Assignment#3-2

시스템 프로그래밍 실습

제출일: 5월 10일 금요일

분 반: 화요일

담당 교수: 신영주

학 번: 2015722025

학 과: 컴퓨터정보공학부

이 름: 정용훈

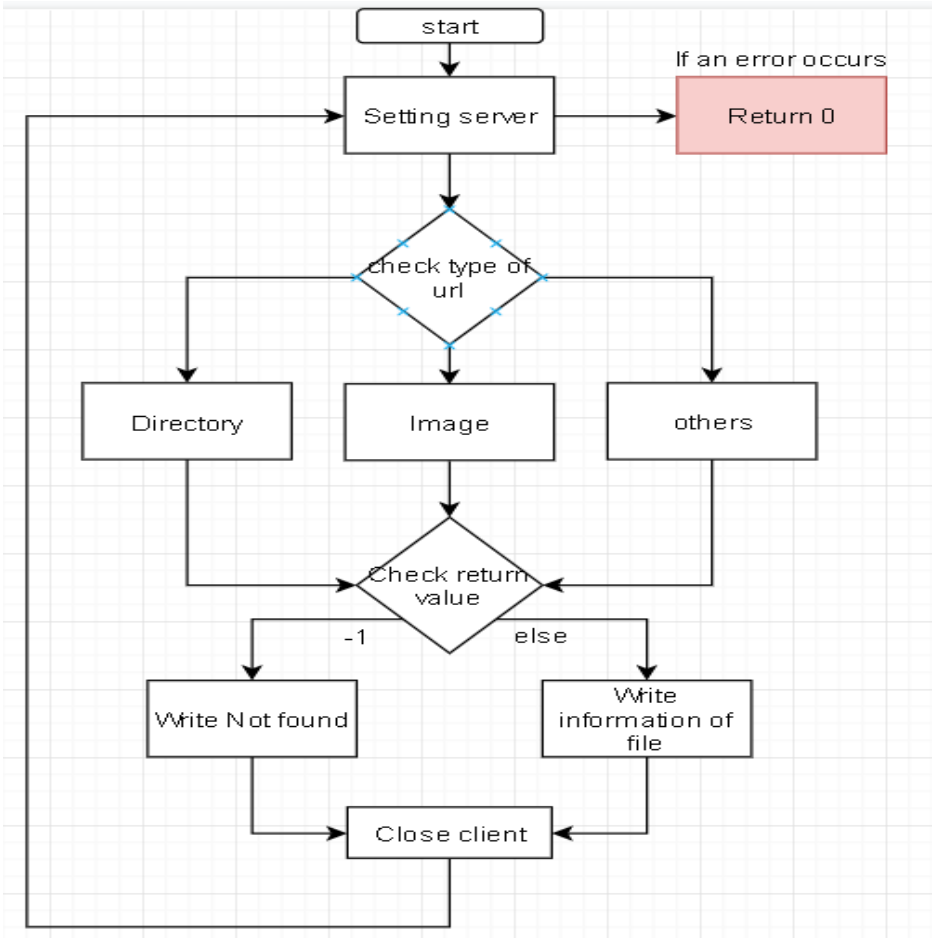
1. Introduction

3-2과제는 전과제인 html파일을 기반으로 서버와 클라이언트의 기본적인 동작을 이해하고 동작을 하기 위한 함수들과 Http 서버의 request message와 response message의 기본적인 구조를 이해하는데 목적을 가지고 있다. 또한 추가적으로 이미지, text, 실행파일들을 어떤 방식으로 클라이언트에게 보낼 수 있는지 알아볼 수 있다.

2. Flow Chart

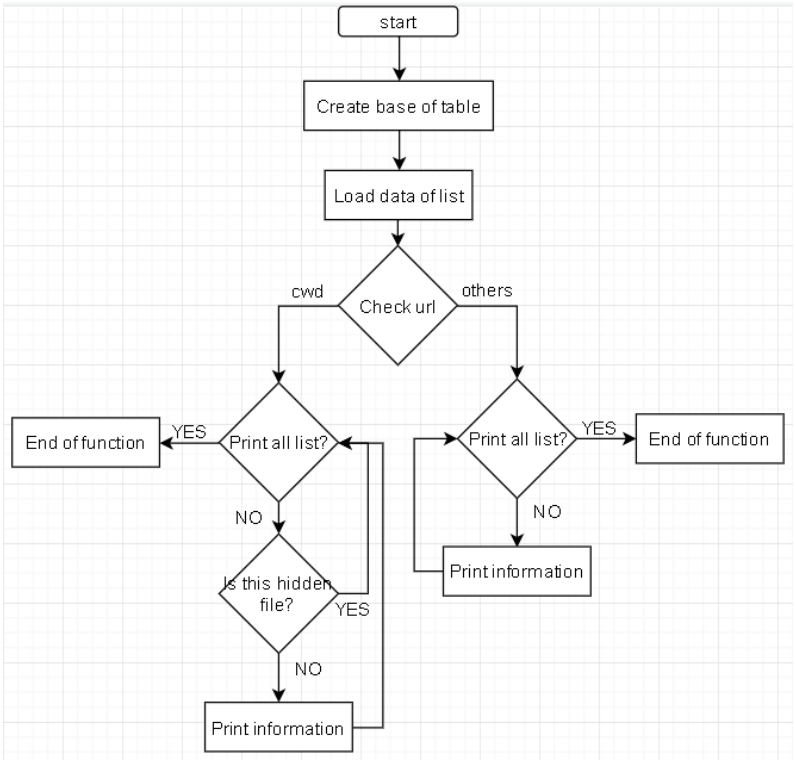
이번 3-2과제에서는 Main함수의 구조적인 변화와 Image파일, Directory파일, 그리고 나머지 파일들을 처리할 수 있는 함수 3가지가 추가적으로 사용되었다. –a와 -l옵션만 쓰이므로 skip되는 코드들이 있다.

Main



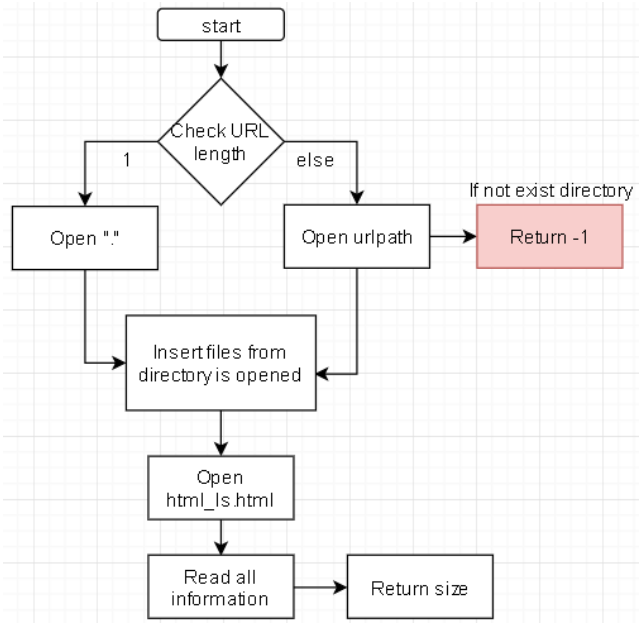
Main함수의 구조는 굉장히 간단하다. 실습시간에 배운 내용을 기반으로 서버를 setting해주고 url의 내용을 판단하여 해당 파일을 directory로 처리할지 이미지로 처리할지 나머지 경우로 처리할지 정해주면 된다.

PrintHTML(create file of html_ls.html)



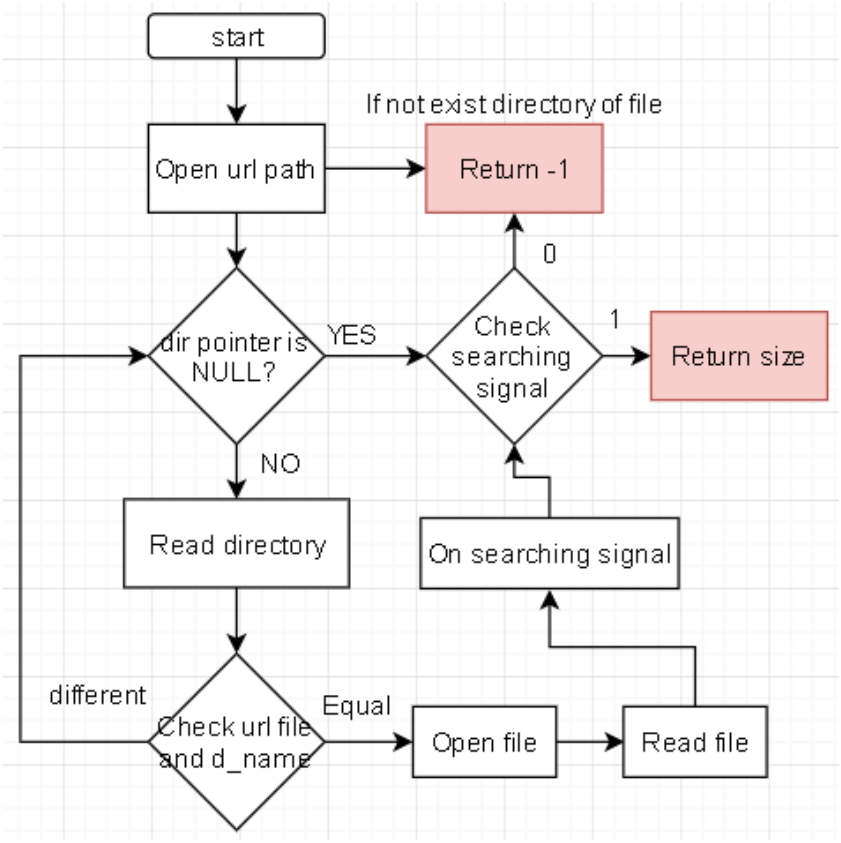
전 과제인 html_ls.hml을 만드는 과정으로 옵션 구현이 –a과 –al만 있기 때문에 구조만 동작 자체는 똑같지만 구조를 변경하게 되었다. 해당 함수는 Html함수에서 call하는 함수로써 아래 Html함수를 참고하면 이해하기 쉽다.

Html



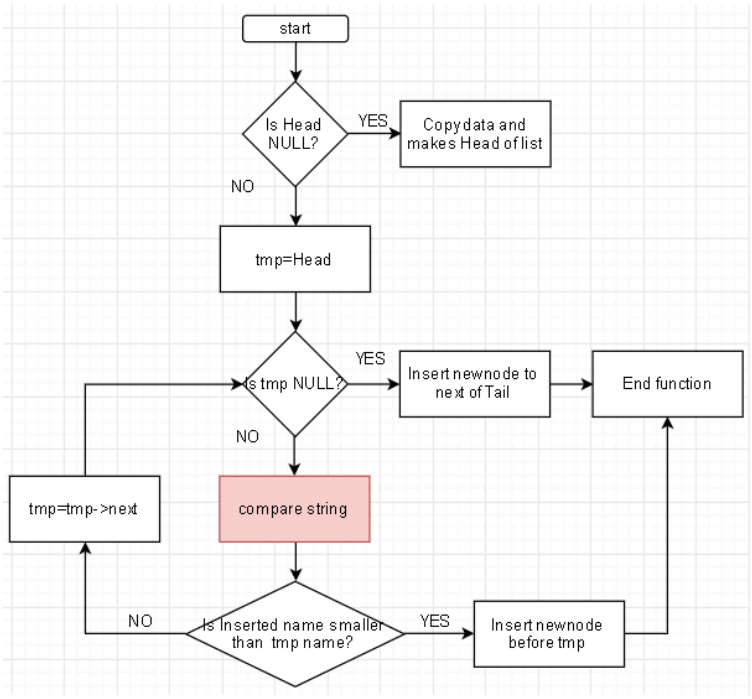
Open html을 하기 전 과정이 printHTML로써 html파일을 생성한 후 실행하게 된다.

Image & Normal (others)



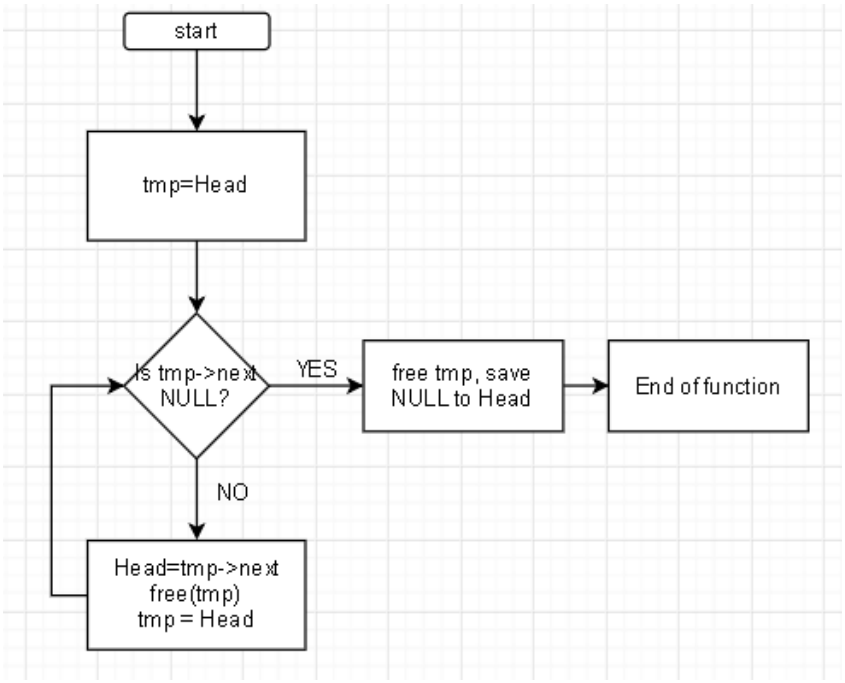
다음 함수는 Image와 나머지 다른 파일들을 처리해주는 함수로써 정보를 binary로 읽어와 write해주는 작업을 하게 된다 사실 image와 나머지 파일은 모두 binary를 통하여 읽을 수 있기 때문에 나눠줄 필요는 없다고 생각된다.

Insertnode



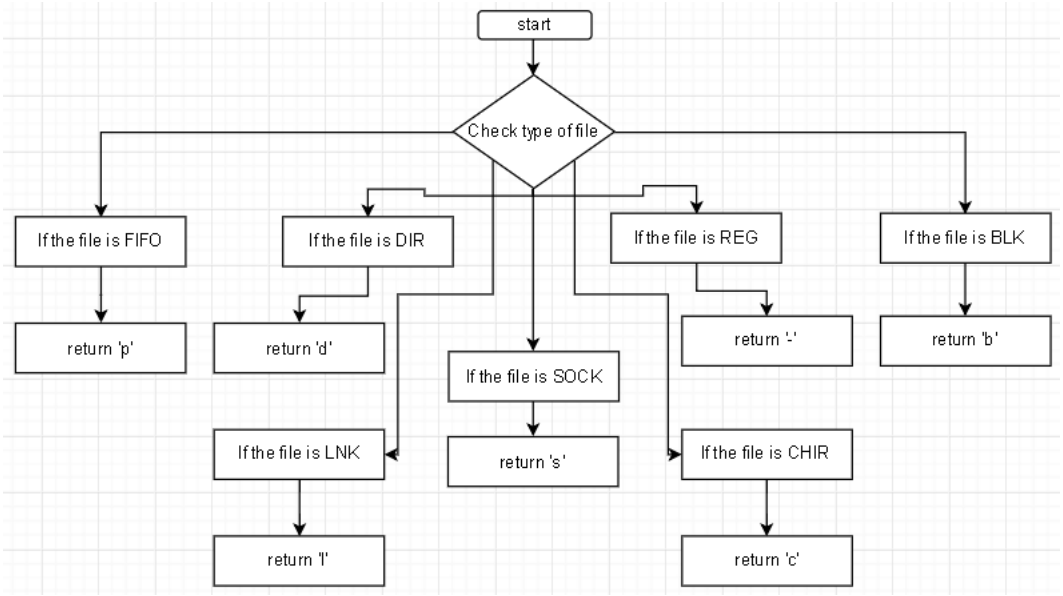
따로 sort함수를 사용하지 않고 insert를 실행 할 때부터 sort가 되며, list가 생성된다. Insert함수는 기존 함수와 동일하게 사용되므로 전체적인 변화는 없지만, S옵션을 사용면 size를 비교해야 하므로 compare를 하는 부분이 변경되게 된다. 해당 문제는 함수는 같고 조건만 바꿔주어 해결하였다.

Deletelist



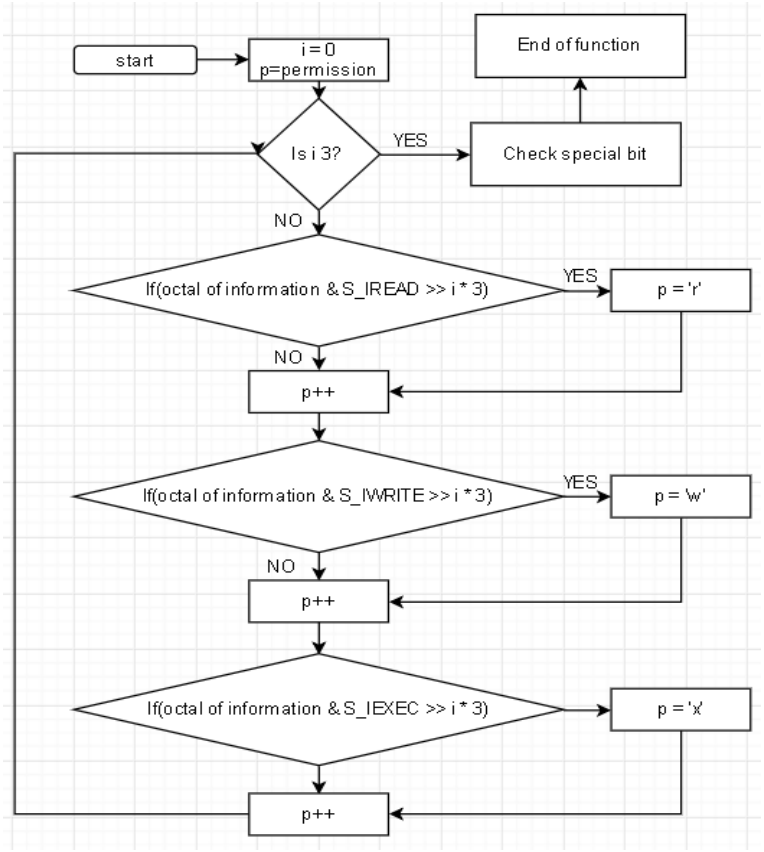
모든 정보 출력 후 linked list를 제거하는 함수다.

printType



파일의 정보를 받아와 st_mode를 통하여 파일의 type을 정의하는 함수다

printPerm



St_mode를 받아와 해당 파일의 permission을 확인하여, 최종적인 permission을 출력할 수 있도록 도와주는 함수다.

3. Pseudo code

Main

```
int main(int argc, char** argv)
{
        Declare value for using main function;

        ////////////////////////Server value////////////////////////
        struct sockaddr_in server_addr, client_addr;
        int socket_fd, client_fd;
        int len, len_out;
        int opt = 1;
        ////////////////////////////////////////////////////////////

        load current working directory;

        ////////////////////////For Connecting Server////////////////////////

        setting socket;
        setting socket opt;

        memset(&server_addr, 0, sizeof(server_addr));
        server_addr.sin_family = AF_INET;
        server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
        server_addr.sin_port = htons(PORTNO);

        practice bind;

        listen(socket_fd, 5);
        ////////////////////////////////////////////////////////////////////////
```

```
while (1) //start server
{
	struct in_addr inet_client_address;
	char Sbuf[BUFSIZE] = { 0, };
	char tmp[BUFSIZE] = { 0, };
	char response_header[BUFSIZE] = { 0, };
	char response_message[BUFSIZE] = { 0, };
	char url[URL_LEN] = { 0, };
	char method[20] = { 0, };
	char * tok = NULL;

	len = sizeof(client_addr);

	client_fd=accept;
	load url;
	////////////////////////////Declare head////////////////////////////////////////////////
	if(The signal is image file)
	{
		long unsigned int filesize=0; //file size
		unsigned char image_message[3000000]={0, }; //message buffer
		filesize=Image(urlname,client_fd,image_message); //return size of file

		if(filesize==-1) //check not found
			send Not found response

		else
			send information of file

		write(client_fd, response_header, strlen(response_header)); //send header
		write(client_fd, image_message, filesize); //send entity
	}
```

```
                else if(The signal is directory)
                {       long unsigned int filesize=0; //file size
                        unsigned char html_message[3000000]={0, }; //message buffer
                        filesize=Html(url,client_fd,html_message); //return size of file

                        if(filesize==-1) //check not found
                                send Not found response

                        else
                                send information of file

                        write(client_fd, response_header, strlen(response_header)); //send header
                        write(client_fd, html_message, filesize); //send entity
                }

                else //type of others
                {
                        long unsigned int filesize=0; //file size
                        unsigned char normal_message[3000000]={0, }; //message buffer
                        filesize=Normal(urlname,client_fd,normal_message); //return size of file

                                if(filesize==-1) //check not found
                                send Not found response

                        else
                                send information of file

                        write(client_fd, response_header, strlen(response_header)); //send header
                        write(client_fd, normal_message, filesize); //send entity

                }

                  close(client_fd); //close client
        }
        close(socket_fd); //close socket
        return 0;
}
```

Normal & Image

```
int Normal & Image(char *urlname,int client_fd,unsigned char *normal_message)
{
        char urlpath[256]={0};
        char urlfile[256]={0};
        char cwd[256]={0};
        char Openpath[256]={0};
        char Dirpath[256]={0};
        int searching=0;

        unsigned char response_message[3000000]={0, };
        struct stat buf;
        DIR *dirp;
        struct dirent *dir;

        load current working directory;
        urlpath = urlname

        Make Open path;

//////////////////////open dir for checking image file//////////////////////////
        if(open directory==NULL)
                return -1;

        change directory;

                do
                {
                        read directory;

                        If(read is NULL)
                                return -1;

                        else if(urlfile is equal d_name)
                        {
                                searching=1;
                                FILE *file=NULL;
                                int ch;

                                file=Open d_name file;

                                while(Repeat read file before EOF)
                                        normal_message[count++]=ch;

                                close file;
                                break;
                        }
                } while (1);
        change directory to home;

        if(searching==1)
                return count;

        return -1;
}
```

Html

```
int Html(char *url,int client_fd,unsigned char *html_message)
{
        DIR *dirp;
        struct dirent *dir;
        struct stat buf;
        struct group *gid;
        struct passwd *uid;
        struct tm *time;
        FILE *htmlfile;
        struct sockaddr_in server_addr, client_addr;
        struct in_addr inet_client_address;

        int total = 0,count=0;
        char response_message[3000000] = { 0, };
        char NULLpath[256]={0, };

        ////////////////////////////////////Open DIR and wirte//////////////////////.
        Open html file that created from printHTML function;

        if(strlen(url)==1)
        {
                Opswitch=1;
                open current directory;
                change directory;

                do
                {
                        read directory;

                        if (read is NULL)
                                break;
```

```
                        else
                        {
                                load information of file to buf;
                                total += buf.st_blocks / 2;
                                Insert node;
                        }

                } while (1);
        }

        else
        {
                Opswitch=0;
                char urlpath[256]={0,};
                make url;

                if(Not exist directory)
                        return -1;
                change working directory;

                do
                {
                        read directory;

                        if (read is NULL)
                                break;

                  else
                  {
                        load information of file to buf;
                        total += buf.st_blocks / 2;
                        Insert node;
                  }

          } while (1);
}


if(root path)
{
        strcpy(response_message, "<h1>Welcome to System Programming Http</h1><br>₩n"); //copy
        fprintf(htmlfile,"<h1>Welcome to System Programming Http</h1><br>₩n"); //write to file
}

else
{
        strcpy(response_message, "<h1>System Programming Http</h1><br>₩n"); //copy
        fprintf(htmlfile,"<h1>System Programming Http</h1><br>₩n"); //write to file
}
```

```
		practice printHTML; //create html file

		deletelist(); //delete list
		change working directory;
		close html file;

		FILE *file=NULL;
		Open html file;

		int ch;
		while(Repeat read file before EOF)
			html_message[count++]=ch;

		close file;

		return count;
}
```

PrintlistHTML

```
void printHTML(int client_fd, FILE* file, int flagA, int flagL, int total, char *dirpath)
{
        struct group *gid;
        struct passwd *uid;
        struct tm *time;
        char buff[256];
        char cwd[256];
        struct stat buf;
        struct dirent *dir;

        int k = 0, m = 0;
        char subpath[256] = { 0 };
        char subdirpath[256] = { 0 };
        char checkpath[256] = { 0 };
        char color[256] = { 0 };

        int createflag = 0;
        char content[BUFSIZE] = { 0, };

        load current working directory;
        For making path;

        Declare information of Directory(path, total);
```

```
        if (createflag == 0)
        {
                Declare format of table;
                createflag = 1;
        }

        for (Node* tmp=Head; tmp; tmp = tmp->next) //Repeat function for printing all list
        {
                if(Opswitch==1&&tmp->filename[0]=='.')
                        continue;
                if(!strcmp(tmp->filename,"html_ls.html"))
                        continue

                lstat(tmp->filename, &buf);
                Declare color of file;
                Make hyperlink;

                if(url's mean is current("."))
                        urlpath[strlen(urlpath)-1]='₩0';
                else
                        strcat(urlpath,tmp->filename); //make full format of url path

                uid = getpwuid(buf.st_uid);
                gid = getgrgid(buf.st_gid);

                time = load information of local time;

                if (File type is Link)
                        Make path use "->"


                Print full format;

        }

        End table and html
        return;
}
```

Nodeinsert

```
void NodeInsert(char* name)
{
        char Ename[256];
        char Etmpname[256];
        Node* tmp = Head;
        Node* prevnode;

        if (Head of list is NULL)
        {
                Makes newnode;
                strcpy(newnode->filename, name);
                newnode->next = NULL;
                newnode->prev = NULL;

                Head = newnode;
                return;
        }

    else
    {
            while (tmp is not NULL)
            {
                    strcpy(Ename, name);
                    strcpy(Etmpname, tmp->filename);

                    if (checkstring(Ename))
                            Eliminate character of '.'
                    if (checkstring(Etmpname))
                            Eliminate character of '.'

                    if Etmpname is bigger than name)
                    {
                            if (tmp is Head)
                            {
                                    Makes newnode;
                                    strcpy(newnode->filename, name);
                                    newnode->prev = NULL;

                                    newnode->next = Head;
                                    Head->prev = newnode;
                                    Head = newnode;
                                    return;
                            }
```

```
                                        else
                                        {
                                                Makes newnode;
                                                strcpy(newnode->filename, name);
                                                newnode->next = tmp;
                                                newnode->prev = tmp->prev;
                                                tmp->prev->next = newnode;
                                                tmp->prev = newnode;
                                                return;
                                        }
                                }
                                else continue;
                        }
                        Move prev to Tail;
                        Makes newnode;

                        strcpy(newnode->filename, name);
                        newnode->next = NULL;
                        prevnode->next = newnode;
                        newnode->prev = prevnode;
                        Tail = newnode;
                        return;

                }
        }
```

Function of others

```
void Eliminate(char *str, char ch)
{
	while (;befor check all character of string;str++)
	{
		if (*str == ch)
		{
			strcpy(str, str + 1);
			str--;
			return;
		}
	}
}

void deletelist()
{
	Node* tmp = Head;
	for (; tmp->next != NULL;)
	{
		Head = tmp->next;
		free(tmp);
		tmp = Head;
	}
	free(tmp);
	Head = NULL;
}
```

```
char printType(mode_t mode)
{
        switch (mode & S_IFMT)
        {
        case S_IFREG:
                return('-');
        case S_IFDIR:
                return('d');
        case S_IFCHR:
                return('c');
        case S_IFBLK:
                return('b');
        case S_IFLNK:
                return('l');
        case S_IFIFO:
                return('p');
        case S_IFSOCK:
                return('s');
        }

        return('?');
}

char *printPerm(mode_t mode)
{
        int i;
        char *p;
        static char perms[10];
        p = perms;
        strcpy(perms, "---------");

        for (i = 0; i < 3; i++)
        {
                if (mode & (S_IREAD >> i * 3))
                        *p = 'r';
                p++;

                if (mode & (S_IWRITE >> i * 3))
                        *p = 'w';
                p++;

                if (mode & (S_IEXEC >> i * 3))
                        *p = 'x';
                p++;
        }
        if ((mode & S_ISUID) != 0)
                perms[2] = 's';

        if ((mode & S_ISGID) != 0)
                perms[5] = 's';

        if ((mode & S_ISVTX) != 0)
                perms[8] = 't';

        return(perms);
}
```

```
void printOph(long int size)
{
	int k=0,flag=0;
	double sub_size;

	sub_size=(double)size;

	for(Repeat until undivided)
	{
		if(sub_size>(double)1024)
			sub_size/=1024;

		else
			break;
	}

	Execute unit alignment process

	if(If the decimal point is not zero) //check first decimal point
		flag=1;

	int size_int=0;

	if(k==0)
	{

		size_int=(int)sub_size;
		printf(integer output of size_int);

	}

	else if(k==1)
	{
		if(flag==1)
		{
			size_int=(int)sub_size;
			printf(integer output of size_int,"K");
		}
		else
			printf(float output of sub_size,"K");
	}

	else if(k==2)
	{
		if(flag==1)
		{
			size_int=(int)sub_size;
			printf(integer output of size_int,"M");
		}
		else
			printf(float output of sub_size,"M");
	}
```

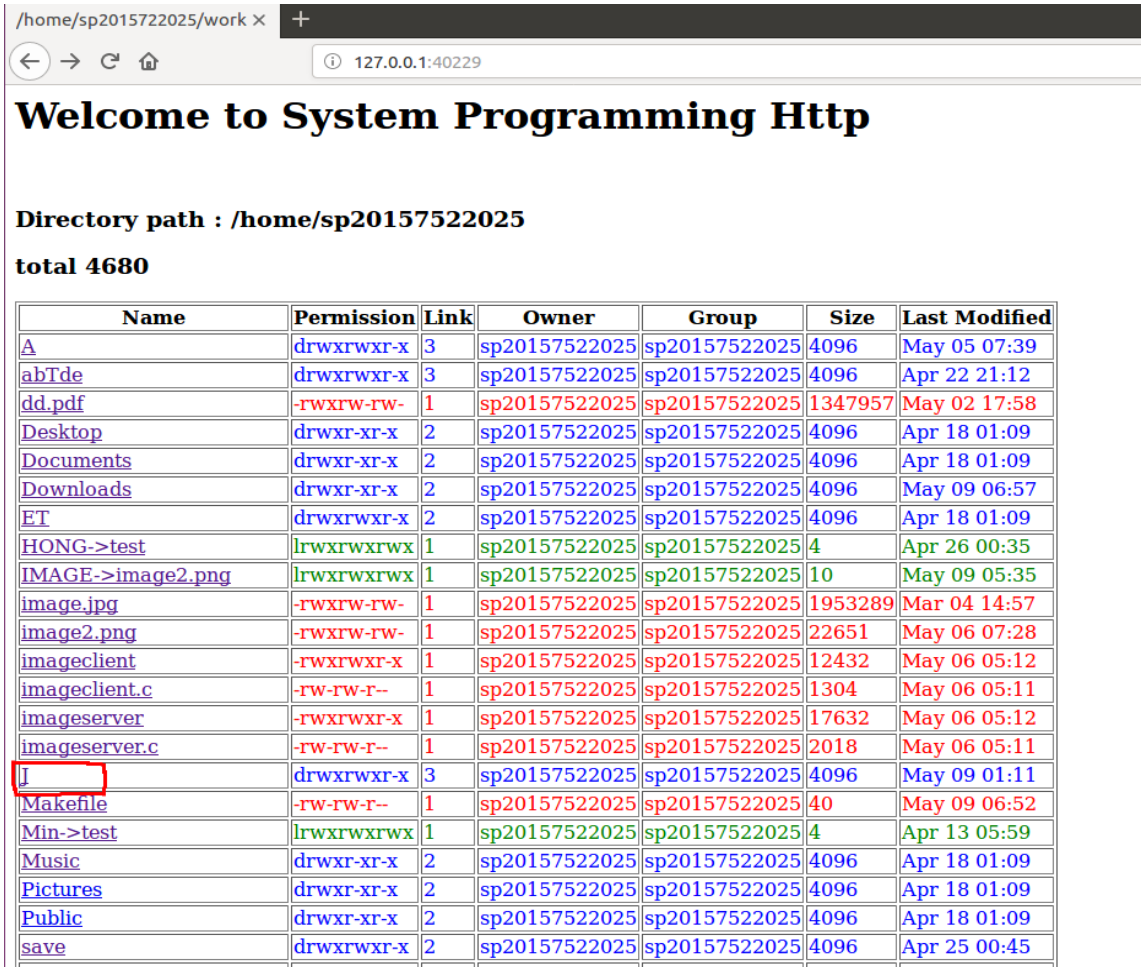
```
        else if(k==3)
        {
                if(flag==1)
                {
                        size_int=(int)sub_size;
                        printf(integer output of size_int,"G");
                }
                else
                        printf(float output of sub_size,"G");
        }

        else if(k==4)
        {
                if(flag==1)
                {
                        size_int=(int)sub_size;
                        printf(integer output of size_int,"T");
                }
                else
                        printf(float output of sub_size,"T");
        }
        return;
}
```

```
int matchfunction(char (*paname)[256], int argc)
{
	char Matchcmd[256][256]={0};
	int matchcount=0;

////////////////////////////////////////////////////////searching match command////////////////////////////////
	while(read all command)
	{
		while(read all character of one string)
		{
			if(string has '*' or '?') //if the command has '*' or '?'
			{
				copy string to command vector
				matchcount++; //increase
				break;
			}
			else if(string has '[x-y]' format) //if the command has 'index'
			{
				copy string to command vector
				matchcount++; //increase
				break;
			}
			else if(string has '[x]' format)

			{
				copy string to command vector
				matchcount++; //increase
				break;
			}
		}
	}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////delete match cmd/////////////////////////////////////////////////////

	Remove extracted commands from existing vectors

////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
int newargc=argc; //integer
int flag=0; //integer
struct dirent *dir;
struct stat buf;
DIR *dirp;
char pathname[256][256]={0};
char cmd[256][256]={0};

while(checking all string) //for check match command
{
        To separate into path and command parts
}


while(Repeat by match count) //for check command
{
        int currentflag=0; //declare flag

        if(string length is 0) //check path name
                dirp=opendir(".");
        else
        {
                currentflag=1; //on flag
                dirp=opendir(pathname[j]); //open pathname directory
        }
        change directory;

    if(In case there is no matching command) //Check null
    {
            Save back to original vector;
            newargc++;
            continue;
    }
    do
    {

            dir = readdir(dirp); //read file
            if (dir == NULL) //check NULL
                    break; //stop repeat functsion

            else
            {
                    if(!fnmatch(pattern[i],file name,0)&&(dir->d_name[0]!='.')) //function of match
                    {
                            Perform a function that makes it the original path;

                            Save the filename that matches the original vector;
                            newargc++;
                            flag=1; //on flag
                    }
            }

    } while (1);


    if(flag==0) //if flag is 0
    {
            Save back to original vector;
            newargc++;
    }
    flag=0; //off flag
    change pointer dirp to start
}
return newargc; //return new vector count

}
```

4. Result

PDF 예시



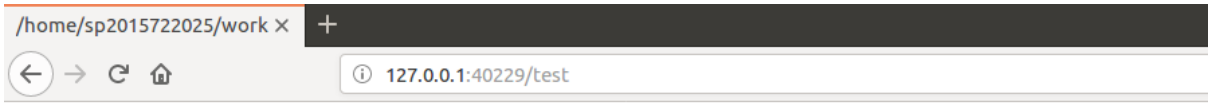
Name	Permission	Link	Owner	Group	Size	Last Modified
A	drwxrwxr-x	3	sp20157522025	sp20157522025	4096	May 05 07:39
abTde	drwxrwxr-x	3	sp20157522025	sp20157522025	4096	Apr 22 21:12
dd.pdf	-rwxrw-rw-	1	sp20157522025	sp20157522025	1347957	May 02 17:58
Desktop	drwxr-xr-x	2	sp20157522025	sp20157522025	4096	Apr 18 01:09
Documents	drwxr-xr-x	2	sp20157522025	sp20157522025	4096	Apr 18 01:09
Downloads	drwxr-xr-x	2	sp20157522025	sp20157522025	4096	May 09 06:57
ET	drwxrwxr-x	2	sp20157522025	sp20157522025	4096	Apr 18 01:09
HONG->test	lrwxrwxrwx	1	sp20157522025	sp20157522025	4	Apr 26 00:35
IMAGE->image2.png	lrwxrwxrwx	1	sp20157522025	sp20157522025	10	May 09 05:35
image.jpg	-rwxrw-rw-	1	sp20157522025	sp20157522025	1953289	Mar 04 14:57
image2.png	-rwxrw-rw-	1	sp20157522025	sp20157522025	22651	May 06 07:28
imageclient	-rwxrwxr-x	1	sp20157522025	sp20157522025	12432	May 06 05:12
imageclient.c	-rw-rw-r--	1	sp20157522025	sp20157522025	1304	May 06 05:11
imageserver	-rwxrwxr-x	1	sp20157522025	sp20157522025	17632	May 06 05:12
imageserver.c	-rw-rw-r--	1	sp20157522025	sp20157522025	2018	May 06 05:11
J	drwxrwxr-x	3	sp20157522025	sp20157522025	4096	May 09 01:11
Makefile	-rw-rw-r--	1	sp20157522025	sp20157522025	40	May 09 06:52
Min->test	lrwxrwxrwx	1	sp20157522025	sp20157522025	4	Apr 13 05:59
Music	drwxr-xr-x	2	sp20157522025	sp20157522025	4096	Apr 18 01:09
Pictures	drwxr-xr-x	2	sp20157522025	sp20157522025	4096	Apr 18 01:09
Public	drwxr-xr-x	2	sp20157522025	sp20157522025	4096	Apr 18 01:09
save	drwxrwxr-x	2	sp20157522025	sp20157522025	4096	Apr 25 00:45

System Programming Http

Directory path : /home/sp20157522025/J

total 12

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	3	sp20157522025	sp20157522025	4096	May 09 01:11
..	drwxr-xr-x	26	sp20157522025	sp20157522025	4096	May 09 17:27
A	-rw-rw-r--	1	sp20157522025	sp20157522025	12	Mar 29 18:41
K	drwxrwxr-x	2	sp20157522025	sp20157522025	4096	May 09 01:09

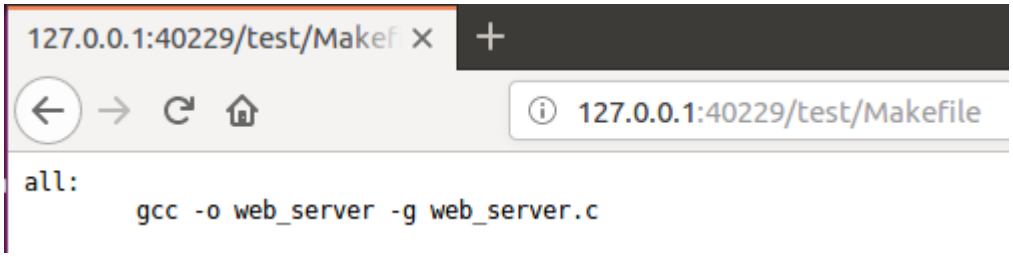


System Programming Http

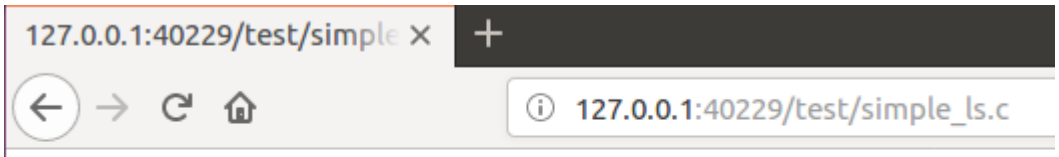
Directory path : /home/sp20157522025/test

total 952

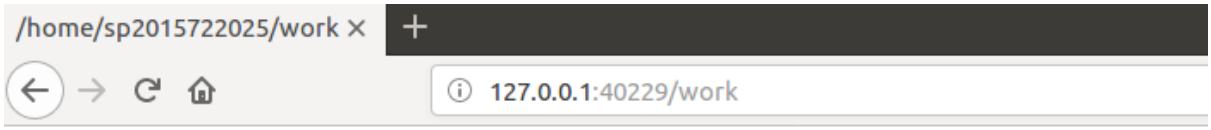
Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	5	sp20157522025	sp20157522025	4096	May 09 11:43
..	drwxr-xr-x	26	sp20157522025	sp20157522025	4096	May 09 17:27
.A.swp	-rw-------	1	sp20157522025	sp20157522025	12288	Mar 27 17:43
CC	drwxrwxr-x	2	sp20157522025	sp20157522025	4096	Apr 26 01:02
HH	drwxrwxr-x	2	sp20157522025	sp20157522025	4096	Apr 26 01:02
Makefile	-rw-rw-r--	1	sp20157522025	sp20157522025	40	May 09 06:52
oo	drwxrwxr-x	2	sp20157522025	sp20157522025	4096	Apr 26 01:02
simple_ls	-rwxrwxr-x	1	sp20157522025	sp20157522025	9840	Apr 18 17:31
simple_ls.c	-rw-rw-r--	1	sp20157522025	sp20157522025	365	Apr 18 17:31
spls_advanced	-rwxrwxr-x	1	sp20157522025	sp20157522025	36640	Apr 20 04:17
tue_2-1_2015722025 (1).tar.gz	-rwxrw-rw-	1	sp20157522025	sp20157522025	365912	Apr 02 05:35
tue_2-1_2015722025.pdf	-rwxrw-rw-	1	sp20157522025	sp20157522025	419771	Apr 02 03:39
web_server	-rwxrwxr-x	1	sp20157522025	sp20157522025	52888	May 09 09:38
web_server.c	-rw-rw-r--	1	sp20157522025	sp20157522025	55665	May 09 11:43




```
all:
	gcc -o web_server -g web_server.c
```




```
#include<stdio.h>
#include<string.h>

int main(int argc,char** argv)
{
	if(strcasecmp(argv[1],argv[2])==0)
		printf("Equal two tring\n");
	else if(strcasecmp(argv[1],argv[2])>0)
		printf("%s is bigger than %s\n",argv[1],argv[2]);
	else if(strcasecmp(argv[1],argv[2])<0)
		printf("%s is smaller than %s\n",argv[1],argv[2]);

	printf("%d   %d\n",argv[1],argv[2]);
}
```

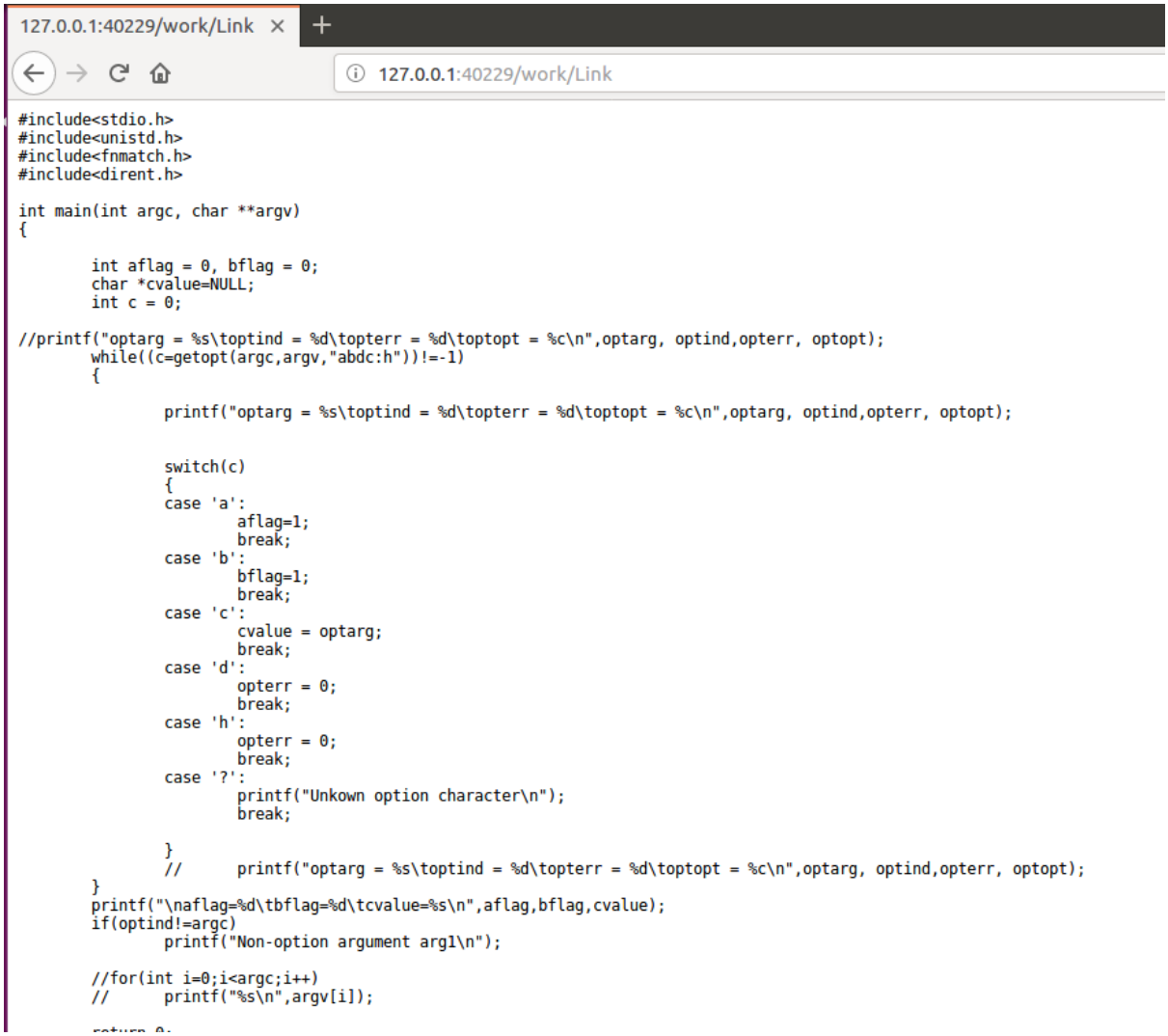


System Programming Http

Directory path : /home/sp20157522025/work

total 16

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	2	sp20157522025	sp20157522025	4096	May 09 17:33
..	drwxr-xr-x	26	sp20157522025	sp20157522025	4096	May 09 17:33
A.c	-rw-rw-r--	1	sp20157522025	sp20157522025	2598	Apr 26 17:32
Link->A.c	lrwxrwxrwx	1	sp20157522025	sp20157522025	3	May 09 17:33
run	-rwxrwxr-x	1	sp20157522025	sp20157522025	8880	Apr 26 17:32




```
#include<stdio.h>
#include<unistd.h>
#include<fnmatch.h>
#include<dirent.h>

int main(int argc, char **argv)
{

        int aflag = 0, bflag = 0;
        char *cvalue=NULL;
        int c = 0;

//printf("optarg = %s\toptind = %d\topterr = %d\toptopt = %c\n",optarg, optind,opterr, optopt);
        while((c=getopt(argc,argv,"abdc:h"))!=-1)
        {

                printf("optarg = %s\toptind = %d\topterr = %d\toptopt = %c\n",optarg, optind,opterr, optopt);


                switch(c)
                {
                case 'a':
                        aflag=1;
                        break;
                case 'b':
                        bflag=1;
                        break;
                case 'c':
                        cvalue = optarg;
                        break;
                case 'd':
                        opterr = 0;
                        break;
                case 'h':
                        opterr = 0;
                        break;
                case '?':
                        printf("Unkown option character\n");
                        break;

                }
                //      printf("optarg = %s\toptind = %d\topterr = %d\toptopt = %c\n",optarg, optind,opterr, optopt);
        }
        printf("\naflag=%d\tbflag=%d\tcvalue=%s\n",aflag,bflag,cvalue);
        if(optind!=argc)
                printf("Non-option argument argl\n");

        //for(int i=0;i<argc;i++)
        //      printf("%s\n",argv[i]);

        return 0;
```

/home/sp2015722025/work ×

127.0.0.1:40229/A

System Programming Http

Directory path : /home/sp20157522025/A

total 180

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	3	sp20157522025	sp20157522025	4096	May 05 07:39
..	drwxr-xr-x	26	sp20157522025	sp20157522025	4096	May 09 17:33
...	-rw-rw-r--	1	sp20157522025	sp20157522025	0	Apr 04 21:06
A	-rw-rw-r--	1	sp20157522025	sp20157522025	0	Apr 04 21:06
a	-rw-rw-r--	1	sp20157522025	sp20157522025	0	Apr 04 21:06
D	-rw-rw-r--	1	sp20157522025	sp20157522025	0	Apr 04 21:06
dddd	drwxrwxr-x	2	sp20157522025	sp20157522025	4096	Apr 25 06:50
run	-rwxrwxr-x	1	sp20157522025	sp20157522025	27224	Apr 26 02:41
spls_advanced	-rwxrwxr-x	1	sp20157522025	sp20157522025	36640	Apr 22 21:11
spls_advanced.c	-rw-rw-r--	1	sp20157522025	sp20157522025	29669	Apr 26 02:41
spls_advanced.o	-rw-rw-r--	1	sp20157522025	sp20157522025	26240	Apr 25 17:38
test_ls	-rwxrwxr-x	1	sp20157522025	sp20157522025	48048	May 02 00:00
X	-rw-rw-r--	1	sp20157522025	sp20157522025	0	Apr 04 21:06

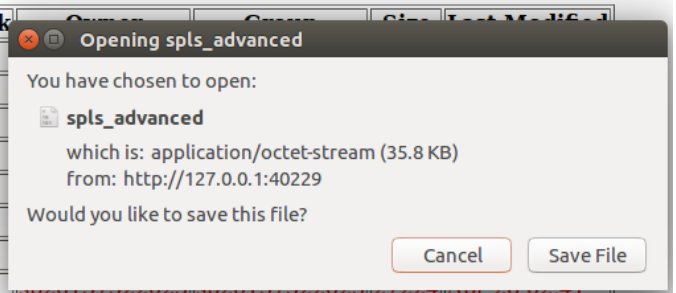
/home/sp2015722025/work ×

127.0.0.1:40229/A

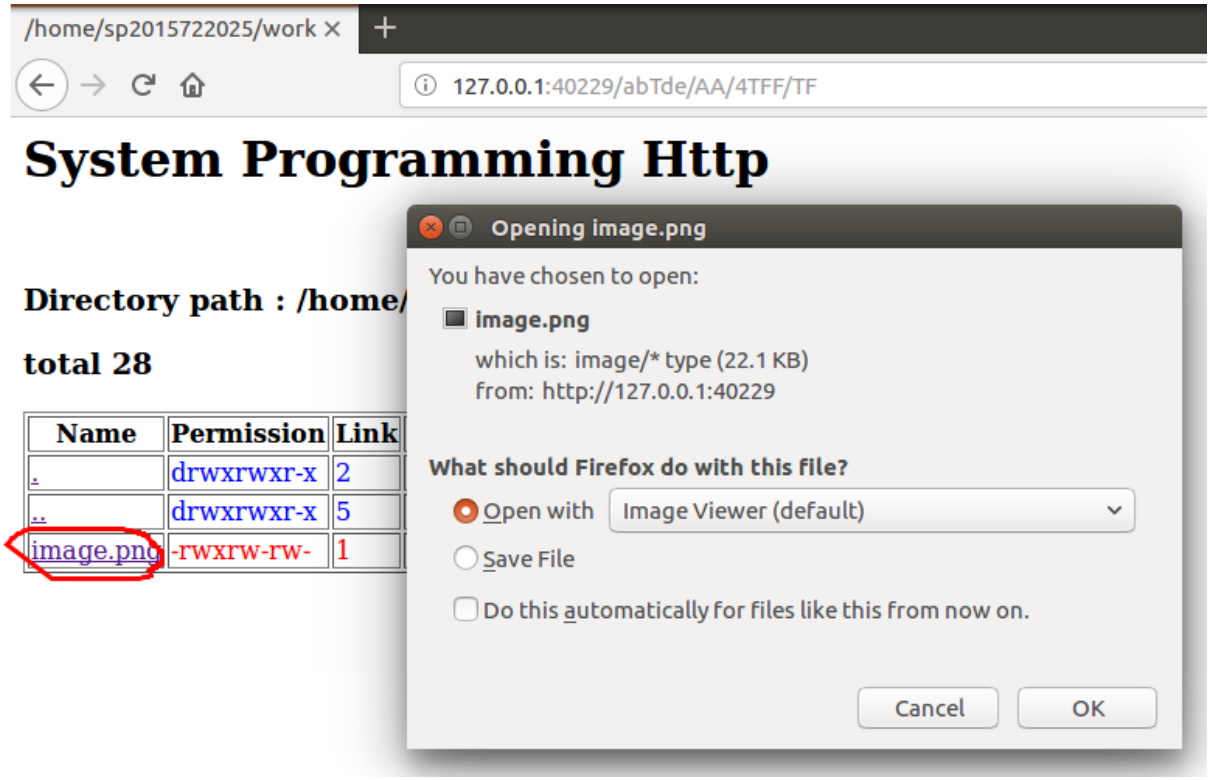
System Programming Http

Directory path : /home/sp20157522025/A

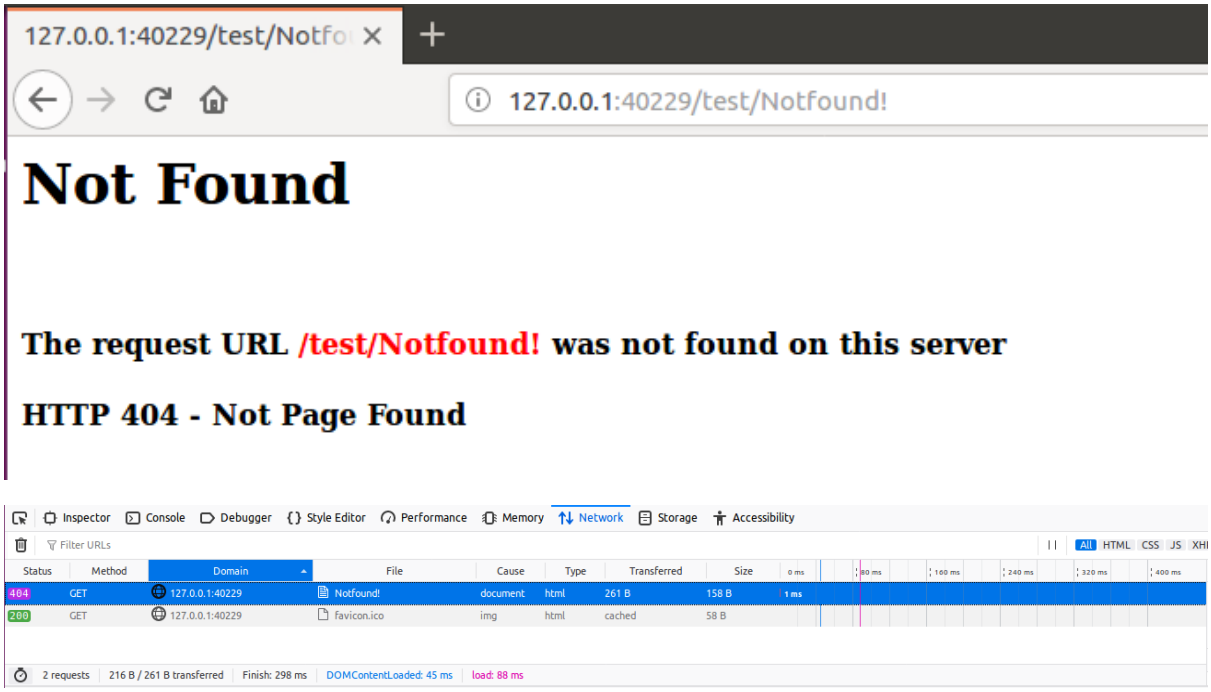
total 180



Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	3				
..	drwxr-xr-x	26				
...	-rw-rw-r--	1				
A	-rw-rw-r--	1				
a	-rw-rw-r--	1				
D	-rw-rw-r--	1				
dddd	drwxrwxr-x	2				
run	-rwxrwxr-x	1				
spls_advanced	-rwxrwxr-x	1	sp20157522025	sp20157522025	36640	Apr 22 21:11
spls_advanced.c	-rw-rw-r--	1	sp20157522025	sp20157522025	29669	Apr 26 02:41
spls_advanced.o	-rw-rw-r--	1	sp20157522025	sp20157522025	26240	Apr 25 17:38
test_ls	-rwxrwxr-x	1	sp20157522025	sp20157522025	48048	May 02 00:00
X	-rw-rw-r--	1	sp20157522025	sp20157522025	0	Apr 04 21:06


/home/sp2015722025/work
127.0.0.1:40229/abTde/AA/4TFF/TF
System Programming Http
Directory path : /home/
total 28
Name
Permission
Link
.
drwxrwxr-x
2
..
drwxrwxr-x
5
image.png
-rwxrw-rw-
1
Opening image.png
You have chosen to open:
image.png
which is: image/* type (22.1 KB)
from: http://127.0.0.1:40229
What should Firefox do with this file?
Open with
Image Viewer (default)
Save File
Do this automatically for files like this from now on.
Cancel
OK


/home/sp2015722025/work
127.0.0.1:40229/abTde/AA/4TFF/TF
System Programming Http
Directory path : /home/s
total 28
Name
Permission
Link
.
drwxrwxr-x
2
..
drwxrwxr-x
5
image.png
-rwxrw-rw-
1
image.png
FIFA ONLINE
4 by EA SP...



5. Conclusion

3-2과제는 지금까지 했던 과제들의 구조를 바꿔야 하는 까다로움이 있었다. 이유로는 response메시지에 파일들의 길이를 입력해주어야 하기 때문에 받아오는 url을 통하여 어떤 방식의 처리를 할지 먼저 정한 후 그에 맞는 동작을 실행하고 얻어오는 파일의 사이즈를 return 해주어야 하기 때문에 전체적인 함수의 순서를 바꾸게 되었다. 또한 이미지 파일을 얻어오는데 있어 많은 어려움이 있었는데 그 이유는 response 메시지를 잘 이해하지 못하고 내용의 길이를 정상적으로 전달해주지 못하여 발생하는 문제였다. 그리고 파일을 열고 읽어서 보내주는 방식이 아니라 하나씩 write를 하는 방법으로 (구)코드를 구현했었는데 이는 코드가 굉장히 불안정하여 제대로 출력되지 않거나 write도중 서버가 튕기는 현상도 생기는 것을 확인할 수 있었다.