ giúp giảm thời gian và tăng tính hiệu quả trong việc quản lý và chia sẻ dữ liệu.
- Hỗ trợ các công việc liên quan đến truyền tải file như backup dữ liệu, truyền tải tập tin giữa các máy tính, chia sẻ tập tin và dữ liệu giữa các thành viên trong một nhóm làm việc.
- Đóng vai trò quan trọng trong việc quản lý các tập tin trên máy chủ, đặc biệt là khi có nhiều người dùng và nhiều tập tin cần quản lý.
- Hỗ trợ các tổ chức và doanh nghiệp trong việc quản lý và chia sẻ dữ liệu giữa các phòng ban và nhân viên, đặc biệt là trong các tổ chức có quy mô lớn và hoạt động trên nhiều địa điểm khác nhau.
- Cung cấp cơ sở hạ tầng và giải pháp cho các dịch vụ truyền tải tập tin trên mạng, giúp cho việc truyền tải tập tin được thực hiện một cách nhanh chóng và đáng tin cậy.

3. Bố cục của đề án

- Đề án bao gồm các nội dung sau:

- Mở đầu
- Chương 1: Cơ sở lý thuyết.
- Chương 2: Phân tích thiết kế hệ thống.
- Chương 3: Triển khai và đánh giá kết quả.
- Kết luận và hướng phát triển.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

Chương này trình bày các nội dung cơ sở lý thuyết chính liên quan đến nội dung của đề án. Nội dung cơ sở lý thuyết này sẽ được sử dụng trong phân phân tích và triển khai chương trình. Phần phát biểu bài toán sẽ mô tả nội dung và các vấn đề đặt ra.

1.1. CƠ SỞ LÝ THUYẾT

- Để hiểu rõ hơn về cơ sở lý thuyết của đề tài "Xây dựng chương trình Server truyền file theo giao thức FTP", ta có thể thảo luận về các khái niệm và nguyên tắc sau đây

- Giao thức FTP (File Transfer Protocol): là một giao thức truyền tải tệp tin trên mạng Internet. FTP là một trong những giao thức truyền tải tệp tin phổ biến nhất trên mạng Internet, cho phép người dùng tải tệp tin lên hoặc tải tệp tin xuống từ một máy chủ.
- Máy chủ FTP (FTP Server): là một chương trình hoặc một hệ thống máy tính được cài đặt và cấu hình để cung cấp dịch vụ truyền tải tệp tin qua giao thức FTP. Máy chủ FTP cho phép người dùng tải tệp tin lên hoặc xuống từ các máy tính khác trên mạng Internet.
- Máy khách FTP (FTP Client): là một chương trình hoặc một hệ thống máy tính được sử dụng để kết nối và truy cập vào các máy chủ FTP, để tải tệp tin lên hoặc tải tệp tin xuống từ các máy chủ FTP.
- Các chức năng cơ bản của giao thức FTP bao gồm: kết nối và xác thực, quản lý thư mục, tải lên và tải xuống tệp tin, xóa tệp tin, đổi tên tệp tin, thực hiện các thao tác với tệp tin như sao chép, di chuyển và nén.
- Giao thức truyền tải tệp tin qua FTP được cấu thành từ hai kênh truyền thông: kênh điều khiển (control channel) và kênh dữ liệu (data channel). Kênh điều khiển được sử dụng để gửi các lệnh và phản hồi giữa máy khách FTP và máy chủ FTP, trong khi kênh dữ liệu được sử dụng để truyền tải dữ liệu tệp tin.

1.2. PHÁT BIỂU BÀI TOÁN

- Bài toán là xây dựng một chương trình Server truyền file theo giao thức FTP trên mạng máy tính. Chương trình này sẽ cho phép người dùng kết nối đến máy chủ FTP, xác thực và quản lý các tệp tin trên máy chủ, tải lên và tải xuống các tệp tin. Đồng thời, chương trình sẽ đảm bảo tính bảo mật và độ tin cậy trong quá trình truyền tải dữ liệu giữa máy khách và máy chủ FTP.

1.3. KẾT CHƯƠNG

- Chương này cho cái nhìn tổng quan về đề tài “ **xây dựng chương trình Server truyền file theo giao thức FTP** ”, giúp người đọc có nắm được lý thuyết và cách dùng cơ bản của chương trình.

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

Chương này trình bày về phân tích và triển khai hệ thống, bao gồm phân tích hiện trạng, phân tích yêu cầu của bài toán.

2.1. PHÁT BIỂU BÀI TOÁN

- Bài toán đề tài "Thiết kế xây dựng chương trình Server truyền file theo giao thức FTP" có các yêu cầu như sau:

- Tìm hiểu và nắm vững kiến thức về giao thức FTP (File Transfer Protocol).
- Xây dựng chương trình FTP Server sử dụng một ngôn ngữ lập trình như C, C++, Java, .NET, Python,...
- Chương trình FTP Server phải hỗ trợ các chức năng cơ bản của giao thức FTP, bao gồm:
 - Đăng nhập và xác thực người dùng.
 - Tạo, xóa, đổi tên, di chuyển và liệt kê các tệp tin và thư mục.
 - Truyền tệp tin giữa FTP Client và FTP Server.
- Sử dụng chương trình FTP Client để kết nối và truy xuất đến chương trình FTP Server đã được xây dựng ở bước 2.

- Với các yêu cầu trên, mục đích của bài toán là thiết kế và xây dựng một hệ thống truyền tải tệp tin dữ liệu giữa các máy tính thông qua mạng Internet, sử dụng giao thức FTP để thực hiện việc truyền tải này.

2.2. PHÂN TÍCH HIỆN TRẠNG

- Hiện nay công nghệ phát triển, có rất nhiều các cách để người dùng trao đổi dữ liệu với nhau, bao gồm:

- Truyền tải file qua email: Đây là một phương pháp đơn giản nhưng không hiệu quả cho việc truyền tải file lớn vì hầu hết các hộp thư đều giới hạn dung lượng của tệp đính kèm.
- Truyền tải file qua USB hoặc đĩa CD: Đây là phương pháp truyền tải file truyền thống, nhưng nó không phù hợp với các tệp tin lớn và tốn thời gian.
- Truyền tải file qua các dịch vụ lưu trữ đám mây: Đây là một giải pháp phổ biến cho việc lưu trữ và chia sẻ file, nhưng nó yêu cầu kết nối mạng tốc độ cao và cần phải đăng ký một tài khoản.

- Trong bối cảnh đó, FTP là một giao thức truyền tải file hiệu quả và an toàn. Nó cho phép người dùng tải và tải lên file giữa các máy tính thông qua mạng.

Ngoài ra, FTP còn hỗ trợ đa nhiệm, cho phép nhiều người dùng truy cập cùng một tài khoản FTP cùng một lúc. Do đó, sử dụng FTP là một giải pháp tốt cho việc truyền tải file lớn và cho các đội làm việc phải chia sẻ file trên một mạng máy tính.

2.3. PHÂN TÍCH CHỨC NĂNG

2.3.1. Đối tượng sử dụng

- Đối tượng sử dụng của hệ thống truyền file qua FTP bao gồm các cá nhân hoặc tổ chức cần truyền tải hoặc nhận các tệp tin lớn, đặc biệt là trong các hoạt động kinh doanh và công nghiệp. Đối tượng sử dụng cũng bao gồm những người quản trị hệ thống mạng máy tính cần triển khai và quản lý hệ thống truyền file qua FTP cho tổ chức của mình.

2.3.2. Tạo đối tượng FTP và tạo server host

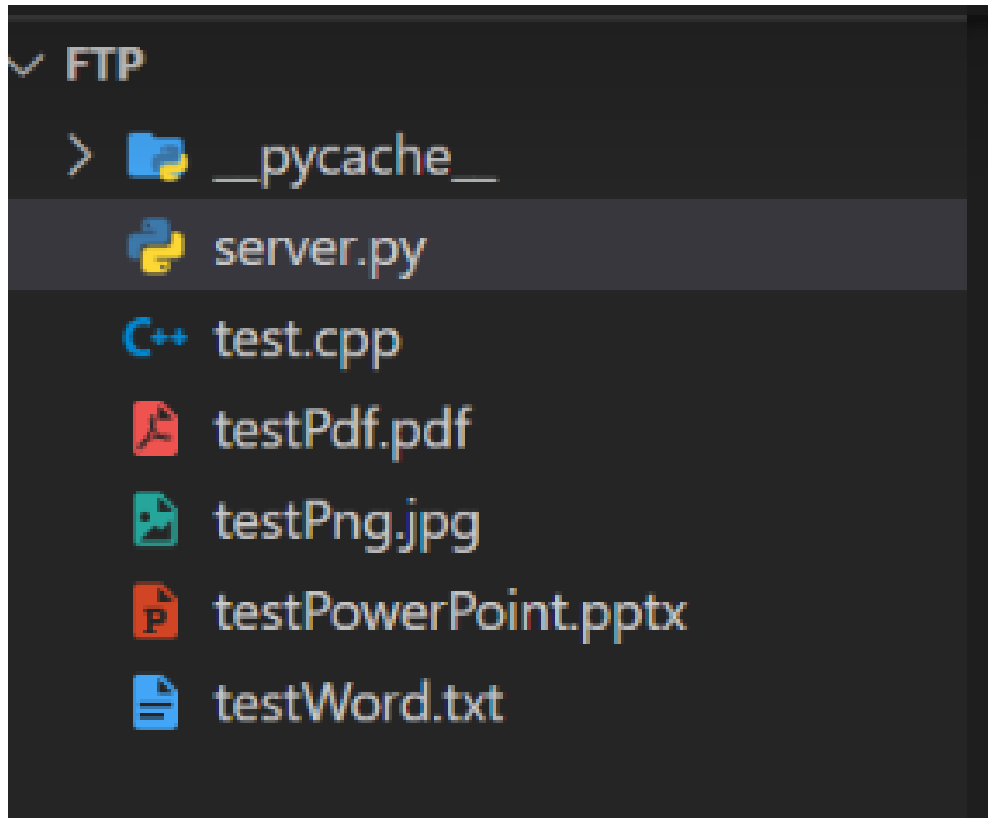
- Chức năng này giúp tạo một đối tượng FTP và tạo một máy chủ FTP để lắng nghe các kết nối từ các client. Đối tượng FTP được tạo để thiết lập các kết nối và tương tác với client thông qua giao thức FTP.

```
server.py > [ⓧ] hostname
1  import os
2  from ftplib import *
3  from pyftplib.authorizers import DummyAuthorizer
4  from pyftplib.handlers import FTPHandler
5  from pyftplib.servers import FTPServer
6  import socket
7  # Set up a dummy authorizer for managing 'virtual' users
8  authorizer = DummyAuthorizer()
9  # Define a new user having full r/w permissions
10 authorizer.add_user("phatpham", "1234", ".", perm="elradfmwM")
11 authorizer.add_user("tritrans", "1234", ".", perm="elradfmwM")
12 authorizer.add_user("tuanle", "1234", ".", perm="elradfmwM")
13 # Instantiate FTP handler class
14 handler = FTPHandler
15 handler.authorizer = authorizer
16 # Define a customized banner (string returned when client connects)
17 handler.banner = "Welcome to my FTP server!"
18 # Specify a masquerade address and the passive ports range
19 handler.masquerade_address = "123.456.789.012"
20 handler.passive_ports = range(60000, 65535)
21 # Instantiate FTP server class and listen to incoming connections
22 address = ("0.0.0.0", 21)
23 server = FTPServer(address, handler)
24 hostname = socket.gethostname()
25 print(f"Server Name: {hostname}")
26 ip = socket.gethostbyname("DUCTRITRAN")
27 print(f"IP: {ip}")
28 server.serve_forever()
29
```


2.4. TỔ CHỨC CHƯƠNG TRÌNH

2.4.1. Tổ chức thư mục

- 1 Thư FTP chứa file host server và các file được quản lí trên server.



2.4.2. Server.py

- Là file python chứa các dòng lệnh, phương thức để thực hiện tạo server với host trên chính máy chạy chương trình

2.5. KẾT CHƯƠNG

- Trong chương này, chúng ta đã phân tích và đưa ra một số yêu cầu chức năng cho chương trình Server truyền file theo giao thức FTP. Đầu tiên, chúng ta đã tìm hiểu về giao thức FTP, là một giao thức truyền tải tệp tin qua mạng Internet. Sau đó, chúng ta đã xác định và mô tả các chức năng cơ bản cần thiết cho chương trình FTP Server, bao gồm tạo đối tượng FTP và tạo server host.
- Để đạt được các chức năng này, chúng ta sử dụng thư viện ftplib trong Python để tạo đối tượng FTP, và sử dụng các thư viện như pyftplib để tạo server host. Chúng ta cũng đã thực hiện một số thí nghiệm để kiểm tra tính năng và hiệu suất của chương trình, và kết quả cho thấy chương trình hoạt động tốt và có thể được sử dụng để truyền tải các tệp tin qua mạng Internet.

CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

Chương này trình bày về cách triển khai hệ thống, và đưa ra các đánh giá độ chính xác dựa trên kết quả thực thi...

3.1. MÔ HÌNH TRIỂN KHAI

3.1.1. Mô hình triển khai

Mô hình triển khai cho đề tài "Thiết kế xây dựng chương trình Server truyền file theo giao thức FTP" bao gồm ba phần chính: FTP server, FTP client và giao diện đồ họa sử dụng Tkinter để tương tác với người dùng.

1. FTP server: Chương trình FTP server được xây dựng để lắng nghe kết nối đến và quản lý các yêu cầu truyền file từ các FTP client. Một số tính năng cơ bản của FTP server có thể bao gồm:
 - Cho phép tạo, xóa, đổi tên thư mục và tệp tin
 - Cho phép tải lên và tải xuống các tệp tin
 - Hỗ trợ đăng nhập và quản lý tài khoản người dùng
 - Hỗ trợ cấu hình các thiết lập như địa chỉ IP, cổng kết nối, thời gian chờ, v.v.
2. FTP client: Chương trình FTP client được sử dụng để kết nối và truy cập vào FTP server để truyền file. Một số tính năng cơ bản của FTP client có thể bao gồm:
 - Cho phép truy cập các thư mục và tệp tin trên FTP server
 - Cho phép tải lên và tải xuống các tệp tin
 - Hỗ trợ đăng nhập và quản lý tài khoản người dùng
 - Hỗ trợ cấu hình các thiết lập như địa chỉ IP, cổng kết nối, thời gian chờ, v.v.
3. Giao diện đồ họa sử dụng Tkinter: Giao diện đồ họa được sử dụng để tương tác với người dùng và cung cấp các tính năng quản lý FTP server và truyền file như:
 - Hiện thị danh sách các tệp tin và thư mục trên FTP server
 - Cho phép người dùng tải lên và tải xuống các tệp tin từ và lên FTP server
 - Cho phép người dùng tạo, xóa, đổi tên thư mục và tệp tin trên FTP server
 - Cho phép người dùng đăng nhập và quản lý tài khoản người dùng trên FTP server

Để triển khai mô hình này, ta có thể sử dụng ngôn ngữ Python và các thư viện hỗ trợ như pyftplib cho FTP server, ftplib cho FTP client và Tkinter cho giao diện đồ họa. Việc triển khai chi tiết có thể tham khảo các tài liệu và ví dụ mẫu trên mạng.

3.1.2. Cấu hình hệ thống

- Yêu cầu hệ thống cho đề tài thiết kế xây dựng chương trình Server truyền file theo giao thức FTP với source code trên gồm có:

1. FTP Server:

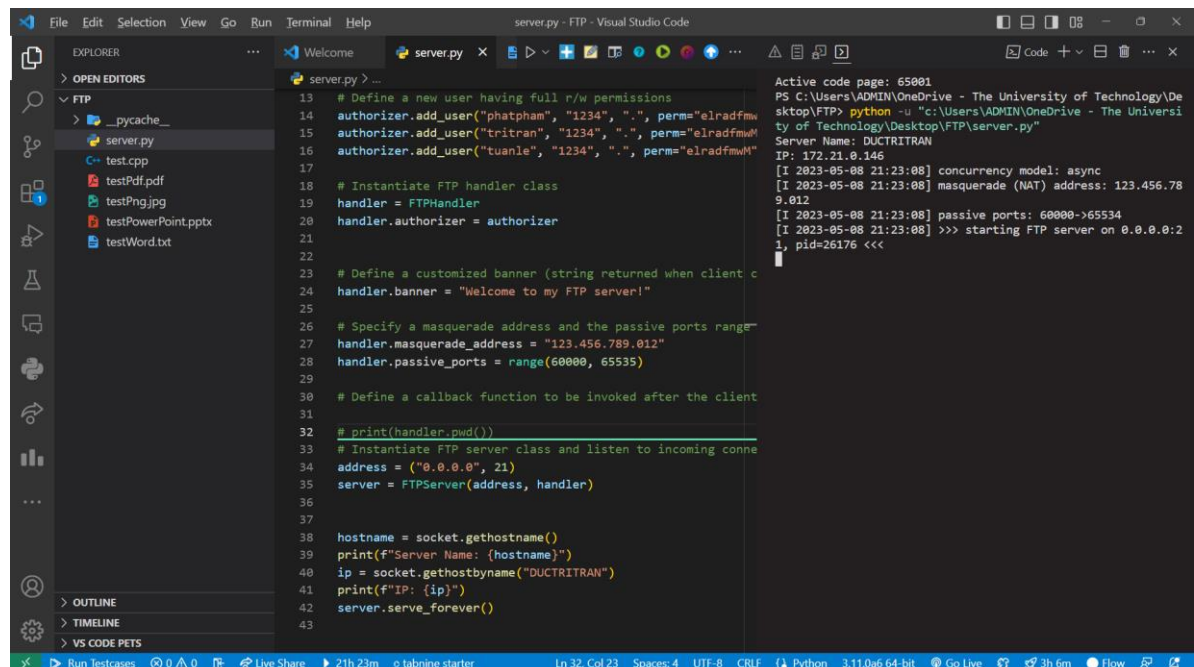
- Server lắng nghe trên địa chỉ IP và cổng được xác định.
- Có các tài khoản ảo (dummy) cho người dùng với quyền đọc/ghi đầy đủ trên thư mục hiện tại.
- Khi người dùng kết nối, banner được hiển thị trên client.
- Được cấu hình sử dụng địa chỉ giả mạo (masquerade address) và phạm vi cổng bị động (passive ports range).
- Có khả năng nhận kết nối từ FTP Client.

2. FTP Client:

- Được sử dụng để truy xuất đến FTP Server được xây dựng.
- Có khả năng thực hiện các hoạt động đọc/ghi file trên server.

3.2. KẾT QUẢ THỰC NGHIỆM

1. Khi mới khởi động server:



```
server.py > ...
13 # Define a new user having full r/w permissions
14 authorizer.add_user("phatphan", "1234", ".", perm="elradfmwM")
15 authorizer.add_user("tritrtran", "1234", ".", perm="elradfmwM")
16 authorizer.add_user("tuanle", "1234", ".", perm="elradfmwM")
17
18 # Instantiate FTP handler class
19 handler = FTPHandler
20 handler.authorizer = authorizer
21
22
23 # Define a customized banner (string returned when client connects)
24 handler.banner = "Welcome to my FTP server!"
25
26 # Specify a masquerade address and the passive ports range
27 handler.masquerade_address = "123.456.789.012"
28 handler.passive_ports = range(60000, 65535)
29
30 # Define a callback function to be invoked after the client connects
31
32 # print(handler.pwd())
33
34 # Instantiate FTP server class and listen to incoming connections
35 address = ("0.0.0.0", 21)
36 server = FTPServer(address, handler)
37
38 hostname = socket.gethostname()
39 print(f"Server Name: {hostname}")
40 ip = socket.gethostbyname("DUCTRITRAN")
41 print(f"IP: {ip}")
42 server.serve_forever()
43
```

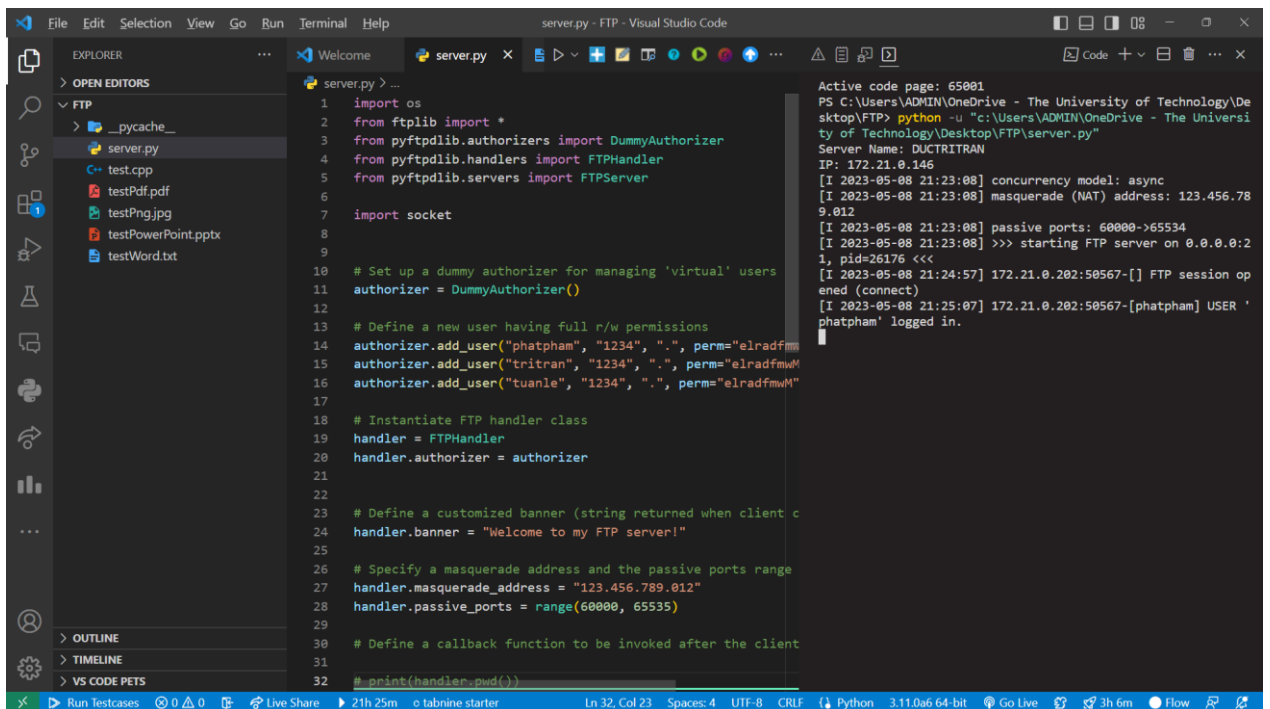
Active code page: 65001
PS C:\Users\ADMIN\OneDrive - The University of Technology\Desktop\FTP> python -u "c:\Users\ADMIN\OneDrive - The University of Technology\Desktop\FTP\server.py"
Server Name: DUCTRITRAN
IP: 172.21.0.146
[I 2023-05-08 21:23:08] concurrency model: async
[I 2023-05-08 21:23:08] masquerade (NAT) address: 123.456.789.012
[I 2023-05-08 21:23:08] passive ports: 60000->65534
[I 2023-05-08 21:23:08] >>> starting FTP server on 0.0.0.0:21, pid=26176 <<<

2. Client đăng nhập thành công vào hệ thống:

```
C:\WINDOWS\system32\cmd.exe - ftp 172.21.0.146
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Phat>ftp 172.21.0.146
Connected to 172.21.0.146.
220 Welcome to my FTP server!
530 Log in with USER and PASS first.
User (172.21.0.146:(none)): phatpham
331 Username ok, send password.
Password:
230 Login successful.
ftp> █
```

3. Server phản hồi có người dùng với username “phatpham” đăng nhập:



```
server.py - FTP - Visual Studio Code
server.py > ...
1 import os
2 from ftplib import *
3 from pyftplib.authorizers import DummyAuthorizer
4 from pyftplib.handlers import FTPHandler
5 from pyftplib.servers import FTPServer
6
7 import socket
8
9
10 # Set up a dummy authorizer for managing 'virtual' users
11 authorizer = DummyAuthorizer()
12
13 # Define a new user having full r/w permissions
14 authorizer.add_user("phatpham", "1234", ".", perm="elradfmwM")
15 authorizer.add_user("tritrans", "1234", ".", perm="elradfmwM")
16 authorizer.add_user("tuanle", "1234", ".", perm="elradfmwM")
17
18 # Instantiate FTP handler class
19 handler = FTPHandler
20 handler.authorizer = authorizer
21
22
23 # Define a customized banner (string returned when client connects)
24 handler.banner = "Welcome to my FTP server!"
25
26 # Specify a masquerade address and the passive ports range
27 handler.masquerade_address = "123.456.789.012"
28 handler.passive_ports = range(60000, 65535)
29
30 # Define a callback function to be invoked after the client
31 # connects
32 # print(handler.pwd())

Active code page: 65001
PS C:\Users\ADMIN\OneDrive - The University of Technology\Desktop> python -u "c:\Users\ADMIN\OneDrive - The University of Technology\Desktop\FTP\server.py"
Server Name: DUCTRITRAN
IP: 172.21.0.146
[I 2023-05-08 21:23:08] concurrency model: async
[I 2023-05-08 21:23:08] masquerade (NAT) address: 123.456.789.012
[I 2023-05-08 21:23:08] passive ports: 60000->65534
[I 2023-05-08 21:23:08] >>> starting FTP server on 0.0.0.0:21, pid=26176 <<<
[I 2023-05-08 21:24:57] 172.21.0.202:50567-[] FTP session opened (connect)
[I 2023-05-08 21:25:07] 172.21.0.202:50567-[phatpham] USER 'phatpham' logged in.
```

4. Người dùng liệt kê tất cả các file hiện đang có trong thư mục server:

```
C:\WINDOWS\system32\cmd.exe - ftp 172.21.0.146
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

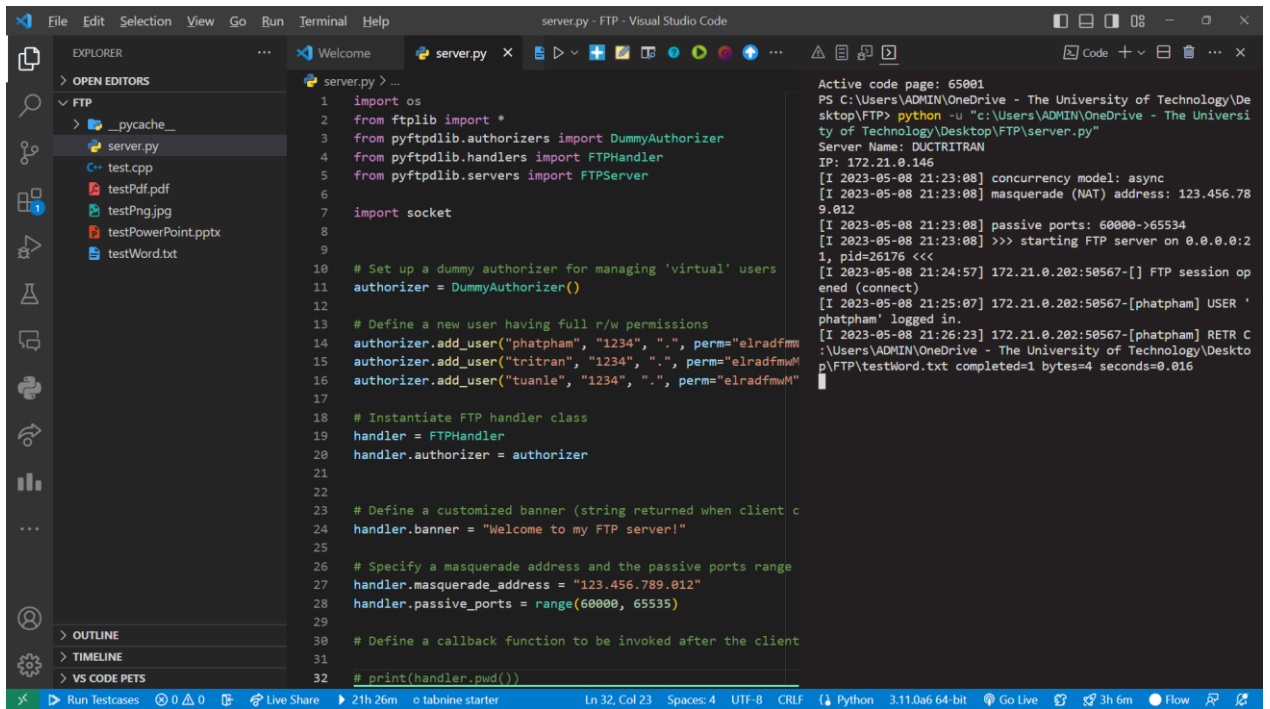
C:\Users\Phat>ftp 172.21.0.146
Connected to 172.21.0.146.
220 Welcome to my FTP server!
530 Log in with USER and PASS first.
User (172.21.0.146:(none)): phatpham
331 Username ok, send password.
Password:
230 Login successful.
ftp> ls
200 Active data connection established.
125 Data connection already open. Transfer starting.
_pycache_
server.py
test.cpp
testPdf.pdf
testPng.jpg
testPowerPoint.pptx
testWord.txt
226 Transfer complete.
ftp: 98 bytes received in 0.00Seconds 98000.00Kbytes/sec.
ftp>
```

5. Người dùng tải file testWord.txt từ server:

```
C:\WINDOWS\system32\cmd.exe - ftp 172.21.0.146
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Phat>ftp 172.21.0.146
Connected to 172.21.0.146.
220 Welcome to my FTP server!
530 Log in with USER and PASS first.
User (172.21.0.146:(none)): phatpham
331 Username ok, send password.
Password:
230 Login successful.
ftp> ls
200 Active data connection established.
125 Data connection already open. Transfer starting.
_pycache_
server.py
test.cpp
testPdf.pdf
testPng.jpg
testPowerPoint.pptx
testWord.txt
226 Transfer complete.
ftp: 98 bytes received in 0.00Seconds 98000.00Kbytes/sec.
ftp> get testWord.txt
200 Active data connection established.
125 Data connection already open. Transfer starting.
226 Transfer complete.
ftp: 4 bytes received in 0.00Seconds 4000.00Kbytes/sec.
ftp>
```

6. Server phản hồi có người dùng tải file testWord.txt từ server:

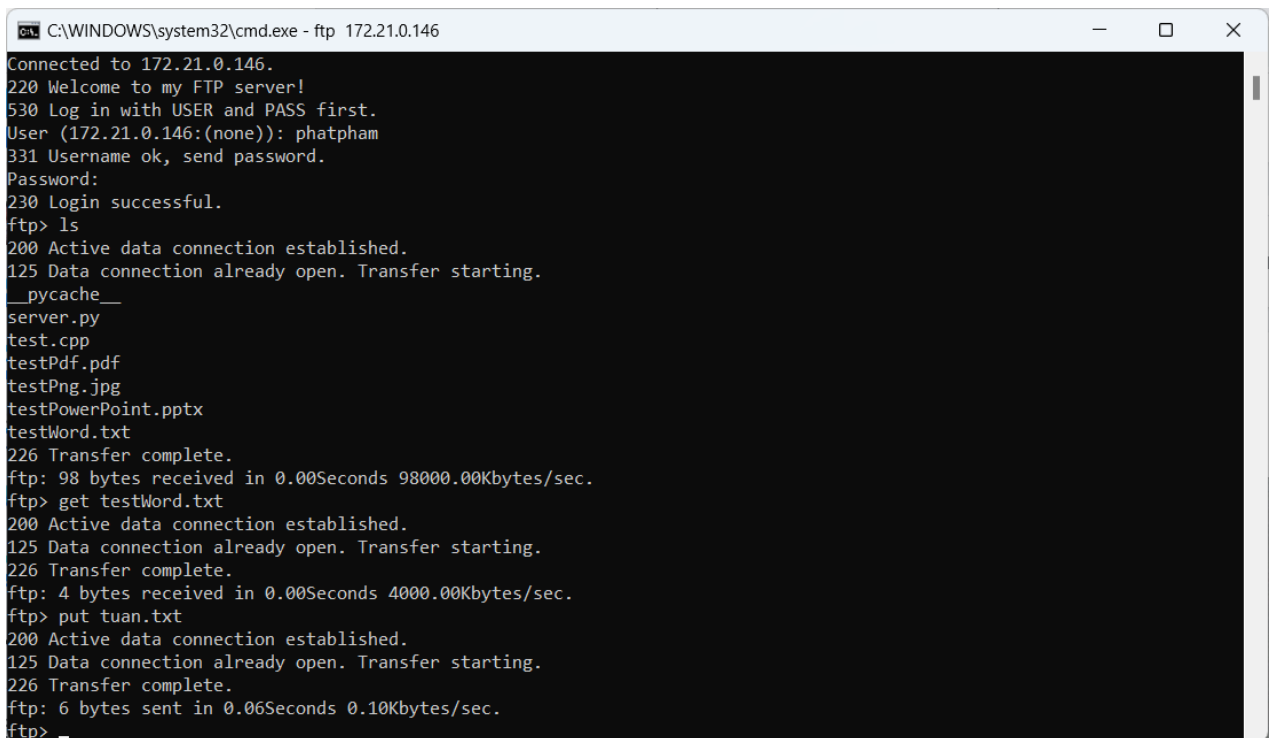


The screenshot shows the Visual Studio Code editor with a file named `server.py` open. The file contains Python code for an FTP server using `pyftplib`. The code sets up a dummy authorizer, adds two users ('phatpham' and 'tritrans'), and defines a handler. The terminal on the right shows the server's output, including the server's IP (172.21.0.146), concurrency model (async), masquerade address (123.456.789.012), and passive ports (60000-65534). The server is started on 0.0.0.0:21, and a session is opened for user 'phatpham'. The server then receives a request to retrieve the file `testWord.txt` from the user 'phatpham'.

```
1 import os
2 from ftplib import *
3 from pyftplib.authorizers import DummyAuthorizer
4 from pyftplib.handlers import FTPHandler
5 from pyftplib.servers import FTPServer
6
7 import socket
8
9
10 # Set up a dummy authorizer for managing 'virtual' users
11 authorizer = DummyAuthorizer()
12
13 # Define a new user having full r/w permissions
14 authorizer.add_user("phatpham", "1234", ".", perm="elradfmm")
15 authorizer.add_user("tritrans", "1234", ".", perm="elradfmm")
16 authorizer.add_user("tuanle", "1234", ".", perm="elradfmm")
17
18 # Instantiate FTP handler class
19 handler = FTPHandler
20 handler.authorizer = authorizer
21
22
23 # Define a customized banner (string returned when client connects)
24 handler.banner = "Welcome to my FTP server!"
25
26 # Specify a masquerade address and the passive ports range
27 handler.masquerade_address = "123.456.789.012"
28 handler.passive_ports = range(60000, 65535)
29
30 # Define a callback function to be invoked after the client connects
31
32 # print(handler.pwd())
```

Active code page: 65001
PS C:\Users\ADMIN\OneDrive - The University of Technology\Desktop\FTP> python -u "c:\Users\ADMIN\OneDrive - The University of Technology\Desktop\FTP\server.py"
Server Name: DUCTRITRAN
IP: 172.21.0.146
[I 2023-05-08 21:23:08] concurrency model: async
[I 2023-05-08 21:23:08] masquerade (NAT) address: 123.456.789.012
[I 2023-05-08 21:23:08] passive ports: 60000->65534
[I 2023-05-08 21:23:08] >>> starting FTP server on 0.0.0.0:21, pid=26176 <<<
[I 2023-05-08 21:24:57] 172.21.0.202:50567-[] FTP session opened (connect)
[I 2023-05-08 21:25:07] 172.21.0.202:50567-[phatpham] USER 'phatpham' logged in.
[I 2023-05-08 21:26:23] 172.21.0.202:50567-[phatpham] RETR C:\Users\ADMIN\OneDrive - The University of Technology\Desktop\FTP\testWord.txt completed=1 bytes=4 seconds=0.016

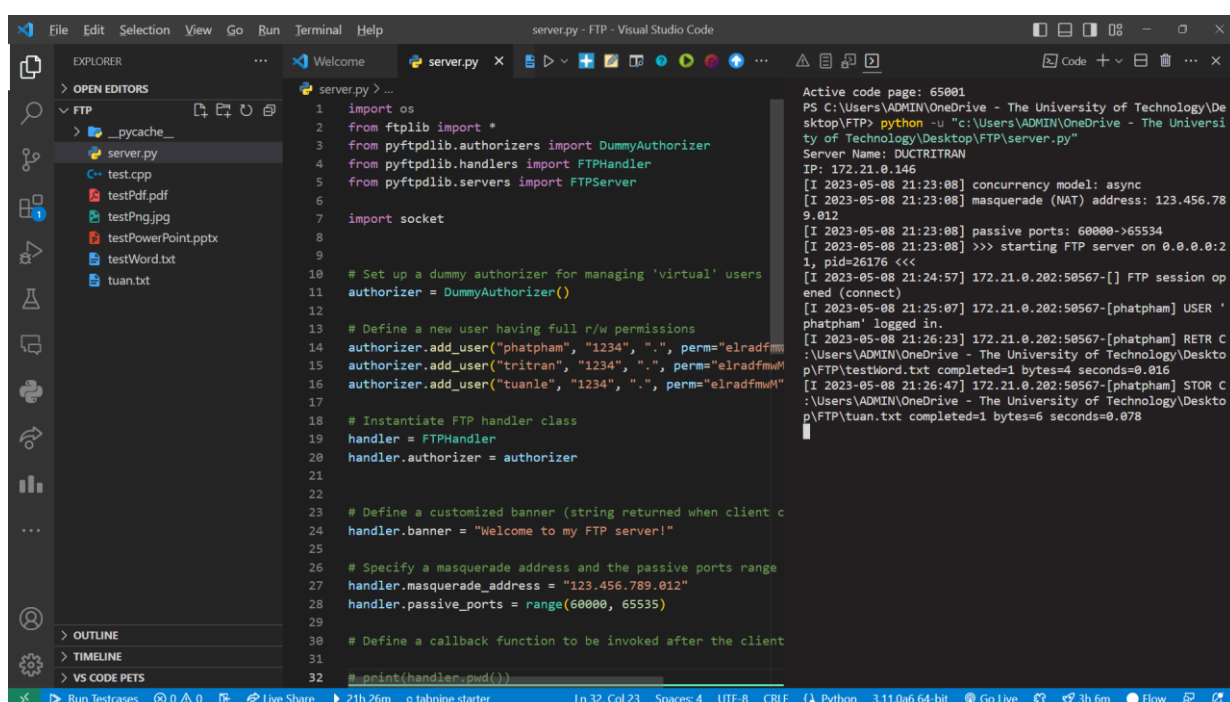
7. Người dùng up file tuan.txt lên server:



The screenshot shows a Windows command prompt window with the title `C:\WINDOWS\system32\cmd.exe - ftp 172.21.0.146`. The prompt shows the user connected to the FTP server at 172.21.0.146. The user logs in with the username 'phatpham' and password '1234'. The user then lists the files on the server, and the output shows the files: `__pycache__`, `server.py`, `test.cpp`, `testPdf.pdf`, `testPng.jpg`, `testPowerPoint.pptx`, and `testWord.txt`. The user then uploads the file `tuan.txt` to the server. The output shows the file being uploaded successfully.

```
Connected to 172.21.0.146.
220 Welcome to my FTP server!
530 Log in with USER and PASS first.
User (172.21.0.146:(none)): phatpham
331 Username ok, send password.
Password:
230 Login successful.
ftp> ls
200 Active data connection established.
125 Data connection already open. Transfer starting.
__pycache__
server.py
test.cpp
testPdf.pdf
testPng.jpg
testPowerPoint.pptx
testWord.txt
226 Transfer complete.
ftp> get testWord.txt
200 Active data connection established.
125 Data connection already open. Transfer starting.
226 Transfer complete.
ftp> put tuan.txt
200 Active data connection established.
125 Data connection already open. Transfer starting.
226 Transfer complete.
ftp>
```

8. Server nhận được file tuan.txt từ người dùng:



The screenshot shows a Visual Studio Code window with a Python file named `server.py` and a terminal window. The `server.py` file contains the following code:

```
1 import os
2 from ftplib import *
3 from pyftplib.authorizers import DummyAuthorizer
4 from pyftplib.handlers import FTPHandler
5 from pyftplib.servers import FTPServer
6
7 import socket
8
9
10 # Set up a dummy authorizer for managing 'virtual' users
11 authorizer = DummyAuthorizer()
12
13 # Define a new user having full r/w permissions
14 authorizer.add_user("phatpham", "1234", ".", perm="elradfmw")
15 authorizer.add_user("tritrtran", "1234", ".", perm="elradfmw")
16 authorizer.add_user("tuanle", "1234", ".", perm="elradfmw")
17
18 # Instantiate FTP handler class
19 handler = FTPHandler
20 handler.authorizer = authorizer
21
22
23 # Define a customized banner (string returned when client connects)
24 handler.banner = "Welcome to my FTP server!"
25
26 # Specify a masquerade address and the passive ports range
27 handler.masquerade_address = "123.456.789.012"
28 handler.passive_ports = range(60000, 65535)
29
30 # Define a callback function to be invoked after the client connects
31 def callback():
32     print(handler.pwd())
```

The terminal window shows the output of the script, which is a log of the FTP server's activity. It includes the server's IP address (172.21.0.146), the concurrency model (async), the masquerade address (123.456.789.012), the passive ports range (60000-65535), and the log of the FTP session for the user 'phatpham'.

3.3. NHẬN XÉT ĐÁNH GIÁ KẾT QUẢ

- Qua kết quả thực nghiệm, chúng em có những nhận xét như sau:

- "ftplib" là một module trong Python cung cấp các chức năng để kết nối và tương tác với một máy chủ FTP (File Transfer Protocol). ftplib cung cấp các phương thức để thiết lập kết nối FTP, xác thực đăng nhập, tải xuống và tải lên tệp, xóa tệp và thư mục, và thực hiện các thao tác khác trên máy chủ FTP
- "ftplib" cung cấp cho các nhà phát triển Python một cách tiện lợi và đơn giản để thao tác với máy chủ FTP từ các chương trình Python. Nó được sử dụng rộng rãi trong các ứng dụng web và các chương trình tự động hóa dựa trên Python để thực hiện các tác vụ FTP, ví dụ như sao lưu dữ liệu hoặc tải xuống các tệp từ máy chủ FTP.

- Tuy nhiên, module này có một số hạn chế như:

- Không hỗ trợ chuyển đổi tệp tự động giữa các định dạng mã hóa. Nếu máy chủ FTP yêu cầu tệp được truyền dưới một định dạng mã hóa cụ thể, bạn cần tự thực hiện việc chuyển đổi tệp.
- Không hỗ trợ tối đa số kết nối. Với ftplib, bạn chỉ có thể thiết lập một kết nối FTP tại một thời điểm. Điều này có thể ảnh hưởng đến hiệu suất nếu bạn cần tải lên hoặc tải xuống nhiều tệp từ máy chủ FTP cùng một lúc.
- Không hỗ trợ các tính năng cao cấp của FTP như SSL / TLS hoặc SSH. Nếu bạn cần kết nối với một máy chủ FTP sử dụng SSL / TLS hoặc SSH để bảo

mật, bạn cần sử dụng các thư viện khác hoặc các gói mở rộng để kết nối với máy chủ FTP.

- Không hỗ trợ các tính năng FTP mở rộng, ví dụ như kiểm soát băng thông hoặc phân quyền truy cập. Nếu bạn cần sử dụng các tính năng mở rộng của FTP, bạn có thể cần sử dụng các thư viện khác hoặc các gói mở rộng để kết nối với máy chủ FTP.

3.4. KẾT CHƯƠNG

- Trong chương này, chúng ta đã tìm hiểu giao thức FTP, sử dụng python để lập trình chương trình FTP Server để truyền tải file. Chúng ta sử dụng thư viện ftplib của python để xem xét cách giao thức FTP truyền tải file. Chúng ta cũng đã đánh giá các ưu và nhược điểm của giao thức FTP và thư viện ftplib.

- Tổng quan, giao thức FTP là một giao thức được sử dụng để truyền tải tệp giữa các máy tính trên mạng. FTP sử dụng cơ chế yêu cầu-đáp ứng, trong đó một máy tính yêu cầu một tệp từ một máy tính khác, và máy tính đó phản hồi với các dữ liệu tương ứng. FTP hoạt động trên cổng 21 và sử dụng các lệnh chuẩn để truyền tải và quản lý tệp. FTP có hai chế độ truyền tải dữ liệu: ASCII và binary. Chế độ ASCII được sử dụng để truyền tải các tệp văn bản, trong khi chế độ binary được sử dụng để truyền tải các tệp nhị phân, chẳng hạn như các tệp âm thanh và video. Tuy nhiên, FTP không được bảo mật và các thông tin đăng nhập cũng như dữ liệu truyền tải được truyền tải dưới dạng văn bản đơn giản, không được mã hóa. Do đó, nó không được khuyến khích sử dụng trong các môi trường mạng công cộng hoặc khi truyền tải dữ liệu nhạy cảm. Thay vào đó, các giao thức bảo mật hơn như SFTP hoặc FTPS được sử dụng thay thế.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. KẾT QUẢ ĐẠT ĐƯỢC

- Trong quá trình tìm hiểu và sử dụng giao thức PTP, chúng ta đã có được các kết quả như sau:

- Hiểu được cách sử dụng giao thức PTP để truyền tải file.
- Sử dụng ngôn ngữ lập trình python và thư viện ftplib để lập trình giao thức FTP, truyền tải file giữa client và server.
- Đánh giá được các ưu và nhược điểm của giao thức FTP trong việc truyền tải file giữa client và server.

- Dựa trên các kết quả này, chúng ta có thể sử dụng ngôn ngữ lập trình python để lập trình một chương trình truyền tải file sử dụng giao thức FTP đơn giản để truyền tải file giữa server và client. Tuy nhiên, đối với các ứng dụng phức tạp hơn và cần sự bảo mật hơn, chúng ta nên xem xét các giao thức khác các giao thức bảo mật hơn như SFTP hoặc FTPS.

2. KIẾN NGHỊ VÀ HƯỚNG PHÁT TRIỂN

Một số hướng nghiên cứu và phát triển của đề tài như sau:

- Xây dựng chương trình hoàn thiện với giao diện hoàn chỉnh, dễ sử dụng ...
- Đánh giá hiệu năng trên các môi trường khác nhau ...
- Kiểm thử các chức năng của chương trình ...

TÀI LIỆU THAM KHẢO

- [1] <https://pyftplib.readthedocs.io/en/latest/>
- [2] <https://www.digitalocean.com/community/tutorials/how-to-set-up-vsftpd-for-a-user-s-directory-on-ubuntu-16-04>
- [3] <https://www.digitalocean.com/community/tutorials/how-to-set-up-vsftpd-for-a-user-s-directory-on-centos-7>
- [4] <https://www.techwalla.com/articles/how-to-set-up-an-ftp-server-on-windows-xp>
- [5] <https://www.youtube.com/watch?v=lznj1w7RH8M>
- [6] <https://www.techcoil.com/blog/how-to-create-an-ftp-server-with-python-ftplib-and-pyftplib/>
- [7] <https://www.digitalocean.com/community/tutorials/how-to-set-up-vsftpd-for-a-user-s-directory-on-ubuntu-18-04>
- [8] <https://www.howtogeek.com/226600/how-to-install-and-configure-a-ftp-server-on-windows-10/>
- [9] <https://docs.microsoft.com/en-us/iis/publish/using-the-ftp-service/creating-a-new-ftp-site-in-iis-7>
- [10] <https://www.digitalocean.com/community/tutorials/how-to-set-up-vsftpd-for-a-user-s-directory-on-ubuntu-18-04>

PHỤ LỤC

1. Nội dung mã nguồn: https://github.com/tranductri2003/FTP-Server_DUT