University of Washington Bothell

CSS 436: Cloud Computing

Program 1: Simple Crawl

Purpose

The programmable Web and cloud is built on HTTP. In this lab we will utilize HTTP programming to "GET" and crawl parts of the HTML based Web. The goal is to familiarize students with HTTP and programmatically searching the web.

Problem Statement

Create an application which takes two arguments:

1) A URL as a starting point
2) The number of hops from that URL (NumHops)

Your application will download the html from the starting URL which is provided as the first argument to the program. It will parse the html finding the first <a href > reference to other absolute URLs, for instance https://www.w3schools.com/tags/att_a_href.asp . Make sure that you have not previously visited this page (if you have then skip and find the next reference). Look for only http and https URLs. The application will then download the html from that page and repeat the operation. If that page is not accessible then continue on the current page looking for the next reference and visit that reference. You will do this NumHops times.

Your app should print out to the console the URL of each hop that you visit. If you encounter a page without any accessible embedded references you should stop there and print out the result.

When your app has visited numHops pages the last page is printed to the console as text.

Problem Statement Details

- **Example:**

  MyWebCrawl.exe http://courses.washington.edu/css502/dimpsey 5

  Prints out to the console the URL and the html from the URM 5 hops from http://courses.washington.edu/css502/dimpsey

Details

- You may build you app either with C# or Java. For this exercise do not use Python.

- The JSoup library should not be used for this assignment.
- Make sure that your program takes in two arguments—*do not query the user for the URL or number of hops.*
- Your program should gracefully handle all input, including bad input.
- An URL with a trailing / should be seen as the same as one without a trailing /
- If a site is unreachable continue down the current page for a URL which is accessible
- You should appropriately handle HTTP requests with return codes in the 300s or 400s
- Building your program should be easy to do by the grader using your Makefile. Part of the grade will be the ease of creating your executable. An IDE should not be required to build the application.
- You can submit a .class or .jar file for you executable (if using java).

Turn In

A .**zip file** which the module named:

- Executable of application
- All code and clear instructions or Makefile on how to build and run the application