TESTING

Almost the same as testing for last assignment, adding up concurrency request

Use some script files to test, have been listed in folder.

- Setting Up Stage - arguments
  - `./httpserver -N x(>0) -c y(>0) -f filename localhost:8000` OK
  - `./httpserver -N x(<=0) -c y(<=0) -f (x)localhost:8000` FAILED
  - The rest is the **same** from last assignment
- Transaction Stage - request
  - `./httpclient s:file:aaaa-40 characters-aaaa & ./httpclient r:aaaa-40 characters-aaaa:test` SUCCEED - 200 OK
  - `./httpclient s:file:aaaa-40 characters-aaaa & ./httpclient s:file:aaaa-40 characters-aaaa & ./httpclient r:aaaa-40 characters-aaaa:test` SUCCEED - 200 OK
  - `./httpclient s:file:aaaa-40 characters-aaaa r:aaaa-40 characters-aaaa:test & ./httpclient s:file:aaaa-40 characters-aaaa & ./httpclient r:aaaa-40 characters-aaaa:test` SUCCEED - 200 OK
  - The rest is the **same** from last assignment
- File Transportation - file
  - 0 B file SUCCEED - 200 OK
  - 30 B file SUCCEED - 200 OK
  - 4 KB file SUCCEED - 200 OK
  - 4 MB file SUCCEED - 200 OK
- Other situations
  - Sending wrong request to both client and server SUCCEED - 200 OK
  - Work with `SimpleHTTPserver` SUCCEED - 200 OK
  - Work with `curl` SUCCEED - 200 OK
  - Work with web browser SUCCEED - 200 OK
  - Work with others' server and client SUCCEED - 200 OK
  - The rest is the **same** from last assignment

QUESTION: Why use size_t (64 bits) for parameters, and then only support 32 bit object lengths internally?

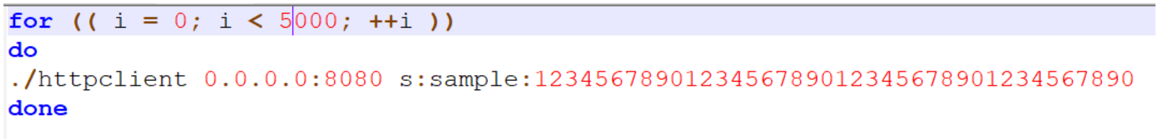
Even though only 32 bits of 64 bits is used in size_t in my program. In my opinion, in order to keep the consistency in the system for other functions to use, 64 bits is much safer to use than only 32 bits. Since we are not sure which functions may use 64 bits rather than 32 bits, it's safe at least for every functions to execute.

QUESTION: Test-750,000 small objects

Didn't have time to test…………………..

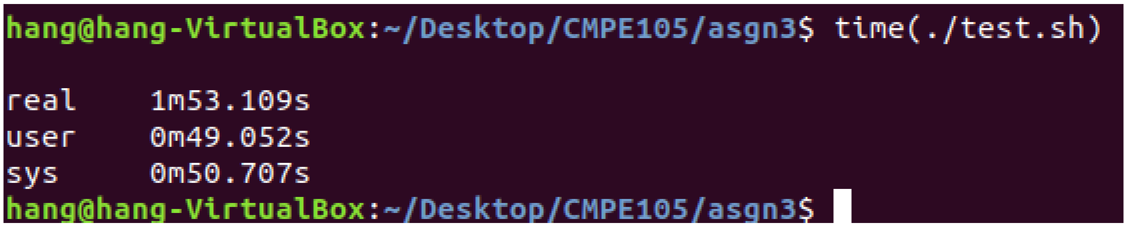
Test writing 100 B file

Script file



```
for (( i = 0; i < 5000; ++i ))
do
./httpclient 0.0.0.0:8080 s:sample:1234567890123456789012345678901234567890
done
```

In ASGN 3



```
hang@hang-VirtualBox:~/Desktop/CMPE105/asgn3$ time(./test.sh)

real	1m53.109s
user	0m49.052s
sys	0m50.707s
hang@hang-VirtualBox:~/Desktop/CMPE105/asgn3$ 
```

In ASGN 2

Takse about 10 minutes to do this

Because in ASGN 2 we open then write to file takes too much time. For each thread which wants to write, we have to lock it until the file is free. And after one thread, we close the fd. In this process, we waste too much time on it.

However, this time, I adopt cache and use only one file. This can save much my time instead of access file again and again.