**HomeWork#2**

20191128 Jian Park

**Environment :**

|  |  |  |
| --- | --- | --- |
| **Hardware** | **OS** | **Used Libraries** |
| MacBook Pro (16-inch, 2019)  2.3GHz 8-core Intel Core i9  16GB 2667 MHz DDR4  Intel UHD Graphics 630 1536MB | macOS Big Sur (11.6) | socket, sys, thread in python3 |

**Compilation commands:**

I’m used python. So pleased command ‘python3 server.py proxy <port number>’

ex) python3 server.py proxy 8001

If port is already using, it will be an exception. Please retry

When you done, please enter ‘^C’ or another key.

**Implementation details:**

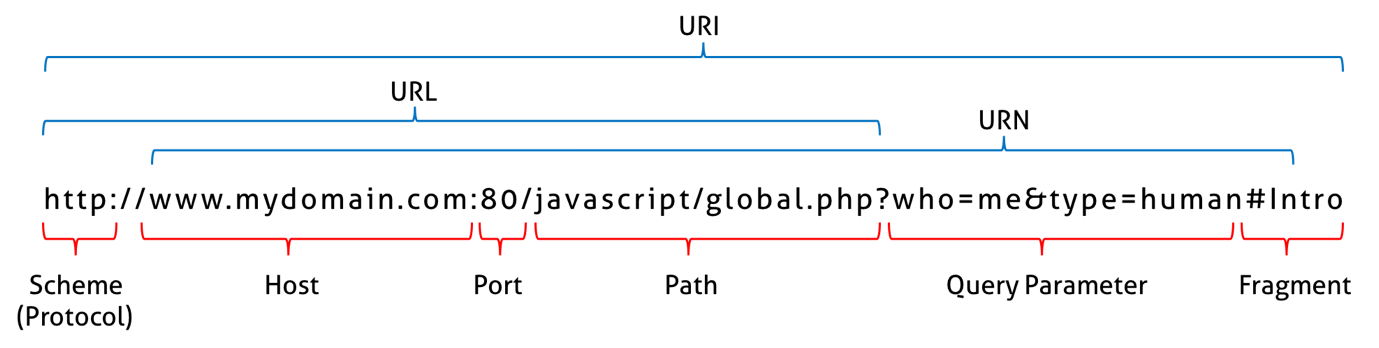
I made 5 functions. The detail of each function is as follows. And please refer to the comments in code.

**main ()**

It is main function. Server initialization (make socket, bind, listen). If port is already using, it will be an exception.

And Client initialization while keyboard interrupt.

**parsing\_request (client\_sock, read\_bytes)**



This function is parsing client request message and determine whether to request to server or not.

I want to export host, port in client request message. So it split protocol first, and split ‘:’ and ‘/’.

And add host, connection option. If connection option is not close, change it.

It parsing is done, check if it cached.

Is valid(HIT), send response to client in cache. Else request to server.

**check\_cache (url)**

It is check is in cache. If file that named url is exist, it cached. Return response message in the cache.

Else retune None

**request\_server(url, domain, port, client\_sock, read\_bytes)**

Reques to sever. Make socket to connection origin server. Receive response as much as the buffer size and send client. When received all message. save in the cache.

**save\_cache(url, server\_Read\_bytes)**

Save response message in cache. Make file that named url, and write message.

**Result:**

Same request “GET http://www.gnu.org/ HTTP/1.0”

텍스트이(가) 표시된 사진

자동 생성된 설명

figure 1. First it MISS, next HIT

텍스트이(가) 표시된 사진

자동 생성된 설명

figure 2. MISS

텍스트이(가) 표시된 사진

자동 생성된 설명

figure 3. HIT

Response message is same



After first request, cache file has been created.