

# Network Assignment 3

2014313303

홍태하

개발환경 : python 3.7.0, chrome에서 test 진행.

## 1. Code 설명

```
from socket import *
import time
import os
from threading import Thread

serverIP = ''
serverPort = 10080
ADDR = (serverIP, serverPort)
chunk_size = 1024 * 4096

serverSocket = socket(AF_INET, SOCK_STREAM)

serverSocket.bind(ADDR)
serverSocket.listen(10)
print('The TCP server is ready to receive.')
```

Ip와 port번호로 address를 만들고 serverSocket을 연결한다.

```
def connect():
    connectionSocket, addr = serverSocket.accept()

|   t = Thread(target = connect)
    t.start()

    login_flag = 0
    cookie_flag = 0
    msg = connectionSocket.recv(chunk_size).decode()
    #print(msg)
    file_split = msg.split('.')
    file_name = file_split[1]

    file_content = msg.split('\r\n')
    for i, data in enumerate(file_content):
        if data.count('Cookie:') > 0:
            cookie_str = data.split('.')
            ID = cookie_str[1].split('=')[1]
            ID = ID[:-1]
            passwd = cookie_str[2].split('=')[1]
            passwd = passwd[:-1]
            cookie_flag = 1
            break

    if file_name == '/' and file_split[0] == 'GET':
        with open('index.html', 'rb') as f:
            file_size = os.path.getsize('index.html')
            connectionSocket.send('HTTP/1.1 200 OK\r\nContent-Type: text/html\r\nContent-Length:{}'.format(file_size).encode())
            while True:
                buffer = f.read(chunk_size)
                if not buffer: break
                connectionSocket.send(buffer)
```

브라우저에서 연결이 올 때마다 새로운 socket을 연다. 소켓을 열고 thread를 이용해서 또 다른 요청이 왔을 때 받을 수 있도록 한다. 브라우저에서 온 http메세지를 msg변수를 통해서 받는다.

먼저 http메시지를 분석해서 cookie가 있는지 없는지 파악하고 있다면 cookie에 id와 password정보가 있으므로 추출해서 변수에 저장한다.

[http://server\\_IP:port\\_number/](http://server_IP:port_number/) 로 접근했을 때 index.html 을 보내줘서 브라우저에 띄운다. Index.html은 로그인 창이다.

```
elif cookie_flag == 1 or file_split[0] == 'POST':
    if file_split[0] == 'POST':
        for i, data in enumerate(file_content):
            if data.count('Content-Disposition:')>0:
                if data[-3]=='\r' and data[-2]=='D':
                    id_str = file_content[i+2]
                else:
                    passwd_str = file_content[i+2]
        with open('secret.html', 'rb') as f:
            file_size = os.path.getsize('secret.html')
            connectionSocket.send('HTTP/1.1 200 OK\r\nSet-Cookie: id_cookie={}; max-age=30\r\nSet-Cookie: passwd_cookie={}; max-age=30\r\nContent-Type: text/html\r\n')
            while True:
                buffer = f.read(chunk_size)
                if not buffer: break
            connectionSocket.send(buffer)

    elif file_split[0] == 'GET':
        file_name = file_name.split('/')[1]
        file_format = file_name.split('.')[-1]

        if file_name == 'cookie.html':
            if cookie_flag == 1:
                connectionSocket.send(b'HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n')
                connectionSocket.send((b'<html><body><h1>Hello '+id_str+b'</h1><br><br><h1>seconds left until your cookie expires.</h1></body></html>').encode())
            else:
                connectionSocket.send(b'HTTP/1.1 403 Forbidden\r\nContent-Type: text/html\r\n')
                connectionSocket.send(b'<html><body><h1>403 Forbidden</h1></body></html>')
        elif not os.path.isfile(file_name):
            connectionSocket.send(b'HTTP/1.1 404 Not Found\r\nContent-Type: text/html\r\n')
            connectionSocket.send(b'<html><body><h1>404 Not Found</h1></body></html>')
        else:
            with open(file_name, 'rb') as f:
                file_size = os.path.getsize(file_name)

            connectionSocket.send('HTTP/1.1 200 OK\r\nContent-Length:{}'.format(file_size).encode())
            while True:
                buffer = f.read(chunk_size)
                if not buffer: break
            connectionSocket.send(buffer)

    else:
        connectionSocket.send(b'HTTP/1.1 403 Forbidden\r\nContent-Type: text/html\r\n')
        connectionSocket.send(b'<html><body><h1>403 Forbidden</h1></body></html>')

connectionSocket.close()
```

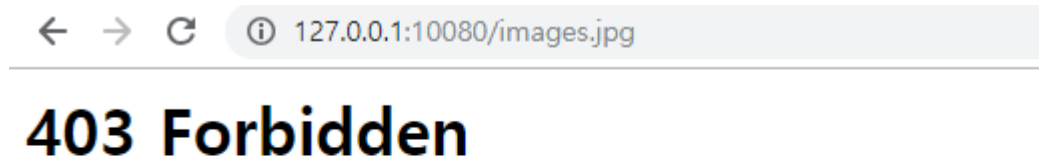
Cookie\_flag는 cookie가 존재하는지 알려주는 flag인데 cookie가 존재한다면 login된 상태이다. Login이 된 상태라면 이미지 파일이나 동영상, pdf 파일을 출력해주는 작업을 할 수 있다. 이 경우 200OK 메시지와 함께 보낼 파일의 크기와 파일을 chunk\_size만큼씩 끊어서 보내게 된다. (1024 \* 4096) 물론 없는 파일을 요청할 경우 404 Not Found 메시지를 보낸다. 그리고 cookie.html을 요청했을 때는 Hello 와 ID를 띄워주고 cookie가 남아 있는 시간을 출력해주어야 하는데 시간은 출력하지 못했다.

또는 cookie가 존재하지 않더라도(login이 완료되지 않았더라도) login 창에서 login버튼을 눌렀을 때 ID, password 정보가 POST method로 server로 가게 된다. POST가 오면 http 메시지를 parsing해서 ID와 password 정보를 얻은 후 secret.html을 보내는데 이때 Set-Cookie를 이용해서 id, password정보도 같이 보낸다. 또 max-age=30으로 설정하여서 30초가 지나면 쿠키가 사라지도록 한다.

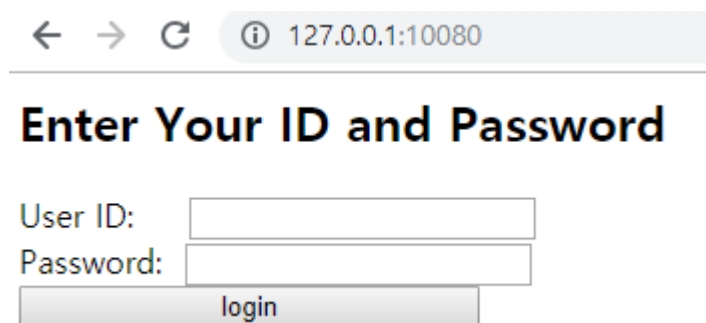
로그인이 되지 않은 상태이고 POST 메시지를 받는 상황도 아니라면 403 Forbidden 메시지를 띄운다.

이 모든 작업들을 하고 난 후 Socket 연결을 끊는다.

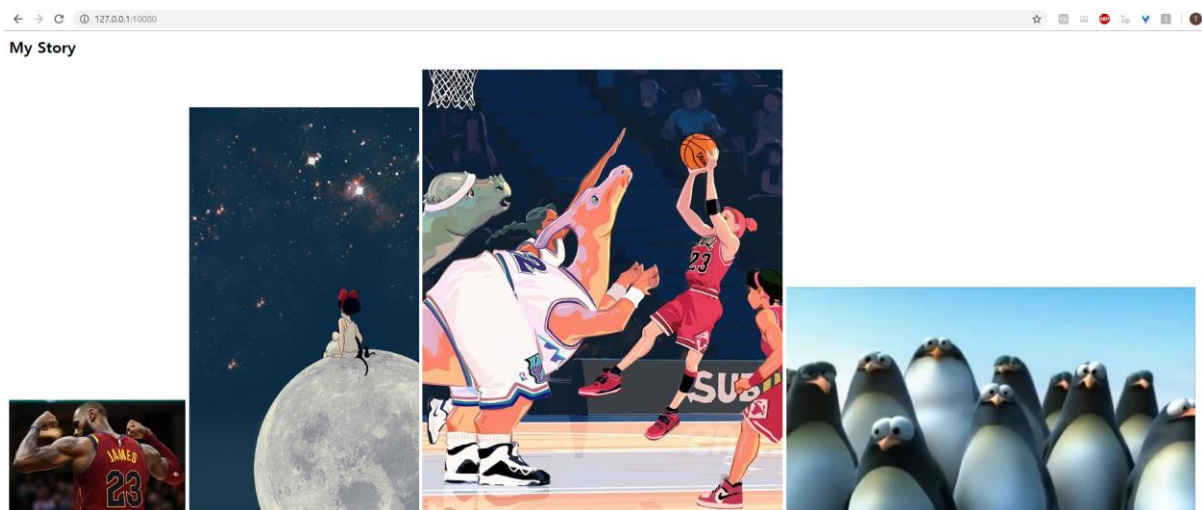
## 2. 실행 결과



로그인이 되기 전 파일을 요청하면 403 Forbidden을 띄운다.



Ip와 port number만 입력하면 로그인창을 띄운다. (index.html)



로그인을 하면 secret.html을 띄운다.

← → ↻ ⓘ 127.0.0.1:10080/cookie.html

# Hello taeha

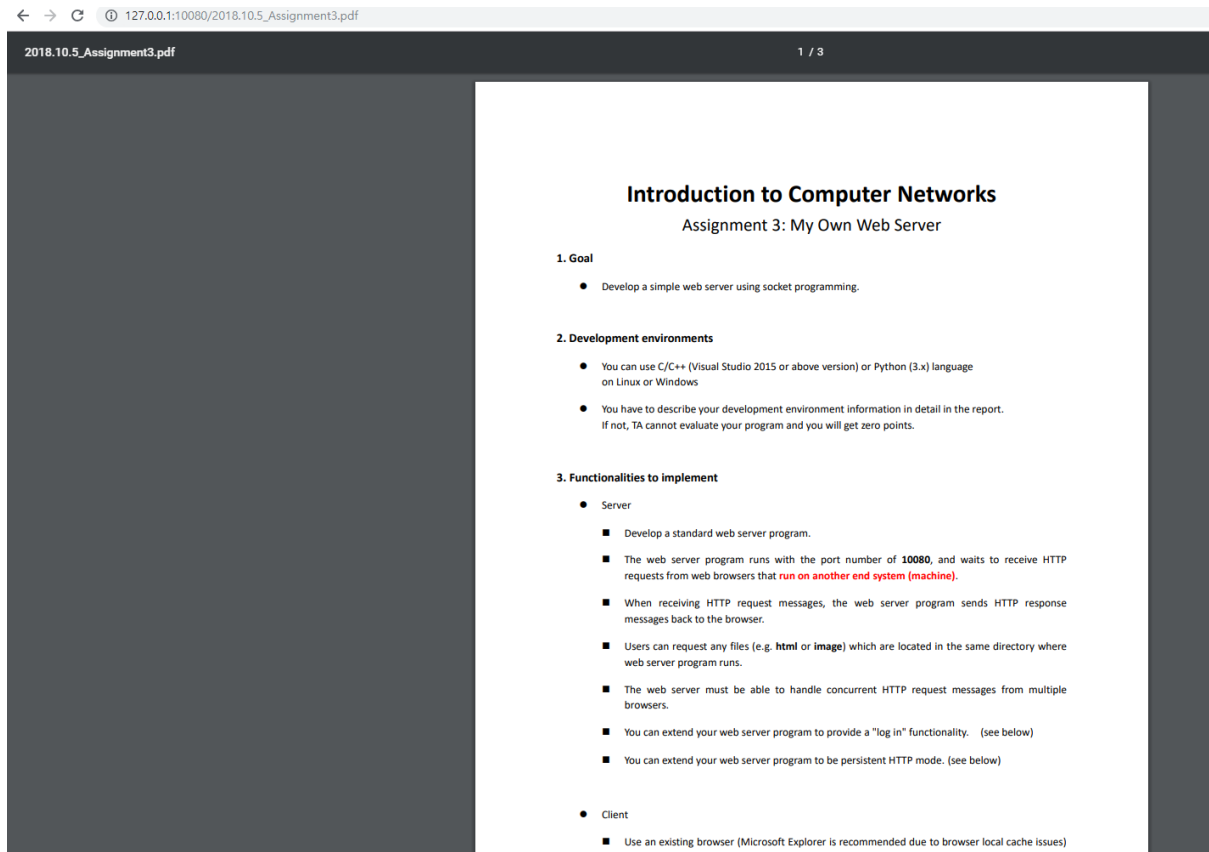
## seconds left until your cookie expires.

쿠키가 사라지기 전 cookie.html을 요청하면 위와 같은 메시지를 띄운다.

← → ↻ ⓘ 127.0.0.1:10080/images.jpg



쿠키가 사라지기 전에 파일을 요청하면 띄워준다.



pdf파일도 띄운다.



## 403 Forbidden

로그인을 하고 30초가 지나서 쿠키가 사라지면 403 Forbidden을 출력한다.



## 404 Not Found

로그인은 되어 있지만 존재하지 않는 파일을 요청하면 404 Not Found를 출력한다.