

## BÁO CÁO ĐỒ ÁN SOCKET

### I/ Giới thiệu sơ lược về nhóm và đồ án:

#### 1/ Các thành viên của nhóm:

STT	Họ và tên	MSSV	Email	Lớp
1	Mạch Quốc Tấn	23127115	mqtan23@clc.fitus.edu.vn	23CLC09
2	Lê Nguyễn Nhật Khánh	23127206	lnnkhanh23@clc.fitus.edu.vn	23CLC09
3	Nguyễn Thành Dâng	23127334	ntdang23@clc.fitus.edu.vn	23CLC09

#### 2/ Giới thiệu sơ lược về đồ án:

##### Tên đồ án:

- Lập trình socket truyền tải file theo mô hình client-server.

##### Mục tiêu của đồ án:

- Hoàn thành các tính năng được yêu cầu đề ra và giúp các thành viên trong nhóm nắm rõ hơn về cách vận hành của giao thức tầng vận chuyển.

##### Các thông tin cơ bản của đồ án:

- Thiết lập giúp cho máy khách (client) có thể tải các file mà máy chủ (server) cung cấp.
- Cài đặt các tính năng cho máy khách như thoát chương trình bằng tổ hợp phím Ctrl + C, hiển thị tiến trình tải file của máy khách bằng cách in ra phần trăm dung lượng đã tải, cập nhật thường xuyên các yêu mới của máy khách.

##### Kết quả cần đạt được:

- Đảm bảo các tính năng được hoàn thiện.

- Các dữ liệu đảm bảo tính toàn vẹn, không thiếu sót trong việc truyền tải.
- Chương trình viết rõ ràng, dễ hiểu.

## II/ Đánh giá mức độ hoàn thành

- Dựa trên các yêu cầu được đề ra cho đề án socket, chúng em sẽ đánh giá mức độ hoàn thành chương trình dựa trên các yêu cầu cho trước.

STT	Yêu cầu	Hoàn thành
Phần I	Client có thể nhận được danh sách các file từ Server và ctrl-c	100%
Phần I	Client có thể nhận lần lượt từng file thành công từ Server. Server có thể gửi file thành công tới Client	100%
Phần I	Hiển thị percent download file và phát hiện những file cần download tiếp theo	100%
Phần II	Client có thể nhận được danh sách các file từ Server và ctrl-c	100%
Phần II	2s quét file input.txt 1 lần	100%
Phần II	Hiển thị percent download files	100%
Phần II	Client có thể nhận files thành công từ Server. Tập tin sau khi download phải đúng và đủ dung lượng	100%
Phần II	Độ ưu tiên CRITICAL, HIGH, NORMAL	100%
	Báo cáo	100%

- Về tổng quan, chúng em nhận thấy rằng chương trình của phần I và phần II đã đáp ứng được đầy đủ các yêu cầu đề ra của đề án. Các tính năng của chương trình như hiển thị phần trăm tiến trình tải file, ngắt kết nối chương trình bằng Ctrl + C, truyền tải file đúng và đủ dung lượng, quét yêu cầu trong file input mỗi 2s, tải file theo thứ tự ưu tiên. Tất cả tính năng trên chúng em thiết lập hoàn chỉnh.
- Ngoài ra, chúng em còn cải tiến thêm cho chương trình về mặt giao diện người dùng. Tuy nhiên, do lần đầu tiếp xúc với việc lập trình giao diện nên có thể nói là còn nhiều mặt hạn chế, cụ thể như giao diện chưa được linh hoạt về kích cỡ của cửa sổ console, chưa thiết lập được thanh kéo trang khi dữ liệu hiển thị quá nhiều.

### **III/ Cách vận hành chương trình**

***Vận hành phần I: Chương trình Client/Server cho phép nhiều Client download file từ 1 Server. Server sẽ phục vụ tuần tự từng Client.***

#### **Phía server:**

- Server chuẩn bị sẵn các địa chỉ IP và cổng để sẵn sàng khởi tạo socket.
- Sau khi khởi tạo socket và bind, server sẽ lắng nghe các kết nối của client.
- Khi có client kết nối tới server, server sẽ bắt đầu việc truyền tải file.
- Đầu tiên, server sẽ có một danh sách các file client có thể tải, lưu các thông tin mỗi file vào một mảng.
- Sau đó, duyệt qua các từng file trong mảng chứa thông tin mỗi file, server sẽ gửi các thông tin cho client lần lượt là tên và kích thước. Khi đã gửi đầy đủ thông tin các file, server sẽ gửi từ khóa "END\_LIST" nhằm báo hiệu cho client rằng đã gửi hết các file server có thể phục vụ.
- Khi đã gửi xong danh sách các file client có thể tải, server sẽ chờ để nhận các yêu cầu của client. Mỗi lần nhận yêu cầu, server sẽ nhận tên của 1 file mà client yêu cầu, và kiểm tra xem file đó server có thể phục vụ không, nếu không thì thông báo về cho

client không thể phục vụ file đó, nếu có thì gửi tín hiệu "START" cho client để bắt đầu việc tải file.

- Trong quá trình tải file, dung lượng file sẽ chia phần các phần nhỏ để gửi qua cho client,
- Đến khi đã tải file xong, server cũng gửi tín hiệu "END" để client biết rằng đã truyền tải hết dữ liệu file yêu cầu.
- Và sau đó server sẽ tiếp tục lắng nghe yêu cầu mới của client, đến khi client không còn yêu cầu nữa và đóng chương trình, server sẽ đóng kết nối với client và chờ để phục vụ các client mới kết nối đến.

### **Phía client:**

- Client sẽ có các thao tác trước khi kết nối đến server. Kiểm tra đã có folder chứa các file tải từ server chưa, nếu chưa thì khởi tạo folder đó. Tiếp đến là nhờ các thông tin có được như địa chỉ IP và cổng của server, client sẽ tạo socket và kết nối đến server.
- Khi đã kết nối đến server, client sẽ nhận lần lượt các tên và kích thước của từng file mà server có thể phục vụ. Sau khi nhận được tín hiệu là "END\_LIST", đồng nghĩa với việc client đã nhận đủ danh sách các file có thể tải từ server, client sẽ in ra màn hình các tên file đó kèm với kích thước cho người dùng.
- Bên phía client cũng có một file txt để ghi nhận các yêu cầu tải file của người dùng, khi người dùng nhập vào các file yêu cầu, client sẽ load các yêu cầu lên và lưu vào trong mảng các file yêu cầu để tiến hành tải các file đó.
- Với từng file trong mảng các file yêu cầu, client sẽ kiểm tra xem rằng file đó đã có trong folder tải về chưa, nếu có thì sẽ bỏ qua và tiếp tục tải các file tiếp theo, nếu chưa thì sẽ bắt đầu tải file đó.
- Bắt đầu việc tải file, client sẽ gửi tên file cần tải cho phía server. Nếu client nhận lại phản hồi từ phía server là "START", thì sẽ bắt đầu việc tải file đó giữa client và server.

- Cho đến khi client nhận được tín hiệu “END” trong các gói tin gửi, việc truyền tải dữ liệu của file đang tải sẽ kết thúc vì đã hoàn thành tải file. Sau đó là hiển thị thông báo đã tải hoàn tất file và tiếp tục tải file tiếp theo.
- Khi đã tải xong các file yêu cầu, chương trình bên phía client sẽ quét 1 lần cuối để xem có file nào cần tải hay không, nếu có thì tiếp tục tải, nếu không thì dừng quá trình tải file và ngắt kết nối với server.
- Nếu người dùng muốn dừng chương trình ngay lập tức thì sẽ nhấn tổ hợp Ctrl + C, chương trình phía client sẽ kết thúc và ngắt kết nối với server.

***Vận hành phần II: Chương trình Client/Server cho phép nhiều Client download file từ Server theo độ ưu tiên. Server phục vụ nhiều Client cùng lúc.***

**Phía server:**

- Server sẽ thiết lập trước các nội dung cần thiết cho quá trình phục vụ tải file từ các client như có trước địa chỉ IP và cổng kết nối, kiểm tra xem trong thư mục chứa các file phục vụ client có những file nào, cập nhật và ghi lên một file chứa các thông tin về các file cho client tải.
- Sau khi đã hoàn thành các công tác chuẩn bị, server sẽ load file chứa thông tin các file phục vụ client vào chương trình và thực hiện khởi tạo socket với địa chỉ IP và số cổng đã chuẩn bị từ trước.
- Với mỗi client kết nối thông qua socket đã khởi tạo từ server, server sẽ tạo một luồng (hay có thể nói là một kết nối dành riêng cho client đó) để có thể phục vụ song song nhiều client cùng một lúc.
- Xem xét quá trình phục vụ cho một client, sau khi kết nối với client, server sẽ gửi cho client số lượng và danh sách các file có thể tải từ phía server.
- Trong quá trình gửi dữ liệu giữa client và server, sẽ có các lần gửi và nhận tín hiệu “ACK” để tránh việc dữ liệu không được truyền như mong muốn (ví dụ như việc bị gửi dồn dữ liệu, 3 lần gửi mà chỉ cần 1 lần nhận).

- Sau khi đã nhận gửi danh sách các file có thể tải, server sẽ chờ đợi các yêu cầu từ phía client.
- Server sẽ nhận được số lượng các file cần tải từ phía client, sau đó lần lượt nhận các thông tin như tên và độ ưu tiên của file đó, lưu vào một mảng dữ liệu để sử dụng cho việc truyền tải file.
- Sau khi đã có được những yêu cầu của client, server sẽ thiết lập một số thứ để chuẩn bị cho việc tải các file client yêu cầu, cụ thể như là mảng kích thước hiện tại của file để cập nhật tiến trình tải file, mảng các con trỏ file để phục vụ cho việc tải song song nhiều file cùng một lúc, đọc một lúc nhiều file.
- Sau các công tác trước việc truyền tải, server sẽ bắt đầu truyền dữ liệu đến phía client. Server sẽ truyền dữ liệu bằng cách truyền từng khúc dữ liệu nhỏ của mỗi file yêu cầu một cách tuần tự. Và mỗi file sẽ có độ ưu tiên khác nhau, tùy vào độ ưu tiên đó mà trong lượt truyền dữ liệu có file đó, server sẽ truyền nhiều lần từng khúc dữ liệu nhỏ của file đó để đảm bảo tính ưu tiên của file đó.
- Trong quá trình truyền tải file, sau mỗi lần duyệt qua các file và đã truyền các khúc dữ liệu nhỏ tương ứng, server sẽ lắng nghe tín hiệu từ client để xem client có yêu cầu thêm các file mới không.
- Nếu có thì server sẽ cập nhật lại các file yêu cầu cần tải, chuẩn bị cho quá trình tải của các file yêu cầu mới, và tiếp tục tải file cho đến khi truyền tải xong dữ liệu của tất cả các file.
- Sau khi đã tải xong 1 đợt các file yêu cầu từ phía client, server sẽ tiếp tục lắng nghe các yêu cầu mới để phục vụ cho client. Nếu có các yêu cầu mới thì lặp lại các bước ở trên và tiếp tục quá trình truyền tải file.
- Cho đến khi client chủ động ngắt kết nối, thì server sẽ ngắt kết nối với client và kết thúc quá trình truyền tải giữa server và client đó.
- Server sẽ bật cho đến khi người dùng bên phía server tắt, để đảm bảo tính “always on host” của server và phục vụ cho client mọi lúc.

### **Phía client:**

- Đầu tiên, client sẽ thiết lập một số thứ trước khi kết nối đến server, như là địa chỉ IP của server và số cổng kết nối, kiểm tra thư mục chứa các file yêu cầu có chưa để khởi tạo.
- Sau đó, client sẽ thực hiện kết nối đến server thông qua địa chỉ IP và số cổng đã chuẩn bị từ trước. Khởi tạo một socket để phục vụ cho việc tiếp nhận các dữ liệu tải về từ phía server.
- Khi đã hoàn tất việc kết nối, client sẽ nhận được danh sách các file có thể tải đến từ phía server, và lúc này các thông tin nhận được sẽ lưu vào một mảng để tiện cho việc xử lý về sau, đồng thời cũng in ra màn hình phía client danh sách đó để người dùng có thể xem và nhập vào các file muốn tải.
- Sau khi đã nhận được danh sách các file có thể tải từ phía server, bên phía client sẽ có một file hỗ trợ là file “input.txt” để người dùng ghi vào đó các file muốn tải và độ ưu tiên cho file đó. Khi đã ghi xong thì lưu bằng tổ hợp phím Ctrl + S.
- Khi đã ghi xong các file muốn tải vào file “input.txt”, chương trình phía client sẽ quét xem các yêu cầu trong file “input.txt” là gì, lưu các yêu cầu đó vào chương trình để xử lý. Ngoài ra, cũng gửi các yêu cầu đó cho phía server để server có thể chuẩn bị cho việc cung cấp các file yêu cầu của phía client.
- Chương trình client đồng thời cũng thiết lập một “đồng hồ”, để sau một khoảng thời gian nhất định (theo yêu cầu đề bài là 2s) thì sẽ kiểm tra xem người dùng có yêu cầu thêm các file mới hay không, để cập nhật và tải thêm các file đó.
- Trước quá trình truyền tải file thì client cũng có một số thứ cần chuẩn bị để phục vụ cho việc tải file như kiểm tra các file nào đã tải rồi, các file không được phục vụ từ phía server, mở con trỏ file của các file yêu cầu để ghi dữ liệu vào, kích thước còn lại của file cần tải để hiển thị tiến trình tải file theo phần trăm ...

- Sau các công tác chuẩn bị đã đề cập ở trên, client bắt đầu quá trình nhận các dữ liệu từ phía server. Quá trình này sẽ tải cùng một lúc tất cả các file yêu cầu của người dùng, và đồng thời cũng cập nhật các yêu cầu mới.
- Tương tự bên server thì client cũng sẽ nhận lần lượt các dữ liệu nhỏ của mỗi file, tùy vào mức độ ưu tiên của file đó mà dữ liệu nhận sẽ tương ứng với độ ưu tiên đó.
- Trong quá trình tải các file thì phía client cũng hiển thị tiến trình tải file thông qua phần trăm được tính bằng lượng dữ liệu đã nhận trên tổng lượng dữ liệu của file đó.
- Khi đã hoàn thành xong 1 đợt yêu cầu tải file, thì người dùng có thể tiếp tục yêu cầu thêm các file khác bằng cách ghi vào file “input.txt” và server sẽ tiếp tục phục vụ việc tải các file theo yêu cầu mới, lặp lại các quá trình đã đề cập ở trên.
- Sau khi nhận đủ các file mong muốn, người dùng phía client có thể kết thúc chương trình bằng tổ hợp phím Ctrl + C.

#### **IV/ Môi trường lập trình**

- Ngôn ngữ sử dụng chính trong quá trình xây dựng chương trình là ngôn ngữ Python. Với các cú pháp đơn giản và dễ dàng thao tác, Python là ngôn ngữ mà nhóm em lựa chọn cho đề án lần này.
- Ngoài ra, chúng em còn sử dụng thêm raylib, một thư viện của ngôn ngữ C++ để tạo nên các giao diện đồ họa, giúp chương trình có thêm tính thân thiện hơn với người dùng.



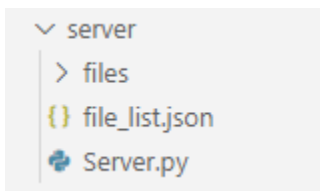
## V/ Hướng dẫn sử dụng các tính năng của chương trình

**Phần I: Chương trình Client/Server cho phép nhiều Client download file từ 1 Server. Server sẽ phục vụ tuần tự từng Client.**

**Hướng dẫn sử dụng trên cửa sổ console không có giao diện:**

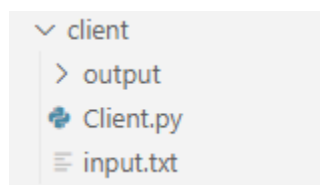
### + Phần chuẩn bị của 2 phía:

- **Server:** có được source code của server, chúng em đặt tên là Server.py và trong thư mục chứa server phải có thư mục chứa các file dữ liệu phục vụ việc tải cho server. Folder chứa các file dữ liệu có tên là “files”.
- Khi chạy server lên thì chương trình phía server sẽ quét xem các thông tin các file có trong folder “files” và ghi các thông tin đó vào một file json có tên là file\_list.json. File file\_list.json tự khởi tạo khi chưa có và nếu có thì sẽ cập nhật lại nội dung file đó với những file có trong folder “files”.



*Hình ảnh bên trái mô tả ví dụ về các tổ chức folder chứa file thực thi server.*

- **Client:** có được source code của client và kèm theo đó là 1 file có tên là “input.txt” để người dùng nhập vào các file mình muốn tải. Ngoài ra sẽ có một folder được tạo ra là folder “output” dùng để chứa các file đã tải về từ phía server, folder này có thể có sẵn hoặc tự khởi tạo nếu chưa có. Folder “output” sẽ nằm cùng thư mục với source code của client.



*Hình ảnh bên trái mô tả cách tổ chức folder chứa file thực thi phía client.*

+ **Phần thực thi chương trình và các tính năng:**

- **Phía server:** Nếu chạy nội bộ trên máy, thì có thể dùng địa chỉ loopback 127.0.0.1 để thực thi chương trình của server, ngoài ra, có thể truyền tải các file giữa các thiết bị cùng kết nối với cùng một đường mạng. Tìm địa chỉ IP của máy và cập nhật trong phần code, ngay biến toàn cục HOST. Số cổng PORT cũng có thể cập nhật phù hợp. Cung cấp địa chỉ IP HOST và số cổng PORT cho client để client có thể kết nối đến.
- Chạy source Server.py bằng câu lệnh “python Server.py” và phía server sẽ hiển thị các thông tin về các client kết nối.
- **Phía client:** Nhận được các địa chỉ IP và số cổng của server, cập nhật vào biến toàn cục HOST và PORT ở phía đầu source code Client.py, sau đó kiểm tra cách tổ chức thư mục để đảm bảo không xảy ra lỗi chương trình.
- Nhập các tên file mà client muốn tải, nếu phía client chưa có thông tin các file server có thì có thể thực thi chương trình client 1 lần với file “input.txt” trống để nhận thông tin các file server có. Chạy bằng lệnh “python Client.py”.
- Sau khi nhận được thông tin các file có thể cung cấp, người dùng muốn tải file nào thì nhập tên các file thành từng dòng trong file “input.txt”, thực hiện tổ hợp Ctrl + S để lưu lại, sau đó chương trình phía client sẽ phát hiện và tải các file người dùng mong muốn.
- Ví dụ về các ghi file “input.txt”:

File1.txt

File2.txt

File3.txt

- Trong quá trình tải file, do tải tuần tự các file nên người dùng có thể thêm các file muốn tải thêm vào cuối danh sách yêu cầu trong file “input.txt” để chờ đến lượt

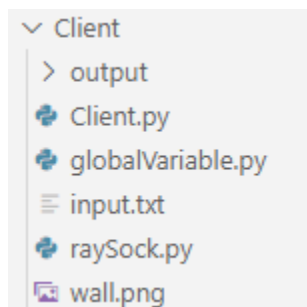
tải file đó. Nếu như đã tải hết các yêu cầu và không cập nhật lại file “input.txt” thì chương trình phía client sẽ đóng lại.

- Lưu ý rằng chương trình sẽ không tải lại các file đã tải trước đó, kể cả những file đã có sẵn bên phía client sẽ không được tải lại.
- Ngoài ra, nếu muốn kết thúc chương trình nhanh hơn người dùng có thể dùng tổ hợp Ctrl + C để ngắt kết nối với server và kết thúc chương trình, các file đang tải có thể bị thiếu sót dữ liệu.

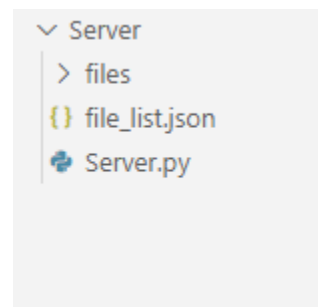
### Hướng dẫn sử dụng trên cửa sổ console có giao diện:

#### + Phần chuẩn bị của 2 phía:

- Hướng dẫn cài đặt thư viện raylib cho python:
  - Sử dụng tổ hợp phím “Windows + R” để mở hộp thoại Run.
  - Nhập “cmd” vào ô Open rồi nhấn OK.
  - Chạy lệnh “python -m pip install --update pip” hai lần để kiểm tra phiên bản của pip, cài đặt và kiểm tra lại lần nữa xem máy bạn đã cài đặt thành công chưa.
  - Chạy lệnh “python -m pip install setuptools”.
  - Và cuối cùng nhập “pip install raylib” để cài đặt thư viện raylib.
- Phần chuẩn bị của 2 phía tương tự như sử dụng không có giao diện, ngoài ra do các hàm và file phục vụ việc thiết lập giao diện người dùng nên ngoài source code của server chính ra còn có các file hỗ trợ.



Cách tổ chức thư mục Client



Cách tổ chức thư mục Server

+ **Phần thực thi chương trình và các tính năng:**

- Sau khi đã hoàn thành xong công tác chuẩn bị các file source, tiếp đến sẽ chạy chương trình client với câu lệnh “python Client.py”.
- Chương trình sau khi chạy lên sẽ hiện lên một cửa sổ giao diện đồ họa như trên.
- Góc trái trên màn hình hiển thị các file người dùng có thể tải, góc trái dưới hiển thị các thông báo liên quan đến việc tải file, hiển thị tiến trình tải file.
- Chương trình cho phép người dùng chọn các file để tải bằng cách bấm vào các nút phía bên trái với các số tương ứng được đánh dấu. Hoặc người dùng có thể đánh tên file mình muốn tải ở ô nhập và nhấn phím enter để tải.
- Chương trình được thiết lập hơi khác một chút so với nguyên bản là sẽ tạo mới lại file input.txt mới lần chạy chương trình client, sau đó sẽ đợi người dùng nhập hoặc ấn các file muốn tải. Khi đã tải xong chương trình sẽ đợi khoảng 2s để xem người dùng có muốn tải thêm file nào nữa không, nếu có thì sẽ tải tiếp và có thể thêm các file muốn tải trong lúc đợi tải các file khác.
- Không có bất kì yêu cầu thêm nào thì chương trình sẽ kết thúc.

***Phần II: Chương trình Client/Server cho phép nhiều Client download file từ Server theo độ ưu tiên. Server phục vụ nhiều Client cùng lúc.***

**Hướng dẫn sử dụng trên cửa sổ console không có giao diện:**

+ **Phần chuẩn bị của 2 phía: tương tự ở phần I**

+ **Phần thực thi chương trình và các tính năng:**

- **Phía server:** Nếu chạy nội bộ trên máy, thì có thể dùng địa chỉ loopback 127.0.0.1 để thực thi chương trình của server, ngoài ra, có thể truyền tải các file giữa các thiết bị cùng kết nối với cùng một đường mạng. Tìm địa chỉ IP của máy và cập nhật trong phần code, thay biến toàn cục HOST. Số cổng PORT cũng có thể cập nhật phù hợp. Cung cấp địa chỉ IP HOST và số cổng PORT cho client để client có thể kết nối đến.

- Chạy source Server.py bằng câu lệnh “python Server.py” và phía server sẽ hiển thị các thông tin về các client kết nối cũng như các yêu cầu từ phía các client.
- **Phía client:** Nhận được các địa chỉ IP và số cổng của server, cập nhật vào biến toàn cục HOST và PORT ở phía đầu source code Client.py, sau đó kiểm tra cách tổ chức thư mục để đảm bảo không xảy ra lỗi chương trình.
- Nhập các tên file mà client muốn tải, nếu phía client chưa có thông tin các file server có thì có thể thực thi chương trình client với file “input.txt” trống để nhận thông tin các file server có. Chạy bằng lệnh “python Client.py”.
- Sau khi nhận được thông tin các file có thể cung cấp, người dùng muốn tải file nào thì nhập tên các file thành từng dòng trong file input.txt kèm với thứ tự ưu tiên của file đó, thực hiện tổ hợp Ctrl + S để lưu lại, sau đó chương trình phía client sẽ phát hiện và tải các file người dùng mong muốn.
- Ví dụ về các ghi file “input.txt”:

File1.zip HIGH File2.zip NORMAL File3.zip CRITICAL
--

- Chương trình của phần 2 cho phép tải nhiều file cùng một lúc nên người dùng có thể nhập một lúc nhiều file để tải cùng lúc với độ ưu tiên phụ thuộc vào nhu cầu. Ngoài ra, có thể phục vụ nhiều client khác cùng một lúc.

### Hướng dẫn sử dụng trên cửa sổ console có giao diện:

- + **Phần chuẩn bị của 2 phía: tương tự ở phần I**
- + **Phần thực thi chương trình và các tính năng:**
  - Sau khi chạy câu lệnh “python Client.py”, cửa sổ giao diện sẽ hiện ra.
  - Tương tự như ở phần I, các thông tin sẽ hiển thị trên cửa sổ giao diện này.

- Ngoài ra, trong phần II có hỗ trợ việc tải file theo mức độ ưu tiên, nên ngoài nhập tên file còn nhập thêm cả độ ưu tiên của file đó trong quá trình tải các file. Nếu click vào các file muốn tải thì sẽ click thêm độ ưu tiên của file đó.
- Các thông tin như tiến độ tải file của sẽ hiển thị trong ô thông báo như phần I để người dùng nắm được tiến trình các file đang tải.
- Kết thúc chương trình thì bấm nút thoát góc phải trên cùng (dấu X).

## VI/ Bảng phân công công việc cho các thành viên trong nhóm

BẢNG PHÂN CÔNG CÔNG VIỆC	
Mạch Quốc Tấn	<ul style="list-style-type: none"> <li>• Phân công công việc cho các thành viên và đảm bảo tiến độ thực hiện.</li> <li>• Nghiên cứu, đề xuất ý tưởng thực thi chương trình.</li> <li>• Tự thử nghiệm và xây dựng chương trình theo yêu cầu của phần I, phần II.</li> <li>• Xây dựng chính các tính năng của phần II, đưa ra kịch bản thực thi chính của chương trình phần II.</li> <li>• Hoàn thiện báo cáo về đồ án.</li> </ul>
Lê Nguyễn Nhật Khánh	<ul style="list-style-type: none"> <li>• Nghiên cứu, đề xuất ý tưởng thực thi chương trình.</li> <li>• Tự thử nghiệm và xây dựng chương trình theo yêu cầu của phần I, phần II.</li> <li>• Xây dựng giao diện cho phía máy khách (client) của phần I và phần II.</li> </ul>
Nguyễn Thành Dâng	<ul style="list-style-type: none"> <li>• Nghiên cứu, đề xuất ý tưởng thực thi chương trình.</li> <li>• Tự thử nghiệm và xây dựng chương trình theo yêu cầu của phần I, phần II.</li> <li>• Xây dựng chính các tính năng của phần I, đưa ra kịch bản thực thi chính của chương trình phần I.</li> <li>• Hỗ trợ các thành viên sửa lỗi chương trình thực thi.</li> </ul>

## VIII/ Các nguồn tham khảo

- AI Chat GPT - tham khảo nhanh về các vấn đề liên quan đến lỗi cú pháp trong python.
- YouTube - tham khảo về các lập trình socket cơ bản và giao diện đồ họa.
- Stack Overflow - tham khảo về các vận hành các hàm trong python.
- GeeksforGeeks - tham khảo về lập trình python cơ bản.
- Các hàm và cách cài đặt của thư viện “Raylib” trong python:
  - + [https://electronstudio.github.io/raylib-python-cffi/pyray.html#pyray.load\\_texture](https://electronstudio.github.io/raylib-python-cffi/pyray.html#pyray.load_texture)
  - + <https://pypi.org/project/raylib/>