

Ricky Zhao
rjzhao@ucsc.edu

For this program, I mostly did script testing, I created a file called shell.sh and made it an executable. Then I ran it to test my httpserver. The following is in my shell.sh

```
Curl -v localhost:8080/qwertyuiopasdfghjklzxcvbnmm >out1& //test GET
Curl -v localhost:8080/thisfiledoesnotexistinheree > out2& //error test file does not exist
Curl -v localhost:8080/badfile > out3 & //test badrequest
Curl -v localhost:8080/thisfiledoesnothavepermissn > out4 & //test forbidden
Curl -v -T qwertyuiopasdfghjklzxcvbnm_ localhost:8080/ >out5& //test PUT
Curl -v -T thisfiledoesnothavepermissn localhost:8080 > out6 //test forbidden
```

I then ran the following on my httpserver

Test 1: ./httpserver localhost 8080
This case test the basic functionality of my server

Test 2: ./httpserver localhost 8080 -N 10
This case test if 10 threads were created

Test 3: ./httpserver localhost 8080 -N 3
This case test if my semaphores work and will the program work with more request than threads

Test 4: ./httpserver -l logfile localhost 8080
This case test the function of getops and if my program logs correctly

Test 5:./httpserver -N 2 localhost -l logfile 8080
This case further test getops. This case also test if the thread and logging functions work while they are together.

Write-up question:

- 1) I ran httpserver and sent 4 GET request to my server from assignment 1. Then, I repeated this on my httpserver after implementing multithreading. My server with multithreading is about 3 times times as fast. Measuring the system time of both, my httpserver with multithreading ran about 1.013 seconds while GETting 4 4Mib files while my httpserver without multithreading ran about 3.873 seconds.
- 2) I think that the bottleneck of my system is dispatcher since we have to wait for the loop to loop back to receiving the buffer. There is concurrency between dispatch and worker since the dispatch can distribute request to the workers and then go back to handling parsing the next request. We can improve on the concurrency between the workers and logging by having separate threads within the worker threads to hand off the job of logging to.

