

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO ĐỒ ÁN

MẠNG MÁY TÍNH

Lập trình ứng dụng Socket truyền tải File

Giảng viên: Cô Chung Thùy Linh

Mã học phần: CSC10008

Nhóm 01 - 23CTT3

Tp. Hồ Chí Minh, ngày 17 tháng 12 năm 2024

MỤC LỤC

MỤC LỤC	1
DANH MỤC HÌNH ẢNH	1
1. TỔNG QUAN	2
1.1. Thông tin nhóm	2
1.2. Thông tin đồ án	2
2. ĐÁNH GIÁ MỨC ĐỘ HOÀN THIỆN CÁC CHỨC NĂNG	2
2.1. Lập trình ứng dụng Socket sử dụng TCP tại tầng Transport	2
2.2. Lập trình ứng dụng Socket sử dụng UDP tại tầng Transport	3
3. KỊCH BẢN GIAO TIẾP CỦA CHƯƠNG TRÌNH	4
3.1. Giao tiếp giữa Server và Client qua TCP	4
3.2. Giao tiếp giữa Server và Client qua UDP	5
4. MÔI TRƯỜNG LẬP TRÌNH VÀ CÁC FRAMEWORK HỖ TRỢ	7
5. HƯỚNG DẪN SỬ DỤNG	7
6. BẢNG PHÂN CÔNG	12
7. TÀI LIỆU THAM KHẢO	12

DANH MỤC HÌNH ẢNH

Hình 1. Giao diện Server đang lắng nghe	8
Hình 2. Giao diện của TCP client	8
Hình 3. Sau khi Client nhận được file có thể tải từ Server	9
Hình 4. Sau khi Client gửi danh sách các file muốn tải cho Server	10
Hình 5. Client trong quá trình đồng thời quét file input và tải file từ server	10
Hình 6. Thanh tiến trình khi tải file đối với giao thức TCP	11
Hình 7. Thanh tiến trình khi tải file đối với giao thức UDP	11
Hình 8. Client thông báo tải các file thành công từ Server	11
Hình 9. Client ngắt kết nối với Server	12

1. TỔNG QUAN

1.1. Thông tin nhóm

STT	MSSV	Họ và tên	Email
1	23120242	Nguyễn Văn Bình Dương	ngduong5124@gmail.com
2	23120243	Trần Nhật Dương	nhatduong0411@gmail.com
3	23120252	Nguyễn Phúc Hậu	phuchau.2005.vlg@gmail.com

1.2. Thông tin đồ án

- Mã học phần: CSC10008
- Tên học phần: Mạng máy tính
- Giảng viên thực hành: cô Chung Thùy Linh
- Tên bài tập: Lập trình Ứng dụng Socket
- Hình thức: Tìm hiểu + triển khai thử nghiệm + thảo luận

2. ĐÁNH GIÁ MỨC ĐỘ HOÀN THIỆN CÁC CHỨC NĂNG

2.1. Lập trình ứng dụng Socket sử dụng TCP tại tầng Transport

STT	Đối tượng	Chức năng	Hoàn thiện	Ghi chú
1	Server	Sử dụng file input.txt để lưu danh sách các file (tên file và dung lượng) cho phép Client tải.	100%	

		Server được nhận yêu cầu từ Client tải file có tên và offset xác định.	100%	
		File chứa trên Server và không cắt thành các file nhỏ hơn.	100%	
		Server có thể phục vụ đồng thời yêu cầu download file từ nhiều client.	100%	
2	Client	Kết nối đến Server.	100%	
		Nhận thông tin danh sách các file từ Server và hiển thị trên màn hình.	100%	
		Ghi nhận các tên file mới cần tải từ file input.txt	100%	
		Duyệt file input.txt 5 giây một lần và không tải những file đã duyệt trước đó.	100%	
		Hiển thị phần trăm download các chunk.	100%	
		Nhiều Client có thể tải đầy đủ các file.	100%	

2.2. Lập trình ứng dụng Socket sử dụng UDP tại tầng Transport

STT	Đối tượng	Chức năng	Hoàn thiện	Ghi chú
-----	-----------	-----------	------------	---------

1	Server	Sử dụng file input.txt để lưu danh sách các file (tên file và dung lượng) cho phép Client tải.	100%	
		Server được nhận yêu cầu từ Client tải file có tên và offset xác định.	100%	
		File chứa trên Server và không cắt thành các file nhỏ hơn.	100%	
		Server chỉ phục vụ duy nhất một Client.	100%	
2	Client	Kết nối đến Server.	100%	
		Nhận thông tin danh sách các file từ Server và hiển thị trên màn hình.	100%	
		Ghi nhận các tên file mới cần tải từ file input.txt	100%	
		Duyệt file input.txt 5 giây một lần và không tải những file đã duyệt trước đó.	100%	
		Hiển thị phần trăm download các chunk.	100%	
		Nhiều client có thể tải đầy đủ các file.	100%	

3. KỊCH BẢN GIAO TIẾP CỦA CHƯƠNG TRÌNH

3.1. Giao tiếp giữa Server và Client qua TCP

- Giao thức sử dụng: TCP
- Cấu trúc thông điệp:

Danh sách file (từ Server gửi đến Client):

- o Cấu trúc: Chuỗi các file (tên file - kích thước) cách nhau bằng dấu ‘,’

- Trình tự:
 - Gửi độ dài chuỗi (4 byte, !I).
 - Gửi nội dung chuỗi đã mã hóa UTF-8.

Yêu cầu tải file (từ Client gửi đến Server):

- Cấu trúc: Chuỗi dạng “name,start,end” (tên file và vị trí bắt đầu/kết thúc chunk).
- Trình tự:
 - Gửi tín hiệu "1" (yêu cầu chunk).
 - Gửi độ dài chuỗi (4 byte, !I).
 - Gửi chuỗi yêu cầu.

Gửi dữ liệu file (từ Server đến Client):

- Cấu trúc: Dữ liệu file gửi theo từng chunk (BUFFER_SIZE).
- Trình tự:
 - Đọc dữ liệu từ start đến end.
 - Gửi từng chunk qua socket cho đến khi hoàn thành.

Thông điệp điều khiển:

- Tín hiệu từ Client:
 - "0": Yêu cầu danh sách file.
 - "1": Yêu cầu tải chunk file.
 - "close": Ngắt kết nối

- Kịch bản giao tiếp:

Khởi động Server

- Server khởi chạy và lắng nghe kết nối trên một địa chỉ và cổng nhất định).
- Server đọc danh sách các file có sẵn (từ **listfile.txt**) và hiển thị

Client kết nối đến Server

- Client kết nối tới địa chỉ IP và cổng của Server.
- Server nhận tín hiệu từ Client và xác định yêu cầu:
 - Nếu tín hiệu là 0: Client yêu cầu danh sách file.
 - Nếu tín hiệu là 1: Client yêu cầu tải file (hoặc một phần của file).
- Truyền danh sách file
 - Server gửi danh sách file có thể tải cho Client dưới dạng chuỗi đã được đóng gói (mã hóa độ dài trước).
 - Client nhận và hiển thị danh sách này trong giao diện.

Client yêu cầu tải file

- Client đọc danh sách file muốn tải từ **input.txt** và kiểm tra với danh sách file từ Server.
- Với mỗi file hợp lệ:
 - Gửi yêu cầu tải file kèm tên file, vị trí bắt đầu và kết thúc của chunk.
 - Server nhận yêu cầu, đọc phần dữ liệu tương ứng trong file và gửi cho Client.
 - Nhận và ghép file trên Client
- Client nhận từng chunk dữ liệu và ghi vào file ở đúng vị trí.
- Tiến trình được cập nhật trong giao diện bằng thanh tiến trình (progress bar).
- Khi hoàn thành, Client lưu tên file vào **downloaded.txt**

Đóng kết nối

Khi Client hoàn tất việc tải file hoặc người dùng dừng chương trình:

- Gửi tín hiệu "**close**" tới Server.
- Server ghi log và đóng kết nối với Client.

3.2. Giao tiếp giữa Server và Client qua UDP

- **Giao thức sử dụng: UDP**
- **Cấu trúc thông điệp:**

Danh sách file từ Server:

| **seq_num (4 bytes)** | **content (<= BUFFER_SIZE bytes)** |

- seq_num: Số thứ tự gói tin.
- content: Dữ liệu danh sách file.

Yêu cầu file từ Client:

| **filename_length (4 bytes)** | **filename (filename_length bytes)** |

- filename_length: Độ dài tên file.
- filename: Tên file cần tải.

Dữ liệu file từ Server:

| **seq_num (4 bytes)** | **checksum (4 bytes)** | **content (<=BUFFER_SIZE bytes)**|

- seq_num: Số thứ tự chunk.

- checksum: Giá trị kiểm tra lỗi.
- content: Dữ liệu chunk.

Phản hồi từ Client (ACK/NACK):

| status (1 byte) |

- status: "1" (ACK) hoặc "0" (NACK).

Thông báo: Chuỗi ASCII như "File not found" hoặc "Connection closed".

- Kịch bản giao tiếp:

Khởi động Server

- Server được khởi tạo tại địa chỉ IP SERVER_HOST và cổng SERVER_PORT, sử dụng giao thức UDP.
- Server kiểm tra thư mục chứa các file (thư mục fordown) và xây dựng danh sách file hiện có để chuẩn bị gửi cho Client.
- Server lắng nghe kết nối từ Client qua socket UDP.

Client gửi yêu cầu danh sách file

- Khi khởi động, Client gửi một tín hiệu ("0") đến Server qua UDP để yêu cầu danh sách các file có sẵn trên Server.

Server gửi danh sách file

- Server nhận yêu cầu từ Client và gửi chiều dài gói tin chứa danh sách file cho Client
- Mỗi gói tin được gán thứ tự ở trước để Client nhận và ghép các gói tin theo thứ tự chính xác

Client nhận và hiển thị danh sách file

- Client nhận dữ liệu từ Server. Nếu có nhiều gói tin, Client sẽ ghép lại các gói tin theo thứ tự chỉ số để tái tạo danh sách file đầy đủ.
- Client hiển thị danh sách file để người dùng biết các file nào có sẵn để tải.

Người dùng nhập danh sách file cần tải (Input File)

- Người dùng nhập tên các file cần tải xuống trong file input.txt.
- Client đọc file input.txt và lọc ra những file cần tải mà Server sẵn có. Đồng thời, Client cũng kiểm tra các file đã tải trước đó trong file downloaded_files.txt để tránh tải lại.

Client gửi yêu cầu tải file

- Với mỗi file trong danh sách yêu cầu
- Nếu file tồn tại trên Server, Client gửi tên file và kích thước file đến Server.
- Nếu file không tồn tại trên Server, Client thông báo file không tồn tại và bỏ qua file đó.

Server gửi dữ liệu file

- Server chia nhỏ file thành các chunk có kích thước tối đa 1024 bytes.
- Để tăng tốc độ, Server sử dụng đa luồng để gửi các phần dữ liệu này đồng thời.
- Dữ liệu được gửi kèm theo check_sum để Client có thể kiểm tra tính bảo toàn của dữ liệu

Client nhận và lưu file

- Client nhận từng phần dữ liệu từ Server, dựa vào check_sum được đính kèm ở đầu để kiểm tra tính bảo toàn của dữ liệu
- Sau khi nhận đủ các phần, Client lưu file vào thư mục hiện tại và cập nhật thông tin file đã tải vào downloaded_files.txt.

Lặp lại quá trình quét file input

- Sau mỗi chu kỳ quét, client tiếp tục kiểm tra input.txt để xem có file mới nào cần tải hay không. Chu kỳ quét diễn ra mỗi 5 giây.

Kết thúc

- Khi người dùng nhấn tổ hợp phím Ctrl + C, client sẽ xử lý tín hiệu kết

thức chương trình:

- Đóng socket UDP.
- Hiện thị thông báo và thoát chương trình.
- **Phương thức truyền tin cậy (RDT):**
 - Checksum: Kiểm tra tính toàn vẹn của dữ liệu. Gửi mã kiểm tra (checksum) kèm theo gói tin và xác nhận lại ở phía nhận.
 - Acknowledgment (ACK):
 - Nếu gói tin hợp lệ (checksum đúng), gửi ACK ("1") để xác nhận.
 - Nếu không, gửi NAK ("0") để yêu cầu gửi lại.
 - Phân mảnh và ghép nối dữ liệu:
 - Chia tệp lớn thành các mảnh nhỏ (fragment) dựa trên kích thước BUFFER_SIZE cụ thể đây là 1024 bytes
 - Phía nhận và ghép lại các mảnh đúng vị trí.

4. MÔI TRƯỜNG LẬP TRÌNH VÀ CÁC FRAMEWORK HỖ TRỢ

Ngôn ngữ và môi trường lập trình:

- Ngôn ngữ lập trình: Python
- Môi trường phát triển: Visual Studio Code.
- Hệ điều hành: Windows.

Các thư viện và framework được sử dụng:

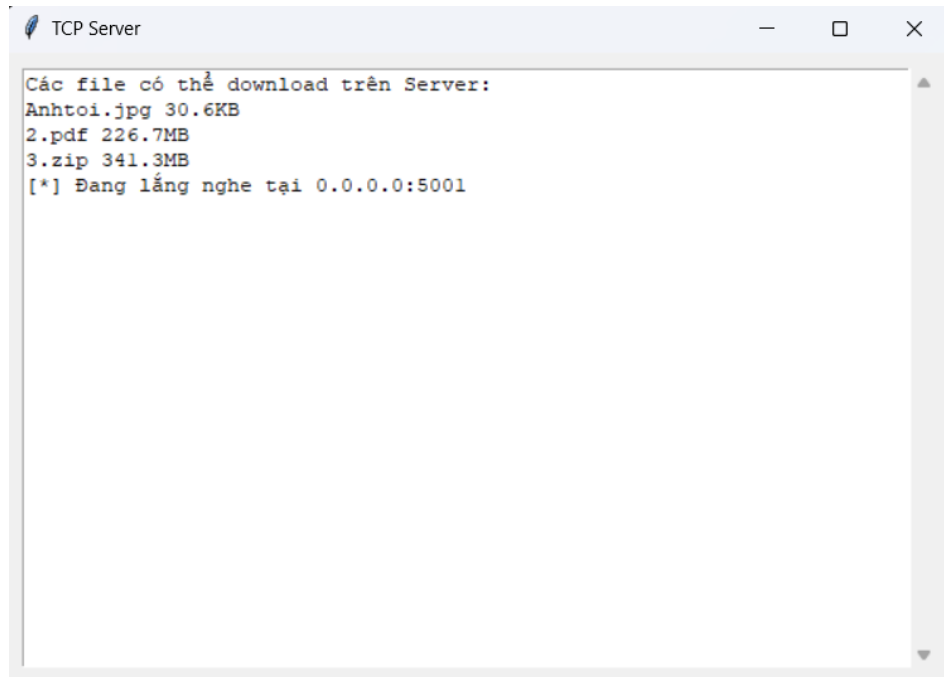
- Thư viện socket: các hàm cơ bản để truyền tin qua socket.
- Thư viện os: lấy thông tin, kích thước dữ liệu của file.
- Thư viện tqdm: sử dụng vẽ thanh tiến trình.
- Thư viện sys: xử lý tổ hợp phím Ctrl + C để dừng chương trình.
- Thư viện system: xử lý tổ hợp phím Ctrl + C để dừng chương trình.
- Thư viện threading: xử lý đa luồng, để Server xử lý cùng lúc các kết nối từ nhiều client.
- Thư viện tkinter: thiết kế giao diện.

5. HƯỚNG DẪN SỬ DỤNG

Hướng dẫn sử dụng chương trình cho người dùng:

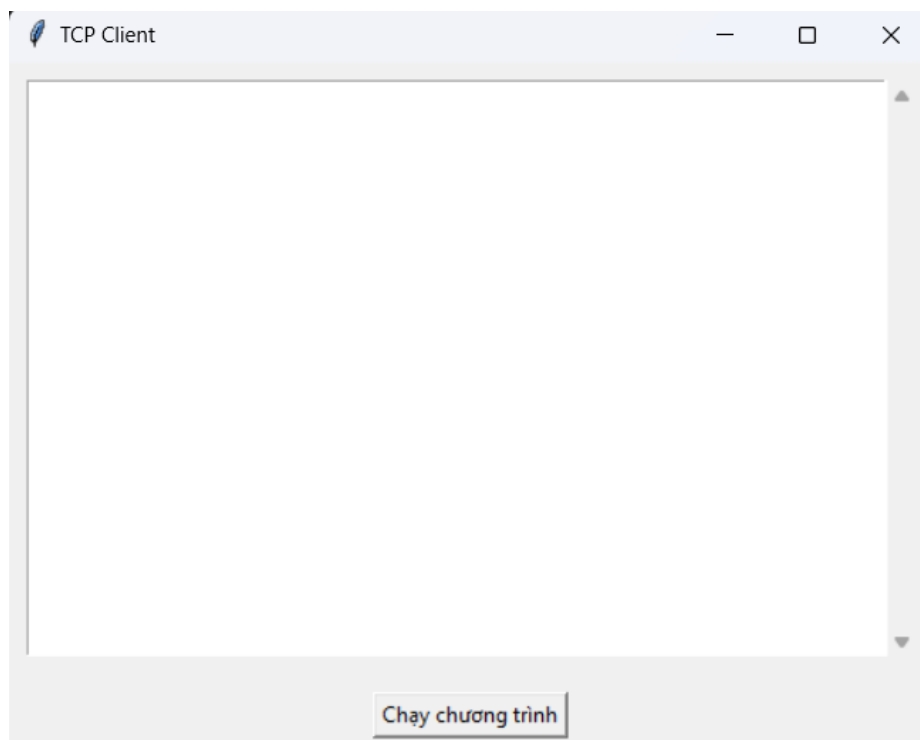
- Khi khởi động chương trình Server, một màn hình giao diện sẽ hiện ra như bên dưới bao gồm các thông tin như: danh sách file có thể tải, SERVER_IP,

SERVER_PORT đang lắng nghe.



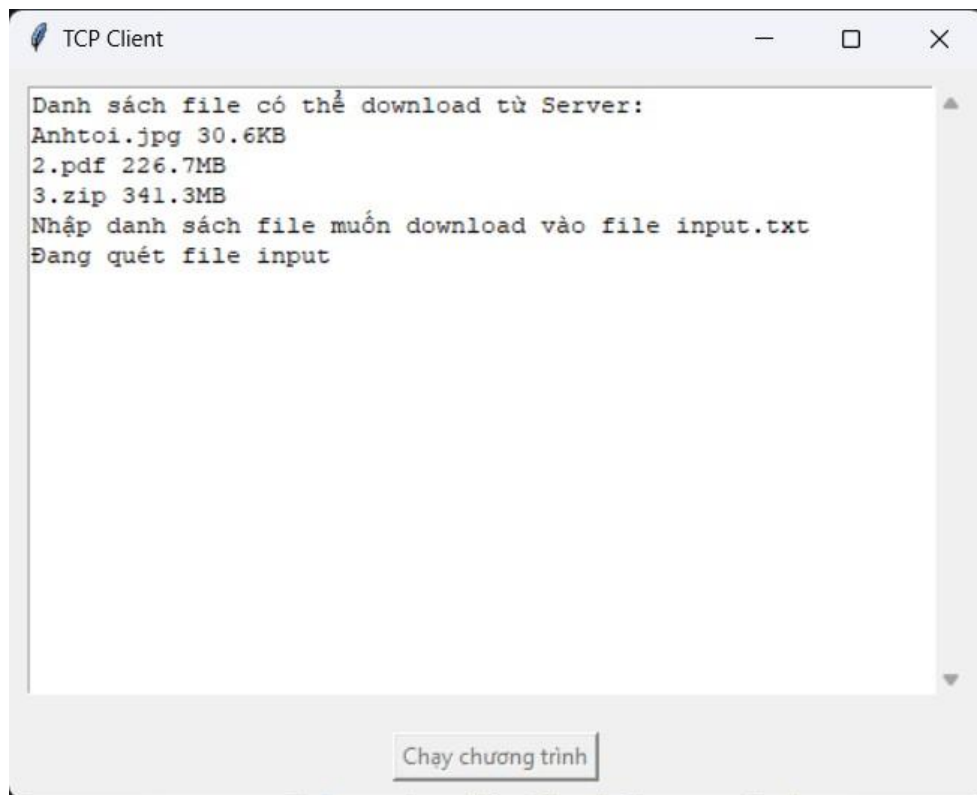
Hình 1: Giao diện Server đang lắng nghe

- Phía người dùng khi khởi động chương trình sẽ hiện ra giao diện phía dưới, đây là một màn hình trống và sẽ hiển thị những thông tin khi người dùng thao tác trên chương trình.



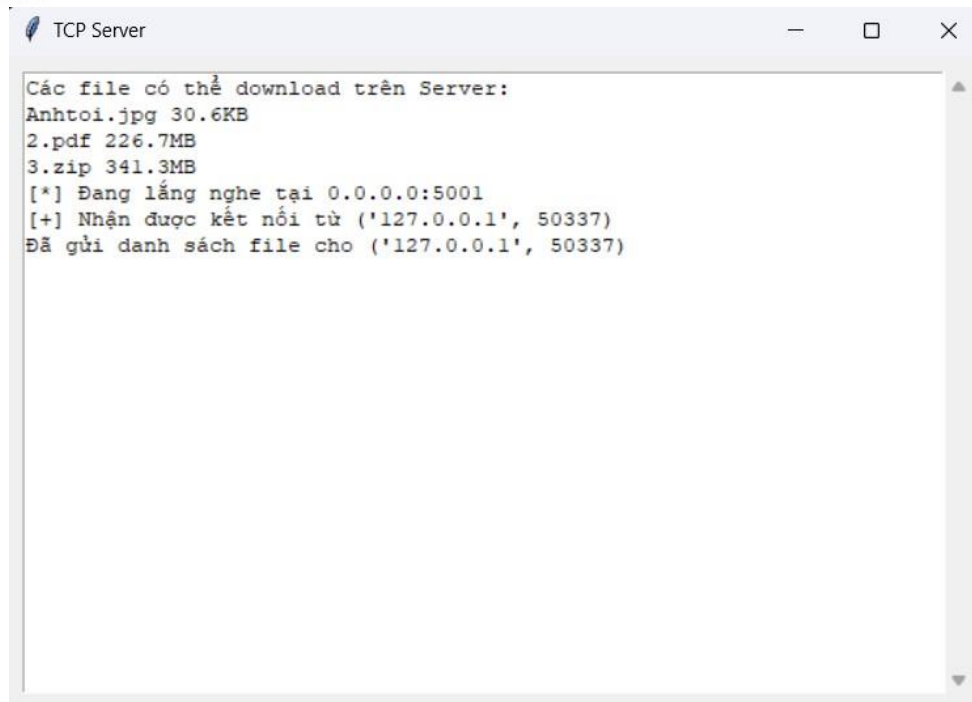
Hình 2: Giao diện của TCP client

- Sau khi nhấn nút “Chạy chương trình”, Server sẽ hiển thị kết nối từ Client và đồng thời Client hiển thị danh sách file có thể download được từ Server.

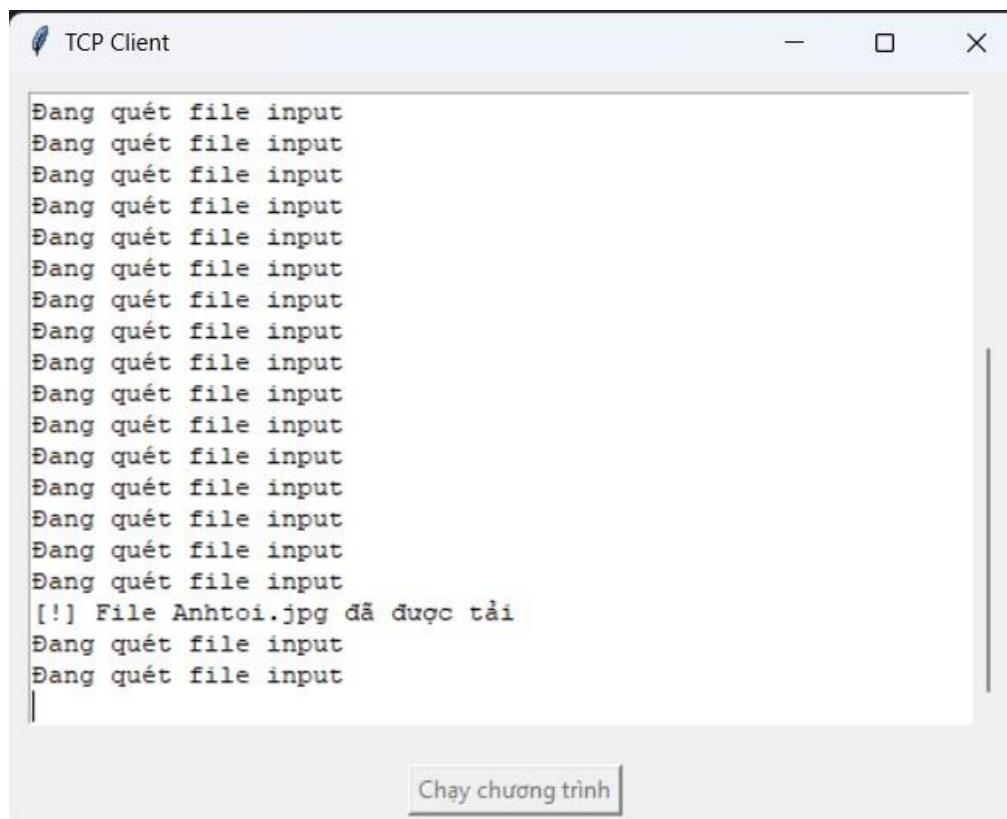


Hình 3: Sau khi Client nhận được file có thể tải từ Server

- Người dùng sẽ nhập các file muốn tải vào file “input.txt” và Client sẽ hiển thị thông báo quét file “input.txt” mỗi 5s, nếu file tồn tại trên Server thì sẽ được tải, ngược lại nếu file đã được tải hoặc không tồn tại thì sẽ hiện thông báo.



Hình 4: Sau khi Client gửi danh sách các file muốn tải cho Server



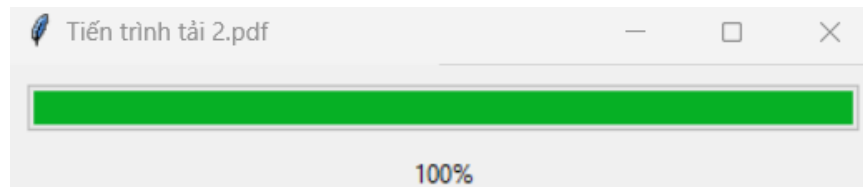
Hình 5: Client trong quá trình đồng thời quét file input và tải file từ server

- Trong quá trình download file, thanh tiến trình sẽ được hiển thị như sau:

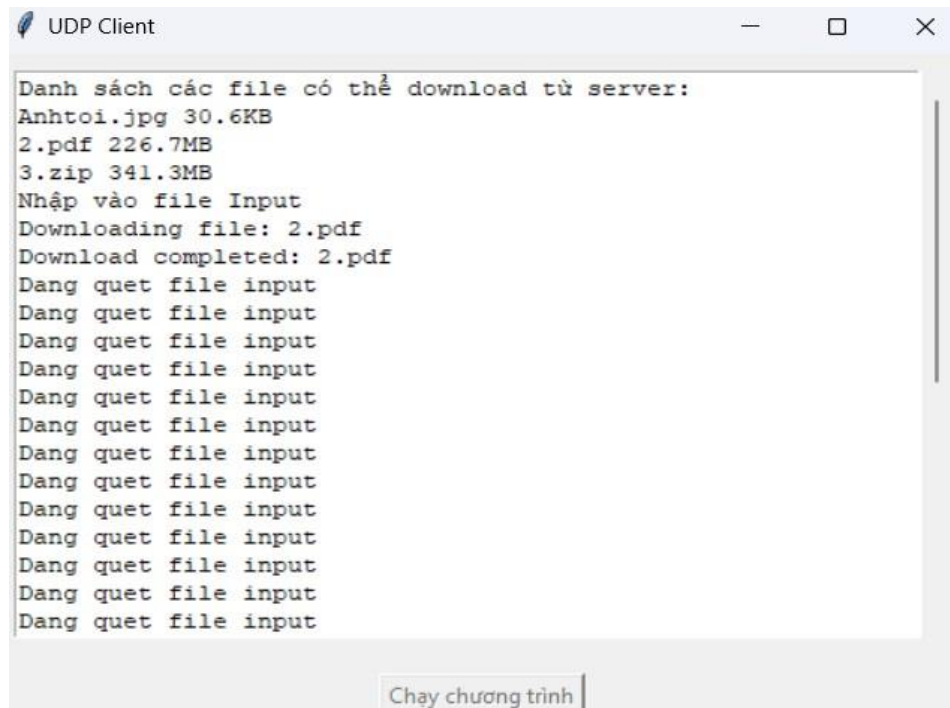
Tiến trình tài 2.pdf

Chunk	Progress
Chunk 1:	100%
Chunk 2:	100%
Chunk 3:	100%
Chunk 4:	100%

Đối với giao thức UDP:

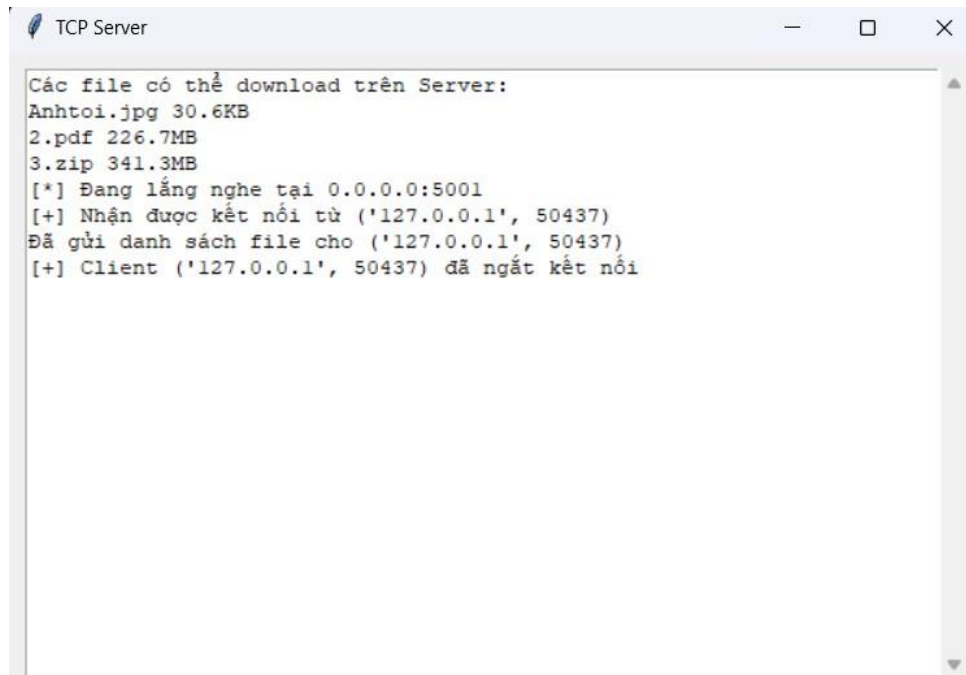


- Sau khi download thành công, Client sẽ thông báo Download completed



13

- Sau khi người dùng kết thúc Client bằng tổ hợp phím Ctrl + C, Server sẽ hiển thị thông báo ngừng kết nối



Hình 9: Client ngắt kết nối với Server

6. BẢNG PHÂN CÔNG

STT	Tên/ MSSV	Nhiệm vụ
1	Nguyễn Văn Bình Dương 23120242	+ Gửi danh sách file từ server đến client. + Client nhận và hiển thị danh sách file từ server. + Xử lý Ctrl + C. + Viết báo cáo.
2	Trần Nhật Dương 23120243	+ Xử lý kết nối ở UDP. + Gửi chunk và xử lý file nhận với UDP. + Tạo cơ chế để truyền dữ liệu tin cậy RDT. + Nhận xét tổng quát.
3	Nguyễn Phúc Hậu 23120252	+ Xử lý kết nối ở TCP. + Xử lý đọc tên file mới và cập nhật tên file cũ. + Tạo kết nối song song khi gửi các chunk. + Giao diện TCP + UDP .

7. TÀI LIỆU THAM KHẢO:

[1] Python Docs: socket — Low-level networking interface

<https://docs.python.org/3/library/socket.html> , access date: 18/11/2024

[2] Geeks For geeks: Socket Programming in Python

<https://www.geeksforgeeks.org/socket-programming-python/> , access date: 18/11/2024

[3] Stack Overflow: [https://stackoverflow.com/questions/7749341/basic-python-client- socket-example](https://stackoverflow.com/questions/7749341/basic-python-client-socket-example) , access date: 21/11/2024

[4] NeuralNine: File Transfer via Sockets in Python

<https://www.youtube.com/watch?v=qFVoMo6OMsQ> , access date: 6/12/2024

[5] Rob Mulla: Easy Python Progress Bars with tqdm

<https://www.youtube.com/watch?v=n4E7of9BINo> , access date: 8/12/2024

[6] Viblo: Đa luồng trong Python (Multithreading)

<https://viblo.asia/p/da-luong-trong-python-multithreading-WAyK8MO6ZxX> , access date: 7/12/2024