



FIT@HCMUS

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG - HCM
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN SOCKET

BỘ MÔN: MẠNG MÁY TÍNH

21120396 - Đào Thị Ngọc Giàu

21120419 - Vũ Thành Công (nhóm trưởng)

21120446 - Kiên Định Mỹ Hạnh

Mục lục

<i>I. Thông tin nhóm-----</i>	2
<i>II. Đánh giá mức độ hoàn thành-----</i>	2
<i>III. Kịch bản giao tiếp của chương trình-----</i>	2
1. Giao thức kết nối giữa client và server -----	2
2. Cấu trúc thông điệp -----	2
3. Kiểu dữ liệu thông điệp-----	4
<i>IV. Môi trường lập trình và các framework hỗ trợ để thực thi ứng dụng.-----</i>	4
1. Môi trường lập trình -----	4
2. Các hàm chính-----	4
<i>V. Hướng dẫn sử dụng các tính năng chương trình-----</i>	6
<i>VI. Bảng phân công công việc-----</i>	9
<i>VII. Các nguồn tài liệu tham khảo-----</i>	9

I. Thông tin nhóm

MSSV	Họ và tên	Ghi chú
21120396	Đào Thị Ngọc Giàu	
21120419	Vũ Thành Công	Nhóm trưởng
21120446	Kiên Định Mỹ Hạnh	

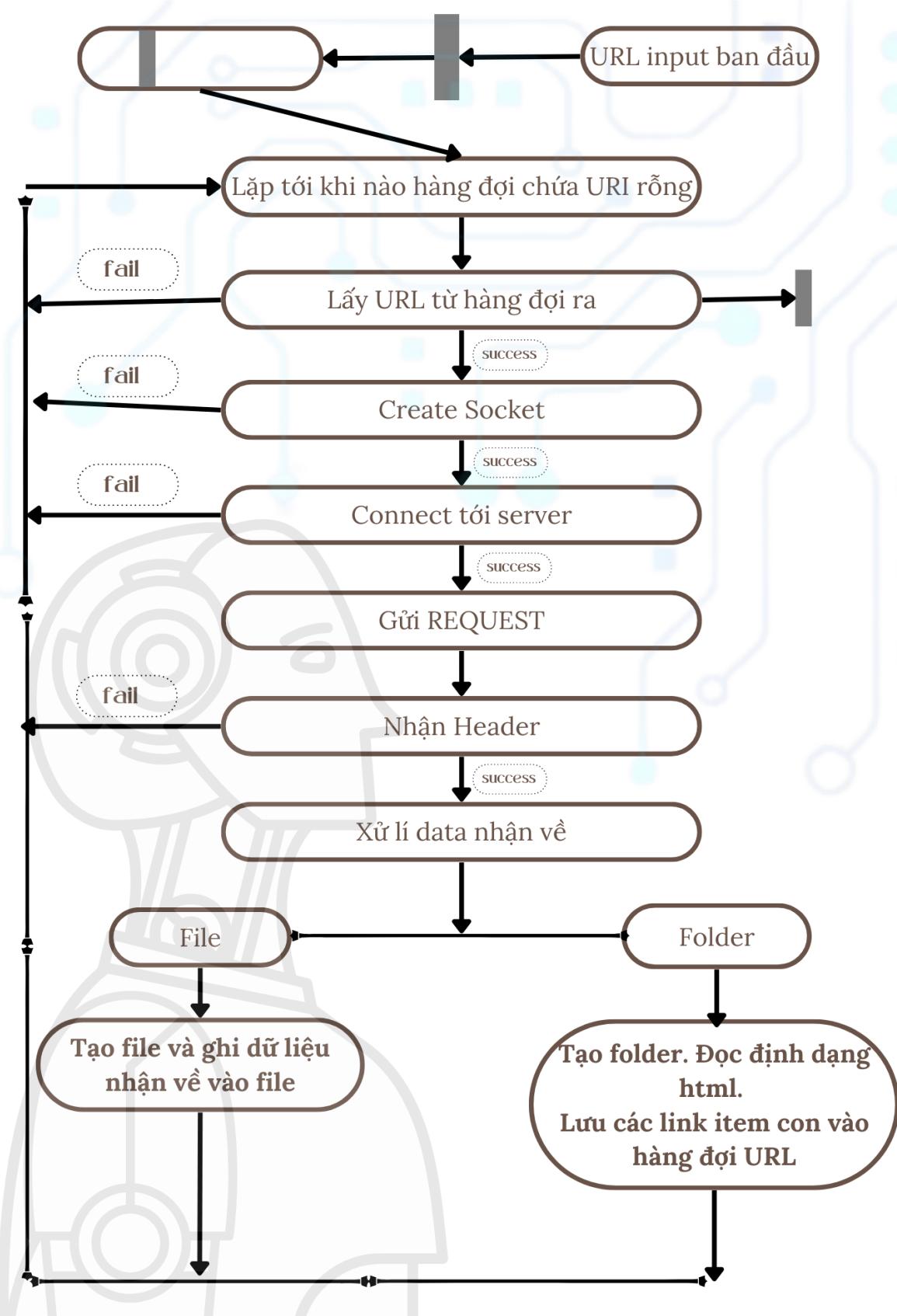
II. Đánh giá mức độ hoàn thành

STT	Chức năng	Mức độ hoàn thành
1	Cho phép client kết nối đến server thông qua kết nối TCP	100%
2	Khi client hoặc server mất kết nối đột ngột, không làm chương trình treo hay xảy ra lỗi	100%
3	Download file thành công	100%
4	Lưu tất cả các file trong folder	100%
5	Gửi nhiều request trong một connect khi download file trong folder	100%
6	Concurrent, handle nhiều kết nối cùng lúc đến các web server	100%

III. Kịch bản giao tiếp của chương trình

1. *Giao thức kết nối giữa client và server:* TCP Socket

2. *Cấu trúc thông điệp*



3. Kiểu dữ liệu thông điệp

- Message data là text

IV. Môi trường lập trình và các framework hỗ trợ để thực thi ứng dụng.

1. Môi trường lập trình: C/C++

2. Các hàm chính:

a. `void get_FileAndFolder(QueueURI&uris, int port, int versionHTTP)`

Hàm thực hiện chức năng chính của chương trình là khởi tạo, lấy dữ liệu (file, folder) từ server, chạy trong vòng lặp khi hàng đợi cho tới khi nào không còn URI nào nữa. Hàm truyền vào 3 tham số là **uris** (tham chiếu, hàng đợi chứa các uri cần get); **port** (số hiệu cổng port sử dụng); **versionHTTP** (version HTTP để GET, 0: HTTP 1.0 và 1: HTTP 1.1).

b. `int Create(int domain = AF_INET, int type = SOCK_STREAM, int protocol = 0)(MyHTTP.h)`

Hàm thực hiện việc khởi tạo socket và các bước khởi tạo dữ liệu cần thiết để kết nối đến server. Hàm trả về socketID nếu thành công và -1 nếu thất bại. Hàm truyền vào 3 tham số là domain (loại định dạng cho URI), type (Loại Stream nào sẽ sử dụng để truyền và nhận dữ liệu), protocol (Loại giao thức nào được sử dụng để truyền và nhận dữ liệu).

c. `int Connect()(MyHTTP.h)`

Hàm kết nối đến server với các giá trị đã khởi tạo sẵn cho đối tượng MyHTTP. Nếu thành công thì trả về 1, thất bại trả về 0.

d. `void MyHTTP::makeRequest(char request[1024], const int&type)(MyHTTP.h)`

Hàm thực hiện việc format chuỗi request tới server dựa vào tham số type là version của HTTP đang sử dụng. Chuỗi request này dùng để gửi yêu cầu tới server.

Ví dụ với một URI:

<http://students.iitk.ac.in/programmingclub/course/lectures/>

request http 1.1:

“**GET/programmingclub/course/lectures/HTTP/1.1\r\nHost:students.iitk.ac.in \r\n”**”
“**Connection: close\r\n”**”
“**\r\n”**”

e. **void Send(char request[1024]); (MyHTTP.h)**

Hàm thực hiện việc gửi chuỗi request tới server.

f. **int Receive(char* pData, int nBytes); (MyHTTP.h)**

Hàm thực hiện việc nhận dữ liệu từ server gửi về. Hàm trả về số byte nhận được. Tham số pData dùng để lưu dữ liệu nhận về và nBytes với ý nghĩa là số bytes muốn nhận về.

g. **int GetHeader(char* header, int pSize); (MyHTTP.h)**

Hàm thực hiện việc lấy header từ server. Hàm trả về số byte của header. Tham số header dùng để lưu dữ liệu của header từ server; pSize là số bytes muốn lấy về

h. **bool isValidResponse(char* data, const int& versionHTTP); (MyHTTP.h)**

Hàm kiểm tra xem gói header có thông báo thành công hay không (200 OK). Nếu thành công thì trả về true, thất bại trả về false. Hàm truyền vào data (dữ liệu header); versionHTTP là loại version của HTTP.

i. **bool hasChunkedEncoding(char* data); (MyHTTP.h)**

Hàm kiểm tra xem gói header có trường nào thông báo gói dữ liệu này sẽ bị mã hóa Chunk hay không? Nếu có trả về true, nếu không trả về false. Tham số data là dữ liệu của header.

j. **void decodeChunk(char* data, char* src, int totalSize); (MyHTTP.h)**

Hàm thực hiện việc giải mã Chunk. Tham số data (chưa dữ liệu được giải mã); src (chứa dữ liệu bị mã hóa chunk); totalSize (chứa kích thước theo bytes của chuỗi bị mã hóa).

k. void writeData(*QueueURI& uris, const int& versionHTTP, const bool& hasChunked*)(*MyHTTP.h*)

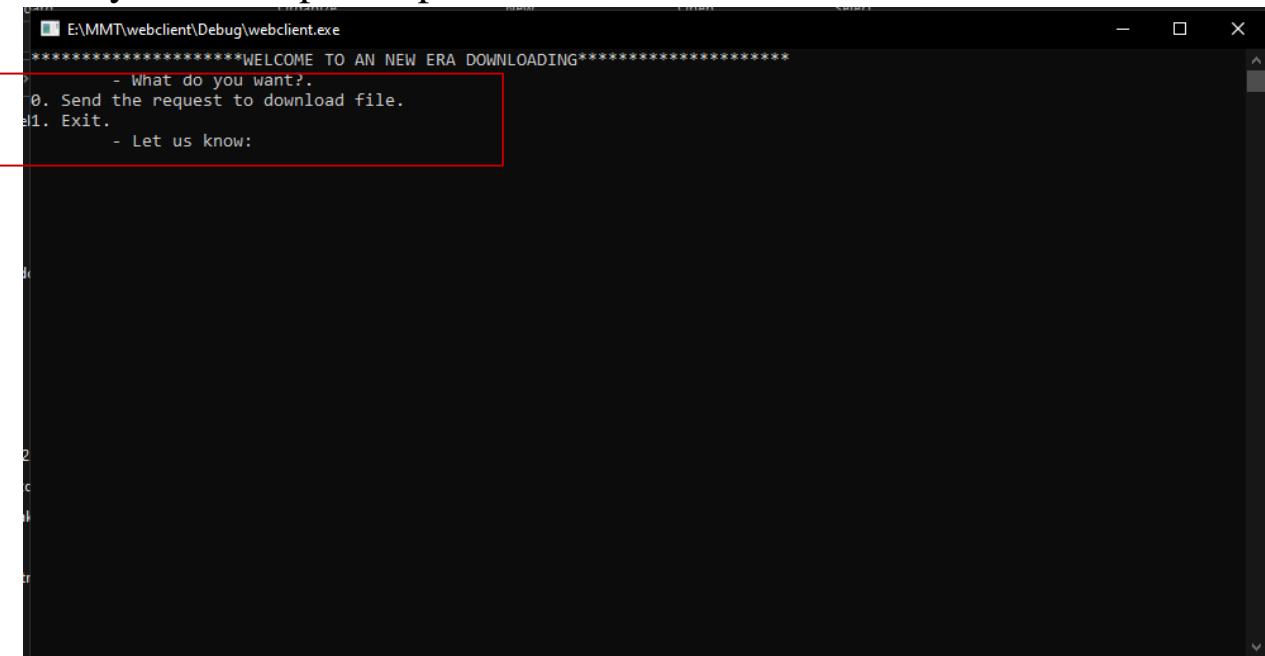
Hàm thực hiện việc nhận dữ liệu và lưu vào máy tính. Nếu dữ liệu nhận về là file thì tạo và lưu file vào máy. Ngược lại nếu là file .html (cấu trúc các subitem), thì thực hiện việc tạo folder và nạp các link vào hàng đợi uris để tiếp tục kết nối và tải về. Tham số 5 versionHTTP (version của HTTP đang dùng); hasChunked (cờ kiểm tra gói dữ liệu nhận về có bị Chunked Encoding hay không?)

V. Hướng dẫn sử dụng các tính năng chương trình

Chương trình được chạy bằng command line (Window)

Khi chạy, chương trình sẽ hiện ra một bảng menu cho bạn lựa chọn.

Hãy lựa chọn phù hợp với mục đích của bạn



Nếu lựa chọn 1, bạn sẽ thoát khỏi chương trình

```
E:\MMT\webclient\Debug\webclient.exe
*****WELCOME TO AN NEW ERA DOWNLOADING*****
- What do you want?.
0. Send the request to download file.
1. Exit.
- Let us know: 1

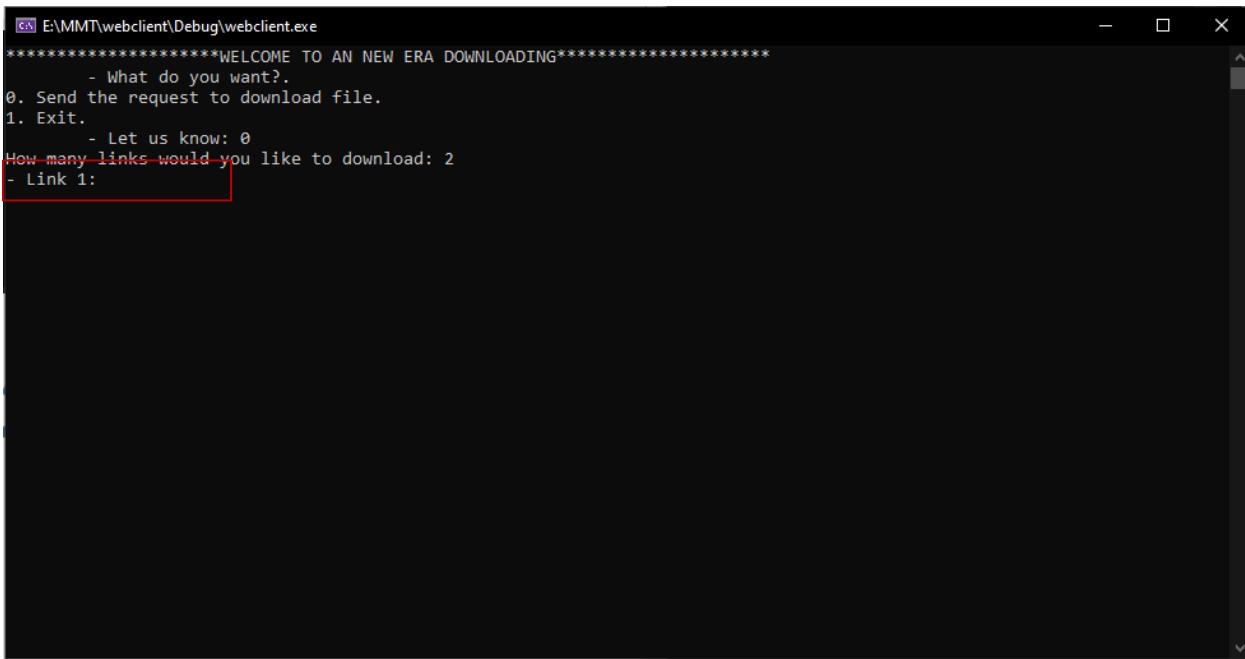
Thank for using our app.
Sincerely.

Press any key to continue . . .
```

Nếu lựa chọn 0, bạn sẽ tiếp tục lựa chọn xem tải 2 link) một lúc, hay tải 1 file (hiện tại chương trình mới đang hỗ trợ tải từ 1 – 2 link.

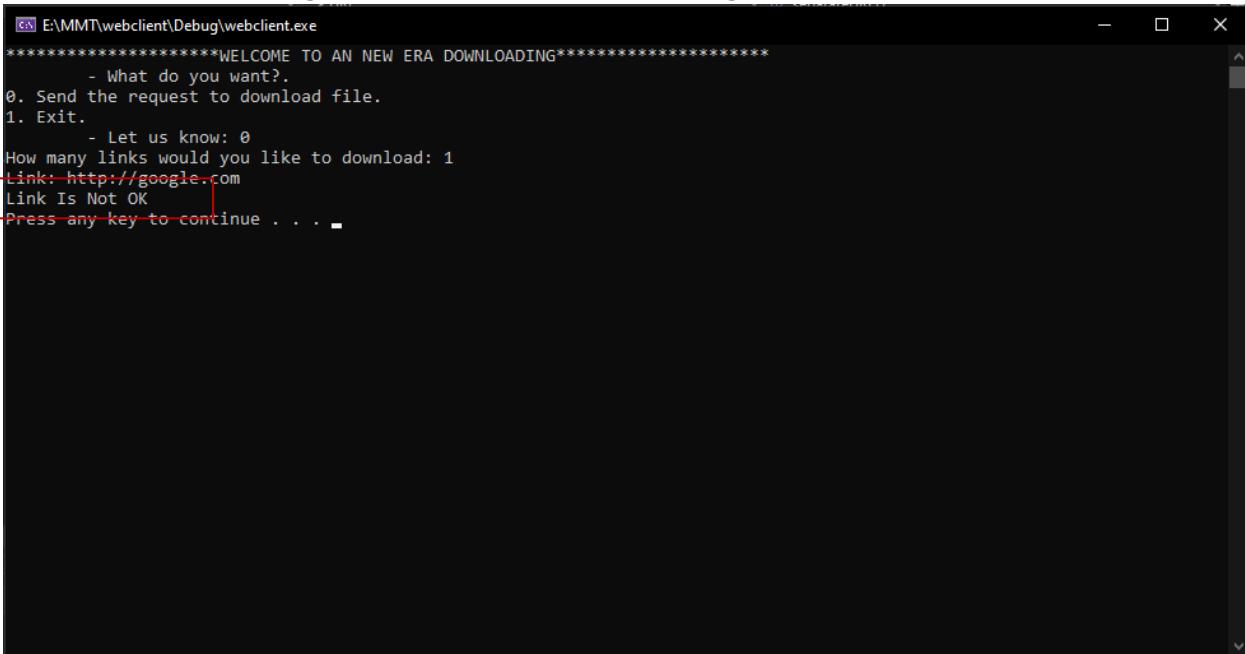
```
E:\MMT\webclient\Debug\webclient.exe
*****WELCOME TO AN NEW ERA DOWNLOADING*****
- What do you want?.
0. Send the request to download file.
1. Exit.
- Let us know: 0
How many links would you like to download: -
```

Cuối cùng hãy nhập link cần tải và chờ chương trình tải xong.



```
GS E:\MMT\wechat\Debug\wechat.exe
*****WELCOME TO AN NEW ERA DOWNLOADING*****
- What do you want?.
0. Send the request to download file.
1. Exit.
- Let us know: 0
How many links would you like to download: 2
- Link 1:
```

Tuy nhiên, do TCP socket chưa hỗ trợ cho toàn bộ các link, nên nếu link bạn cần tải không thuộc diện được hỗ trợ hoặc bạn nhập sai đường link, sẽ có thông báo: “Link is not OK” gửi cho bạn.



```
GS E:\MMT\wechat\Debug\wechat.exe
*****WELCOME TO AN NEW ERA DOWNLOADING*****
- What do you want?.
0. Send the request to download file.
1. Exit.
- Let us know: 0
How many links would you like to download: 1
Link: http://google.com
Link Is Not OK
Press any key to continue . . .
```

Ngoài ra, sẽ có một vài thông báo gửi đến bạn như “Init failed” (dành cho việc không thể khởi tạo MyHTTP), “connect failed” (dành cho việc không thể kết nối được tới server)

VI. *Bảng phân công công việc*

MSSV	Họ và tên	Công việc
21120396	Đào Thị Ngọc Giàu	Tìm hiểu về giao thức HTTP, các gói header của HTTP. Viết các class NodeQueue, QueueURI, URI Làm báo cáo.
21120419	Vũ Thành Công	Tìm hiểu về giao thức HTTP, các gói header của HTTP. Viết các class MyHTTP với công việc cài đặt giao tiếp giữa server và client bằng api socket, thực hiện lưu file, lưu folder. Viết hàm main.
21120446	Kiên Đinh Mỹ Hạnh	Tìm hiểu về giao thức HTTP, các gói header của HTTP. Tìm hiểu về chunked encoding và viết code bắt và giải mã chunked encoding. Làm báo cáo.

VII. *Các nguồn tài liệu tham khảo*

- (Wikipedia, Last edited on 25 July 2022)
https://en.wikipedia.org/wiki/Chunked_transfer_encoding
- (bunny.net)
<https://bunny.net/academy/http/what-is-chunked-encoding/>
- (BogoToBogo, 2020)
https://www.bogotobogo.com/cplusplus/sockets_server_client.php
- (Moon, August 10, 2020)

<https://www.binarytides.com/code-a-simple-socket-client-class-in-c/>

5. (Geeksforgeeks, Last Updated : 19 Sep, 2022)

<https://www.geeksforgeeks.org/socket-programming-cc/>