**Mạng máy tính**

**Đồ án thực hành 1**

C:\Users\tdqua_000\Dropbox\SS-Slides\DeCuong-CDIO\Template CDIO v4.2\Templates\Hinh anh\LogoTruong.png

Bộ môn Mạng máy tính

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

# Mục lục

[1 Mục lục 2](#_Toc500506274)

[2 Thông tin nhóm 3](#_Toc500506275)

[3 Phân công công việc 4](#_Toc500506276)

[4 Các hàm chức năng chính và thuật toán. 5](#_Toc500506277)

[4.1 Các hàm chức năng chính 5](#_Toc500506278)

[4.2 Mô tả thuật toán: 6](#_Toc500506279)

[5 Các chức năng và mức độ hoàn thành 6](#_Toc500506280)

[6 Cách chạy 7](#_Toc500506281)

[7 Tài liệu tham khảo 8](#_Toc500506282)

# Thông tin nhóm

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ Tên** | **Email** | **Điện thoại** |
| 1512112 | Tôn Thất Tâm Định | [1512112@student.hcmus.edu.vn](mailto:1512112@student.hcmus.edu.vn) |  |
| 1512628 | Nguyễn Anh Tuân | 1512628@student.hcmus.edu.vn |  |
| 1512075 | Võ Khắc Duy | [1512075@student.hcmus.edu.vn](mailto:1512075@student.hcmus.edu.vn) |  |

# Phân công công việc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | STT | Công việc | Người thực hiện | Ghi chú |
| Download file | 1 | Tải file từ đường dẫn dùng giao thức HTTP/1.0. | Định |  |
| 2 | Tải file từ đường dẫn dùng giao thức HTTP/1.1 | Định & Duy |  |
| Download folder | 1 | Tải file từ đường dẫn dùng giao thức HTTP/1.0. | Định |  |
| 2 | Tải file từ đường dẫn dùng giao thức HTTP/1.1, xử lý chunked trong http/1.1. | Định & Duy |  |
| Download folder và các subfolder | 1 | Tải file và các subfolder từ đường dẫn dùng giao thức HTTP/1.0. | Định |  |
| 2 | Tải file và các subfolder từ đường dẫn dùng giao thức HTTP/1.1. | Định |  |
| Bắt gói tin WireShark | 1 | Bắt gói tin và ghi kết quả | Tuân |  |

# Các hàm chức năng chính và thuật toán.

## Các hàm chức năng chính

* Để chuẩn bị cho việc download file nhóm chia chương trình thành các hàm con để xử lý các chức năng riêng biệt.
* Đầu tiên nhóm thiết kế hai hàm phục vụ hai chức năng riêng biệt là download file và download foloder. Hàm int downloadfile(char \*url, string path); nhận tham số đầu vào là url và địa chỉ đường dẫn để lưu file vào sau đó sẽ thực hiện chức năng download và ghi vào file theo như đường dẫn. Nếu thực hiện thành công thì hàm trả về kết quả là 0, ngược lại trả về -1 để báo lỗi. Tương tự, hàm int downloadfolder(char \*url, string path); thực hiện chức năng download folder từ url và lưu vào foldername.
* Hàm void mParseUrl(char \*mUrl, string &serverName, string &filepath, string &filename) dùng để phân chia url nhận được thành các chuỗi serverName – tên của host server, filepath - địa chỉ đường dẫn của url và chuỗi filename chứa tên của file cần download.
* Hàm SOCKET connectToServer(char \*szServerName, WORD portNum); dùng để tạo Socket từ phía Client để giao tiếp với server, 2 tham số đầu vào là tên server và số hiệu của port tương ứng.
* Hàm char \*readUrl2(char \*szUrl, long &bytesReturnedOut, char \*\*headerOut); thực hiện việc download và tiến hành lưu kích thước của file tải được vào byteReturnedOut, lưu dữ liệu phần header của file vào headerOut, và trả về kết quả là phần nội dung của file download được. Hàm này sẽ tiến hành tạo socket lắng nghe ở port 80, dùng giao thức TCP để download. Sau đó gửi request GET cho server, và nhận kết quả trả về lưu vào tmpBuffer. Cuối cùng hàm sẽ dựa vào dấu hiệu phân tách để chia tmpBuffer nhận về thành 2 thành phần headerbuffer và memBuffer và trả về kết quả.

## Mô tả thuật toán:

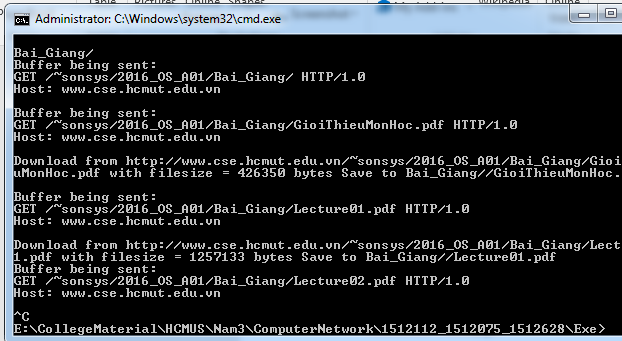
1. Đầu tiên ta sẽ tiền xử lí chuỗi url để có được giá trị của đường dẫn cần lưu.
2. Nếu url kết thúc là ký tự ‘/’ thì đây là dấu hiệu downloadfolder và ta sẽ tiến hành download url lưu về folder theo đường dẫn để tìm được ở trên, ngược lại ta sẽ tiến hành download file.
3. Trong hàm download folder đầu tiên ta sẽ download toàn bộ source của trang web về và tiến hành tìm kiếm các link con(sublink) trong trang web. Nếu link con đó là 1 sub folder thì ta sẽ tiến hành download subfolder, nếu không thì ta sẽ tiến hành bắt link và download file của sublink mới tìm được.

# Các chức năng và mức độ hoàn thành

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | STT | Công việc | Mức độ hoàn thành | Ghi chú |
| Download file | 1 | Tải file từ đường dẫn dùng giao thức HTTP/1.0. | 100% |  |
| 2 | Tải file từ đường dẫn dùng giao thức HTTP/1.1 | 90% |  |
| Download folder | 1 | Tải file từ đường dẫn dùng giao thức HTTP/1.0. | 100% |  |
| 2 | Tải file từ đường dẫn dùng giao thức HTTP/1.1 | 80% |  |
| Download folder và các subfolder | 1 | Tải file từ đường dẫn dùng giao thức HTTP/1.0. | 100% |  |
| 2 | Tải file từ đường dẫn dùng giao thức HTTP/1.1. | 80% |  |

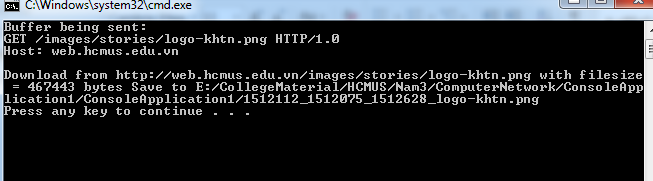
# Cách chạy

* Vào Start 🡪 Run 🡪cmd để tiến hành mở cmd lên.
* Tiến hành gõ địa chỉ của thư mục chứa file exe ta được như hình vẽ.

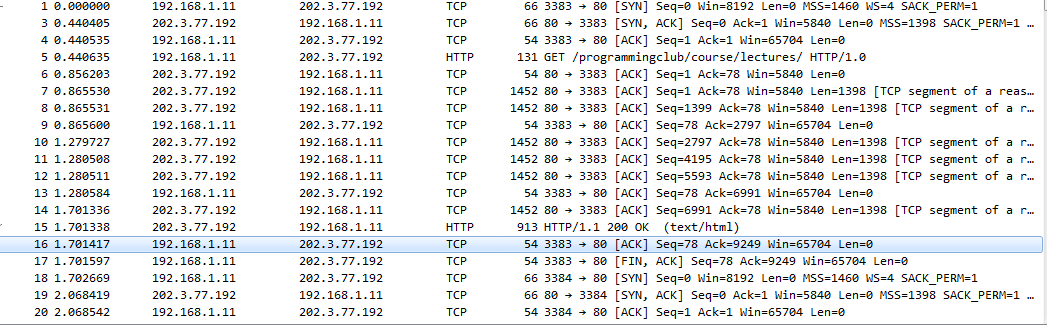


* Gọi chương trình thực thi theo cú pháp <tên chương trình> <địa chỉ url cần download> <giao thức HTTP sử dụng>. Ví dụ: 

Chương trình sẽ chạy tự động bắt link và download dữ liệu về. Màn hình sẽ hiện thị http request được gửi đi cho server, thông báo download dữ liệu và địa chỉ đường dẫn của file được lưu trong máy sau khi đã download xong.



# Mô tả bắt gói tin:



|  |  |
| --- | --- |
| Client | Server |
| 1. Bắt tay 3 bước: | |
| Đầu tiên, client tạo một socket để giao tiếp với server tại port 80, dùng protocol TCP, đồng thời gửi gói tin SYN đến Server |  |
|  | Server gửi trả gói tin SYNACK sau khi nhận được gói SYN từ client. |
| Client nhận được SYNACK từ server và gửi trả SYNACK để hoàn tất thủ tục bắt tay 3 bước(three-way handshake). |  |
| 1. Gửi và nhận thông điệp GET: | |
| Client dùng protocol HTTP gửi yêu cầu GET cho server. |  |
|  | Server gửi trả ACK cho client sau khi nhận được yêu cầu GET. |
| 1. Server gửi dữ liệu cho client: | |
|  | Server chia nhỏ dữ liệu và gửi từng đoạn cho client. |
| Trong quá trình nhận file, client gửi trả ACK thông báo để cho server biết về việc file đã được nhận hay chưa |  |
|  | Server gửi gói tin HTTP/x 200 OK để báo hiệu kết thúc gửi file http. |
| 1. Hoàn tất việc tải 1 file: | |
| Sau khi đã tải xong file theo yêu cầu thì Client gửi gói tin FINACK để yêu cầu đóng kết nối cho server. Client sẽ lặp lại thủ tục bắt tay với server để tải file tiếp theo nếu vẫn chưa tải đến file cuối cùng, ngược lại nó không thực hiện bắt tay |  |
|  | Server sẽ làm thủ tục bắt tay với client để chuẩn bị cho lượt truyền dữ liệu tiếp theo nếu file vừa truyền không phải là file cuối cùng trong danh sách truyền, ngược lại nó sẽ gửi trả ACK và đóng kết nối. |

# Tài liệu tham khảo

* Stackoverflow.com
* Tài liệu hướng dẫn thực hành
* <https://msdn.microsoft.com/en-us/library/windows/desktop/ms738545(v=vs.85).aspx>