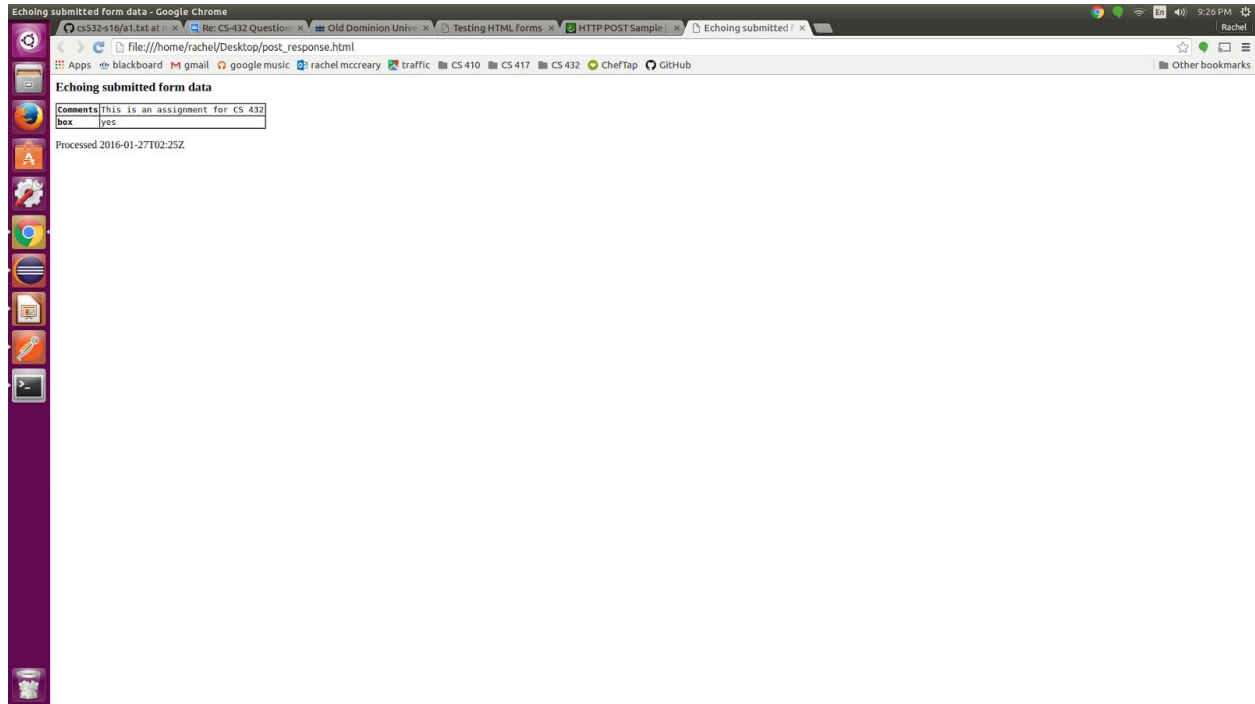


PART 1

[illegible]

and received this response:



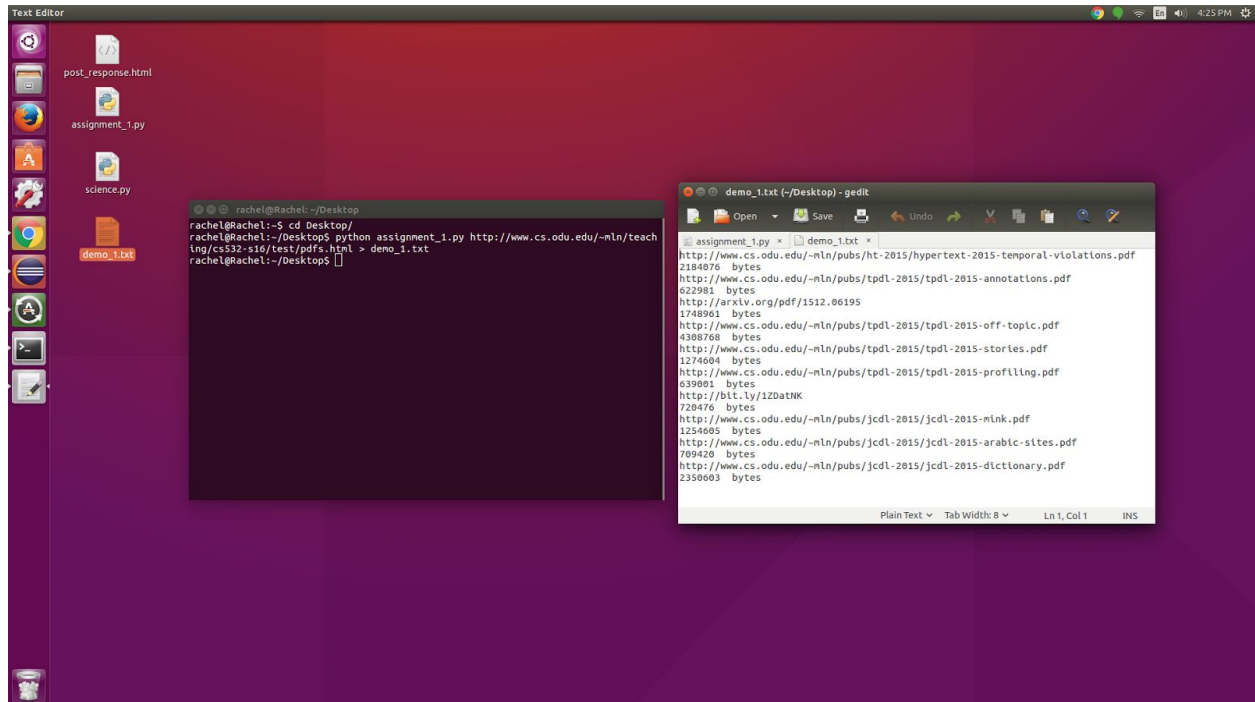
PART 2

This program takes a web page as a command line argument, extracts all the links from that page, lists all links that result in PDF files, and prints out the size (in bytes) for each of the links. This was my first time programming in Python, so feedback is appreciated. (The actual file will also be attached)

```
# RACHEL MCCREARY
# CS 432 ASSIGNMENT 1
from bs4 import BeautifulSoup
import urllib2
import urlparse
import sys
if __name__ == "__main__":
    for arg in sys.argv[1:]:
        uri = arg
        page = urllib2.urlopen(uri)
        soup = BeautifulSoup(page.read(), 'html.parser')
        for link in soup.find_all('a'):
            href = link.get('href')
            if href != None:
                if href.startswith("http") == False:
                    href = urlparse.urljoin(uri, href)
                    response = urllib2.urlopen(href)
                    status_code = response.info().getheader('Status')
                    content_type = response.info().getheader('Content-Type')
                    if content_type == "application/pdf":
                        print href
                        size_of_pdf = response.info().getheader('Content-Length')
                        print size_of_pdf, " bytes"
```

Libraries used: sys (for command line arguments)
BeautifulSoup (for extracting links)
urllib2 (for opening links and getting content type & length from headers)
urlparse (for dealing with redirects)

Trying it out on three different URIs:

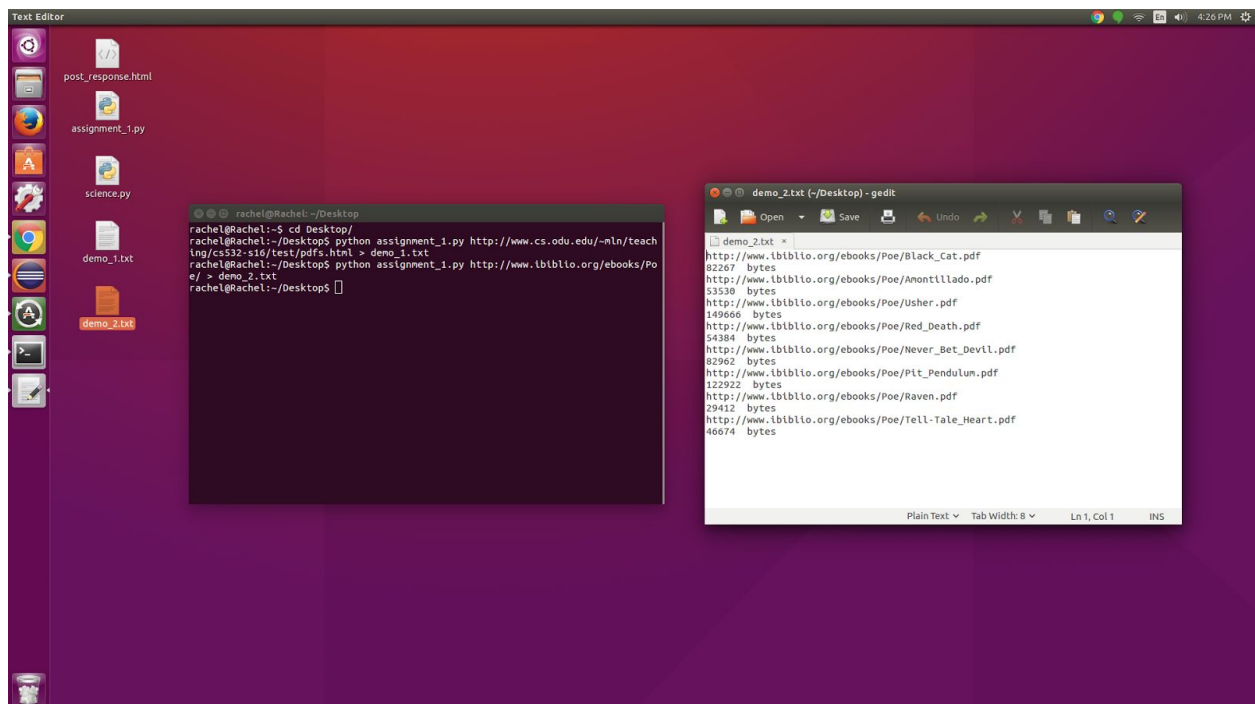


The screenshot shows a Linux desktop with a purple background. On the left is a vertical dock with icons for various applications. The main workspace contains two windows. The first window is a terminal titled 'rachel@Rachel: ~/Desktop'. It shows the following commands and output:

```
rachel@Rachel:~/Desktop$ cd Desktop/
rachel@Rachel:~/Desktop$ python assignment_1.py http://www.cs.odu.edu/~nln/teaching/cs332-s16/test/pdfs.html > demo_1.txt
rachel@Rachel:~/Desktop$
```

The second window is a text editor titled 'demo_1.txt (-/Desktop) - gedit'. It displays a list of URLs and their corresponding content lengths in bytes:

```
http://www.cs.odu.edu/~nln/pubs/ht-2015/hypertext-2015-temporal-violations.pdf
2184876 bytes
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-annotations.pdf
622981 bytes
http://arxiv.org/pdf/1512.06195
1748961 bytes
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-off-toplc.pdf
4308768 bytes
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-stories.pdf
1274604 bytes
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-profiling.pdf
639081 bytes
http://bit.ly/1ZDatNK
728476 bytes
http://www.cs.odu.edu/~nln/pubs/jcdl-2015/jcdl-2015-mnk.pdf
1254605 bytes
http://www.cs.odu.edu/~nln/pubs/jcdl-2015/jcdl-2015-arabic-sites.pdf
709429 bytes
http://www.cs.odu.edu/~nln/pubs/jcdl-2015/jcdl-2015-dictionary.pdf
2350603 bytes
```

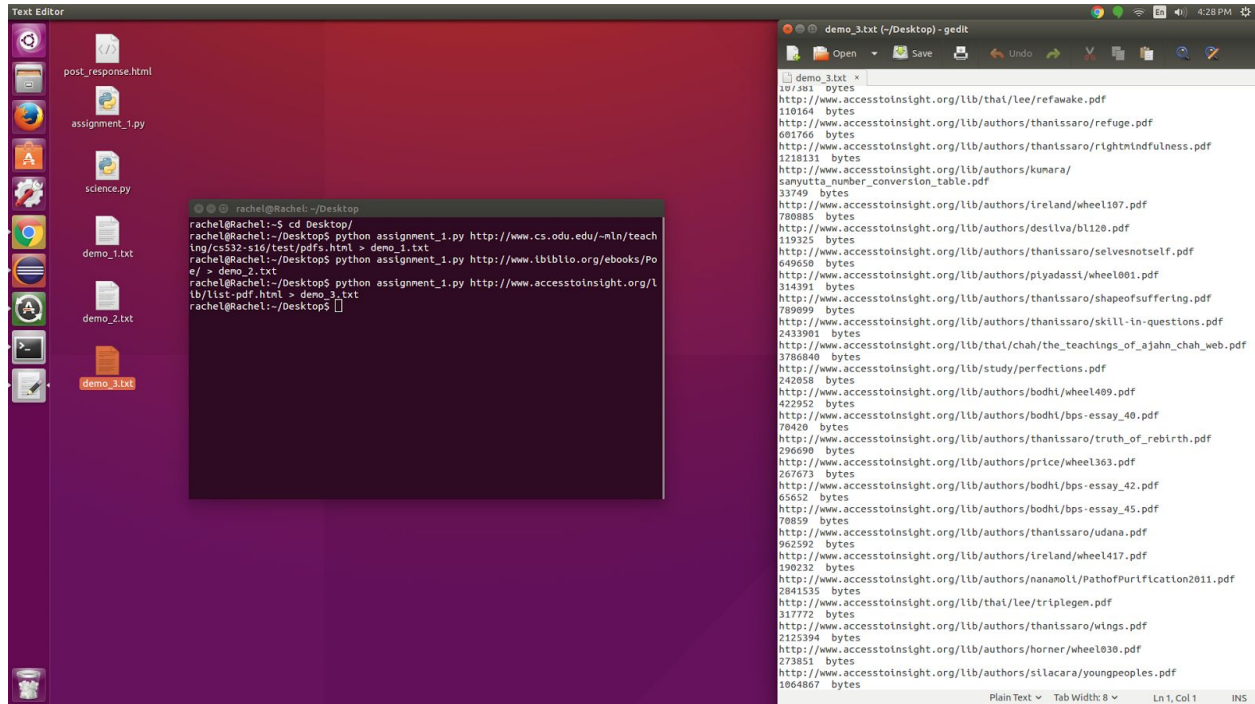


The screenshot shows the same Linux desktop environment. The terminal window now shows the following commands and output:

```
rachel@Rachel:~/Desktop$ cd Desktop/
rachel@Rachel:~/Desktop$ python assignment_1.py http://www.cs.odu.edu/~nln/teaching/cs332-s16/test/pdfs.html > demo_1.txt
rachel@Rachel:~/Desktop$ python assignment_1.py http://www.lbblo.org/ebooks/Poe/ > demo_2.txt
rachel@Rachel:~/Desktop$
```

The text editor window, titled 'demo_2.txt (-/Desktop) - gedit', displays a list of URLs and their content lengths in bytes:

```
http://www.lbblo.org/ebooks/Poe/Black_Cat.pdf
82267 bytes
http://www.lbblo.org/ebooks/Poe/Anontillado.pdf
53530 bytes
http://www.lbblo.org/ebooks/Poe/Usher.pdf
149666 bytes
http://www.lbblo.org/ebooks/Poe/Red_Death.pdf
54384 bytes
http://www.lbblo.org/ebooks/Poe/Never_Bet_Devil.pdf
82962 bytes
http://www.lbblo.org/ebooks/Poe/Plt_Pendulum.pdf
122922 bytes
http://www.lbblo.org/ebooks/Poe/Raven.pdf
29412 bytes
http://www.lbblo.org/ebooks/Poe/Tell-Tale_Heart.pdf
46674 bytes
```



PART 3

For this part, I mainly used these snippets in the the Broder et al. paper for reference:

“Figure 9: Connectivity of the web: one can pass from any node of IN through SCC to any node of OUT. Hanging off IN and OUT are TENDRILS containing nodes that are reachable from portions of IN, or that can reach portions of OUT, without passage through SCC. It is possible for a TENDRIL hanging off from IN to be hooked into a TENDRIL leading into OUT, forming a TUBE -- a passage from a portion of IN to a portion of OUT without touching SCC.”

“This connected web breaks naturally into four pieces. The first piece is a central core, all of whose pages can reach one another along directed links -- this "giant strongly connected component" (SCC) is at the heart of the web. The second and third pieces are called IN and OUT. IN consists of pages that can reach the SCC, but cannot be reached from it - possibly new sites that people have not yet discovered and linked to. OUT consists of pages that are accessible from the SCC, but do not link back to it, such as corporate websites that contain only internal links. Finally, the TENDRILS contain pages that cannot reach the SCC, and cannot be reached from the SCC.”

I ended up with these values:

IN: **M, O, P**
 SCC: **A, B, C, G**
 OUT: **D, H**
 TENDRILS: **I, J, K, L, N**
 TUBES: **N**
 DISCONNECTED: **E, F**

My bow-tie graph:

