Zhihua Zhang & KaiXiang Wang CSOC 560

Homework 1

1. Single connection mode (serial) and multiple connection mode (parallel with multiple threads)

In our code, ten threads have been initialized.

```python
import BaseHTTPServer
import CGIHTTPServer ## This line enables CGI error reporting
from SocketServer import ThreadingMixIn
import threading
import time
import socket


if __name__ == '__main__':
    addr = ('', 5000)
    handler = CGIHTTPServer.CGIHTTPRequestHandler
    print("Server started, listen to port 5000!")
    server = BaseHTTPServer.HTTPServer(addr, handler)
    for i in range(10):  # create 10 threads
        server_thread = threading.Thread(target=server.serve_forever)
        server_thread.daemon = True
        server_thread.start()
    time.sleep(9e9)
```

2. HTTP GET requests with query and header parsing

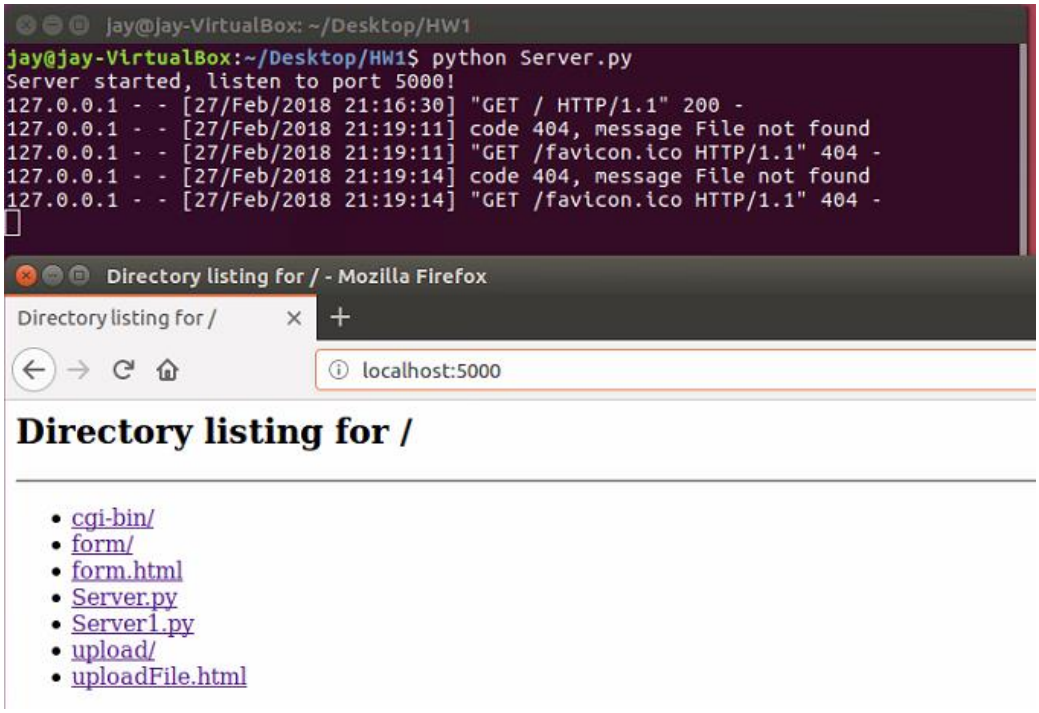
When the server is running, use the curl command in Linux, the result of HTTP GET requests are showed below. Also, use GET command in Linux, you can get the content of webpage.

```
jay@jay-VirtualBox: ~
jay@jay-VirtualBox:~$ curl -I -L http://localhost:5000
HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/2.7.13
Date: Wed, 28 Feb 2018 02:34:59 GMT
Content-type: text/html; charset=UTF-8
Content-Length: 444
```

```
jay@jay-VirtualBox:~$ GET http://localhost:5000
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><htm
<title>Directory listing for /</title>
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href="cgi-bin/">cgi-bin/</a>
<li><a href="form/">form/</a>
<li><a href="form.html">form.html</a>
<li><a href="Server.py">Server.py</a>
<li><a href="Server1.py">Server1.py</a>
<li><a href="upload/">upload/</a>
<li><a href="uploadFile.html">uploadFile.html</a>
</ul>
<hr>
</body>
</html>
```

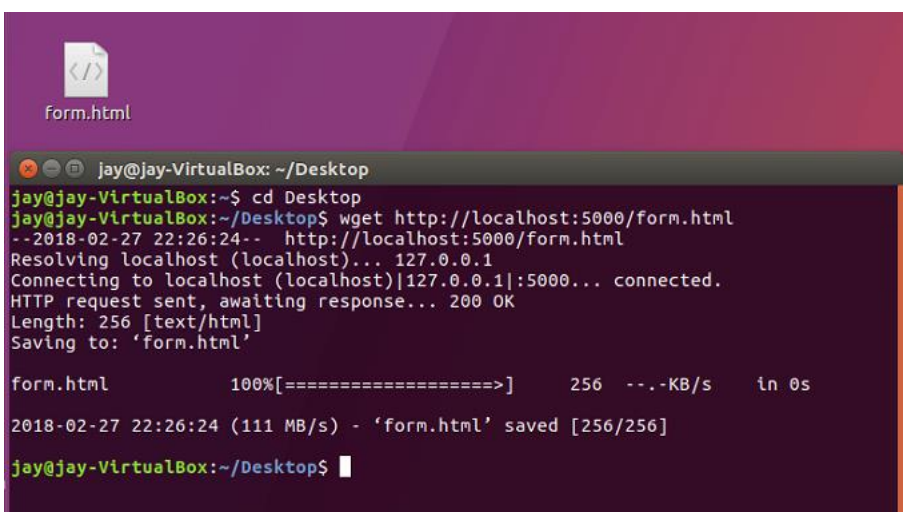
3. Automatic directory listing

Run the Server: python Server.py, the directory listing is showed. Need to note, the cgi-bin/ directory cannot be accessed, because the CGI scripts are in this directory, which may not be read or modified by clients.

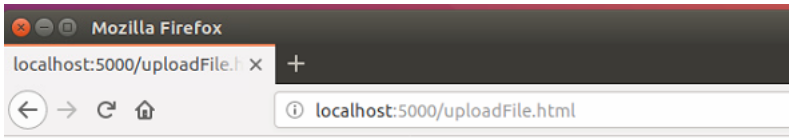


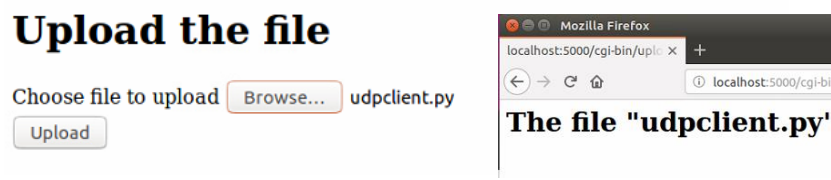
4. Static file transport

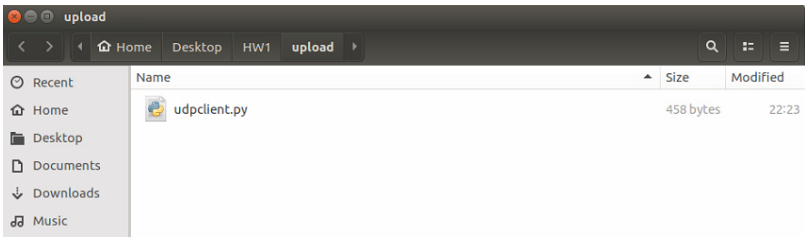
For downing files from the server, we use the wget command:
wget thttp://localhost:5000/form.html



For uploading the files to the server, open the uploadFile.html, then simply choose the file to upload from our local function.

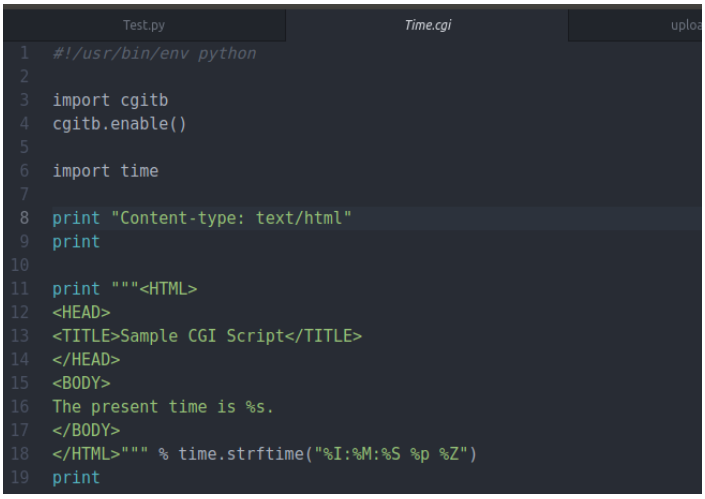


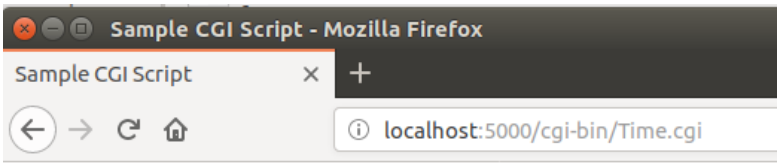




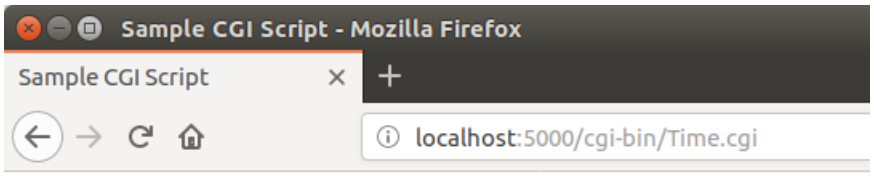
5. Basic CGI support by running a sample CGI script on the server side

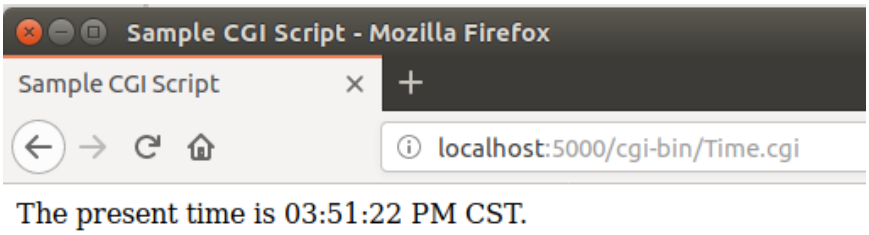
We created a simple test CGI script: Time.cgi. When we browser to the http:\\localhost:5000\cgi-bin\Time.cgi, the CGI script will be executed by the server, the current time will be displayed in the page. Note: To run the CGI script, you need to make the code executable, like chmod +x Time.cgi





The present time is 03:49:10 PM CST.





Extra credit: implement basic user interaction on submitting a form.

We first created the form.html, then when the form is submitted, the formSave.py CGI script will be executed, the form data will be saved to the form directory named with formData.txt.