**BÁO CÁO**

**BÀI THỰC HÀNH 1**

MÔN: **Khoa học về web**.

Giáo viên hướng dẫn: **Nguyễn Hoàng Khai**

**MỤC LỤC**

[**I.** **Giới thiệu nhóm.** 2](#_Toc477717598)

[**II.** **Trả lời câu hỏi** 2](#_Toc477717599)

1. **Giới thiệu nhóm.**

|  |  |  |
| --- | --- | --- |
| MSSV | Họ tên | Đóng góp |
| 1412363 | Trần Thị Nhã | 30% |
| 1412477 | Đoàn Hiếu Tâm | 40% |
| 1412548 | Nguyễn Đình Tiến | 30% |

1. **Trả lời câu hỏi**
2. Mô tả sơ lược về thư viện và cách cài đặt chương trình.

* Về thư viện:

Chương trình **Simple Py Web Crawler** không dùng framework hỗ trợ crawling các website (như **Scrapy**,..). Chương trình tạo ra nhờ sự hỗ trợ của các module thư viện như:

* [**urlparse**](https://docs.python.org/3/library/urllib.parse.html) của **urllib.parse** để parse đường dẫn của các web pages.
* [**HTMLParser**](https://docs.python.org/3/library/html.parser.html) của html.parser để phân rã nội dung của các web pages nhằm ghi ra file và lấy các đường link chứa trong thẻ **anchor element <a>.**
* [**urllib.request**](https://docs.python.org/3.0/library/urllib.request.html) để gửi request và nhận về các response. Sau đó lấy header từ response để kiểm ra **Content-Type** nhằm chắc chắn rằng web pages đó là một pages chứa text chứ không phải là đường link của một tấm ảnh hay một file media.
* Các hàm của **Python 3.6.0**: [**threading**](https://docs.python.org/3/library/threading.html) để cài đặt đa tiến trình crawl các web pages, [**queue**](https://docs.python.org/3/library/queue.html) dùng làm hàng đợi chứa các link trong frontier của Crawler, [**os**](https://docs.python.org/3/library/os.html) dùng thực hiện các thao tác đọc/ghi files.
* Cách cài đặt chương trình:

Cài đặt các hàm thao tác với tệp tin và thư mục trong file **file\_utils.py**, sau đó dùng nó để ghi nội dung của web pages vừa crawl được ra file HTML, đồng thời sử dụng để đọc/ghi các đường link đã đưa vào frontier và đã crawl từ file lên set (Python) và ngược lại.

Cài đặt các hàm phân tách các web pages đã crawl để tìm thêm các đường link khác chứa trong các thẻ <a> của page đó trong file **finder.py**. Sau đó lưu các đường link vừa tìm được vào hàng đợi của frontier.

Cài đặt **WebSpider** trong file **web\_spider.py**. Class này gọi lại các hàm ở trên theo đúng công đoạn làm việc của một web crawler cơ bản là: Đặt homepage của Website cần crawl vào frontier, sử dụng http request để lấy nội dung của webpage, ghi ra tệp HTML, sau đó dùng link\_finder để parse nội dung của webpage đó, lấy các đường link tìm được đưa vào hàng đợi frontier. Cập nhật lại hàng đợi frontier và các set chứa các đường link, gọi hàm trong file\_utils để ghi ra tệp tin. Và trở lại bước đầu, lấy một link từ hàng đợi frontier ra để xử lý….

File **main.py** dùng để khởi tạo **WebSpider**, nhập xuất website và lưu các giá trị toàn cục.

1. Mô tả kết quả chạy thực nghiệm:

* Số lượng trang web tải được là: 10000 ~ (đã chạy được đến 10 000 và vẫn chưa thấy có vấn đề gì làm crawler ngừng chạy.)
* Tải 5000 trang web mất thời gian: 1 giờ 01 phút (61 phút)
* Mức sử dụng bang thông của Crawler:
  + Tốc độ tải: Trung bình 110kB/s
  + Chiếm 80% băng thông đang sử dụng (Vì khi đang chạy crawler chỉ đọc tin tức trên các trang báo)
  + Chiếm 6-10% băng thông nhà mạng cung cấp (VNPT 15Mbs)