

컴퓨터네트워크

HW #2 (30 Points)

Due date : 2023/5/2 (eCampus)

제출물 : hw2_학번.c, hw 2_학번.cpp, hw 2_학번.java, hw 2_학번.py (하나의 파일에 모든 것을 다 구현함) 중 하나.

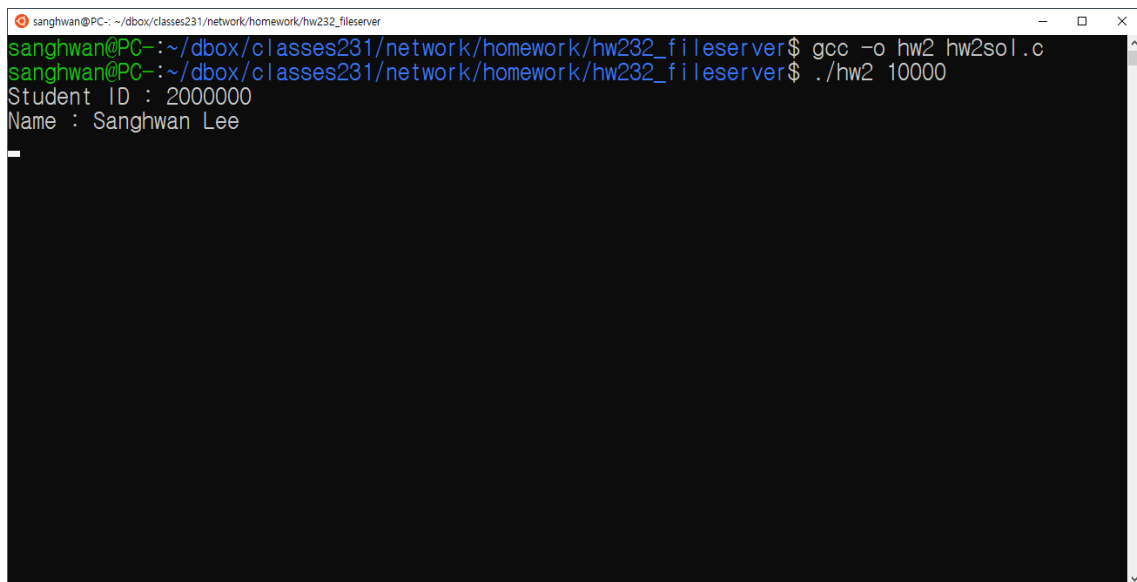
절대로 .zip 파일을 제출하면 안됨.

이 과제는 간단한 파일서버를 구현하는 것이다. 이 서버를 이용하여 사용자는 3가지 서비스를 수행할 수 있다.

- 특정 확장자를 가진 파일 목록 다운로드
- 주어진 이름의 파일 다운로드
- 파일 업로드

아래는 실행 예이다. 소스를 컴파일해서 hw2라는 실행파일을 만들어 냈다.

그리고 파일 서버 프로그램인 “hw2”를 실행하는데, 파라미터로 포트번호를 제공한다. 이 예에서는 포트번호가 10000이다. 프로그램은 학번과 이름을 출력한다.



```
sanghwan@PC: ~/dbox/classes231/network/homework/hw232_filesaver
sanghwan@PC:~:/dbox/classes231/network/homework/hw232_filesaver$ gcc -o hw2 hw2sol.c
sanghwan@PC:~:/dbox/classes231/network/homework/hw232_filesaver$ ./hw2 10000
Student ID : 2000000
Name : Sanghwan Lee
```

기능 설명:

- 파일서버가 실행되면 주어진 포트 번호로 연결이 들어오길 기다린다.
- 클라이언트가 tcp 연결을 요청하면 받아들인다.

- Tcp연결이 설정된 후에, 클라이언트는 다음 3가지 명령을 보낼 수 있다. 이 명령은 모두 한 줄로 이루어진다. 즉 서버는 클라이언트가 보내는 첫번째 라인을 읽어서 명령을 해석해야 한다.

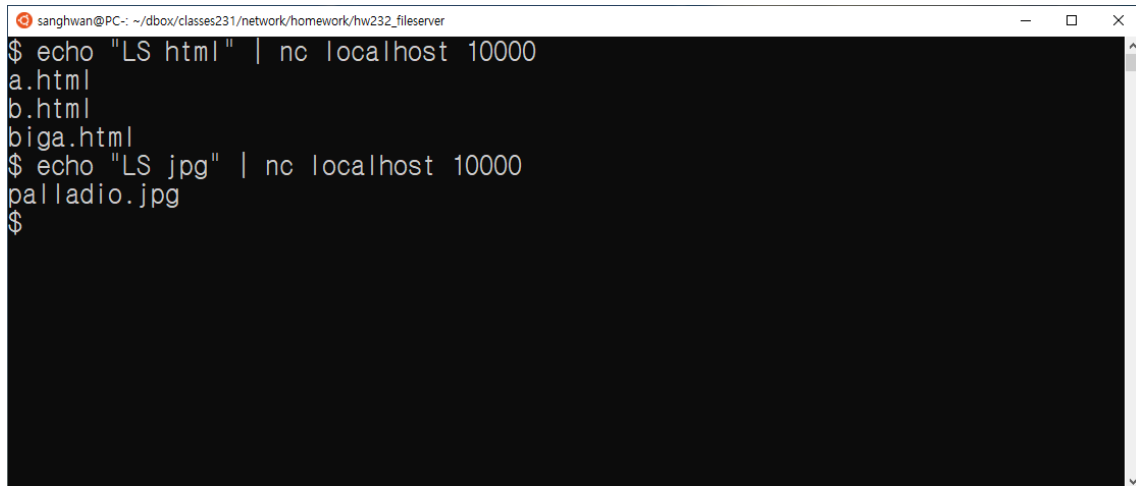
- 1) GET filename: 이 명령이 들어오면 서버는 filename에 해당하는 파일을 찾아서 tcp 연결을 통해 전송한다. 만약 존재하지 않으면 "FILE NOT FOUND"을 클라이언트로 전송한다.

```
sanghwan@PC: ~/dbbox/classes231/network/homework/hw232_fileserv
$ ls *.htm
ls: cannot access '*.htm': No such file or directory
$ echo "GET biga.html" | nc localhost 10000 > a.htm
$ ls a.htm
a.htm
$ diff biga.html a.htm
$ echo "GET kkk.html" | nc localhost 10000 > kkk.htm
$ cat kkk.htm
FILE NOT FOUND
$
```

- 2) PUT filename: 이 명령이 들어오면 서버는 filename 이라는 파일을 현재 디렉토리에 생성한 후, 이 tcp 연결을 통해 명령 라인 이후에 들어오는 모든 내용을 이 filename 파일에 쓴다.

```
sanghwan@PC: ~/dbbox/classes231/network/homework/hw232_fileserv
$ ls *.html
a.html biga.html
$ echo "PUT b.html" | cat - biga.html | nc localhost 10000
^C
$ ls *.html
a.html b.html biga.html
$ diff biga.html b.html
$
```

- 3) LS ext: 이 명령이 들어오면 서버는 현재 디렉토리에서 파일 확장자가 ext인 파일을 찾아서 tcp 연결을 통해 전송한다. 전송시에는 한 라인에 한 파일명을 전송한다. 구체적으로는 라인 구분을 “WrWn”으로 해서 각 파일에 대해서 “파일명WrWn”의 형태로 전송하면 된다.

A terminal window titled 'sanghwan@PC: ~/dbox/classes231/network/homework/hw232_fileserver'. It shows two commands being executed. The first command is '\$ echo "LS html" | nc localhost 10000', which results in the output 'a.html', 'b.html', and 'biga.html'. The second command is '\$ echo "LS jpg" | nc localhost 10000', which results in the output 'palladio.jpg'. The prompt '\$' is visible at the end of each command line.

```
sanghwan@PC: ~/dbox/classes231/network/homework/hw232_fileserver
$ echo "LS html" | nc localhost 10000
a.html
b.html
biga.html
$ echo "LS jpg" | nc localhost 10000
palladio.jpg
$
```

참고. 혹시 연결이 안될 경우 방화벽에서 사용하는 포트 번호를 허용하도록 변경하세요.

참고.

자동화테스트를 위하여

첨부한 labtest.sh, .html, .jpg 파일을 소스코드가 있는 곳에 복사한다.

그리고 구현한 파일 확장자에 따라 다음 명령 중 하나를 실행. 아래 그림 참조.

```
$ bash labtest.sh hw2sol.c 20000
```

```
$ bash labtest.sh hw2sol.java 20000
```

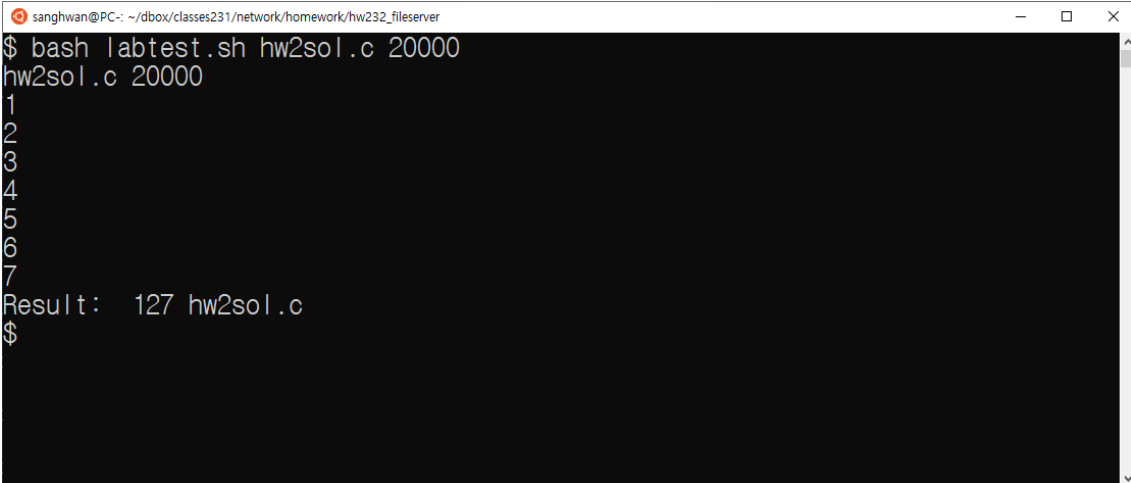
```
$ bash labtest.sh hw2sol.py 20000
```

```
$ bash labtest.sh hw2sol.cpp 20000
```

최종 라인에 “Result: 127”이 나와야 한다.

이 labtest.sh는 Ubuntu linux와 MacOS 에서 동작하도록 구현되어 있다.

참고로 이 자동화테스트는 **실행파일을** 강제로 죽이기 때문에 **stdout으로 출력하는 내용이 있으면 flush 를 시켜야 정상적으로 출력 내용이 출력이 될 수도 있음.**



```
sanghwan@PC: ~/dbbox/classes231/network/homework/hw232_fileserver
$ bash labtest.sh hw2sol.c 20000
hw2sol.c 20000
1
2
3
4
5
6
7
Result: 127 hw2sol.c
$
```