Project - 2

Akash Waghela (13064)

Date : *31st August, 2016*

Browser Used: *Google Chrome*

How to run

> Make all

> ./proxy PortNumber

> python proxy_tester.py proxy {port number optional}

Implemented Options:

1. Supports Get Request only.
2. Handles ill formed requests, without getting stuck.
3. Concurrent Fetching supported.
4. Every response in sent from proxy server having Connection close.
5. Error checking.

Design Choices made:

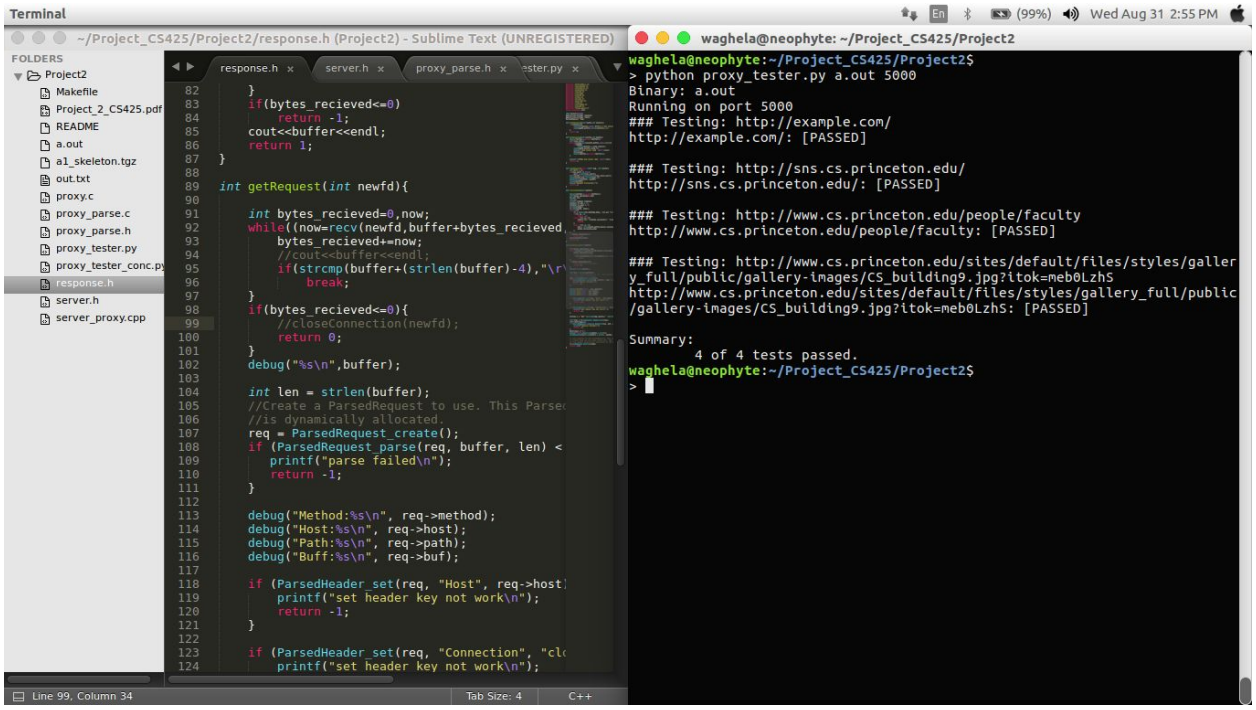
6. Request length is less than 4096 bytes.
7. Server should be given no more than 20 simultaneous/ parallel requests.
8. All parsing error and buffer overflows are replied with “500 Internal Server Error”.
9. The server reads a request until there is something to read and then quits if request end is observed([CLRF][CLRF]) else waits for a fixed amount of time before quitting.
10. Signal is used to keep track of number of childs, a handler is defined that is called on exit of every child process .

Explanation of Testing Results

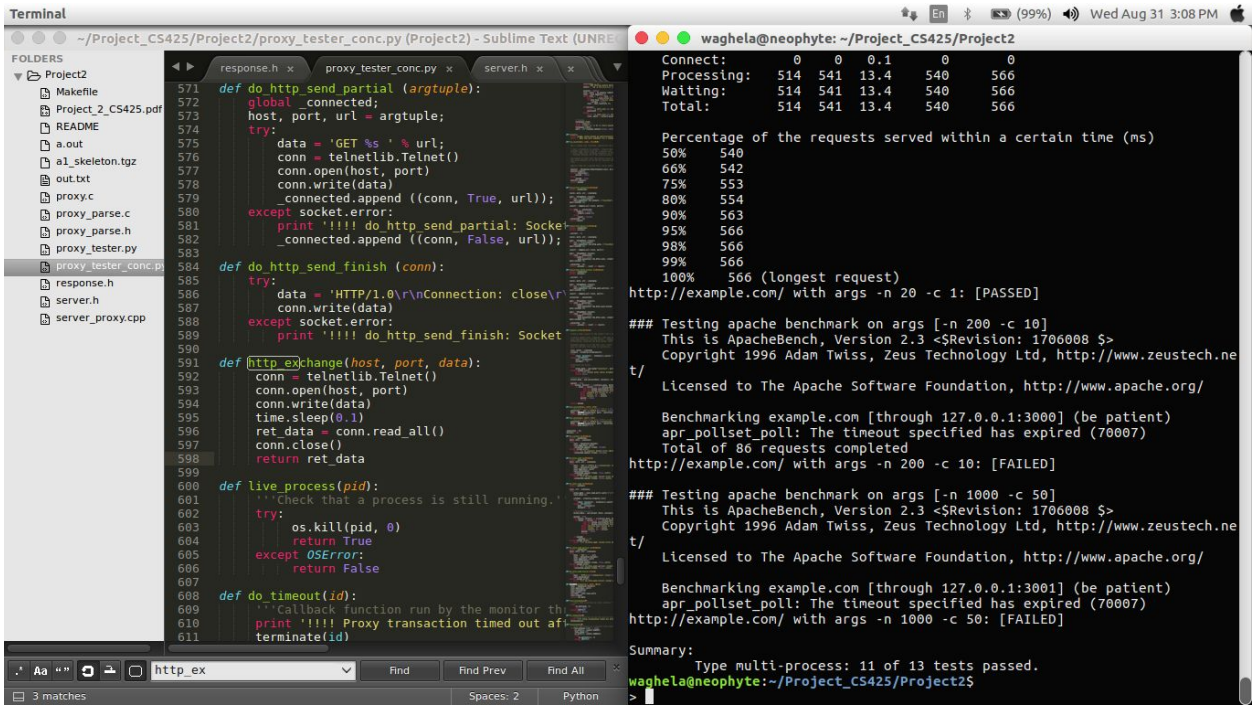
1. **Script1**
    - I added an extra wait in the script to pass the last test case, as it was immediately trying to get response.
2. **Script2**
    - 11/13 test cases passed, 1 of which was the 4th one, I did not add any wait in this script hence got failed.
    - The other was apache benchmark, it did not passed due to slow connection at hostel network and yield connection timeout, works fine on CSE network.
3. **Browser Response**
    - A request to 9gag.com was sent on port 3000 which processed successfully (localhost2 proxy setting was server:locahost port:3000).
    - The “parse failed” is due to the dynamic components in other web pages that are trying to send various request.
4. **Malformed Response**
    - To test 500 Internal Error

Testing Results

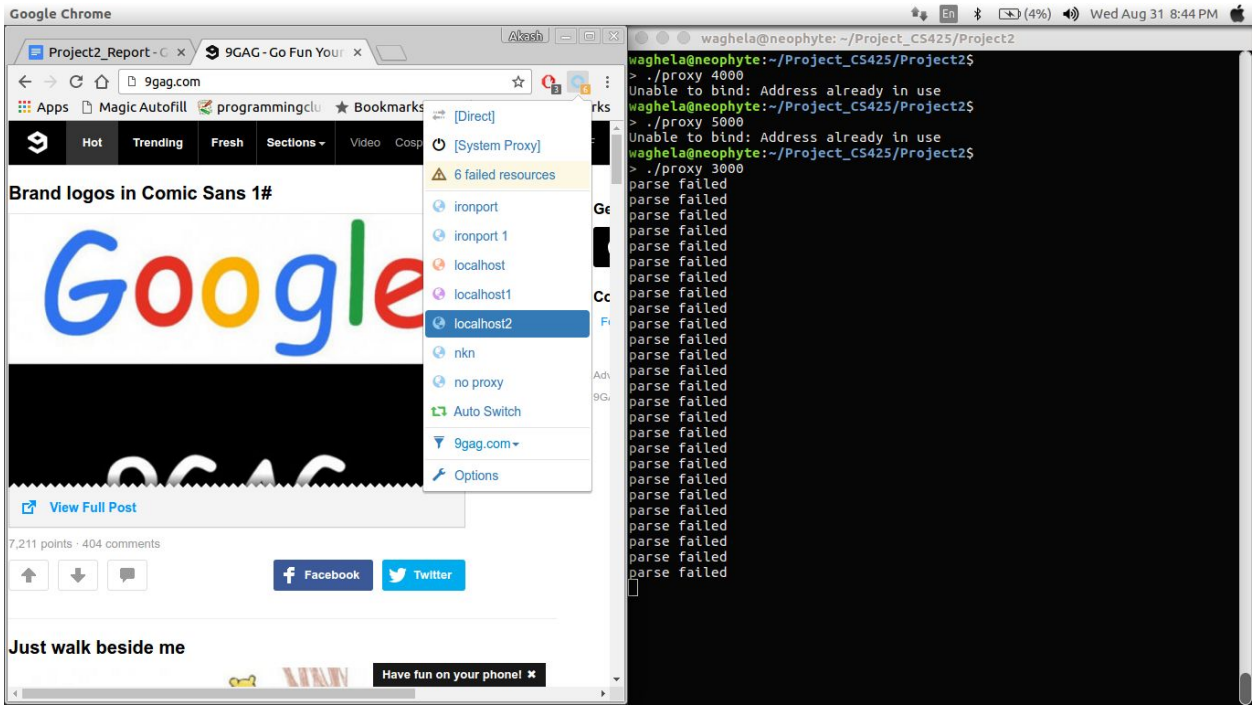
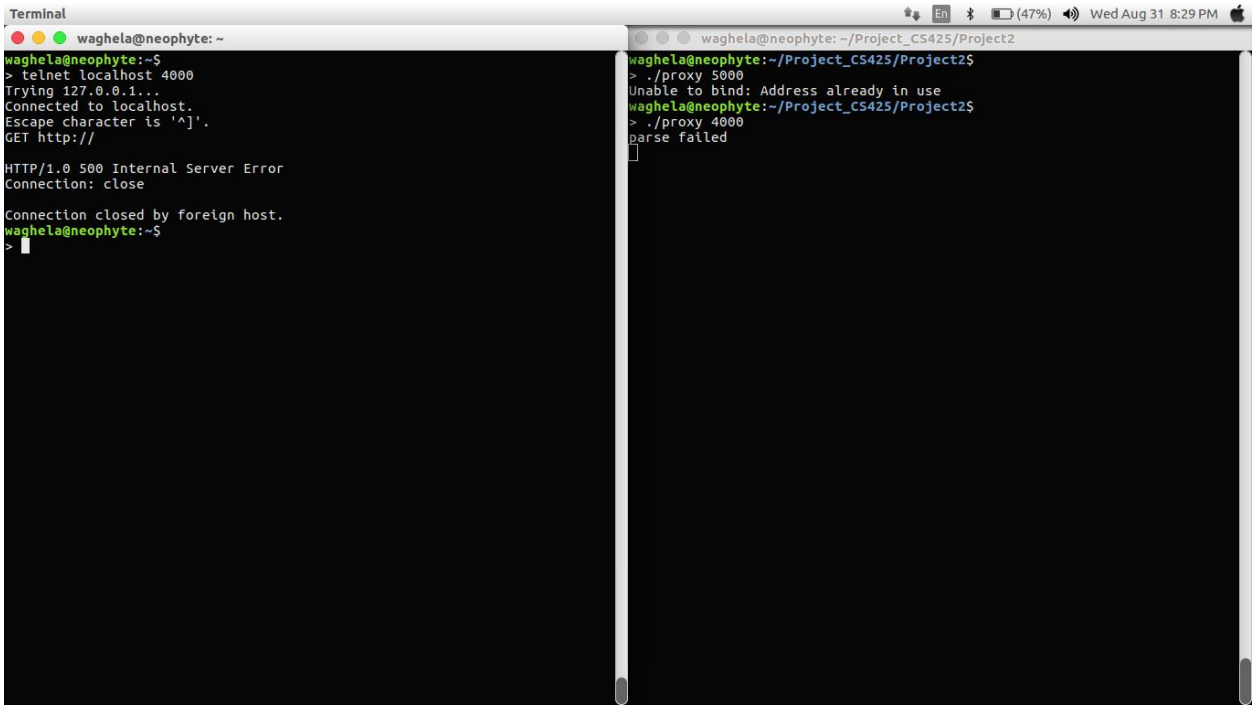
1. OutPut of Script1 (proxy_tester.py)



2. OutPut of Script2 (proxy_tester_conc.py)



3. Browser Response



4. Malformed Request