Srividya Majeti

Assignment 1

CS 532: Introduction to Web Science
Dr. Michael Nelson
Spring 2016

January 28, 2016

Contents

1

Question 1

Demonstrate that you know how to use "curl" well enough to correctly POST data to a form. Show that the HTML response that is returned is "correct". That is, the server should take the arguments you POSTed and build a response accordingly. Save the HTML response to a file and then view that file in a browser and take a screen shot.

To POST data to a form using cURL, I did the following:

- I created a form using PHP with name as an input element and a submit button.
- When we open this PHP form in the browser and type any name, it displays a response saying "Welcome followed by the text".
- By using the following cURL command we can POST data to the form which is received by the server and generates a response with the same message that we see on the browser. By using -o followed by parameter, I am outputting the HTML response to a file.

```
curl -d name=Srividya ``www.cs.odu.edu/~smajeti/postForm.php''
-o output.html
```

```
<?php
        if( $_POST["name"]) {
      if (preg_match("/[^A-Za-z'-]/",$_POST['name'] )) {
         die ("invalid name and name should be alpha");
      }
      echo "Welcome ". $_POST['name'];
          echo "\n";

          exit();
         }
?>
<form method ="POST" action="<?php echo $_SERVER['PHP_SELF
    '];?>">
        <input type="text" name="name">
        <input type="submit" name="submit" value="OK">
</form>
```

Listing 1.1. "PHP script"

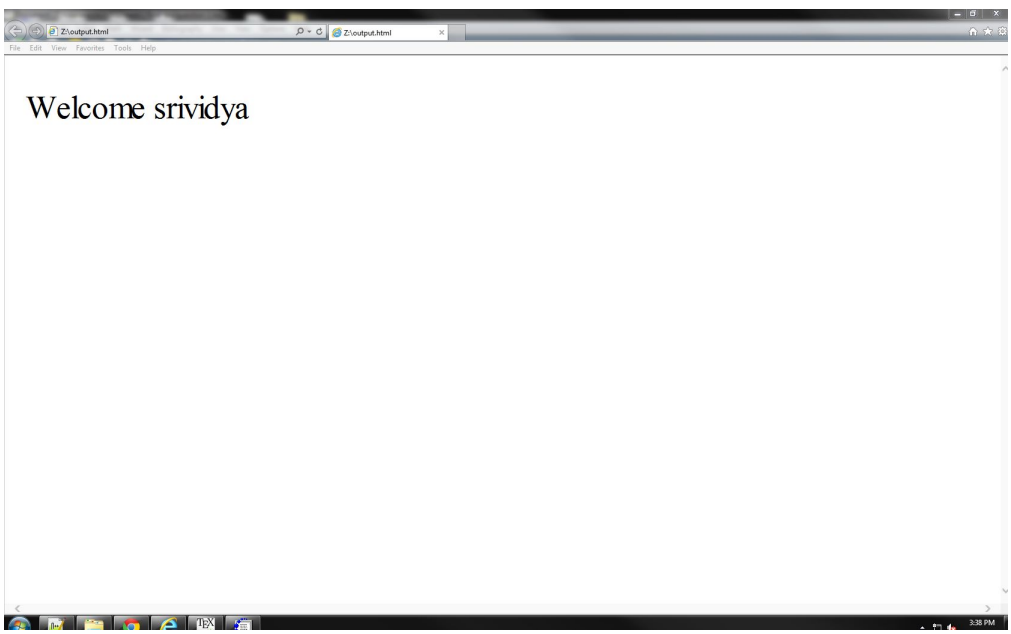


Fig. 1.1. HTML response saved into a file and viewed in browser.

2

Question 2

Write a Python program that:

- **takes as a command line argument a web page**
- **extracts all the links from the page**
- **lists all the links that result in PDF files, and prints out the bytes for each of the links. (note: be sure to follow all the redirects until the link terminates with a "200 OK".)**
- **show that the program works on 3 different URIs, one of which needs to be:**`http://www.cs.odu.edu/~mln/teaching/cs532-s16/test/pdfs.html`

For solving the above problem I have written a program using Python. Following are the steps I've taken to solve the problem:

- The python script takes the web page as command line argument, finds all the links with "a" tag and passes its href to the function getsize().
- This function fetches the content-length, content-type and status code for each link and extracts all the links that result in PDF files by checking if the content-type is "application/pdf" and if the status code is "200".
- The output of the program prints the links that are PDF files and its size.

```python
import urllib2
import sys #to pass link as command line arguement
from bs4 import BeautifulSoup,SoupStrainer

uri= sys.argv[1]
request= urllib2.Request(uri)
response= urllib2.urlopen(request)
response.getcode()
soup = BeautifulSoup(response,"html5lib")

def getsize(link):
        file= urllib2.urlopen(link)
        size=file.headers.get("content-length")
        type=file.headers.get("content-type")
        status=file.getcode()
        if status == 200 and type == "application/pdf":
                print "found a url with pdf in the link:",
                    link
                print "size:",size,"bytes"
                print "status:",status
        file.close()

for link in soup.findAll('a'):
        getsize(link['href'])
```

Listing 2.1. "Python code for extracting links that are PDF files"

- To run the code, go to the folder where the python code is located. Type python extractPDF.py <link> to see the output.
- I tried to run the code for URIs which use relative path in the href, but my program wasn't returning any URI with PDFs as the href gets appended to the base URI. I did not handle this case. This can be handled by appending this relative href to the base URI.
- I ran the program on three different URIs.

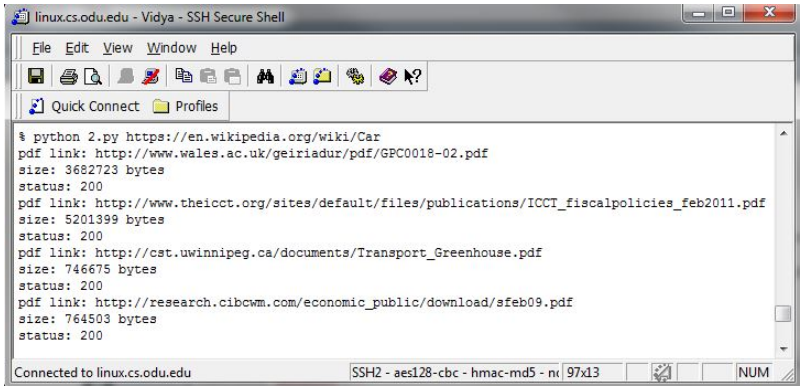


Fig. 2.1. Output for URI `https://en.wikipedia.org/wiki/Car`

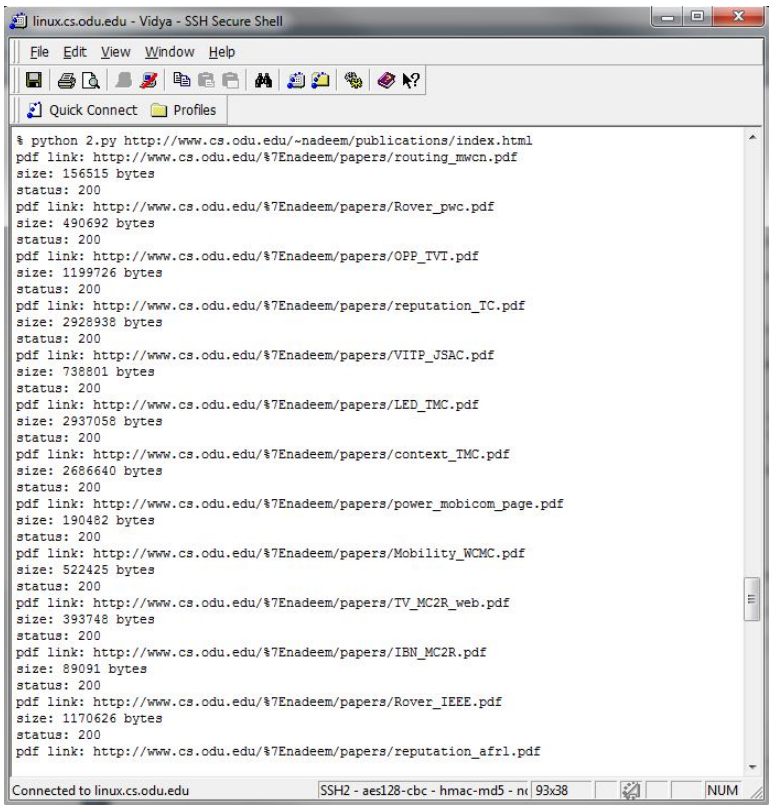


Fig. 2.2. Output for URI `http://www.cs.odu.edu/~nadeem/publications/index.html`

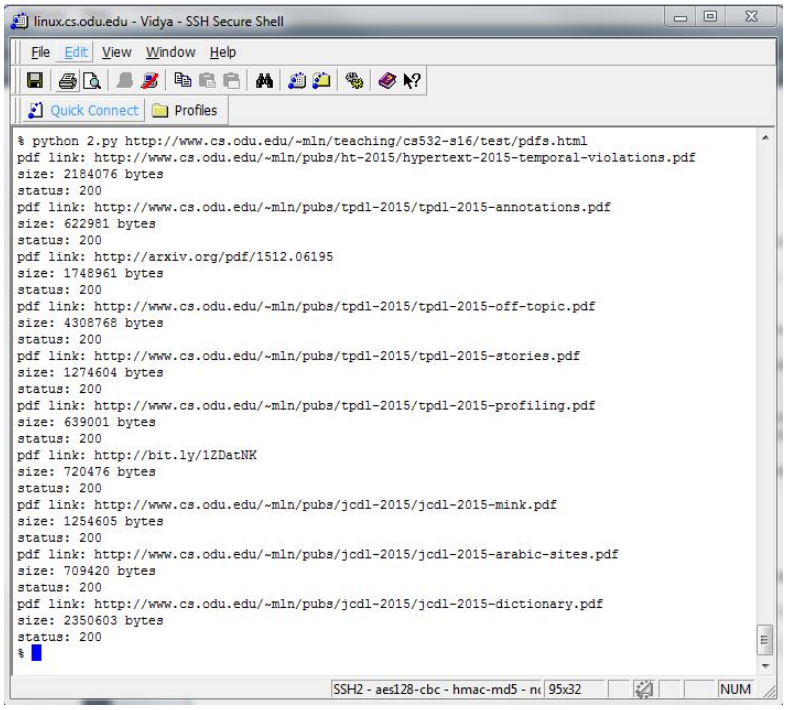


Fig. 2.3. Output for URI `http://www.cs.odu.edu/~mln/teaching/cs532-s16/test/pdfs.html`

3

Question 3

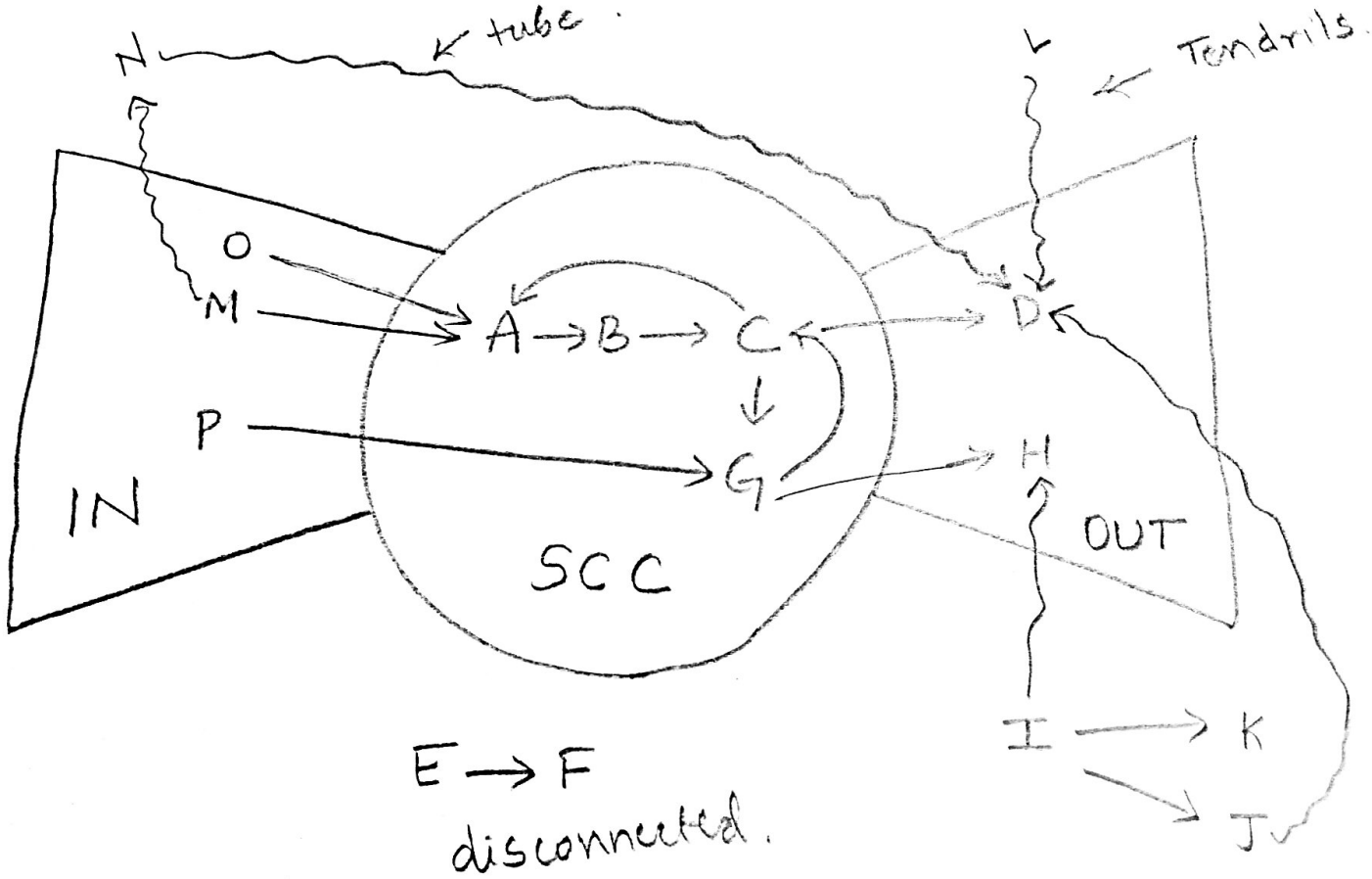
Consider the "bow-tie" graph in the Broder et al. paper (fig 9): `http://www9.org/w9cdrom/160/160.html`

Now consider the following graph:

A $\longrightarrow$ **B**
B $\longrightarrow$ **C**
C $\longrightarrow$ **D**
C $\longrightarrow$ **A**
C $\longrightarrow$ **G**
E $\longrightarrow$ **F**
G $\longrightarrow$ **C**
G $\longrightarrow$ **H**
I $\longrightarrow$ **H**
I $\longrightarrow$ **J**
I $\longrightarrow$ **K**
J $\longrightarrow$ **D**
L $\longrightarrow$ **D**
M $\longrightarrow$ **A**
M $\longrightarrow$ **N**
N $\longrightarrow$ **D**
O $\longrightarrow$ **A**
P $\longrightarrow$ **G**

For the above graph, give the values for:
IN:
SCC:
OUT:
Tendrils:
Tubes:
Disconnected:

- **SCC(heart of the web):** In our graph A, B, C, G are the strongly connected components. If we select any 2 nodes among A, B, C, G there exists a path between them.
- **IN:** O, M, P belongs to IN. These nodes can access SCC but they cannot be accessed from SCC.
- **OUT:** D, H belongs to OUT. These pages cannot access SCC but can be accessed from SCC.
- **TENDRILS:** I, J, L, N are tendrils.These pages cannot reach SCC and cannot be reached from SCC. N is a node that is reachable from portion of IN and I, J, L are nodes that can reach portions of OUT, without passing through SCC.
- **TUBES:** M, N, D form a tube.
- **DISCONNECTED:** E, F, K are disconnected nodes.


tube.
Tendrils.
N
L
O
M
P
A
B
C
D
G
H
I
K
J
IN
SCC
OUT
E → F
disconnected.

References

1. Inserting code listings in latex https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings
2. Extract links using BeautifulSoup http://stackoverflow.com/questions/3075550/how-can-i-get-href-links-from-html-code
3. Getting content-length http://stackoverflow.com/questions/12317493/urllib2-urlopen-getting-the-size-of-the-content
4. Inserting PDF in latex http://tex.stackexchange.com/questions/149443/how-to-include-pdf-file-in-latex-doc-from-folder-location