BÁO CÁO MẠNG MÁY TÍNH

ĐỒ ÁN CUỐI KỲ

Thành viên:

* 1512406 – Nguyễn Vĩnh Phúc
* 1512474 – Vòng Chí Tài

Nội dung

[1. Phân công 3](#_Toc499561654)

[2. Các hàm chức năng chính 4](#_Toc499561655)

[vector<string> split(string data, string token); 4](#_Toc499561656)

[Link::Link(string link); 4](#_Toc499561657)

[ResponseHeaderAnalyzer::ResponseHeaderAnalyzer(string responseHeader) 4](#_Toc499561658)

[string constructRequest(Link link); 4](#_Toc499561659)

[struct sockaddr\_in getAddr(Link link); 4](#_Toc499561660)

[void establishConnection(CSocket & cs, Link link); 4](#_Toc499561661)

[void startDownload(CSocket &cs, Link link); 4](#_Toc499561662)

[void downloadFile(CSocket &cs, Link link, string subFolder); 5](#_Toc499561663)

[void updateProgress(int &recvDat, int &recvLen, int downloadSize); 5](#_Toc499561664)

[void transferDataNormal(CSocket &cs, ofstream &os, char\* buffer, int start, int recvLen, int contentLength); 5](#_Toc499561665)

[void transferDataChunked(CSocket &cs, ofstream &os, char\* buffer, int start, int recvLen); 5](#_Toc499561666)

[vector<string> getCurrentFileList(string targetLink); 5](#_Toc499561667)

[3. Các tính năng đã làm/ chưa làm 7](#_Toc499561668)

[4. Cách chạy chương trình 8](#_Toc499561669)

[5. Quá trình gửi nhận TCP segment data – ACK 10](#_Toc499561670)

# Phân công

|  |  |  |
| --- | --- | --- |
| MSSV | Họ và tên | Công việc |
| 1512406 | Nguyễn Vĩnh Phúc | Viết các hàm hỗ trợ phân tích đường dẫn, header download tập tin kiểu bình thường |
| 1512474 | Vòng Chí Tài | Nghiên cứu xử lý đọc dữ liệu kiểu chunked và download thư mục và ghép nối các thành phần. |

# Các hàm chức năng chính

## vector<string> split(string data, string token);

Tham số cần truyền vào: chuỗi cần phân tách data, chuỗi dùng để phân tách token

Chức năng: tách chuỗi data thành cách chuỗi con, phân tách bằng chuỗi token

Kết quả trả về: mảng lưu các chuỗi con

## Link::Link(string link);

Tham số cần truyền vào: đường dẫn tập tin hoặc thư mục

Chức năng: phân tích các thông tin có thể sử dụng từ đường dẫn

Kết quả trả về: đối tượng Link

## ResponseHeaderAnalyzer::ResponseHeaderAnalyzer(string responseHeader)

Tham số cần truyền vào: header của phản hồi từ máy chủ

Chức năng: phân tích các thông tin có trong header

Kết quả trả về: đối tượng ResponseHeaderAnalyzer

## string constructRequest(Link link);

Tham số cần truyền vào: đường dẫn đã được phân tích Link

Chức năng: xây dựng request cần thiết kế tải tập tin/ thư mục

Kết quả trả về: chuỗi request đã được xây dựng

## struct sockaddr\_in getAddr(Link link);

Tham số cần truyền vào: đường dẫn đã được phân tích Link

Chức năng: chuyển đổi đường dẫn Link thành đối tượng sockaddr\_in để sử dụng cho các hàm khác

Kết quả trả về: đối tượng sockaddr\_in được xây dựng

## void establishConnection(CSocket & cs, Link link);

Tham số cần truyền vào: đối tượng Csocket đã được tạo, đường dẫn đã được phân tích Link

Chức năng: thiết lập kết nối từ máy tính đến địa chỉ bằng Csocket với thông tin trong đường dẫn Link

Kết quả trả về: không có

## void startDownload(CSocket &cs, Link link);

Tham số cần truyền vào: đối tượng Csocket đã được tạo, đường dẫn đã được phân tích Link

Chức năng: tùy thuộc loại đường dẫn (thư mục/tập tin), chương trình sẽ tùy chọn phương thức xử lý

Kết quả trả về: không có

## void downloadFile(CSocket &cs, Link link, string subFolder);

Tham số cần truyền vào: đối tượng Csocket đã được tạo, đường dẫn đã được phân tích Link, đường dẫn thư mục con(nếu có)

Chức năng: Gửi request và phân tích phản hồi để lựa chọn cách xử lý dữ liệu

Kết quả trả về: không có

## void updateProgress(int &recvDat, int &recvLen, int downloadSize);

Tham số cần truyền vào: dung lượng dữ liệu đã nhận recvDat, dung lượng dữ liệu vừa nhận recvLen, dung lượng tập tin dự kiến downloadSize

Chức năng: In tiến độ ra màn hình để người sử dụng theo dõi

Kết quả trả về: không có

## void transferDataNormal(CSocket &cs, ofstream &os, char\* buffer, int start, int recvLen, int contentLength);

Tham số cần truyền vào: đối tượng Csocket đã được tạo, luồng xuất dữ liệu ra file, bộ nhớ đệm buffer, địa chỉ bắt đầu của dữ liệu start, số dữ liệu đã nhận recvLen, dung lượng dự kiến contentLength

Chức năng: Nhận dữ liệu theo phương thức trực tiếp cho đến khi đủ dung lượng yêu cầu, thông báo liên tục cho người dùng biết tiến độ

Kết quả trả về: không có

## void transferDataChunked(CSocket &cs, ofstream &os, char\* buffer, int start, int recvLen);

Tham số cần truyền vào: đối tượng Csocket đã được tạo, luồng xuất dữ liệu ra file, bộ nhớ đệm buffer, địa chỉ bắt đầu của dữ liệu start, số dữ liệu đã nhận recvLen

Chức năng: Nhận dữ liệu theo phương thức chunked cho đến khi dung lượng chunk bằng 0 hoặc máy chủ đóng kết nối, thông báo liên tục cho người dùng biết tiến độ

Kết quả trả về: không có

## vector<string> getCurrentFileList(string targetLink);

Tham số cần truyền vào: đường dẫn tập tin thư mục chứa đường dẫn các tập tin bên trong thư mục

Chức năng: lọc tên các tập tin và thư mục con từ trong tập tin dữ liệu thư mục

Kết quả trả về: danh sách tên tập tin

# Các tính năng đã làm/ chưa làm

Các tính năng đã làm

* Download tập tin bằng giao thức HTTP 1.0
* Download tập tin bằng giao thức HTTP 1.1 (bằng phương thức thường và chunked)
* Download tập tin trong thư mục bằng giao thức HTTP 1.0
* Download tập tin trong thư mục bằng giao thức HTTP 1.1

Mức độ hoàn thành: 100%

# Cách chạy chương trình

Chương trình sẽ chạy bằng command line

$ 1512406\_1512474 http://stream7.r17s100.vcdn.vn/filenhac.mp3 --http1.0

$ 1512406\_1512474 http://stream7.r17s100.vcdn.vn/filenhac.mp3 --http1.1

hoặc:

$ 1512406\_1512474 --http1.0 http://stream7.r17s100.vcdn.vn/filenhac.mp3

$ 1512406\_1512474 --http1.1 http://stream7.r17s100.vcdn.vn/filenhac.mp3

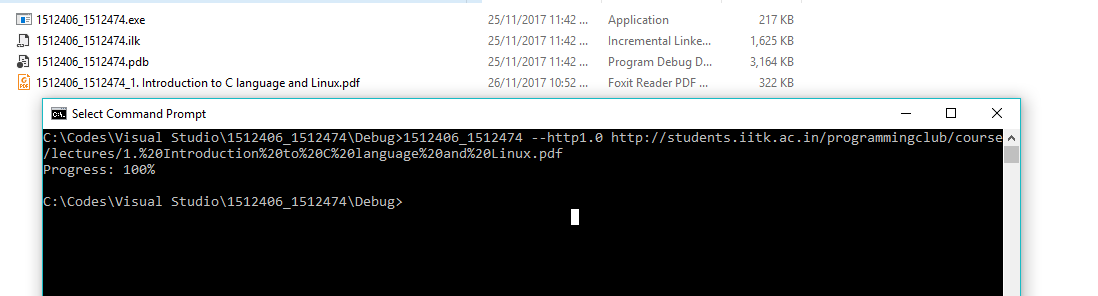
Trong đó

http1.0 dùng HTTP version 1.0

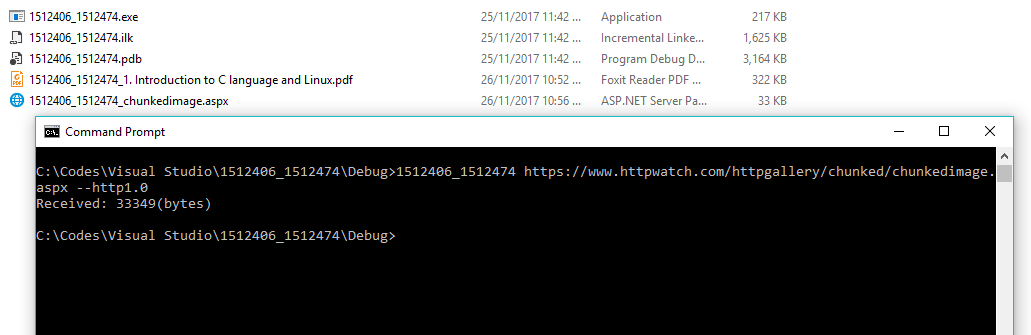
http1.1 dùng HTTP version 1.1

Kết quả chạy:

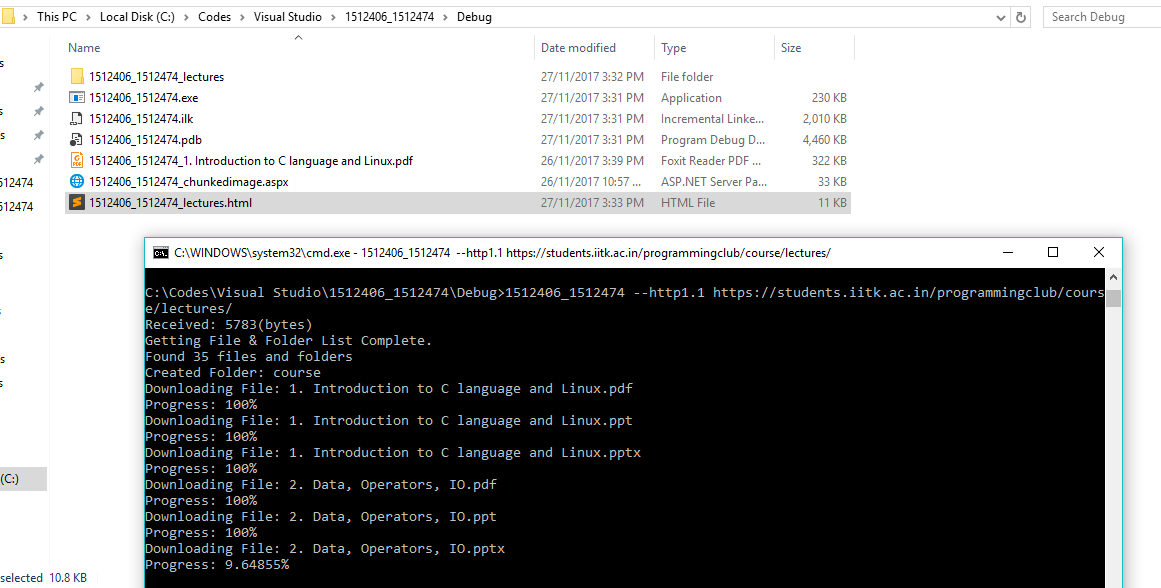
Phương thức truyền thông thường



Phương thức truyền chunked

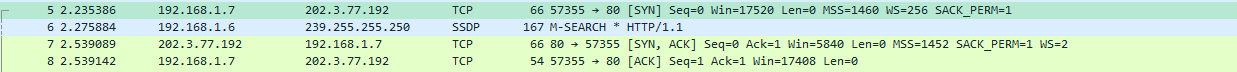


Tải xuống thư mục



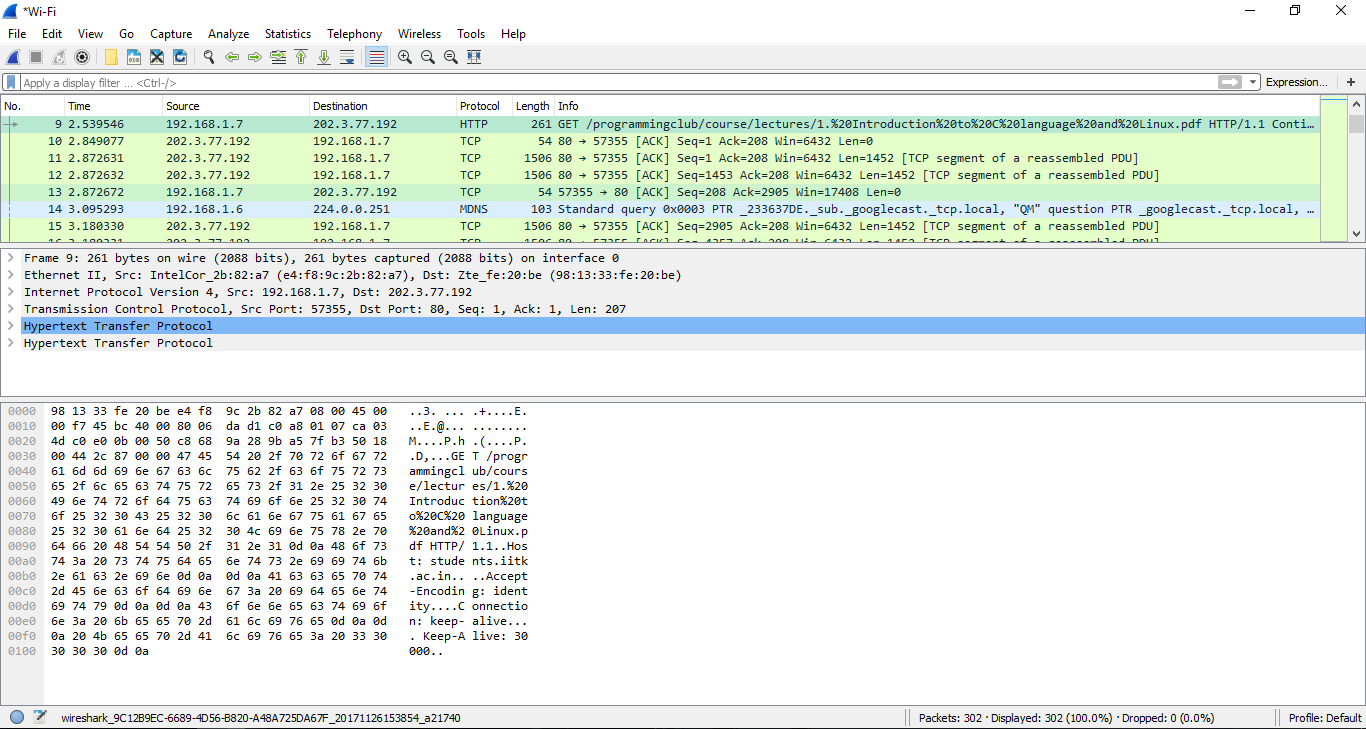
# Quá trình gửi nhận TCP segment data – ACK

Mở kết nối



* Chương trình gửi tín hiệu xin kết nối đến máy chủ [SYN]
* Máy chủ phản hồi tín hiệu chấp thuận [SYN, ACK]
* Chương trình gửi lại tín hiệu báo hiệu đã nhận được [ACK]

Gửi yêu cầu download tập tin:



* Chương trình gửi request yêu cầu download tập tin đến máy chủ
* Máy chủ phản hồi đã nhận được tín hiệu và gửi phản hồi nội dung gồm http response header và dữ liệu tập tin nằm ở http body

Đóng kết nối



* Chương trình gửi yêu cầu đóng kết nối [FIN,ACK] đến máy chủ
* Máy chủ phản hồi đã nhận tín hiệu và tiến hành đóng kết nối [ACK]