



시스템프로그래밍실습
Assignment2-3 과제

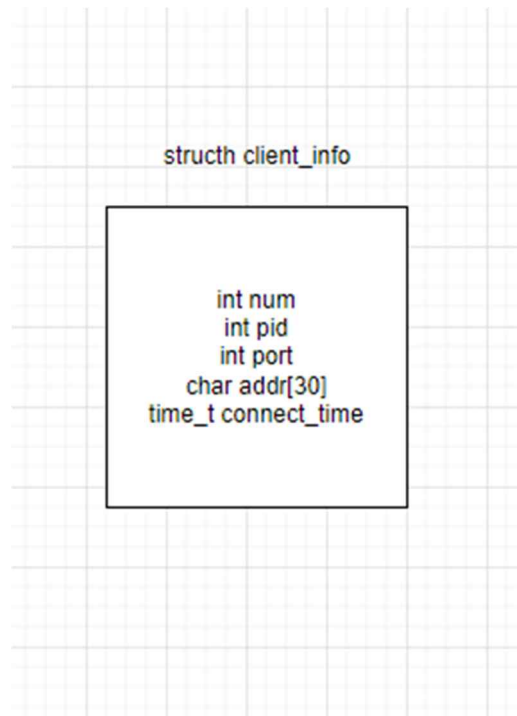
수업 명 : 시스템프로그래밍실습
과제 이름 : assignment2-3
담당 교수님 : 김태석 교수님
학 번 : 2019202005
이 름 : 남종식

과제 소개

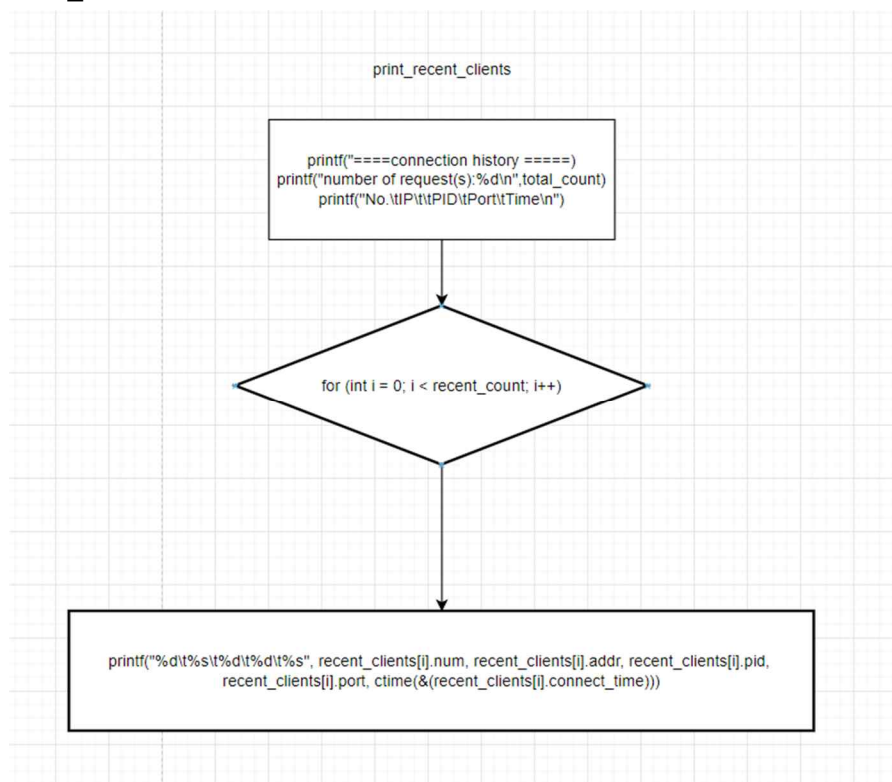
이번 과제는 다중 접속과 접근 제어를 지원하는 웹 서버 프로그램을 작성하는 과제입니다. 먼저 다중 접속을 지원하기 위해서는 `fork()`함수를 이용하여 자식 프로세스를 생성해 진행할 수 있습니다. 클라이언트와 접속 연결 해제시 마다 ip 주소와 port번호를 출력해야 하며 10초에 한번씩 연결된 클라이언트의 정보를 출력해야 합니다. 이를 10초마다 출력하기 위해서는 `alarm`함수를 통해 처리할 수 있습니다. 만약 서버와 클라이언트의 접속이 총 10번을 넘어간다면 최근기록을 포함해 최대 10개만 출력해야 합니다. 접근 제어 부분은 `accessible usr`에 적히지 않은 ip주소가 입력될 시에 접근 제한 메시지와 함께 ip주소를 출력해야 합니다.

Flow Chart

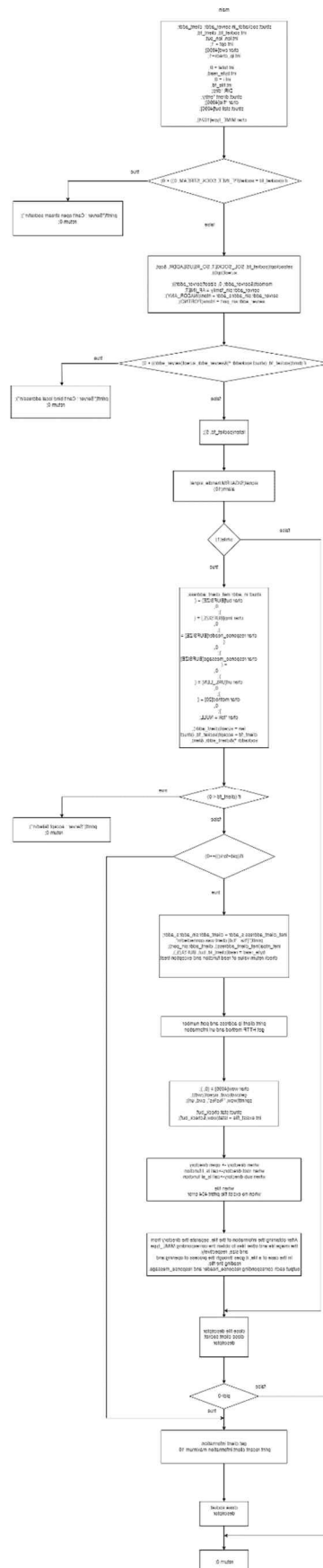
struct client_info



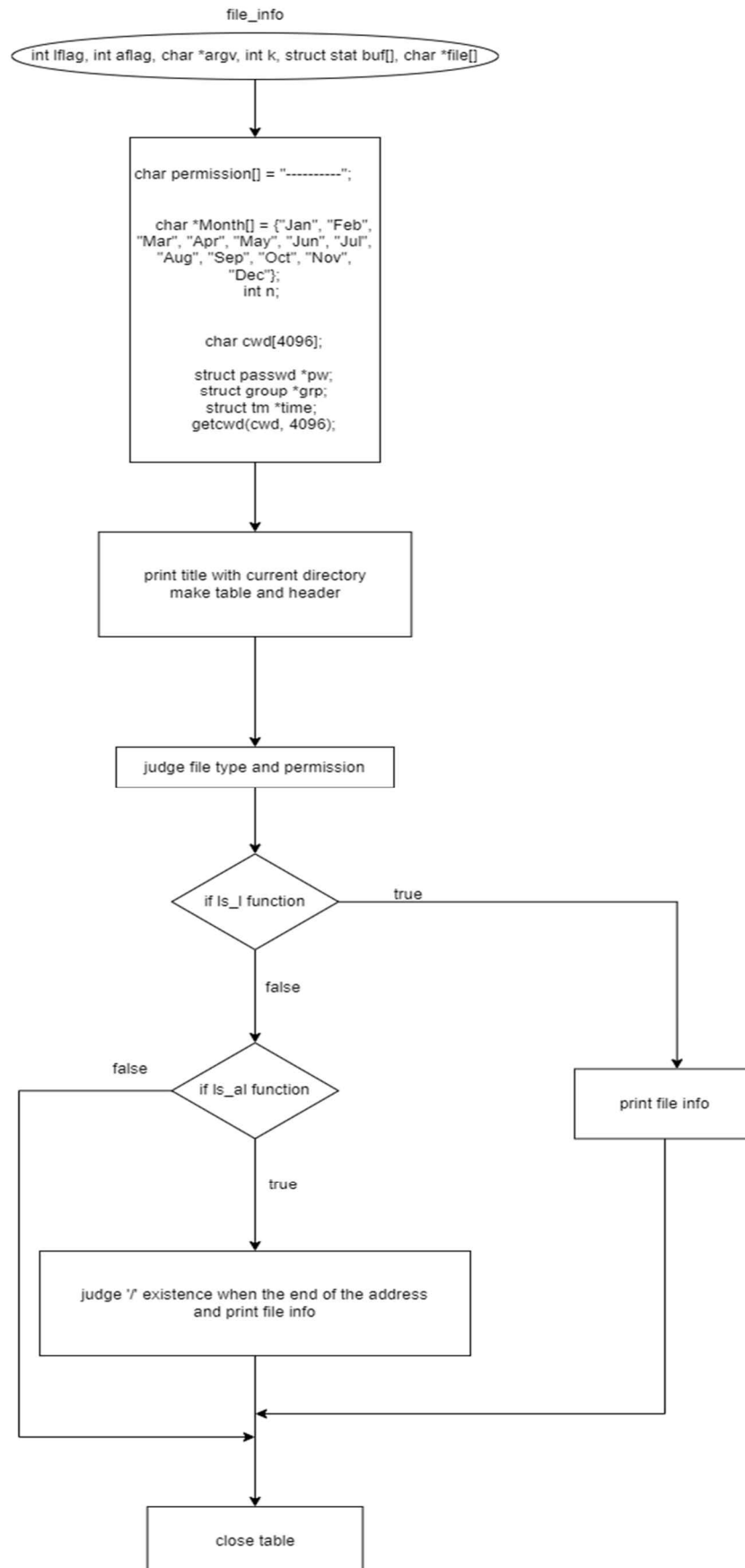
print_recent_clients function



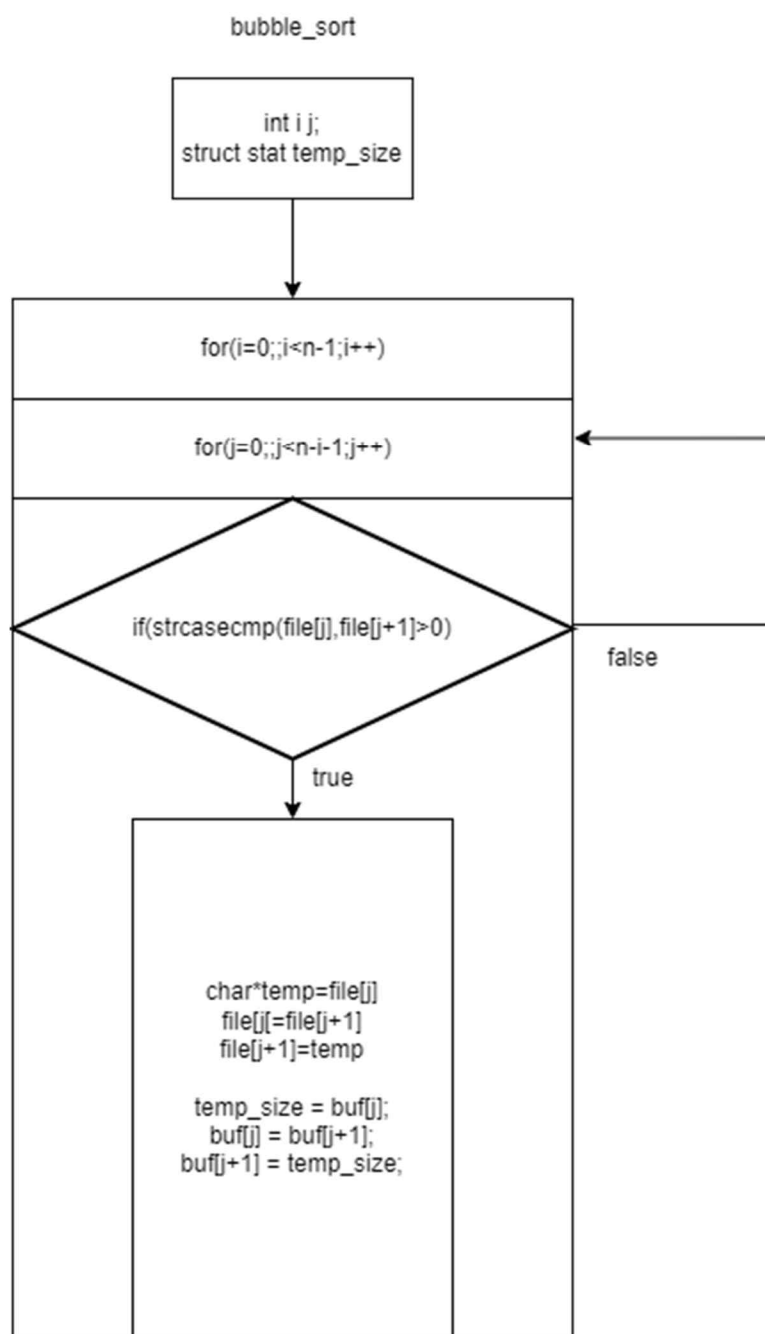
Main fuction



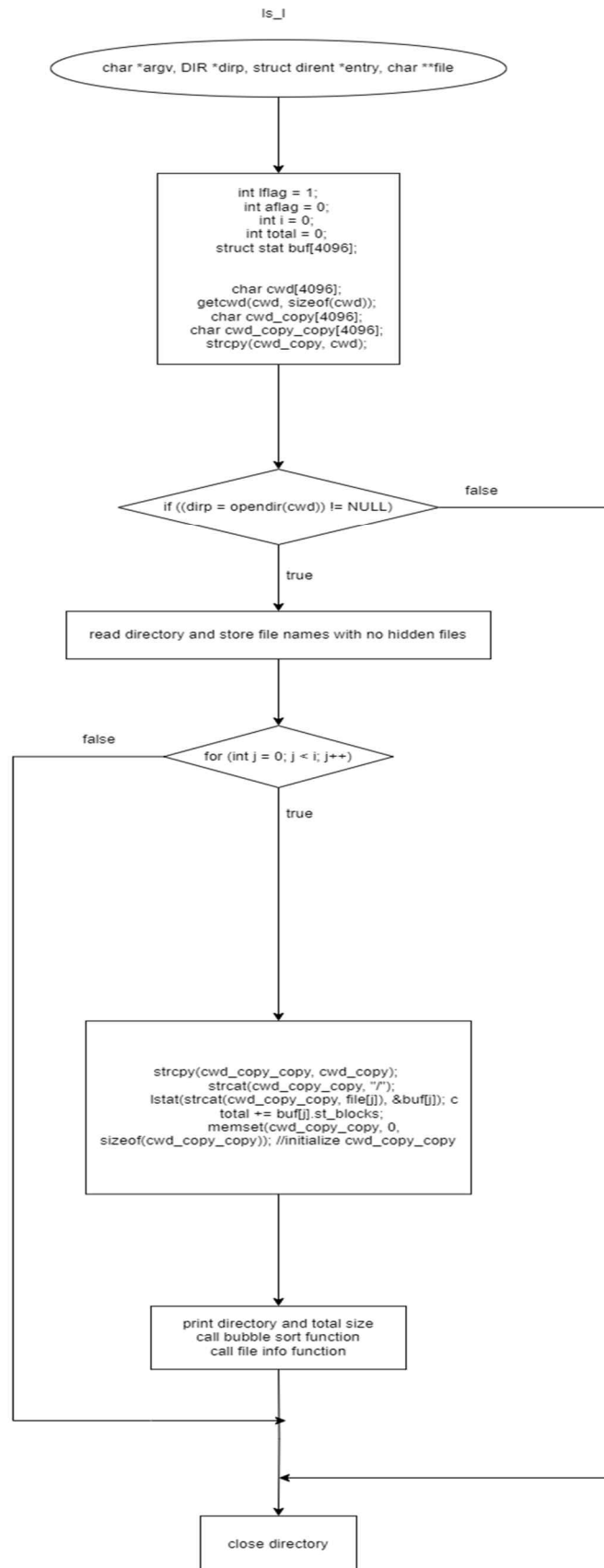
File_info function



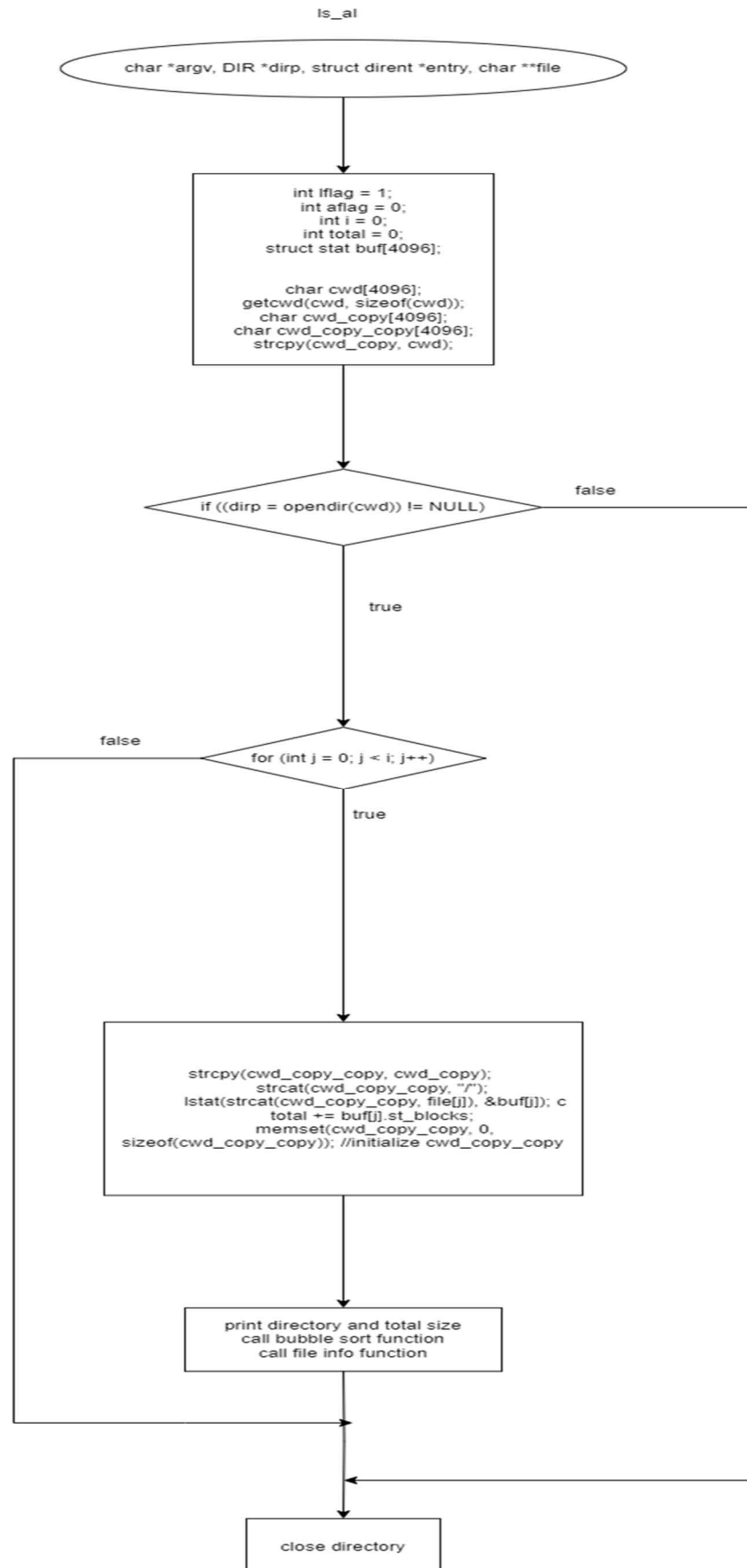
Bubble sort function



Ls_l function



Ls_al function



Pseudo Code

```
#define _GNU_SOURCE
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <string.h>
#include <unistd.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <pwd.h>
#include <grp.h>
#include <time.h>
#include <fnmatch.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <fcntl.h>
#include <sys/wait.h>
#include <signal.h>
```

```
#define URL_LEN 256
#define BUFSIZE 1024
#define PORTNO 40000
```

```
define file info function
define bubble sort
define ls -l function
define ls -al function
```

define struct array

define print_recent_clients

define handle_signal

Main function

{

declare server address, client address

declare socket descriptor, client descriptor

declare pointer to directory stream and Pointer to a dirent structure

declare file list stat structure that stores information about a file or director

Call the socket function to create a socket file descriptor

Call setsockopt to enable reuse of previously used port numbers

Initialize server_addr structure

Save address information for the server socket

Bind a socket with socket_fd to the IP address and port number specified in the server_addr structure

Socket wait to accept incoming connections from client sockets.

signal handler

alarm in 10 seconds

when no exist file print 404 not found

while (1)

{

Define variables required to process client requests

define size of client address

get the socket file descriptor associated with the client

fork()

when child process

{

store ip address

Make an exception when read function return 0

Copy HTTP method information and url information

get current working directory

get path of wow (wow=cwd+url)

Open the file to read all lines, compare it to the ip address, and proceed if the corresponding ip address exists, and if there is no ip address, print an access restriction message

get information of files

open directory

{

when root directory

{

call ls -l function

}

when sub directory

{

call ls -al function

}

}

get file info

```
when directory
{
    MIME_type=text/html
    get size of response message
    print response header
    write response header
    write response message
}
when image file
{
    MIME_type=image/*
    get file size
    file open
    file read
    print response header
    write response header
    write response message
    close file descriptor
}
when file
{
    MIME_type=text/plain
    get file size
    file open
    file read
    print response header
    write response header
    write response message
    close file descriptor
}
print the client's IP address and port number
```

```

        close client socket descriptor
    }
    when parent process
    get client information
    call waitpid function
}
close socket descriptor
exit
when parent process
{
    get client information
    add client information to client struct array
    wait until child process die
}
}
define print_recent_clients function
{
    print client information
    alarm(10)
}
define handle_signal
{
    call print_recent_clients function
}
define file_info function about directory
{
    get current directory

    print title current directory
    make table
    make header

```

```

for (n = 0; n < k; n++)
{
    judge file mode and permission

    get File Owner's Information
    get the information of the file's owning group
    get the last modified time of that file
    function for Parsing Time Information

    when root directory
    {
        when directory hyperlink with blue color :file[n]
        when link file hyperlink with green color :file[n]
        when file hyperlink with red color :file[n]
    }
    when sub directory
    {
        When the '/' is at the end of the address
        {
            when directory hyperlink with blue color :url file[n]
            when link file hyperlink with green color :url file[n]
            when file hyperlink with red color :url file[n]
        }
        When there is no '/' at the end of the address
        {
            when directory hyperlink with blue color :url/file[n]
            when link file hyperlink with green color :url/file[n]
            when file hyperlink with red color :url/file[n]
        }
    }
}
close table

```

```

}
define bubble sort function
{
    initialize value
    Bubble sort the file names alphabetically
}
define ls -l function
{
    get current directory
    get current directory+url

    when open directory
    {
        Repeat read directory
        {
            store file names without hidden files
        }

        for (int j = 0; j < i; j++)
        {
            get current directory+url+/
            get info about current directory+url+/+file
            get block size
            initialize current directory+url+/+file
        }
        print directory path & total size
        call bubble_sort function
        print file info
    }

    close directory
}

```

```

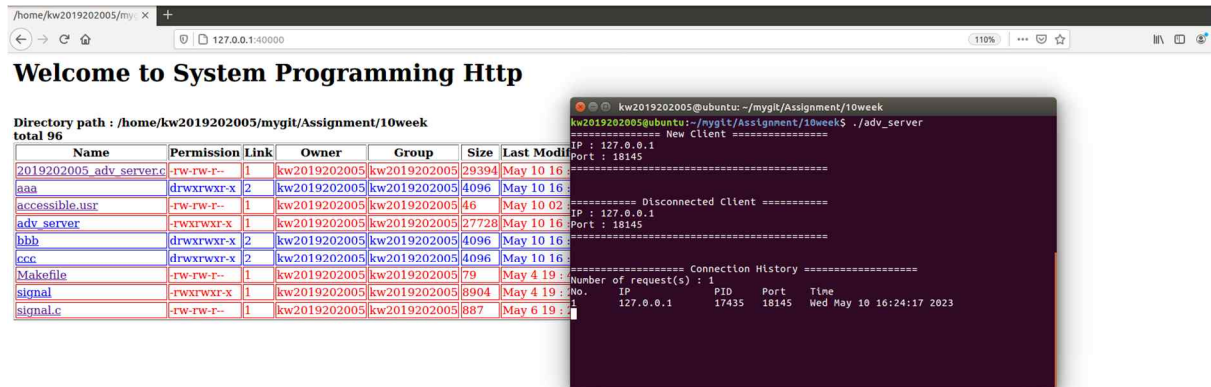
define ls -al function
{
    get current directory
    get current directory+url

    when open directory+url
    {
        Repeat read directory
        {
            store file list with hidden file
        }

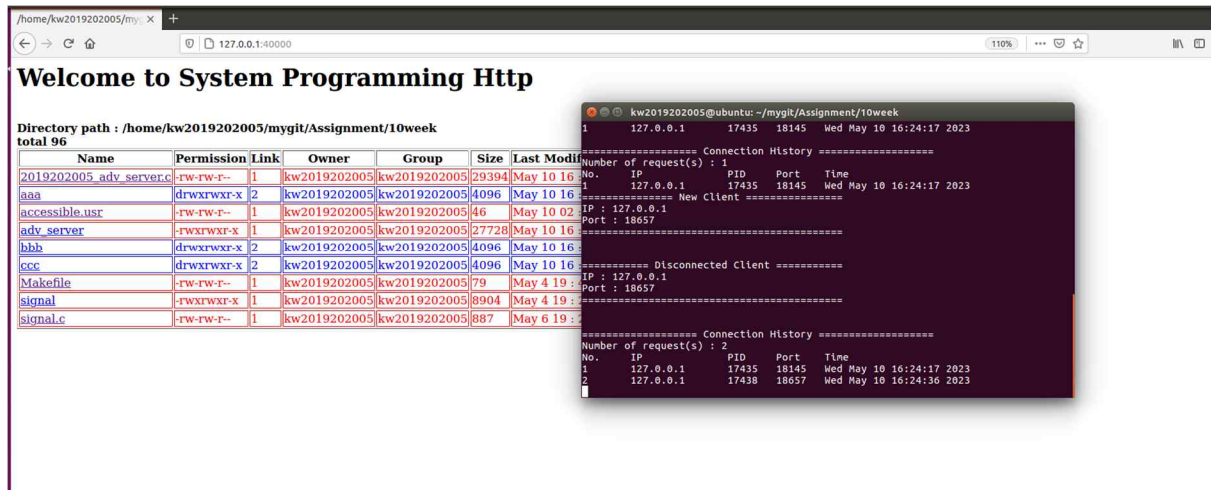
        for (int j = 0; j < i; j++)
        {
            get current directory+url+/
            get info about current directory+url+/+file
            get block size
            initialize current directory+url+/+file
        }
        print directory path & total size
        call bubble_sort function
        print file info
    }
    close directory
}

```

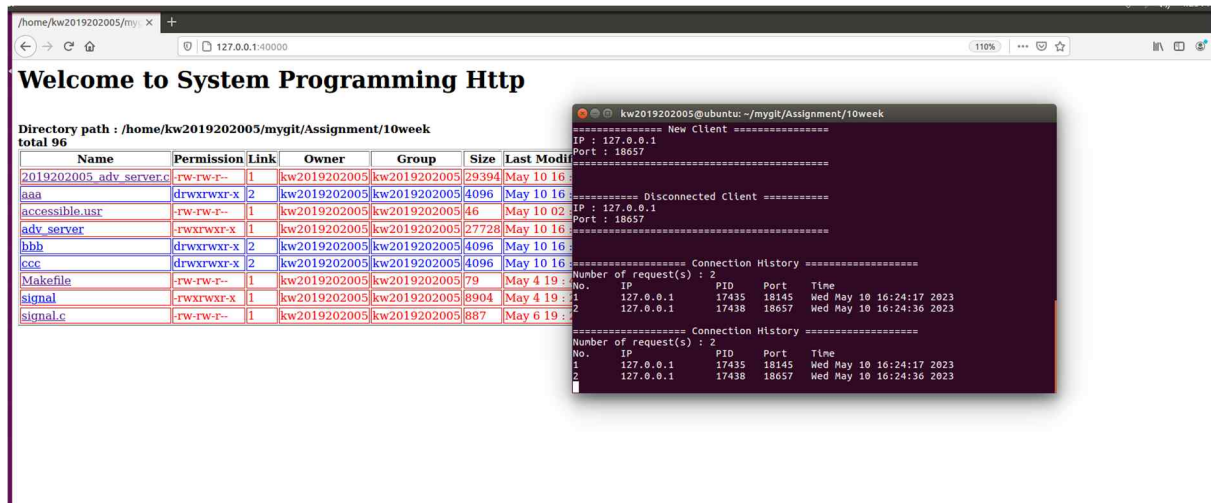
결과화면



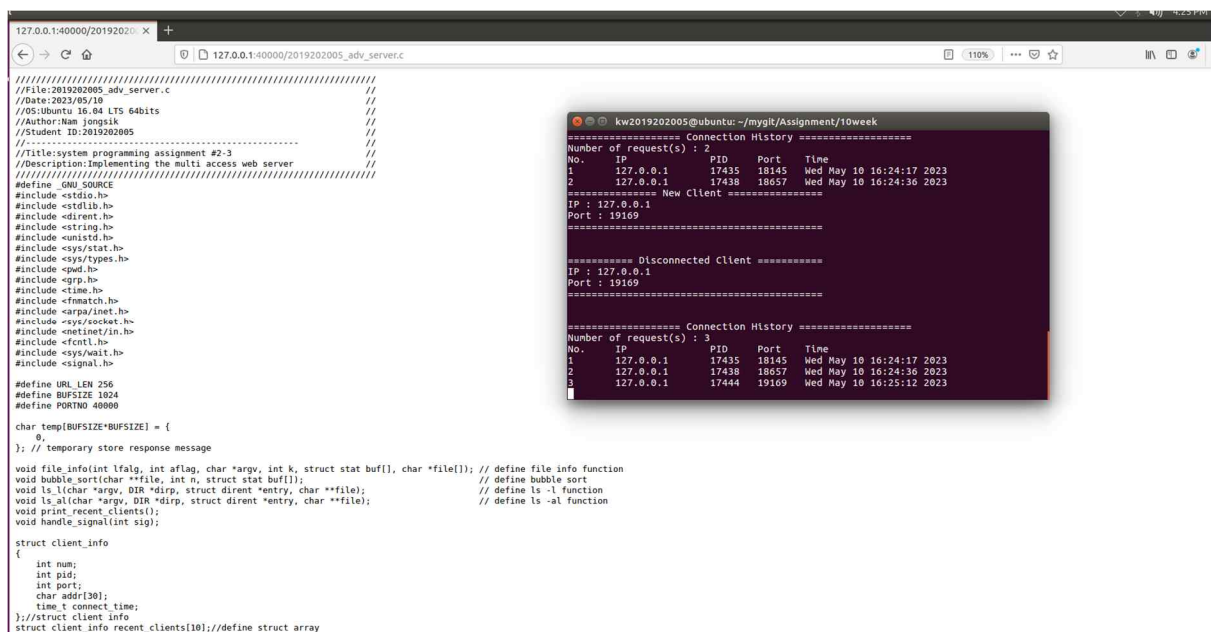
다음은 client의 request가 1일 때 출력되는 화면입니다.
클라이언트와 서버가 연결되었을 때 ip주소와 port번호가 출력되는 모습을 확인할 수 있으며 10초에 한번씩 연결되었던 클라이언트의 정보가 출력됩니다. 총 request의 개수를 먼저 출력하며 다음 줄에는 해당 클라이언트가 몇 번째로 연결되었는지, 클라이언트의 ip주소, 프로세스id, port번호, 서버와 클라이언트가 연결된 시간을 출력합니다.



다음은 request가 2번일 때 출력되는 화면이며 먼저 연결되었던 클라이언트의 정보가 첫번째로 출력되고 나중에 연결된 클라이언트의 정보가 두번째로 출력되는 모습을 확인할 수 있습니다.



다음은 request를 하지 않았을 때 출력 화면입니다.
10초 후 이전의 접속 기록을 그대로 출력하는 모습을 확인할 수 있습니다.



다음은 request가 3번일 때 출력 화면입니다. 시간 순으로 정렬되어 가장 나중에 연결된 클라이언트의 정보가 가장 마지막으로 출력되는 모습을 확인할 수 있습니다.

2019202005/my: X

127.0.0.1:40000

Welcome to System Programming Http

Directory path : /home/kw2019202005/mygit/Assignment/10week
total 96

Name	Permission	Link	Owner	Group	Size	Last Modified
2019202005_adv_server.c	-rw-rw-r--	1	kw2019202005	kw2019202005	29394	May 10 16:19
aaa	drwxrwxr-x	2	kw2019202005	kw2019202005	4096	May 10 16:19
accessible_usr	-rw-rw-r--	1	kw2019202005	kw2019202005	46	May 10 02:26
adv_server	-rwxrwxr-x	1	kw2019202005	kw2019202005	27728	May 10 16:19
bbb	drwxrwxr-x	2	kw2019202005	kw2019202005	4096	May 10 16:19
ccc	drwxrwxr-x	2	kw2019202005	kw2019202005	4096	May 10 16:19
Makefile	-rw-rw-r--	1	kw2019202005	kw2019202005	79	May 4 19:26
signal	-rwxrwxr-x	1	kw2019202005	kw2019202005	8904	May 4 19:26
signal.c	-rw-rw-r--	1	kw2019202005	kw2019202005	887	May 6 19:28

```
kw2019202005@ubuntu: ~/mygit/Assignment/10week
Port : 35041
=====
===== Disconnected Client =====
IP : 127.0.0.1
Port : 35041
=====
===== Connection History =====
Number of request(s) : 10
No. IP PID Port Time
1 127.0.0.1 17549 30433 Wed May 10 16:31:33 2023
2 127.0.0.1 17553 30945 Wed May 10 16:31:41 2023
3 127.0.0.1 17556 31457 Wed May 10 16:31:51 2023
4 127.0.0.1 17559 31969 Wed May 10 16:32:02 2023
5 127.0.0.1 17567 32481 Wed May 10 16:32:13 2023
6 127.0.0.1 17570 32993 Wed May 10 16:32:21 2023
7 127.0.0.1 17573 33505 Wed May 10 16:32:32 2023
8 127.0.0.1 17578 34017 Wed May 10 16:32:42 2023
9 127.0.0.1 17583 34529 Wed May 10 16:32:52 2023
10 127.0.0.1 17586 35041 Wed May 10 16:33:03 2023
```

다음은 총 10번의 request가 진행될 때 출력되는 화면입니다.

2019202005/my: X

127.0.0.1:40000

Welcome to System Programming Http

Directory path : /home/kw2019202005/mygit/Assignment/10week
total 96

Name	Permission	Link	Owner	Group	Size	Last Modified
2019202005_adv_server.c	-rw-rw-r--	1	kw2019202005	kw2019202005	29394	May 10 16:19
aaa	drwxrwxr-x	2	kw2019202005	kw2019202005	4096	May 10 16:19
accessible_usr	-rw-rw-r--	1	kw2019202005	kw2019202005	46	May 10 02:26
adv_server	-rwxrwxr-x	1	kw2019202005	kw2019202005	27728	May 10 16:19
bbb	drwxrwxr-x	2	kw2019202005	kw2019202005	4096	May 10 16:19
ccc	drwxrwxr-x	2	kw2019202005	kw2019202005	4096	May 10 16:19
Makefile	-rw-rw-r--	1	kw2019202005	kw2019202005	79	May 4 19:26
signal	-rwxrwxr-x	1	kw2019202005	kw2019202005	8904	May 4 19:26
signal.c	-rw-rw-r--	1	kw2019202005	kw2019202005	887	May 6 19:28

```
kw2019202005@ubuntu: ~/mygit/Assignment/10week
Port : 36577
=====
===== Disconnected Client =====
IP : 127.0.0.1
Port : 36577
=====
===== Connection History =====
Number of request(s) : 13
No. IP PID Port Time
1 127.0.0.1 17559 31969 Wed May 10 16:32:02 2023
2 127.0.0.1 17567 32481 Wed May 10 16:32:13 2023
3 127.0.0.1 17570 32993 Wed May 10 16:32:21 2023
4 127.0.0.1 17573 33505 Wed May 10 16:32:32 2023
5 127.0.0.1 17578 34017 Wed May 10 16:32:42 2023
6 127.0.0.1 17583 34529 Wed May 10 16:32:52 2023
7 127.0.0.1 17586 35041 Wed May 10 16:33:03 2023
8 127.0.0.1 17592 35553 Wed May 10 16:33:22 2023
9 127.0.0.1 17593 36065 Wed May 10 16:33:24 2023
10 127.0.0.1 17598 36577 Wed May 10 16:33:43 2023
11 127.0.0.1 17592 35553 Wed May 10 16:33:22 2023
12 127.0.0.1 17593 36065 Wed May 10 16:33:24 2023
13 127.0.0.1 17598 36577 Wed May 10 16:33:43 2023
```

2019202005/my: X

127.0.0.1:40000

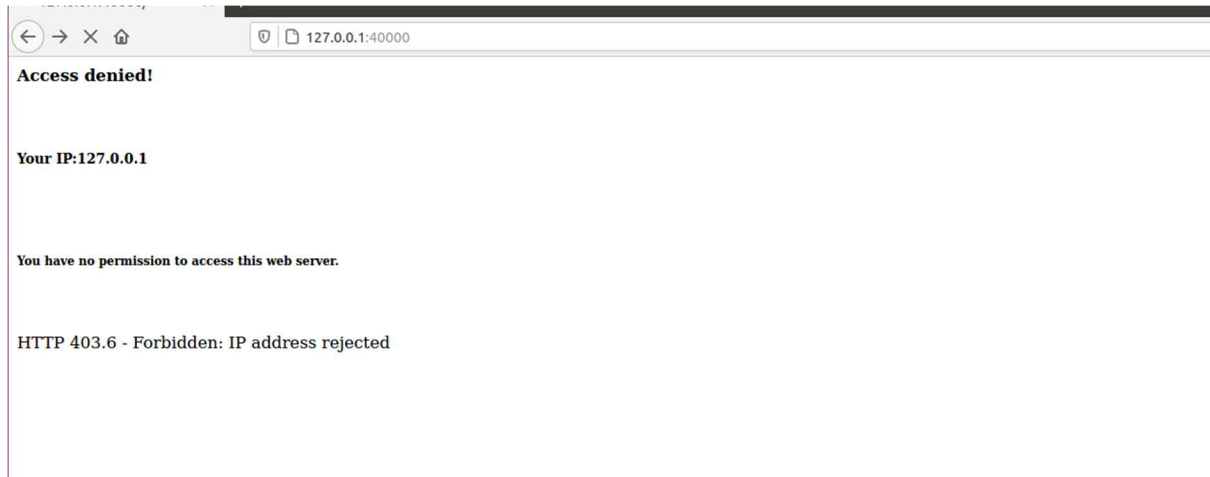
Welcome to System Programming Http

Directory path : /home/kw2019202005/mygit/Assignment/10week
total 96

Name	Permission	Link	Owner	Group	Size	Last Modified
2019202005_adv_server.c	-rw-rw-r--	1	kw2019202005	kw2019202005	29394	May 10 16:19
aaa	drwxrwxr-x	2	kw2019202005	kw2019202005	4096	May 10 16:19
accessible_usr	-rw-rw-r--	1	kw2019202005	kw2019202005	46	May 10 02:26
adv_server	-rwxrwxr-x	1	kw2019202005	kw2019202005	27728	May 10 16:19
bbb	drwxrwxr-x	2	kw2019202005	kw2019202005	4096	May 10 16:19
ccc	drwxrwxr-x	2	kw2019202005	kw2019202005	4096	May 10 16:19
Makefile	-rw-rw-r--	1	kw2019202005	kw2019202005	79	May 4 19:26
signal	-rwxrwxr-x	1	kw2019202005	kw2019202005	8904	May 4 19:26
signal.c	-rw-rw-r--	1	kw2019202005	kw2019202005	887	May 6 19:28

```
kw2019202005@ubuntu: ~/mygit/Assignment/10week
Port : 38625
=====
===== Disconnected Client =====
IP : 127.0.0.1
Port : 38625
=====
===== Connection History =====
Number of request(s) : 17
No. IP PID Port Time
8 127.0.0.1 17578 34017 Wed May 10 16:32:42 2023
9 127.0.0.1 17583 34529 Wed May 10 16:32:52 2023
10 127.0.0.1 17586 35041 Wed May 10 16:33:03 2023
11 127.0.0.1 17592 35553 Wed May 10 16:33:22 2023
12 127.0.0.1 17593 36065 Wed May 10 16:33:24 2023
13 127.0.0.1 17598 36577 Wed May 10 16:33:43 2023
14 127.0.0.1 17605 37089 Wed May 10 16:34:08 2023
15 127.0.0.1 17606 37601 Wed May 10 16:34:08 2023
16 127.0.0.1 17609 38113 Wed May 10 16:34:10 2023
17 127.0.0.1 17612 38625 Wed May 10 16:34:19 2023
```

다음은 request가 10번이 넘어갔을 때 출력되는 화면입니다. 접속 기록은 최대 10개만 출력해야 하며 최근 기록들만 출력해야 합니다. request가 10번 이후에 연결된 클라이언트의 개수만큼 이전의 연결된 클라이언트의 접속 기록을 출력하지 않고 request 10개 이후의 클라이언트의 정보를 출력합니다.



다음은 127.0.0.1의 ip주소가 accessible.usr파일의 목록에 들어있지 않은 경우입니다. 접근 제한 메시지를 출력하는 모습을 확인할 수 있습니다.

고찰

이번 과제는 이전 과제에서 작성한 웹 서버에 다중 접속 및 접근 제어 기능을 추가하는 것입니다. 다중 접속 지원을 위해 클라이언트와 소켓 연결 이후의 작업을 새로운 프로세스에서 수행하도록 구현해야 합니다. 이를 위해 fork() 함수를 사용할 수 있습니다.

그리고 10초에 한 번씩 연결되었던 클라이언트의 정보를 일괄 출력하기 위해서는 alarm함수를 이용했고 정보를 출력하는 함수를 만들어 signal handling을 통해 함수 호출 후 출력해주었습니다.

접근 제어 기능은 미리 허용한 IP를 가진 사용자만 서버에 접속할 수 있도록

하는 것입니다. 이를 위해 서버 프로그램 실행 이전에 접근 가능한 클라이언트의 목록을 파일로 저장하고, 연결을 요청한 클라이언트의 IP를 기준으로 판단하여 접근 가능 여부를 판단해야 합니다. 이를 처리하기 위해서는 fnmatch함수를 이용하여 처리하였습니다. fnmatch함수를 이용하여 클라이언트의 ip주소와 파일 안에 있는 ip주소가 겹친다면 기존의 작업을 수행하고 만약 겹치지 않는다면 접근 제한 메시지를 출력해주었습니다.

위에서 (pid=fork())==0 이렇게 자식 프로세스 생성 후 자식 프로세스에 코드를 작성하고 마지막에 부모 프로세스에서 코드를 작성할 때 (pid=fork())>0 이렇게 작성하고 계속 진행해서 제가 원하는 값이 나오지 않았습니다. fork는 위에서 자식 프로세스 만들 때 한번 했으니 pid>0 이렇게 작성하는 것이 맞는데 실수로 fork를 한번 더 해서 결과 값에 오류가 있었습니다. 그래도 수정하여 원하는 결과 값을 얻을 수 있었습니다.

Reference

2023년 1학기 시스템프로그래밍 & 시스템 프로그래밍 실습 강의자료

Assignment 2-3

2023년 1학기 시스템 프로그래밍 1학기 강의자료 3. Files and directories

2023년 1학기 시스템 프로그래밍 1학기 강의자료 6. sockets

2023년 1학기 시스템 프로그래밍 실습 10주차 강의자료 Advanced web server