

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Khoa Công nghệ thông tin



BỘ MÔN: MẠNG MÁY TÍNH
BÁO CÁO ĐỒ ÁN SOCKET



Giáo viên hướng dẫn : Thầy Lê Hà Minh

Sinh viên thực hiện : Doãn Anh Khoa – 21127076
Nguyễn Chí Lương – 21127643
Hồ Thế Phúc - 2112670

Lớp: 21CLC03

Đề số: 2 – Web Client

Contents

1. Đánh giá mức độ hoàn thành từ 0 – 100% (Chú thích rõ những mục làm được, chưa làm được và còn bị lỗi) + Bảng phân công công việc và cho biết rõ ràng ai làm việc gì một cách rõ ràng.....	3
2. Kịch bản giao tiếp của chương trình:	4
3. Môi trường lập trình và các framework hỗ trợ để thực thi ứng dụng.	5
4. Hướng dẫn sử dụng các tính năng chương trình.....	6
5. Các nguồn tài liệu tham khảo.....	6

1. Đánh giá mức độ hoàn thành từ 0 – 100% (Chú thích rõ những mục làm được, chưa làm được và còn bị lỗi) + Bảng phân công công việc và cho biết rõ ràng ai làm việc gì một cách rõ ràng.

Chức năng	Thành viên thực hiện	Nhiệm vụ	Mức độ hoàn thành
Kết nối	21127670	Tạo một TCP socket, kết nối đến port 80 của web server Yêu cầu sử dụng HTTP/1.1 và "Connection: keep-alive"	100%
Quản lí kết nối	21127670	Khi client hoặc server mất kết nối đột ngột, không làm chương trình treo hay xảy ra lỗi	100%
Tải và lưu thành file dạng "Content-length"	21127076	Viết một hàm riêng với mục tiêu cắt header riêng ra khỏi phần nội dung chính. Sử dụng header để xác định kiểu truyền dữ liệu. Nếu nó thuộc kiểu "content-length" thì sẽ tìm độ dài của content nằm trong header và qua đó tải và lưu file dạng content-length	100%
Tải và lưu thành file dạng "Transfer-Encoding: chunked"	21127076	Viết một hàm riêng với mục tiêu cắt header riêng ra khỏi phần nội dung chính. Sử dụng header để xác định kiểu truyền dữ liệu. Nếu nó thuộc kiểu "Transfer-Encoding: chunked", thực hiện tải file theo từng "miếng". Đọc file theo format để lấy độ dài nội dung từng "miếng" của gói tin. Cuối cùng ghép các	100%

		miếng đó thành một tập tin hoàn chỉnh	
Tải các file trong folder	2112764343	Xác định có đang tải folder bằng cách phân tích url được cung cấp, nếu phần cần được tải có là folder, nó sẽ không có phần đuôi chỉ định kiểu tập tin. Thứ ta tải về ban đầu là file html chứa tên các file tương ứng nằm trong folder được nêu. Ghép tên các file này với url được cung cấp và gọi đệ quy hàm client_service để tải các folder và file nằm trong url được cấp ban đầu. Sau khi tải xong, sẽ lưu những thứ vừa tải vào thư mục được chọn.	100%
Multiple requests	21127643	Nhờ vào “Connection: keep-alive” để gửi nhiều request	100%
Multiple connection	21127670	sử dụng đa tiến trình để kết nối với servlet cùng một lúc, được sử dụng để cho phép kết nối nhiều request cho mỗi kết nối	100%
Report	21127076	Viết báo cáo trình bày chi tiết những yêu cầu	100%

2. Kịch bản giao tiếp của chương trình:

- Giao thức trao đổi giữa client và server
 - + Trước khi trao đổi client và server có thể trao đổi với nhau, chúng ta cần tạo ra một cổng Socket để “bắt tay 3 chiều” và thiết lập kết nối. Ví dụ trong mã nguồn: s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
 - + AF_INET: dùng để xác định hạ tầng mạng, nơi diễn ra giao tiếp của socket.

- + SOCK_STREAM (tham số truyền vào trong `socket.socket()`): là giao thức TCP / IP và là giao thức trao đổi giữa client và server. TCP hoạt động với giao thức Internet (IP) để chỉ định cách dữ liệu được trao đổi trực tuyến
- + Khởi tạo kết nối TCP, ta sẽ kết nối với nó với địa chỉ IP, Port Number của client và địa chỉ IP, Port Number của server bằng cách chương trình thực hiện DNS query. Từ một đường dẫn, mã nguồn sẽ xác định tên miền, sau đó sẽ thiết lập kết nối bằng cách truy vấn các máy chủ DNS để lấy địa chỉ IP liên quan đến miền đó.
- + Ví dụ trong mã nguồn thực hiện kết nối là `s.connect((domain_name, 80))` với `domain_name` là tên miền mà trước đó mã nguồn xác định, 80 là Port Number của web server.
- Cấu trúc thông điệp:
 - + Sau khi khởi tạo kết nối, client sẽ bắt đầu gửi yêu cầu đến web server bằng cách dùng hàm `s.sendall(request)` với request là đoạn thông điệp được mã hóa. Nội dung của request là phương thức GET, HTTP/1.1, "Connection: keep-alive" được sử dụng cho yêu cầu của đồ án
 - + Hàm `s.recv()` nhận dữ liệu mà server gửi phản hồi.
- Kiểu dữ liệu của thông điệp:
 - + Dữ liệu của yêu cầu được mã hóa thành chuỗi các byte để gửi đến web server.
 - + Sau khi gửi yêu cầu đến server, client sẽ nhận được gói tin phản hồi trong đó dữ liệu đã được mã hóa thành những chuỗi byte.
 - + TCP đảm bảo rằng các byte được truyền theo thứ tự mà chúng được gửi mà không có lỗi hoặc thiếu sót nào.

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

- Môi trường lập trình: Python
- Các framework hỗ trợ
 - + Thư viện socket: hỗ trợ sử dụng các hàm tạo cổng Socket, kết nối với web server, gửi request và nhận nội dung từ web server

- + thư viện concurrent.futures
- + thư viện sys: khả năng sử dụng các đối số, được truyền vào trong command line, sử dụng được các tham số dòng lệnh
- + thư viện os: cung cấp các chức năng được sử dụng để tương tác với hệ điều hành và cho phép chúng ta làm việc với các tập tin, thư mục.

4. Hướng dẫn sử dụng các tính năng chương trình.

- Chạy mã nguồn bằng cách mở command line, thêm các đường dẫn đến trang web để request đến web server
- Ví dụ: Tên mã nguồn: Client.py và chạy dòng lệnh trên cmd như sau
 - + python Client.py
<http://web.stanford.edu/class/cs231a/project.html>
 - + python Client.py <http://web.stanford.edu/class/cs224w/slides/01-intro.pdf> <http://www.bing.com>
 - + python Client.py <http://web.stanford.edu/class/cs224w/slides/>

5. Các nguồn tài liệu tham khảo.

- Kiến thức về thư viện os: <https://biquyetxaynha.com/thu-vien-os-trong-python>
- **internal / pointers, “Making HTTP requests with sockets in Python”, 2022,[Online].Available:
<https://www.internalpointers.com/post/making-http-requests-sockets-python>
- Wikipedia, “Chunked transfer encoding”, [Online]. Available:
https://en.wikipedia.org/wiki/Chunked_transfer_encoding
- bunny.net, “What Is HTTP Chunked Encoding? How and When Is It Used?”, [Online]. Available:
<https://bunny.net/academy/http/what-is-chunked-encoding/>
- Stack Overflow, “How to download image from HTTP server[Python/sockets]”, 2017, [Online]. Available:
<https://stackoverflow.com/questions/43408325/how-to-download-image-from-http-server-python-sockets>
- GeeksforGeeks, “Command Line Arguments in Python”, 2022, [Online]. Available:

<https://www.geeksforgeeks.org/command-line-arguments-in-python/>

- GeeksforGeeks, “OS Module in Python with Examples”, 2022, [Online]. Available:

<https://www.geeksforgeeks.org/os-module-python-examples/>

- GeeksforGeeks, “Multithreading in Python | Set 1”, 2022, [Online]. Available:

<https://www.geeksforgeeks.org/multithreading-python-set-1/>