The testing I used on this assignment was getting the GET requests working first for the cache. And the way I did this was just making sure that the cache was being populated so I would just print out the contents first. Then when I had that working, I tried overpopulating the cache to test my replacement code. After I had GET working, I moved on to PUT and had to implement dirtying the cache. So first I made sure that the dirty vector was putting dirty values where it needed to and would print these values out too. And then I would overpopulate the cache for PUT requests and see if replacement worked and writing back to disk worked as well. And finally, I tested if you had different content for a file that's already in the cache and see if it would replace the content properly.

Using your new httpserver with caching, perform an experiment to demonstrate how caching can improve performance (latency and/or throughput). Do a test with caching turned on and compare it with the same test but with caching turned off.

Caching enabled increases performance as I am able to directly read from a smaller memory source that is faster than reading from the larger but slower disk. When caching was off, I would have to go through my get_file and write_file function code which takes much more time compared to just reading or writing out the data directly.