

# REPORT

[시스템프로그래밍  
Assignment 2-3]



학과	컴퓨터정보공학부
----	----------

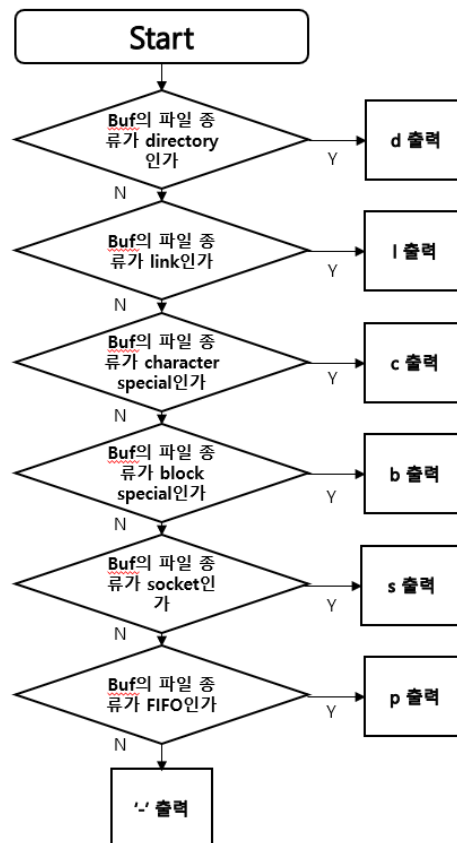
학번	2018202018
이름	유승재
담당 교수	김태석 교수님
제출일	2023.05.10

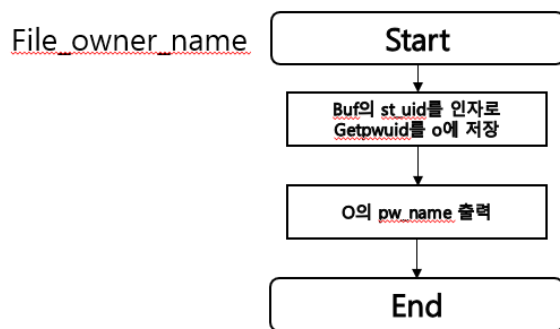
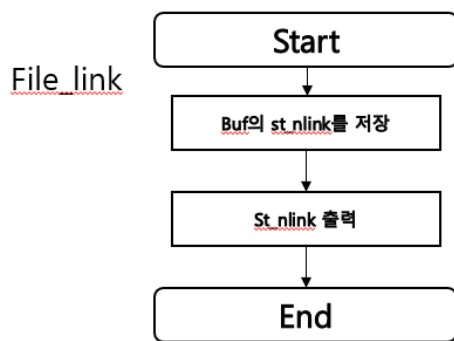
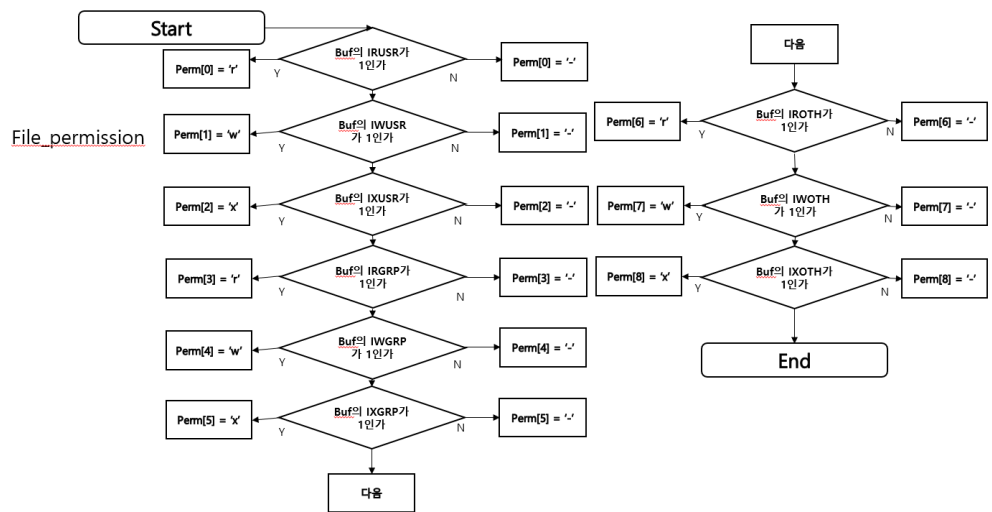
## 1. Introduction

이번 과제는 저번 주의 web\_server 를 활용하여 여러 client 에서 다중 접속이 가능한 web\_server 를 구현하는 것입니다. 이번 과제에서는 서버에 access 가능한 ip 를 설정하여 허가되지 않은 ip 에서 접속할 시에 새로운 response message 를 출력하는 기능과 fork 를 이용하여 기존의 기능을 child process 에서 구현하게 되고, parent process 에서는 access 된 기록을 전부 받아서 정해진 시간마다 출력해주도록 하는 기능 두 가지를 추가로 구현해주어야 합니다.

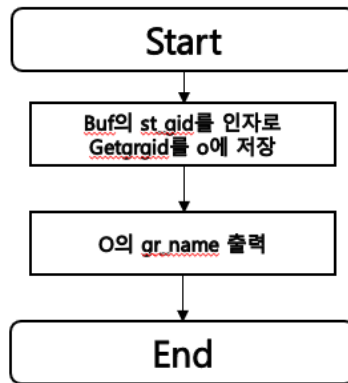
## 2. Flow Chart

File\_check

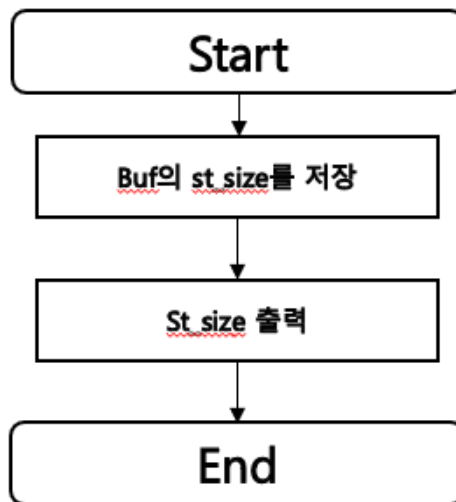




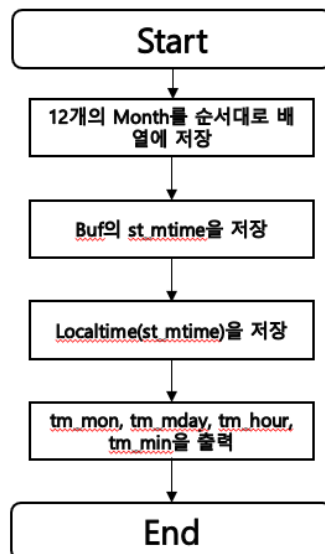
File\_group\_name

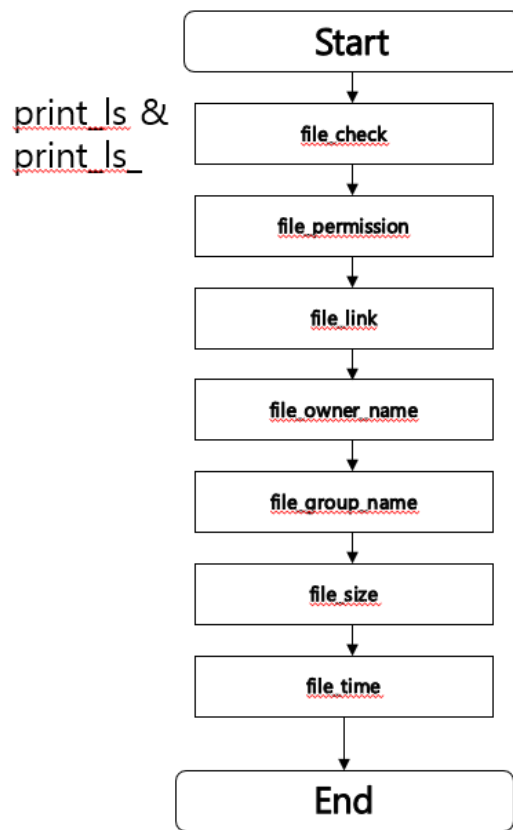


File\_size

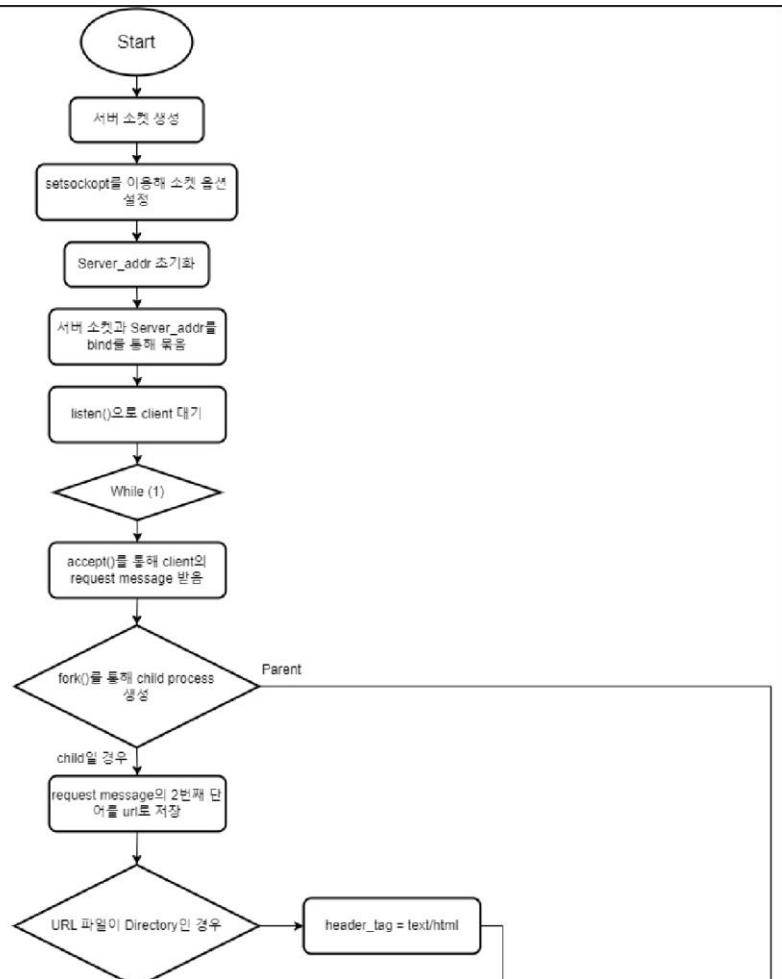


File\_time

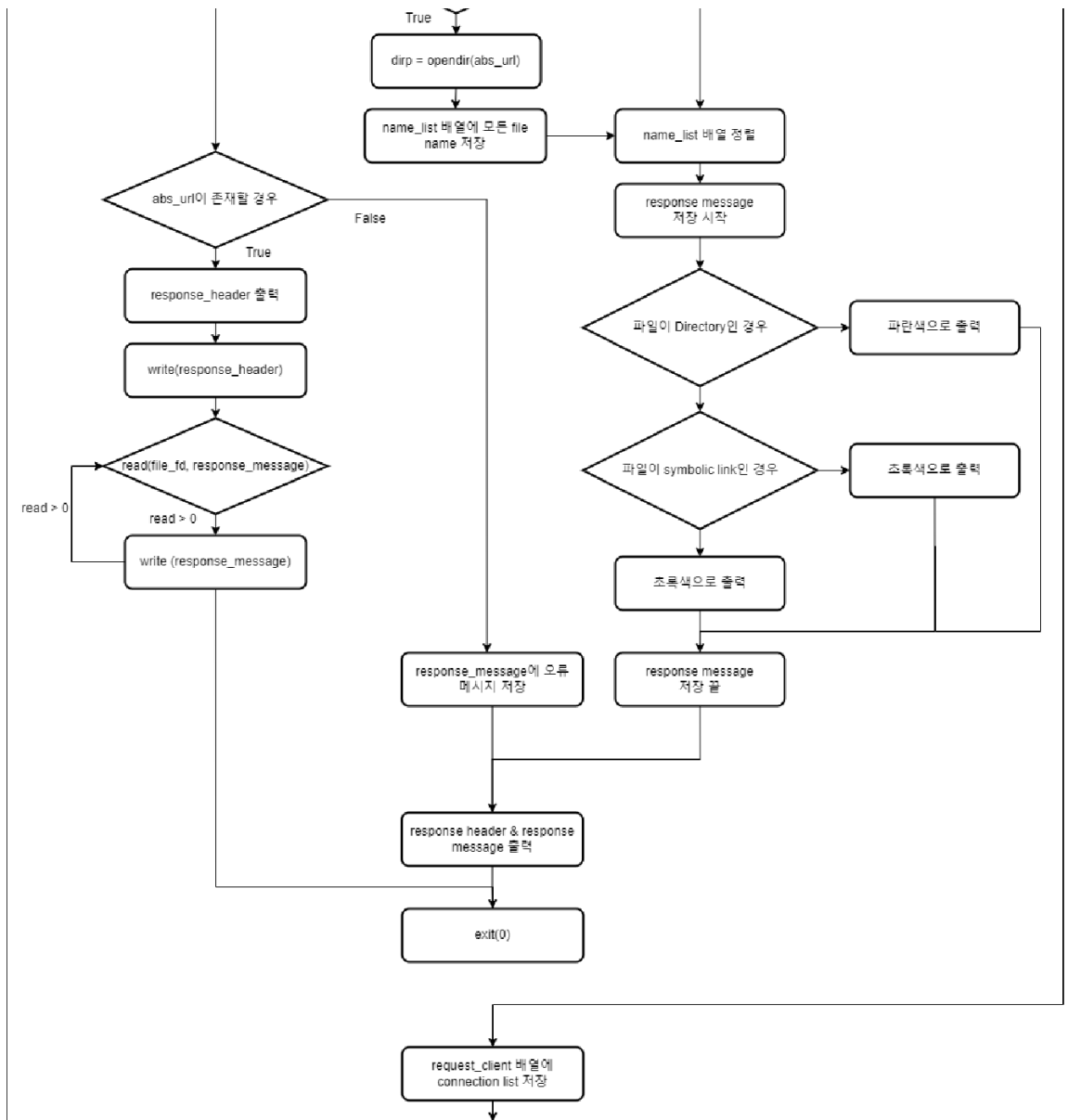




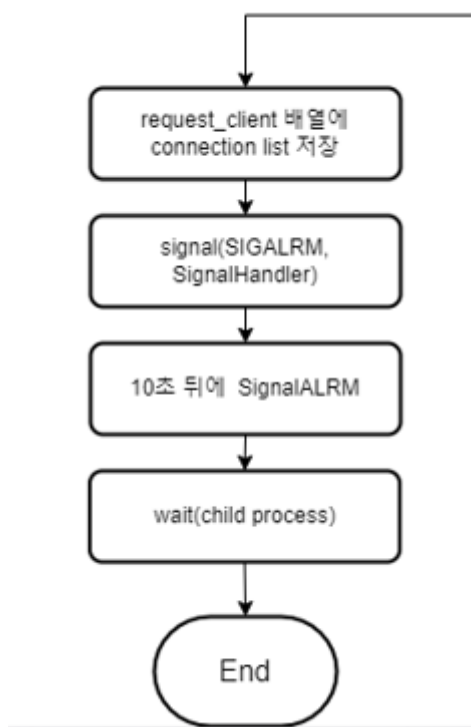
Main 함수











### 3. Pseudo Code

```
void file_check(struct stat ){  
    char 선언;  
    table 행 시작;  
    if (디렉토리){  
        d 출력
```

```

    }

    else if (링크){

        l 출력

    }

    else if (character special){

        c 출력

    }

    else if (block special){

        b 출력

    }

    else if (socket){

        s 출력

    }

    else if (FIFO){

        p 출력

    }

    else if (regular){

        - 출력

    }

}

void file_permission(struct stat ){

```

Perm[10];

Stat 의 st\_mode 를 가져와 권한에 맞게 perm 의 자리에 char 배치

Perm 출력

}

void file\_link(struct stat ){

Stat 의 st\_nlink 를 가져와 link 개수 확인

Link 개수 출력

}

void file\_owner\_name(struct stat ) {

Stat 의 getpwuid 로 st\_uid 를 가져와 struct passwd 형태에 저장

Struct passwd 의 pw\_name 출력

}

void file\_group\_name(struct stat ){

Stat 의 getgrgid 로 st\_gid 를 가져와 struct group 형태에 저장

Struct group 의 gr\_name 출력

}

void file\_size(struct stat ){

Off\_t 변수 0 으로 초기화;

변수에 st\_size 로 대입;

St\_size 출력

}

```
void file_time(struct stat ){
```

12 개의 month 를 배열 크기 12 개의 문자열 배열에 1 월부터 12 월까지 저장;

Struct stat 의 st\_mtime 을 localtime 함수를 이용해 struct tm 에 저장;

Struct tm 형식의 tm\_mon, tm\_mday, tm\_hour, tm\_min 출력

Table 행 끝

```
}
```

```
void print_ls(struct stat ){
```

File\_check, file\_permission, file\_link, file\_owner\_name, file\_group\_name,

File\_size, file\_time 한번에 struct stat 을 넣어줘 출력

```
}
```

```
void SignalHandler(int Sig) {
```

월 배열 생성;

요일 배열 생성;

if (NO 가 10 보다 작을 경우) {

No 만큼 출력;

```
}
```

else {

No - 10 부터 No 까지 출력

```
}
```

alarm(10);

```
}
```

```

int main(){

    Socket_fd 생성;

    Setsockopt 로 socket_fd 설정;

    Struct sockaddr_in Server_addr 초기화;

    Server_addr 과 socket_fd 바인딩;

    Listen(socket_fd);

    Client_fd = Accept(socket_fd);

    read(client_fd 를 읽을 경우 잘못 읽었을 때){

        while 문 continue;

    }

    자식 프로세스 생성;

    while (1){

        if (자식 프로세스일 경우){

            Client_fd 를 read 해서 request message 출력;

            Request message 에서 url 확인;

            url 이 directory 일 경우

                header_tag = text/html;

            url 이 이미지파일 일 경우

                header_tag = image/*

            그 외일 경우

                Header_tag = text/plain

```

```

if (url 이 "/" (cwd)일 경우){

    cwd 의 non-hidden 파일들 배열에 저장;

    파일 배열 정렬;

    파일배열 출력 형식을 HTML_Is 형식에 맞추어
response_message 에 저장;

}

else if (url 이 cwd 가 아닌 경우) {

    if (url 이 directory 인 경우){

        url 에 위치한 파일들 전부 배열에 저장;

        파일 배열 정렬;

        파일 배열 출력 형식을 HTML_Is 형식에 맞추어
response_message 에 저장;

    }

    else if (url 이 파일인 경우){

        url 의 파일 정보 전부 read;

        read 한 데이터 전부 response_message 에 저장;

        response_header client_fd 에 출력;

    }

    response_message client_fd 에 출력;

    else (존재하지 않는 파일인 경우){

        404 Error 양식에 맞추어 출력;

```

```

        url 이 directory 이거나 image 파일 인 경우

        response_header 출력;

        이전에 저장한 response_message 출력;

    }

}

}

else (부모 프로세스일 경우) {

    if (url 이 /favicon.ico 일 경우) (favicon 예외 처리) {

        continue;

    }

    if (IP 가 허가되지 않은 IP 일 경우) {

        continue;

    }

    if (abs_url 이 존재하지 않는 경로일 경우(404 Error)) {

        continue;

    }

    if (이외의 경우) {

        struct Point 에 현재 access 한 client 의 정보들 저장;

        이를 struct Point 배열에 저장;

        Signal Handler 로 출력;

```

```

    }

}

}

}

```

#### 4. 결과화면

The screenshot shows a web browser at 127.0.0.1:40000 displaying a directory listing for /home/kw2018202018/ls\_a. The directory contains 16520 bytes of files. A terminal window in the foreground shows the server logs for the request.

Name	Permission Link	Owner	Group	Size
2018202018_advanced_js.c	-rw-rw-r--	1	kw2018202018	24125
2018202018_final_js.c	-rw-rw-r--	1	kw2018202018	50672
2018202018_html_js.c	-rw-rw-r--	1	kw2018202018	61294
2018202018_spls_advanced.c	-rw-rw-r--	1	kw2018202018	23768
2018202018_web_server.c	-rw-rw-r--	1	kw2018202018	23899
A.AA	-rw-rw-r--	1	kw2018202018	0
a.aa	-rw-rw-r--	1	kw2018202018	0
a.sdf	-rw-rw-r--	1	kw2018202018	0
aa.a	-rw-rw-r--	1	kw2018202018	0
aaa	-rw-rw-r--	1	kw2018202018	0
accessible_usr	-rw-rw-r--	1	kw2018202018	63
adv_server.c	-rw-rw-r--	1	kw2018202018	893
advanced_js	-rw-rw-r--	1	kw2018202018	22368
airplane.jpg	-rw-rw-r--	1	kw2018202018	4440
alink	lrwxrwxrwx	1	kw2018202018	3
asdf	lrwxrwxrwx	1	kw2018202018	6
aub	drwxrwxr-x	3	kw2018202018	4096
az	-rw-rw-r--	1	kw2018202018	0
B.aa	-rw-rw-r--	1	kw2018202018	0
cli	-rw-rw-r--	1	kw2018202018	9128
cli.c	-rw-rw-r--	1	kw2018202018	1272
fdf	-rw-rw-r--	1	kw2018202018	0
html_js	-rw-rw-r--	1	kw2018202018	47336

```

kw2018202018@ubuntu:~/ls_a$ ./srv_request2
===== New Client =====
IP : 127.0.0.1
Port : 19172
===== Disconnected Client =====
IP : 127.0.0.1
Port : 19172
===== Connection History =====
No. IP PID PORT TIME
1 127.0.0.1 73301 19172 Wen May 10 16:47:7 2023

```

Root 위치에 client 가 첫 access 했을 때의 결과 화면입니다. 현재 제 IP 와 Port 번호가 정확하게 출력되는 것을 확인할 수 있습니다.

The screenshot shows a web browser at 127.0.0.1:40000 displaying a directory listing for /home/kw2018202018/ls\_a/aub. The directory contains 56 bytes of files. A terminal window in the foreground shows the server logs for the request.

Name	Permission Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	3	kw2018202018	4096	May 1 16:59
..	drwxrwxr-x	4	kw2018202018	4096	May 10 16:43
dir1	drwxrwxr-x	2	kw2018202018	4096	May 1 16:59
file1	-rw-rw-r--	1	kw2018202018	0	Apr 3 12:47
file2	-rw-rw-r--	1	kw2018202018	14	May 1 9:32
link1	lrwxrwxrwx	1	kw2018202018	5	Apr 19 15:45
link2	lrwxrwxrwx	1	kw2018202018	5	May 1 9:32
slime.png	-rw-rw-r--	1	kw2018202018	37551	Apr 30 22:23
temp.txt	-rw-rw-r--	1	kw2018202018	0	May 1 16:58

```

kw2018202018@ubuntu:~/ls_a$ ./srv_request2
===== New Client =====
IP : 127.0.0.1
Port : 19684
===== Disconnected Client =====
IP : 127.0.0.1
Port : 19684
===== New Client =====
IP : 127.0.0.1
Port : 20196
===== Disconnected Client =====
IP : 127.0.0.1
Port : 20196
===== Connection History =====
No. IP PID PORT TIME
1 127.0.0.1 73353 20196 Wen May 10 16:49:28 2023
2 127.0.0.1 73351 19684 Wen May 10 16:49:25 2023

```



The browser shows a directory listing for `/home/kw2018202018/ls_a/aub` with a total of 56 items. The terminal window shows the connection history for the server, listing IP addresses, PIDs, ports, and connection times.

Name	Permission	Link	Owner	Group	Size	Last Modified
..	drwxrwxr-x	3	kw2018202018	kw2018202018	4096	May 1 16:59
..	drwxrwxr-x	4	kw2018202018	kw2018202018	4096	May 10 16:43
dir1	drwxrwxr-x	2	kw2018202018	kw2018202018	4096	May 1 16:59
file1	-rw-rw-r--	1	kw2018202018	kw2018202018	0	Apr 3 12:47
file2	-rw-rw-r--	1	kw2018202018	kw2018202018	14	May 1 9:32
link1	lrwxrwxrwx	1	kw2018202018	kw2018202018	5	Apr 19 15:45
link2	lrwxrwxrwx	1	kw2018202018	kw2018202018	5	May 1 9:32
slime.png	-rw-rw-r--	1	kw2018202018	kw2018202018	37551	Apr 30 22:23
temp.txt	-rw-rw-r--	1	kw2018202018	kw2018202018	0	May 1 16:58

```

IP : 127.0.0.1
Port : 20196
===== Disconnected Client =====
IP : 127.0.0.1
Port : 20196
===== Connection History =====
No. IP PID PORT TIME
1 127.0.0.1 73353 20196 Wen May 10 16:49:28 2023
2 127.0.0.1 73351 19684 Wen May 10 16:49:25 2023
===== Connection History =====
No. IP PID PORT TIME
1 127.0.0.1 73353 20196 Wen May 10 16:49:28 2023
2 127.0.0.1 73351 19684 Wen May 10 16:49:25 2023
===== Connection History =====
No. IP PID PORT TIME
1 127.0.0.1 73353 20196 Wen May 10 16:49:28 2023
2 127.0.0.1 73351 19684 Wen May 10 16:49:25 2023
===== Connection History =====
No. IP PID PORT TIME
1 127.0.0.1 73353 20196 Wen May 10 16:49:28 2023
2 127.0.0.1 73351 19684 Wen May 10 16:49:25 2023

```

Client 가 여러 번의 access 를 시도했을 때 입니다. Connection History 가 10 초마다 출력되며 이는 위부터 가장 최신의 access 부터 출력됩니다.

The file `accessible.user` in the `~/ls_a` directory contains the following IP addresses:

```

128.134.52.61
192.168.*.1
128.*.*.62
192.168.126.145

```

The browser shows an "Access denied!" message with the following details:

```

Your IP : 127.0.0.1
You have no permission to access this web server.
HTTP 403.6 - Forbidden:IP address reject

```

The terminal window shows the command `./srv_request2` being executed.

현재 제 IP 를 허가하지 않았을 경우 입니다. 127.0.0.1 은 accessible.user 에 포함되지 않은 IP 이기 때문에 Access denied! 메시지가 출력됩니다.

또한 connection list 에도 포함되지 않는 것을 확인할 수 있습니다.

## 5. 고찰

이번 과제는 이론으로만 배웠었던 `fork()` 함수의 실제 사용법을 익히게 되었고, parent process 와 child process 가 어떠한 방식으로 실행되는지를 정확하게 이해함으로써 과제를 마무리할 수 있었습니다. 하지만 이번 과제에서는 예상하지 못했던 의도치 않은 신호들로 인해 발생하는 오류들이 꽤 많이 발생하였습니다. /favicon.ico 또는 .. directory 를 들어가게 되었을 때 발생하는 알 수 없는 비어 있는 request message 등등이 발생하게 되었고, 이는 제가 코드를 직접 실행해보면서 캐치하기 어려운 오류들이었고, 이로 인해 코드의 오류를 잡아내기 위해 많은 시간을 할애했습니다. 이번 오류들을 잡아내기 위해서 코드를 직접 디버깅하는 과정에서 모든 저장되는 데이터를 출력하는 방식으로 잡아내게 되었습니다. 또한 이번에도 여러 함수를 응용하여 구현하게 되었지만, 아직 C 언어를 공부하면서 배우게 된 기초적인 함수들이 아닌, Linux 에서 서버를 구현하는데 필요로 하는 새로운 함수들 특히 struct 형태의 기본적으로 주어지는 자료들이나 함수들의 응용력이 많이 떨어지는 것을 느끼게 되었고, 함수의 응용을 완벽하게 못하는 것은 곧 비효율적인 코드 작성으로 이어지게 되었습니다. 이번 과제를 통해 여러 함수를 기억하고, 알맞은 자리에서 함수를 이용하여 코드를 구현하는 것이 중요하다는 것을 알게 되었습니다.

## 6. Reference