

Computer network Project1(SOCKET)

소프트웨어학부 2018044384 김서권

1. 실행

```
gimseogwon@gimseogwon-ui-MacBookPro project1_2018044384_Kim_Seogwon % make
gcc -c -o server.o server.c
server.c:57:66: warning: passing 'int *' to parameter of type 'socklen_t *' (aka 'unsigned int
 *') converts between pointers to integer types with different sign [-Wpointer-sign]
    client = accept(server, (struct sockaddr*) &client_addr, &addr_len);
                                   ^
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/sys/socket.h:700:73: note: pas
sing argument to parameter here
int      accept(int, struct sockaddr * __restrict, socklen_t * __restrict)
                                   ^
1 warning generated.
gcc -o server_exe server.o
gimseogwon@gimseogwon-ui-MacBookPro project1_2018044384_Kim_Seogwon % ls
Makefile      rsc           server.c      server.o      server_exe
gimseogwon@gimseogwon-ui-MacBookPro project1_2018044384_Kim_Seogwon %
```

Makefile을 이용하여 컴파일 한 모습

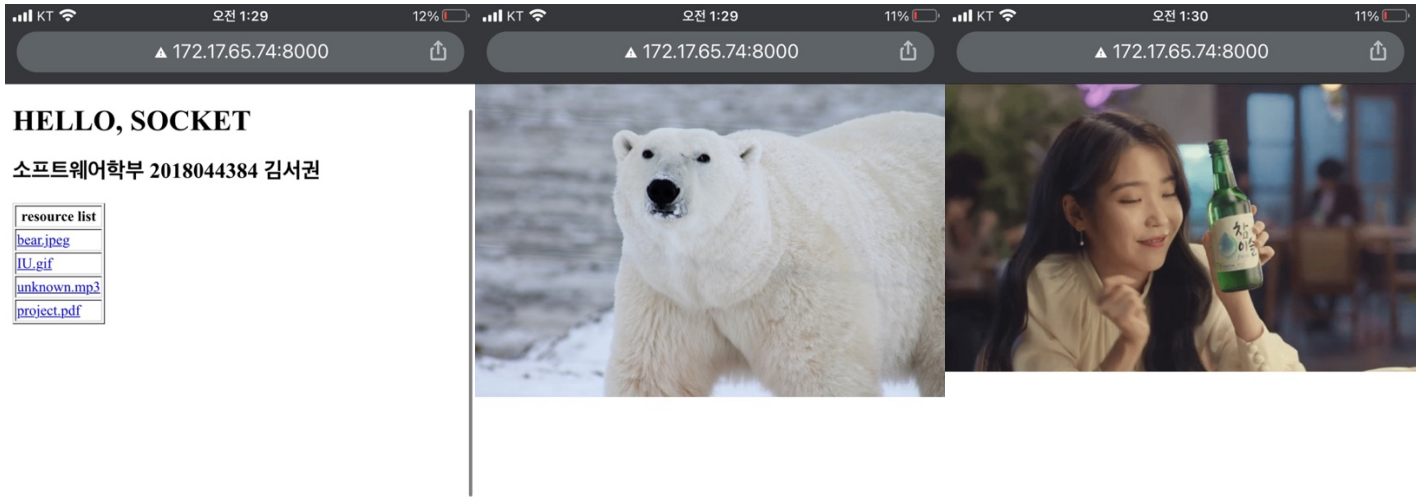
```
gimseogwon@gimseogwon-ui-MacBookPro project1_2018044384_Kim_Seogwon % ./server_exe
unavailable port number%
gimseogwon@gimseogwon-ui-MacBookPro project1_2018044384_Kim_Seogwon % ./server_exe 8000
server open successfully.
accepting request...
```

올바른 포트번호로 실행해야 작동됩니다.

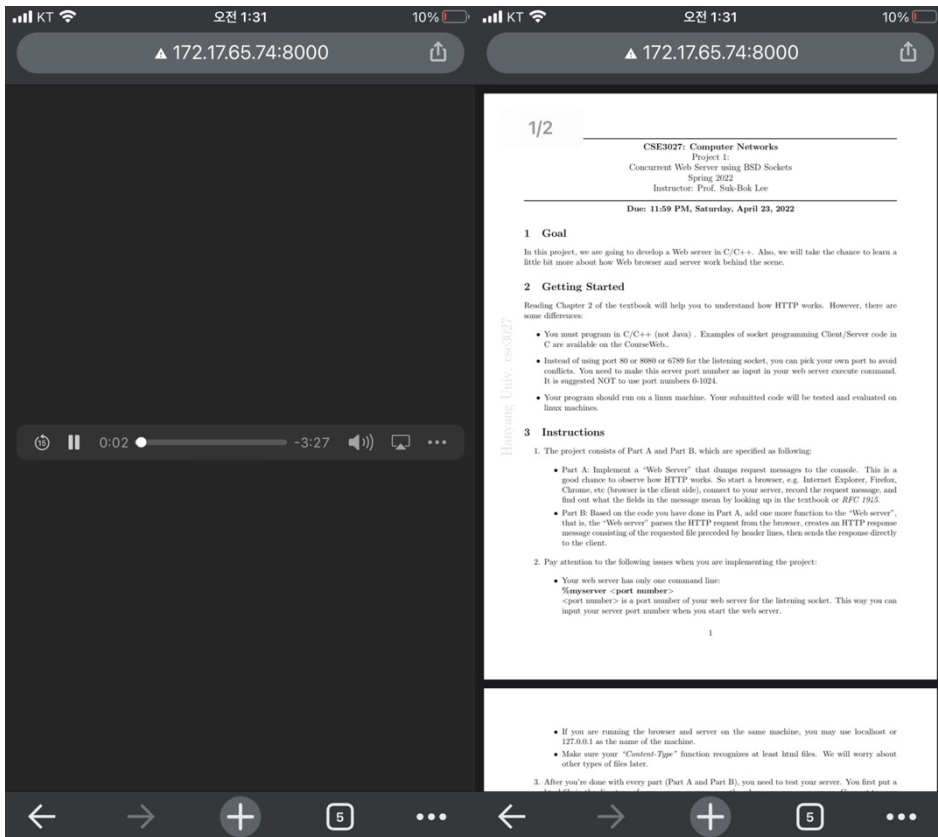
```
accepting request...
HIT FILE
path : /Users/gimseogwon/CLionProjects/Socket_server/project1_2018044384_KimSeog
won/rsc/IU.gif
open
type : image/GIF
=====send success=====
accepting request...
HIT FILE
path : /Users/gimseogwon/CLionProjects/Socket_server/project1_2018044384_KimSeog
won/rsc/bear.jpeg
open
type : image/jpeg
=====send success=====
^C
=====ending server program.=====
client socket closes successfully.
server socket closes succussfully.
=====success to end server program.=====
gimseogwon@gimseogwon-ui-MacBookPro project1_2018044384_KimSeogwon %
```

각 request에 대한 log를 이러한 형식으로 표현했고,

signal처리를 통해 자원해제이슈를 해결했습니다.



해당 wifi아이피주소와 포트번호로 들어간 모습. 각각 jpeg, gif, mp3, pdf가 문제 없이 실행됩니다.



2. 느낀점, 어려웠던 점

1. 클라이언트 소켓과 포크

처음에 공부를 하기 위해 많은 사이트를 찾아보았는데 대부분의 사람들이 포크를 하여 자식프로세스가 클라이언트를 처리하는 것을 보고 서버의 안정성을 위해서 이게 적합하다 생각하여 해보았습니다. 그랬더니 accept가 일어나는 위치가 잘못되어 포크가 무한히 되는 등 많은 착오를 겪고 성공적으로 fork를 할 수 있게 되었습니다. 그후에도 성공적으로 되는 줄 알았더니 포크된 자식프로세스가 not found에러가 발생할 경우 정상적인 종료가 되지 않는 것을 확인하여, 기존 부모가 해주던 waitpid함수의 세번째 인자를 WNOHANG에서 0으로 바꿈으로써 해당이슈를 해결했습니다.

2. Response

처음에는 헤더를 보내지 않고 파일만 보냈더니 type정의를 내려주지 않아 웹브라우저에서 파일이 보이지 않았습니다. 처음에는 왜 그런지 몰라서 고난을 겪던 중에 같은 수업을 듣는 동기가 헤더에 컨텐츠타입과 사이즈를 같이 보내줘야 한다고 조언을 해주어서 열심히 코딩, 디버깅을 하면서 성공적으로 과제에서 필요로 하는 모든 파일을 웹에서 보이게 코드를 완성 할 수 있었습니다.

3. 느낀점

이번 과제가 처음에는 간단할 거라 생각했는데 하다 보니 생각보다 보고서에 쓴 내용 외로도 어려운 점이 꽤 있었습니다. 하지만 인터넷과 강의 pdf의 http부분을 정독하고 코드를 완성하면서 c에 대한 공부도 하고 더불어 http의 통신 방법에 대해서 이해할 수 있게 되는 좋은 기회가 되었습니다.