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 -l logfile
This case test if the logging function is logging the results properly

Test 3: ./httpserver localhost -c 8080 -l logfile
This case test if getops parsing the arguments correctly and if we are logging the cache correctly

I then changed the shell.sh
Curl -v localhost:8080/qwertyuiopasdfghjklzxcvbnmm >out1&
Curl -v localhost:8080/qwertyuiopasdfghjklzxcvbnmm > out2&
Curl -v localhost:8080/qwertyuiopasdfghjklzxcvbnmm > out3&
Curl -v localhost:8080/qwertyuiopasdfghjklzxcvbnmm > out4 &
Curl -v -T qwertyuiopasdfghjklzxcvbnm_ localhost:8080/ >out5&
Curl -v -T qwertyuiopasdfghjklzxcvbnm_ localhost:8080/ >out6&
Curl -v -T qwertyuiopasdfghjklzxcvbnm_ localhost:8080/ >out7&
Curl -v -T qwertyuiopasdfghjklzxcvbnm_ localhost:8080/ >out8

Test 4: ./httpserver localhost -c 8080 -l logfile
This case tests if the caching function is working properly and further test if the logging function logs the cache correctly

Test 5: ./httpserver localhost -c 8080 -l logfile
I change the content of the file qwertyuiopasdfghjklzxcvbnm_ and ran this test again. This is to test the put function and whether or not the PUT function updates the cache.

Write-up question:
To test how my cache improve the performance of my server, I ran the second script with the cache turned off and the cache turned on. With the cache turned off, the program ran about 1.46 seconds. With the cache turned on, the program ran about 3.47 seconds. This result is based on the system time of the server.

I think that the cache improves the latency of the GET function since if the file requested is in the cache, we do not need to read from disk. However I do not think this will improve the performance of the PUT function that much since we are not multithreading. Since we have to ensure that the file sent through the client gets written to disk and we are not doing multithreading, we have to write back to disk for every PUT request we get. Since without multithreading, the server only handles one request at a time.