시스템프로그래밍 보고서

실험제목: assignment2-3 과제

제출일자: 2023년 05월 7일 (일)

학 과: 컴퓨터공학과

담당교수: 이기훈 교수님

실습분반: 금요일 5 6

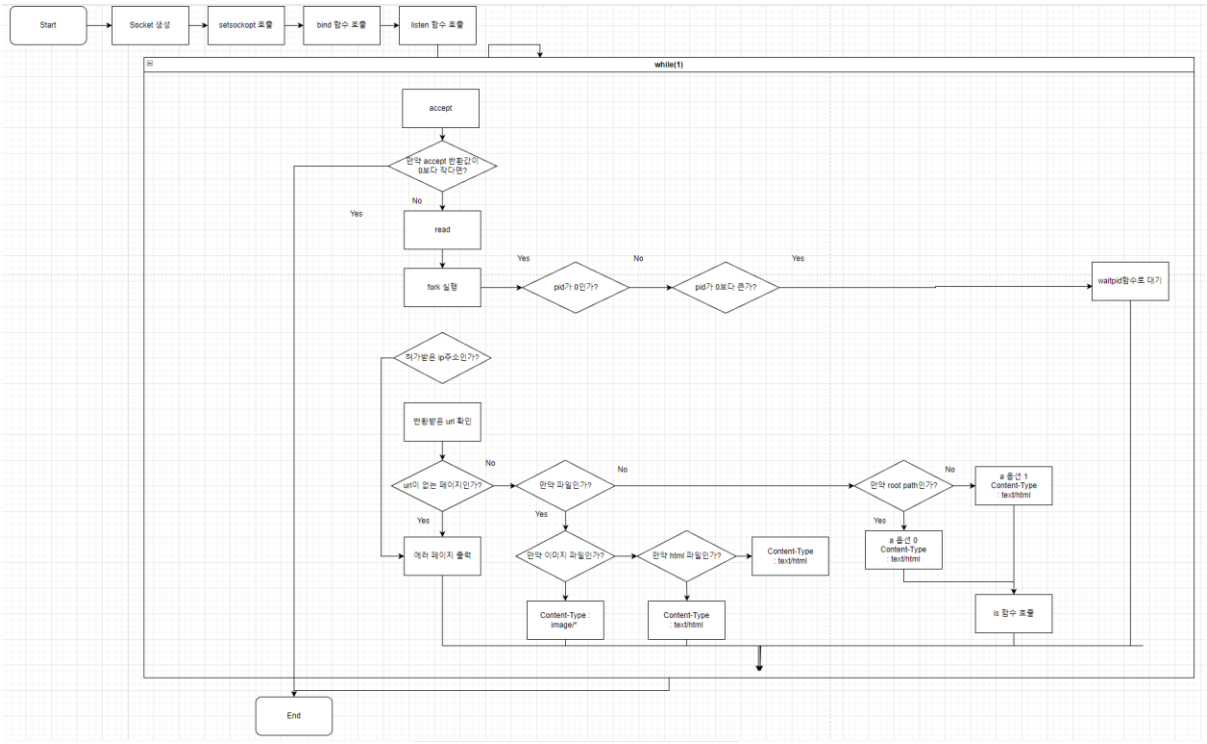
학 번: 2019202031

성 명: 장형범

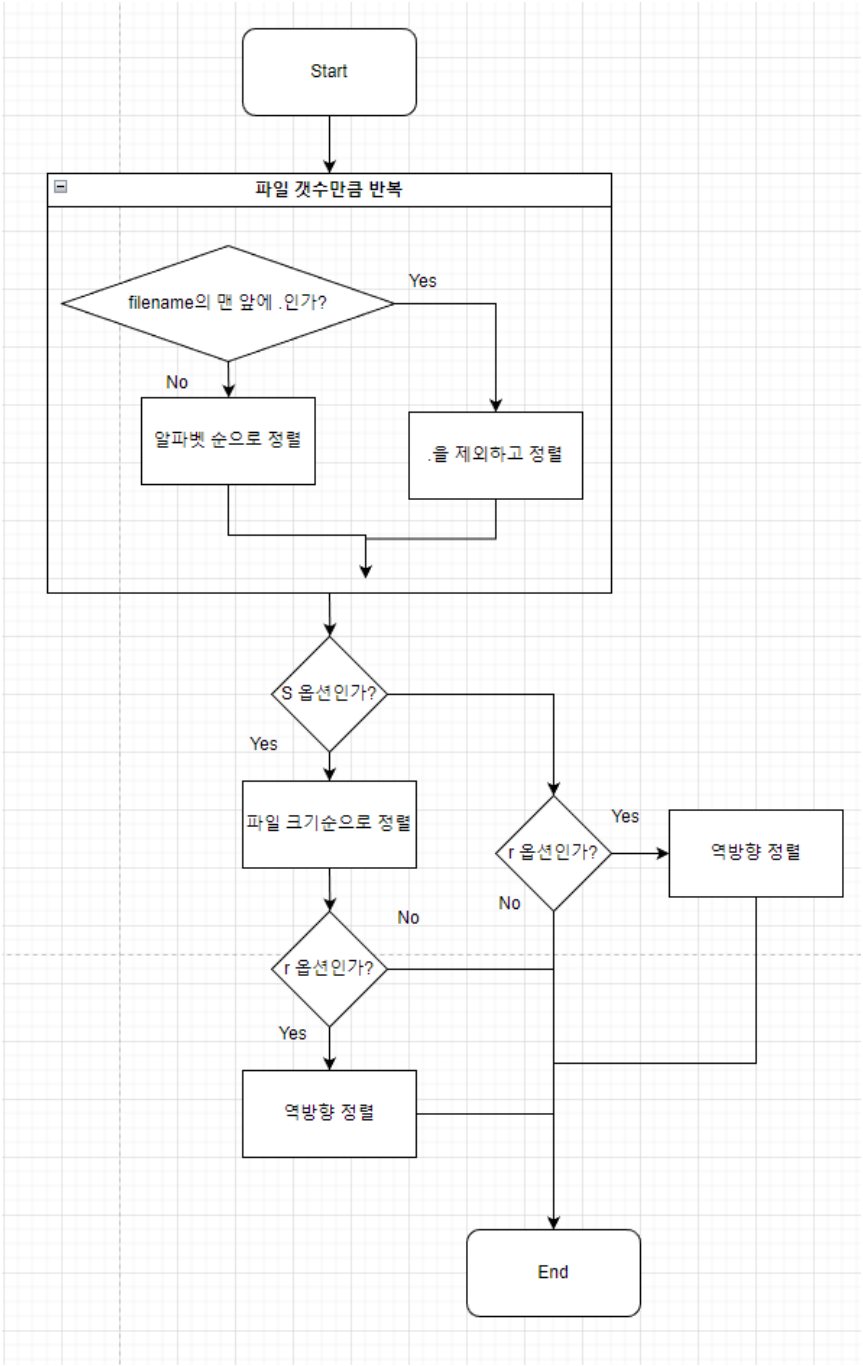
1. Introduction

2-2 에서 구현한 서버에서 다중 접속과 접근 제어를 지원하는 웹 서버 프로그램을 구현하는 것이다. 클라이언트와 소켓 연결된 이후의 작업을 새로운 프로세스에서 수행하여 동시에 다수의 클라이언트가 접속할 수 있도록 지원하도록 한다. 접속 연결 및 해제 시 클라이언트 정보를 출력하며 10 초마다 연결되었던 프로그램을 출력한다.

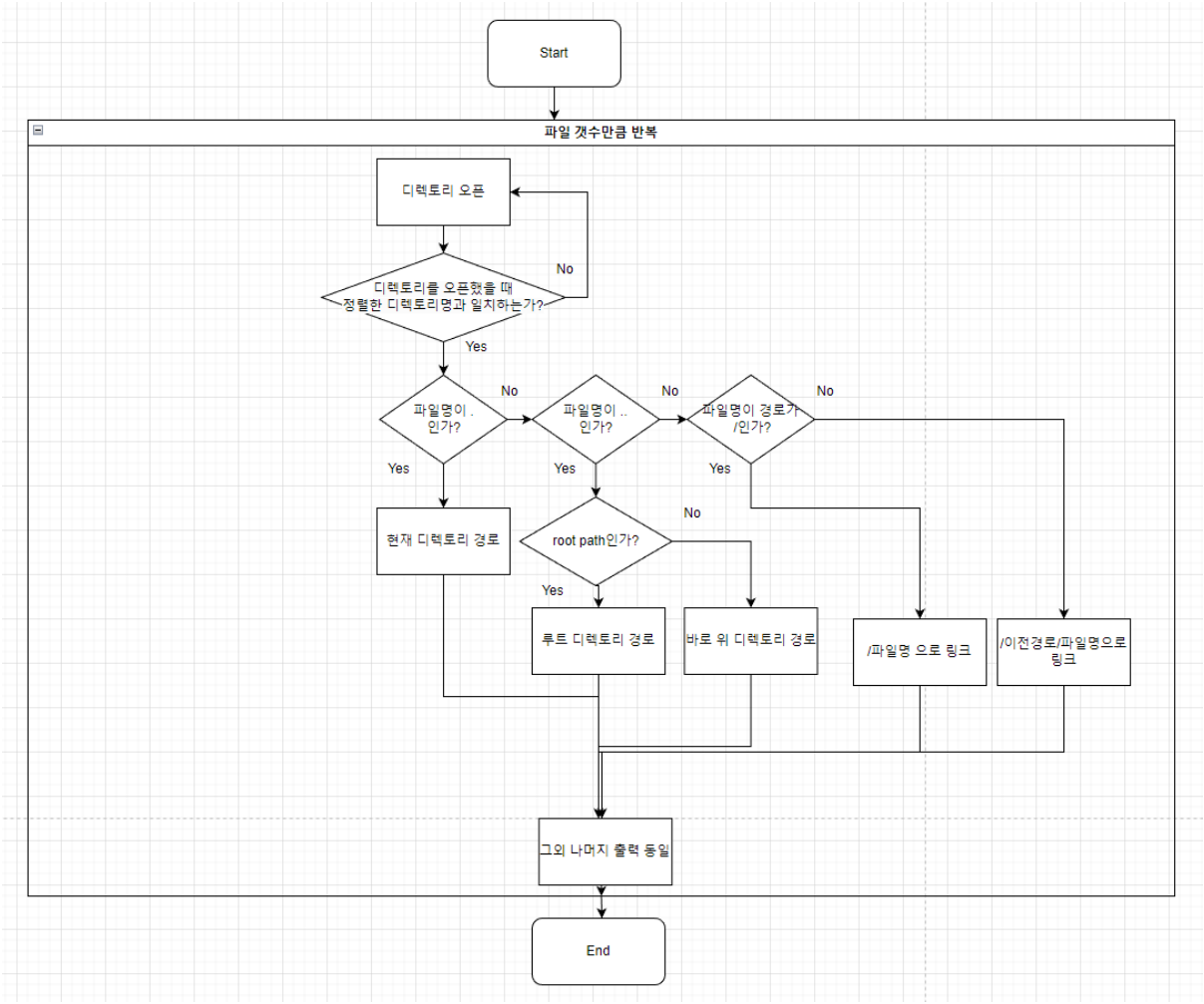
2. Flow chart



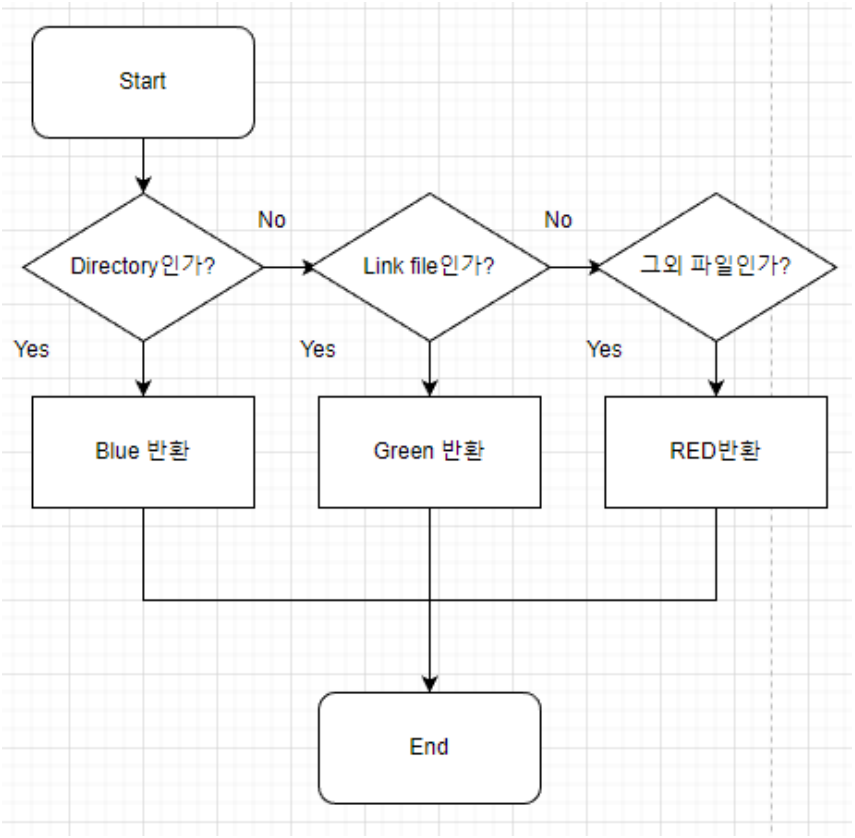
main 함수의 flow chart 이다.



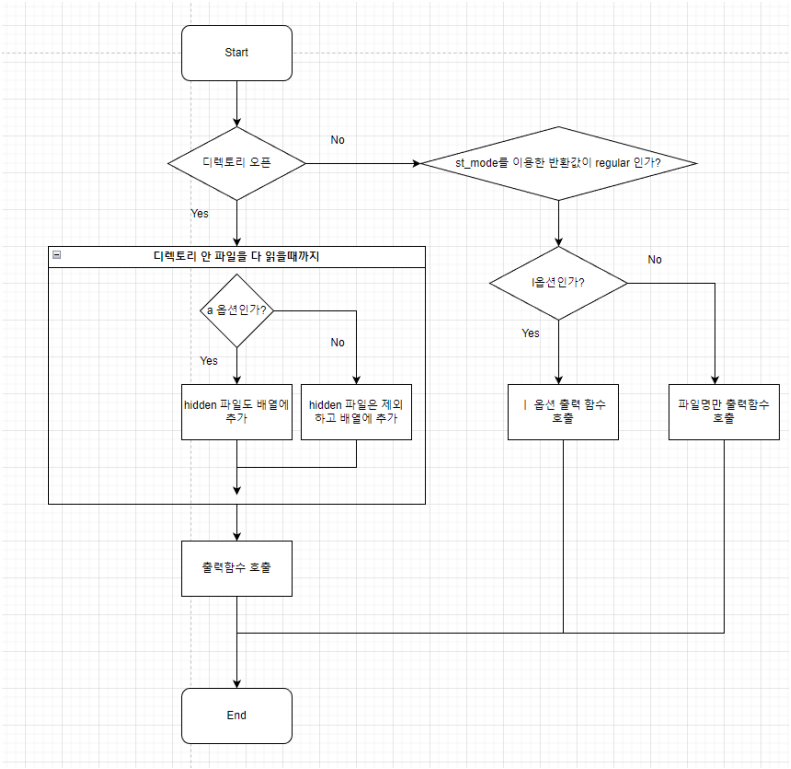
sorting 함수의 순서도이다.



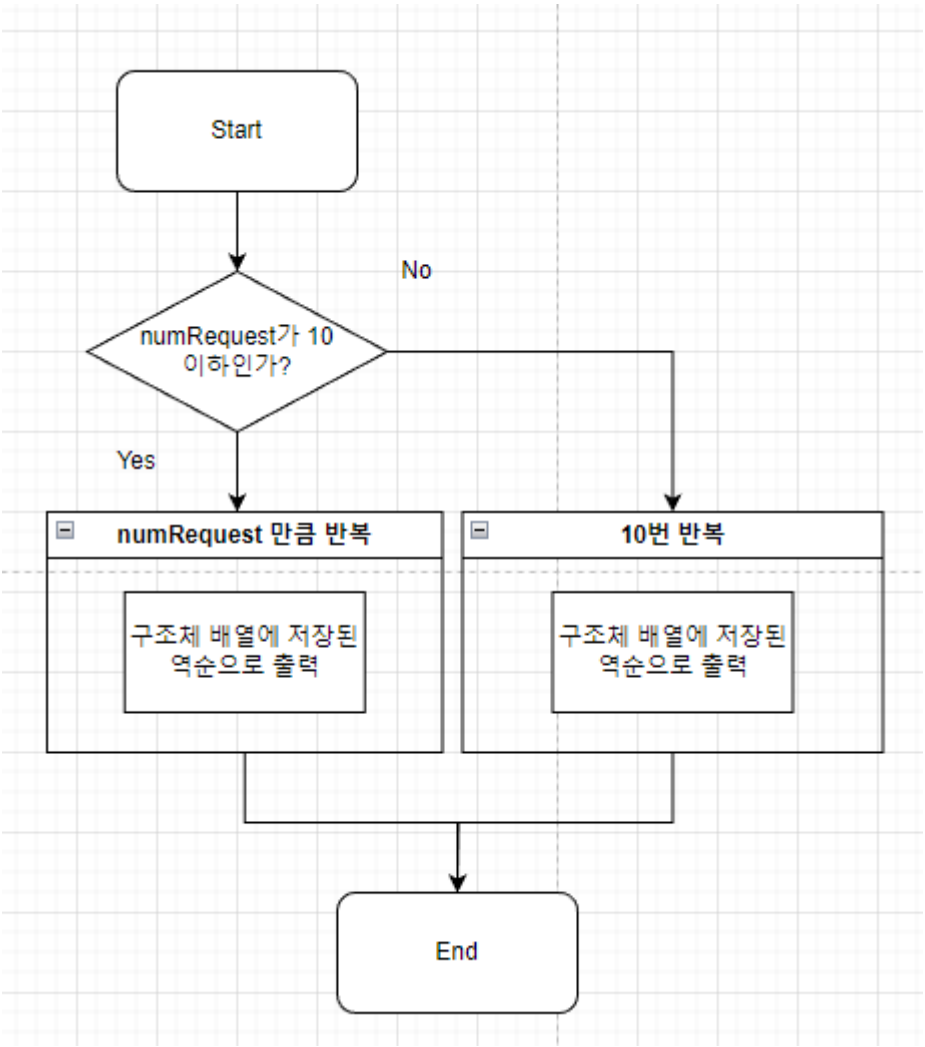
print_file_info 함수의 순서도이다.



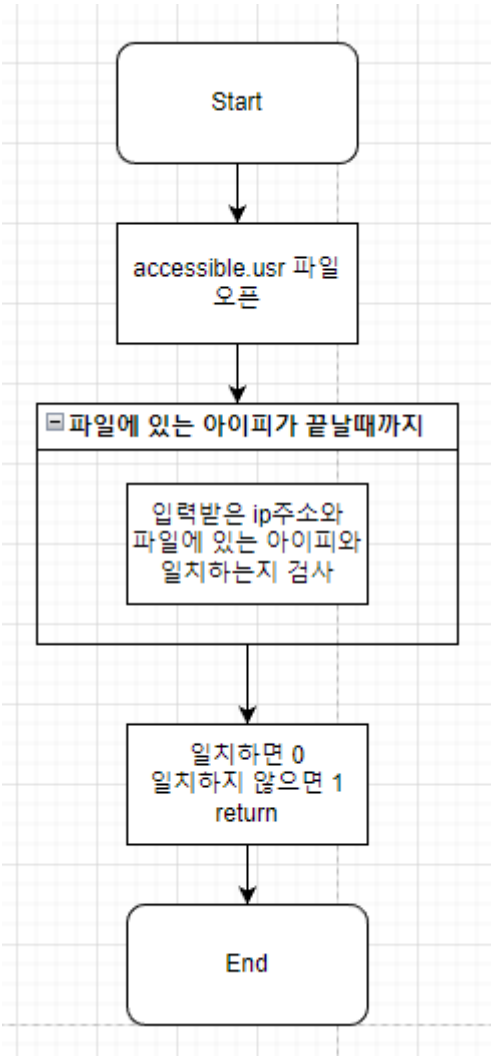
print_filetype 함수의 순서도이다.


Start
디렉토리 오픈
No
st_mode를 이용한 반환값이 regular 인가?
Yes
디렉토리 안 파일을 다 읽을때까지
a 옵션인가?
Yes
No
hidden 파일도 배열에 추가
hidden 파일은 제외하고 배열에 추가
l옵션인가?
No
Yes
l 옵션 출력 함수 호출
파일명만 출력함수 호출
출력함수 호출
End

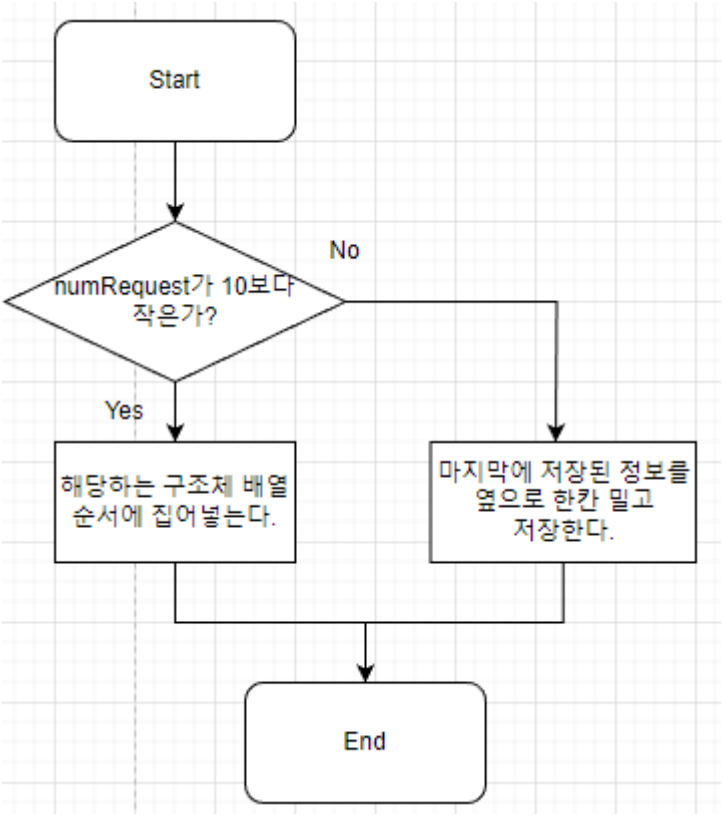
ls 함수의 순서도이다.



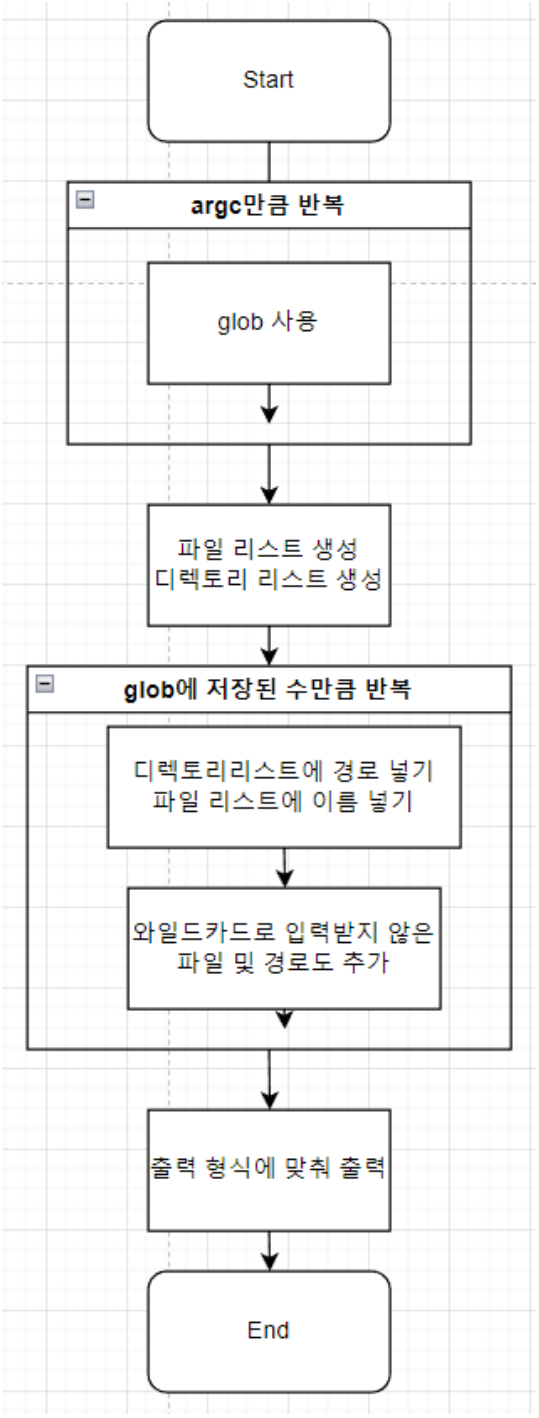
printHistory 함수이다.



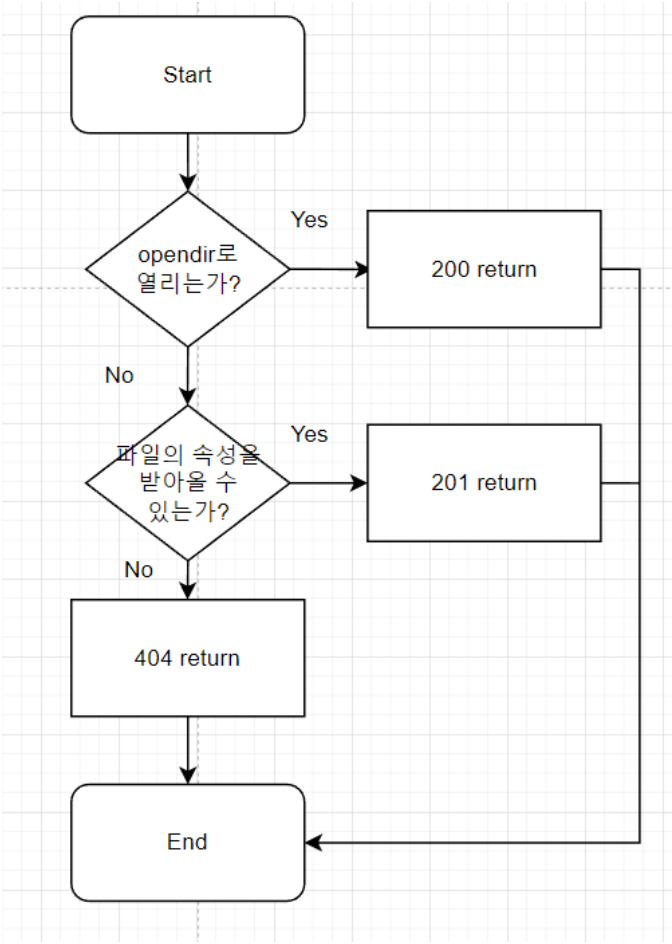
is_ip_allowed 함수이다.



addHistory 함수이다.



wild card 의 순서도이다.



check_404 의 순서도이다.

3. sudo code

```
int main(int argc, char** argv)
{
	소켓 함수 호출
	setsockopt 로 셋팅
	bind 함수로 소켓과 구조체 정보 binding
	listen 함수로 client 로부터 connection 을 위해 대기상태로 변경
	while(1)
	{
		자식 프로세스
		{
		10 초마다 출력되는 알람 설정
		accept 로 서버가 클라이언트 접속 허용
		read 로 정보 받음
		 fork 함수 사용
```

```
            if 허가된 ip list 에 없으면 access denied 출력
            if(없는 페이지)
                   404 에러 출력
            이미지, text, 일반 파일이면 조건에 맞게 출력
            root path 라면 a 옵션 off
            root path 가 아니라면 a 옵션 on
            }
            부모 프로세스
            {
            구조체 배열에 정보 전달
            자식이 종료되기까지 대기 후 closed
            }
       }
    return 0;
}
void alarmHandler(int signum)
{
    if (SIGCHILD 가 신호로오면)
    {
       waitpid 함수를 이용한 자식 프로세스 pid > 0 일때까지 반복
    }
    else if (signal == SIGALRM)
    {
        구조체 배열 선언
    }
    alarm(10); // set 10 second
}

void printHistory()
{
    if (저장된 history 가 10 개 이하라면) { //process number < 10
        저장된 것 만큼 반복해서 역순으로 출력
    }
    else
    {
        for (10 번 반복)
        {
            역순으로 출력
    }
}

int is_ip_allowed() {
    accessible.usr 파일 오픈
    fnmatch 함수를 이용해 파일에 있는 ip 주소들 검사
    같은게 있다면 거짓 반환 (함수 호출을 위해)
    같은게 없다면 참 반환
}
```

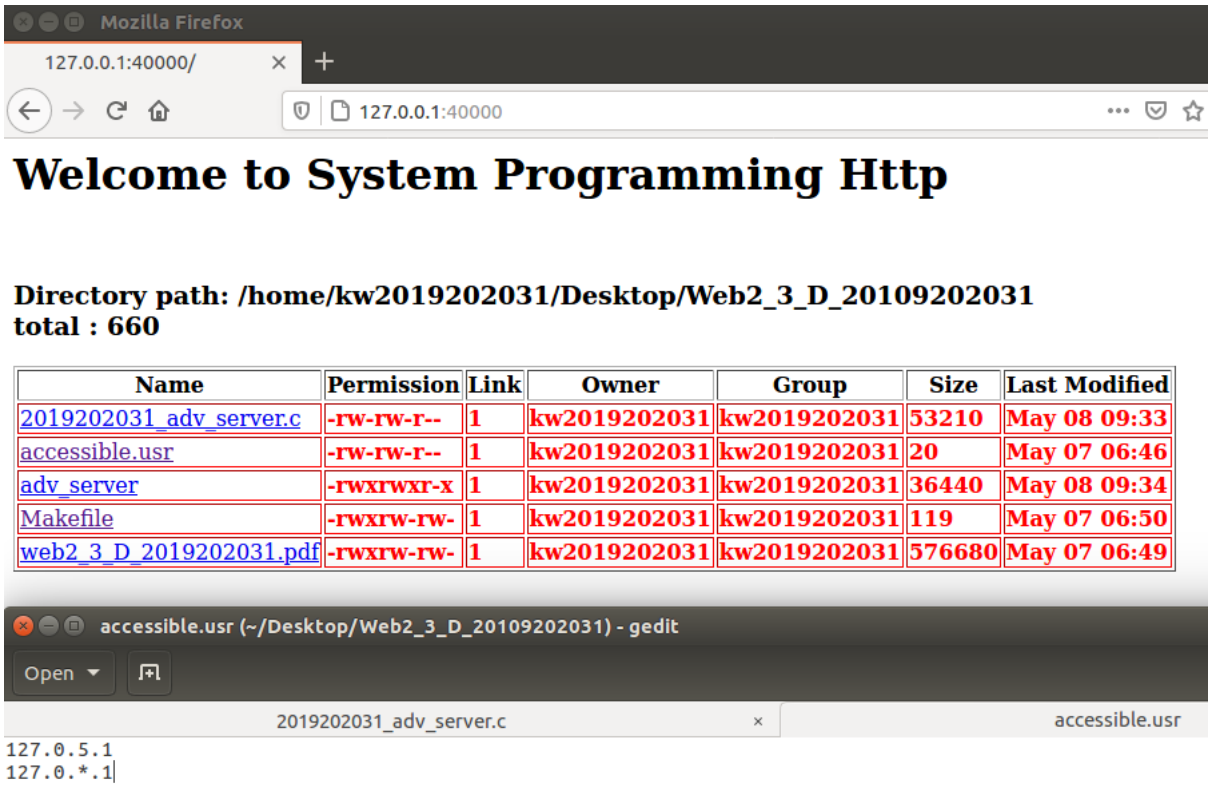
```
    char line[256];
    while (fgets(line, sizeof(line), file) != NULL) { //get ip address
        line[strcspn(line, "\n")] = '\0'; // cut \n word
        if (fnmatch(line, client_ip, FNM_CASEFOLD) == 0) { //compare word
            fclose(file);
            return 0;  //return allow ip
        }
    }
    fclose(file);
    return 1;  // return not allow ip
}



void addHistory() {
    if (저장된 개수가 10 개보다 적다면) {
        그다음 위치에 프로세스 정보 저장

    } else {
        기존에 저장된 배열을 한칸 민다.
        밀어서 생긴 공간에 프로세스 정보 저장한다.
    }
    numRequests++;
}

int check_404
{
    디렉토리라면
        return 200; //exist
    }
    파일이라면
    {
        return 201; //exist
    }
    else
        return 404; //not exist
}

void wild_card
{
    argc 만큼 반복문을 사용해 glob 함수 사용
    파일과 디렉토리 갯수 체크
     파일과 디렉토리 배열 생성
    배열에 정보 저장
    와일드카드가 아니었던 경로나 파일을 배열에 추가
    테이블 생성
    값 write 하기
}
```

```
void no_dir
{
    테이블 생성
    디렉토리가 아닌 파일의 출력 -l 옵션일때 사용
    write 하기
}

void ls
{
   디렉토리를 열고 안에 있는 파일 read
   파일의 size 합산, 배열에 파일명 넣기
   옵션에 맞는 파일 정렬
}

void sorting
{
       알파벳 순서로 배열 정렬 만약 파일의 이름 맨 앞이 .이라면 그 다음 알파벳
       부터 비교
       S 옵션이 1 이라면 dir 를 파일의 크기로 정렬
       r 옵션이 1 이랑면 역순으로 파일 정렬

}

char* check_filetype
{
    type 반환
}

void print_file_info
{
       테이블 생성
       l 옵션이라면 파일의 세부 정보 html 테이블에 입력

       옵션이 없다면 파일의 이름만 테이블에 입력
    }

}
```

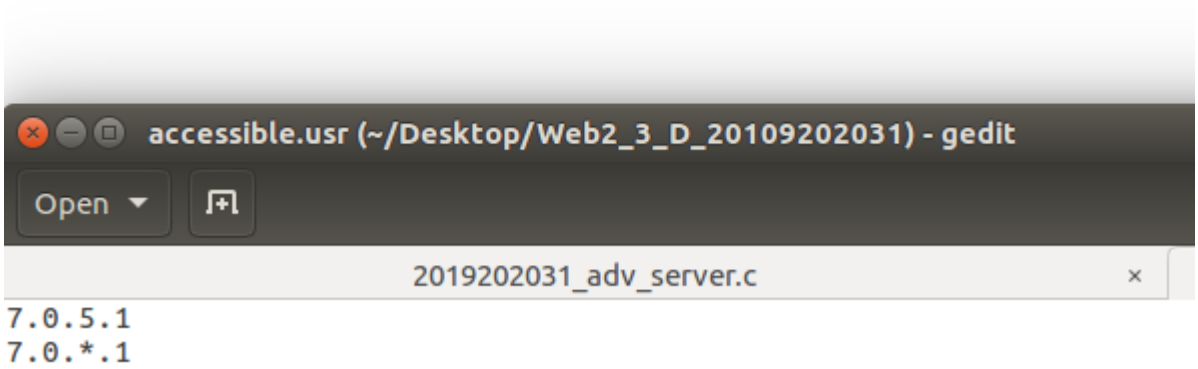
결과화면

```
kw2019202031@ubuntu:~/Desktop/Web2_3_D_20109202031$ ./adv_server
=============== New Client ===============
IP : 127.0.0.1
Port : 20143
==========================================
========== Disconnected Client ===========
IP : 127.0.0.1
Port : 20143
==========================================
=============== New Client ===============
IP : 127.0.0.1
Port : 20655
==========================================
========== Disconnected Client ===========
IP : 127.0.0.1
Port : 20655
==========================================
============ Connection History ==========================================
Number of request(s) : 2
No.     IP              PID     PORT    TIME
2       127.0.0.1       2463    20655   Mon May  8 09:34:27 2023
1       127.0.0.1       2442    20143   Mon May  8 09:34:21 2023
```

접속한 ip 주소와 port 번호가 뜬다.



accessible.usr 에 ip 를 이렇게 넣어둔 후 실행하면 ip 가 127.0.0.1 이기 때문에 페이지가 정상적으로 나온다.



accessible.usr 에 7.0.5.1 과 7.0.*.1 을 넣으면 ip 주소가 허용되지 않기 때문에 Access denied 가 뜬다.

고찰

fork 를 하면서 자식 프로세스가 일을 전부 하는 거라고 처음에는 생각하지 못 하고 부모는 127.0.0.1 에서 루트 디렉토리를 여는 것을 부모

프로세스가 하는 것이라 처음 생각했는데 강의자료를 자세히 읽어보다보니 그렇게 하는 것이 아니라는 것을 깨달았다. 또한 전역변수를 사용하고 자식 프로세스에서 전역변수에 값을 쓰려면 부모에서 값을 보내야 하는 것을 알게 되었다. vfork 를 썼다가 vfork 는 다중 접속 지원이 안 되는 것 같아 fork 로 바꾸게 되었다.

reference

강의자료 참고