

Assignment#3-1

시스템 프로그래밍 실습

제출일: 5월 03일 금요일

분 반: 화요일

담당 교수: 신영주

학 번: 2015722025

학 과: 컴퓨터정보공학부

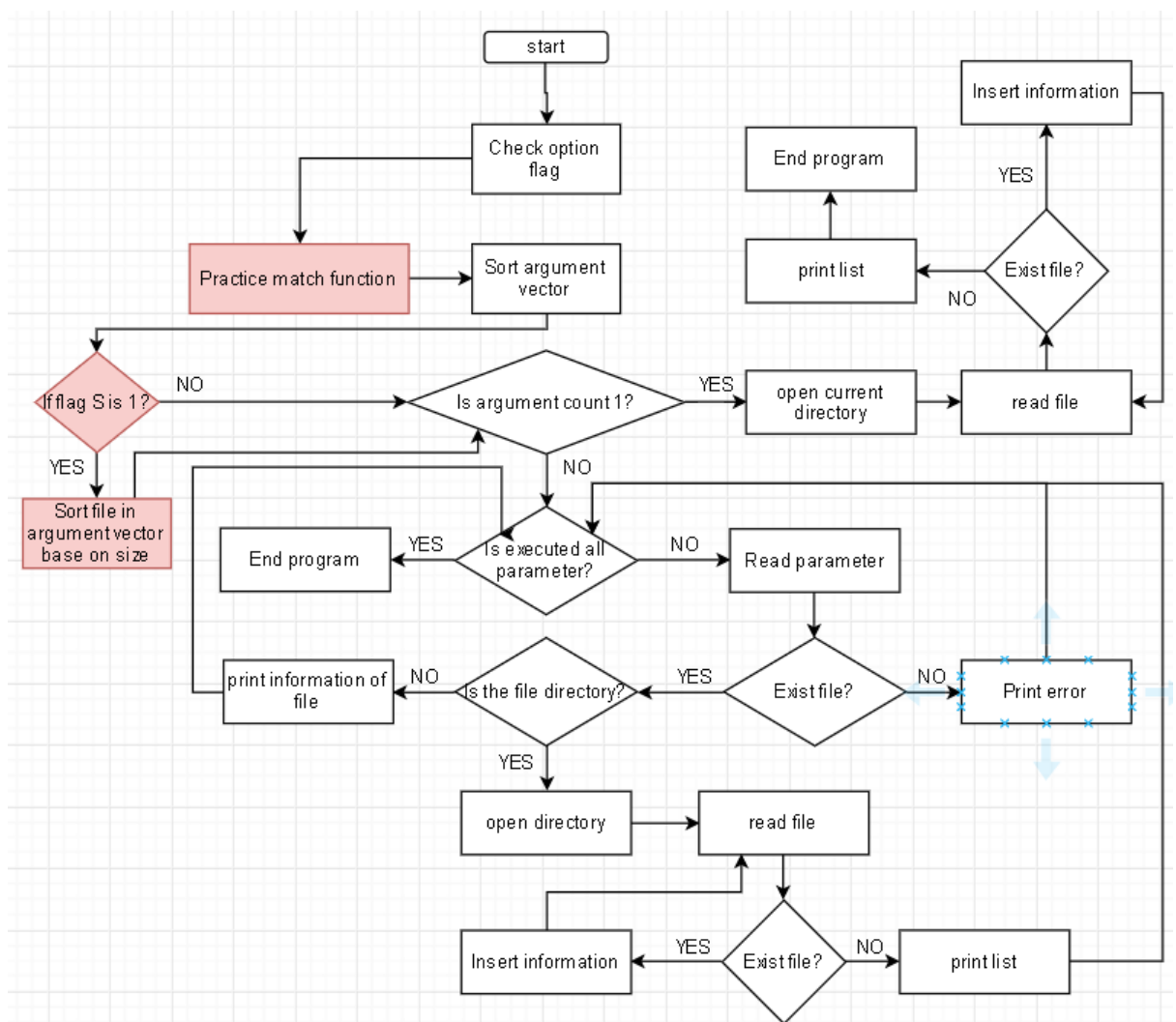
이 름: 정용훈

1. Introduction

3-1과제는 단순 html출력을 익히는 과제다. fopen함수를 통하여 파일을 오픈 하여 html출력 형식에 맞추어 html파일을 생성하고 해당 파일을 실행시키면 평소 콘솔에서 확인하는 결과가 아닌 제공된 pdf처럼 출력형식을 나오게 하면 된다. 출력을 위한 알고리즘이나 함수는 전 과제와 동일하며, 함수들의 상세 동작은 아래 목록에서 확인할 수 있다.

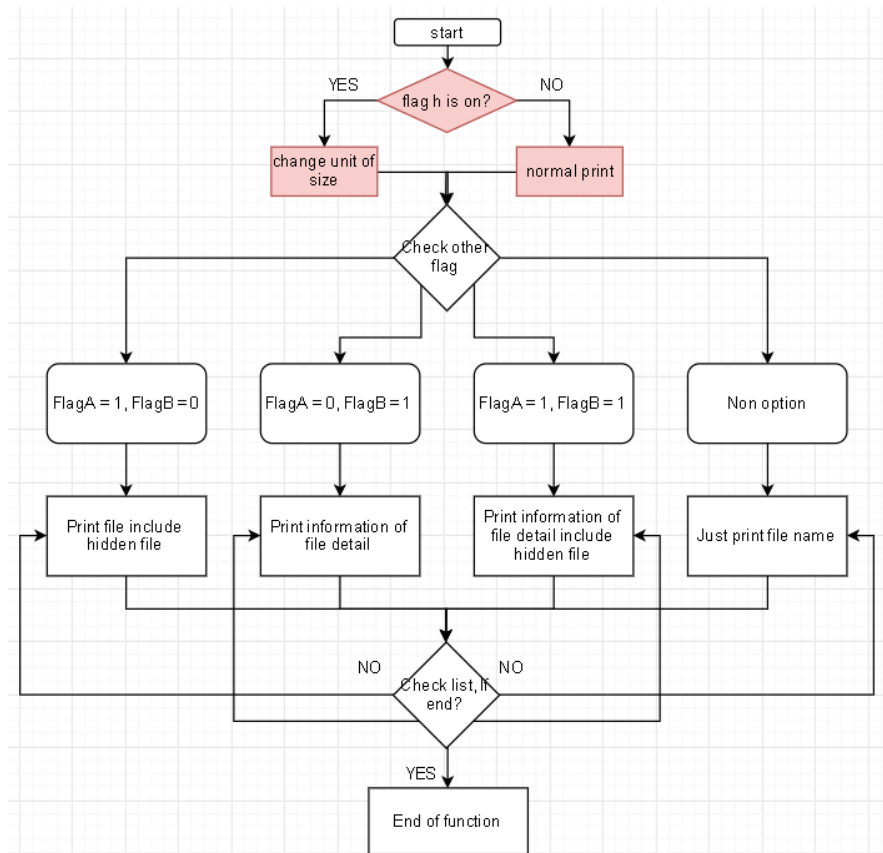
2. Flow Char

Main



메인함수의 전체적인 동작을 나타낸 flow chart다. 이전 과제(final_ls)와 다른 점으로는 Print 함수가 콘솔출력이 아닌 html format에 알맞은 출력으로 변경되었다는 것이며, 그러므로 콘솔에서는 직접적으로 값을 확인할 수 없다.

PrintlistHTML



2-3차 과제까지 콘솔에 출력을 해주는 함수였다. 이번 과제로 넘어오면서 File open을 통하여 file을 열고 fprintf함수를 사용하여 HTML파일에 직접 write하여 browser를 통하여 결과를 확인할 수 있도록 설계되어있다. 출력 format에만 변경사항이 있으며, 나머지 파일을 추출하는 알고리즘은 전 과제와 동일하게 설계되어있다.

PrintlistHTML (More about option r)

다음은 옵션 r의 관련된 내용이다. Print 함수의 전체 동작은 위와 같지만 r 옵션이 들어오면, print자체를 반대로 해야 하기 때문에 포인터의 이동이 다음과 같이 변경된다. 구현 자체를 이중 링크드리스트로 했기에 방문순서를 변경 하는 코드는 간단하다.

기존 print



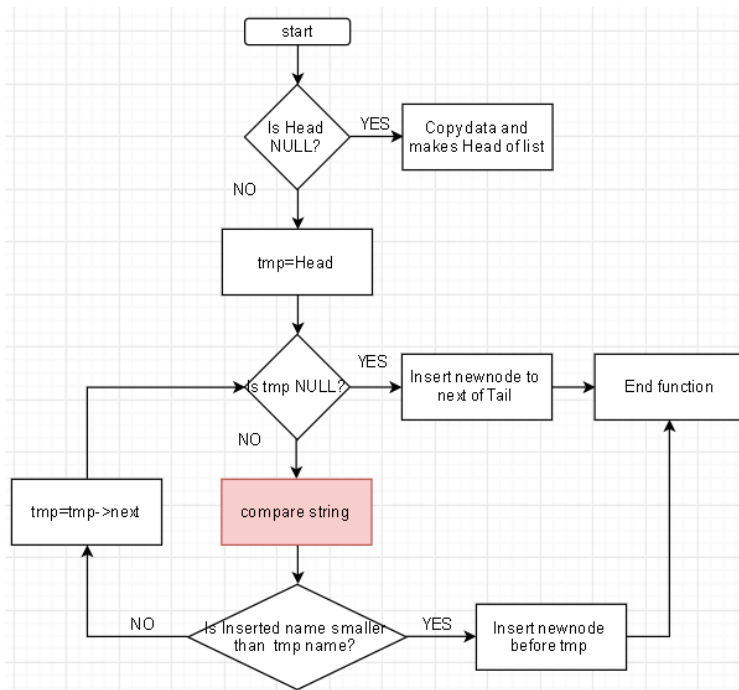
출력 순서 -> Head, 2, 3, 4, Tail

변경된 print



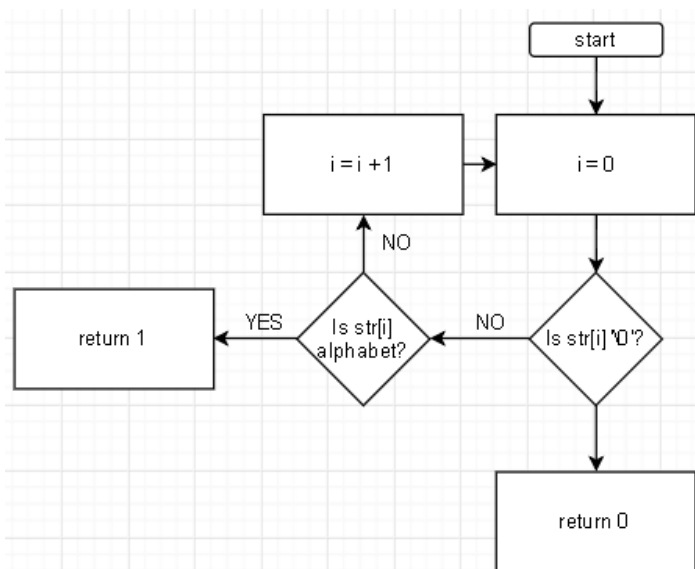
출력 순서 -> Tail, 4, 3, 2, Head

Insertnode



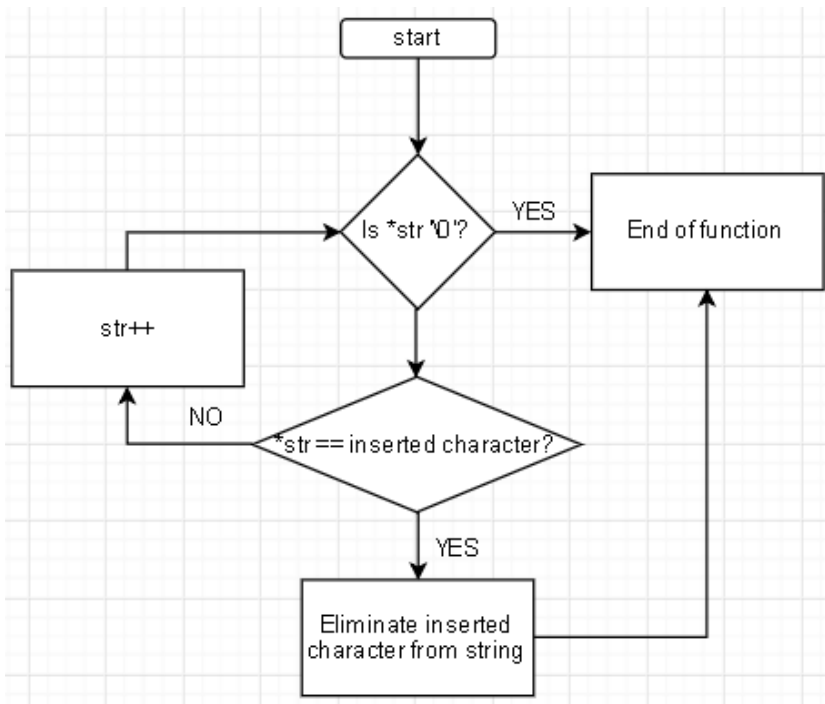
따로 sort함수를 사용하지 않고 insert를 실행 할 때부터 sort가 되며, list가 생성된다. Insert함수는 기존 함수와 동일하게 사용되므로 전체적인 변화는 없지만, S옵션을 사용하면 size를 비교해야 하므로 compare를 하는 부분이 변경되게 된다. 해당 문제는 함수는 같고 조건만 바꿔주어 해결하였다.

Checkstring



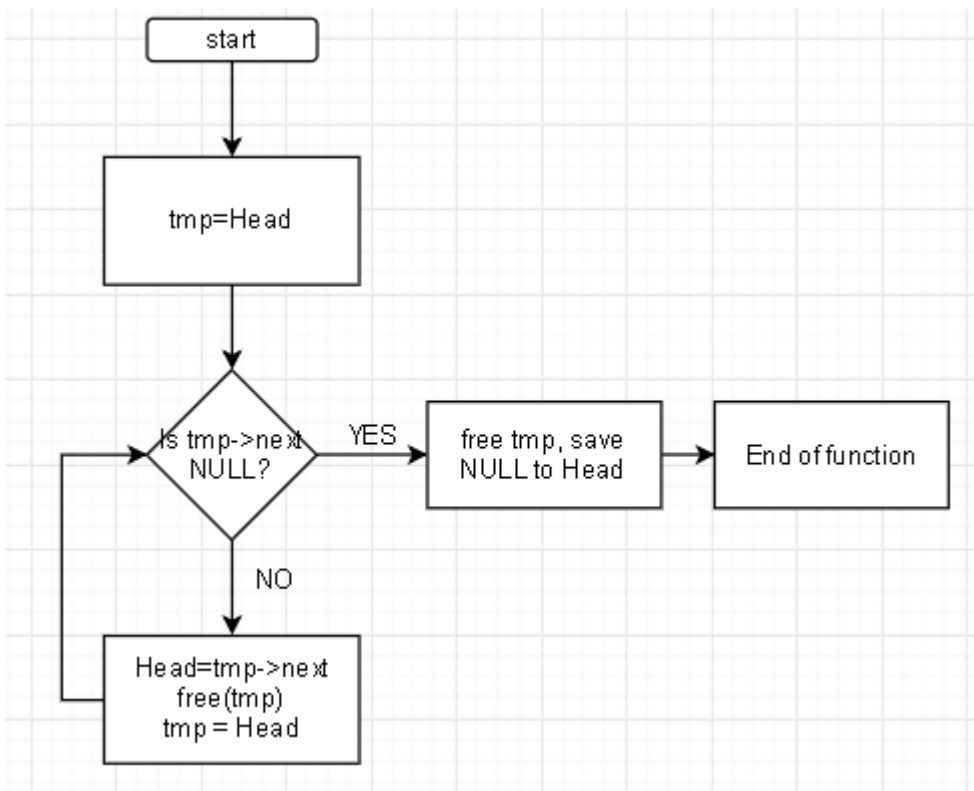
Eliminate와 연계되는 함수로써 알파벳이 아닌 문자로만 이루어진 string일 경우 '.'을 제거하지 않기 위한 함수다. 해당 함수도 변경사항이 없다.

Eliminate



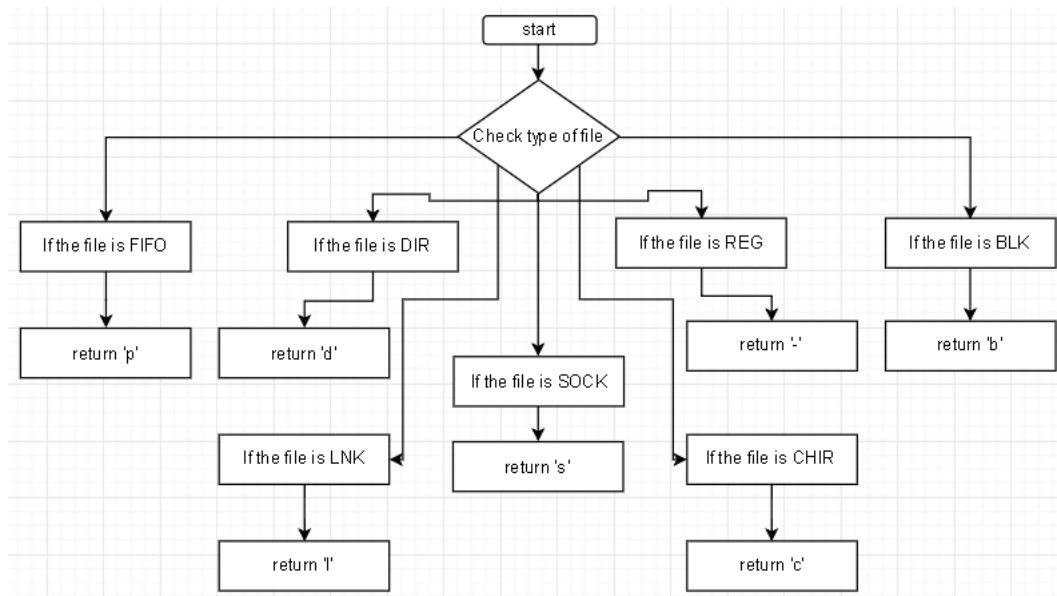
Hidden file의 이름을 비교하기 위해 '.'문자나 '/'문자를 제거하는 함수다.

Deletelist



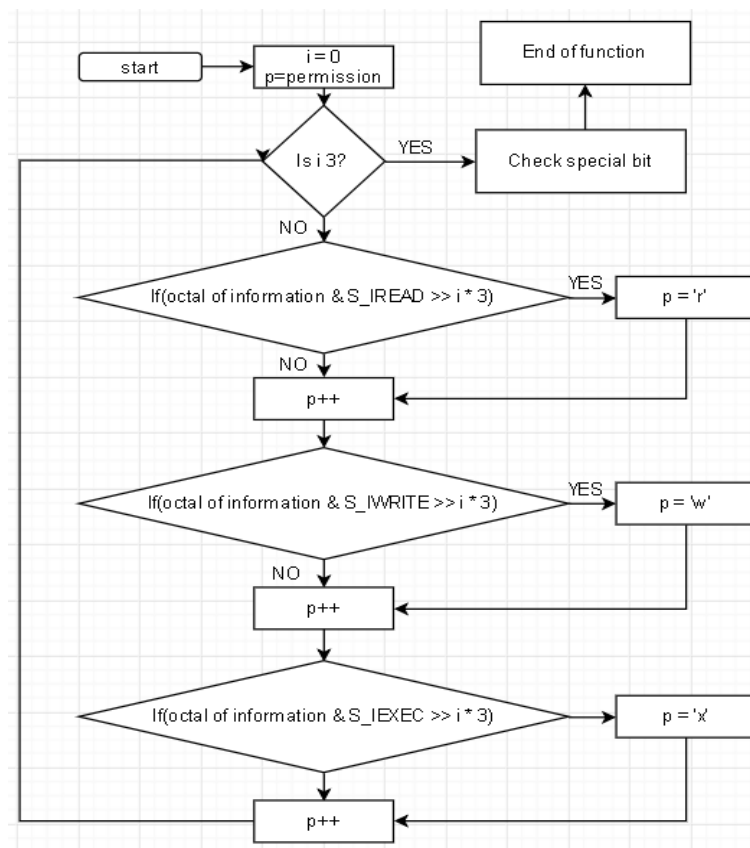
모든 정보 출력 후 linked list를 제거하는 함수다.

printType



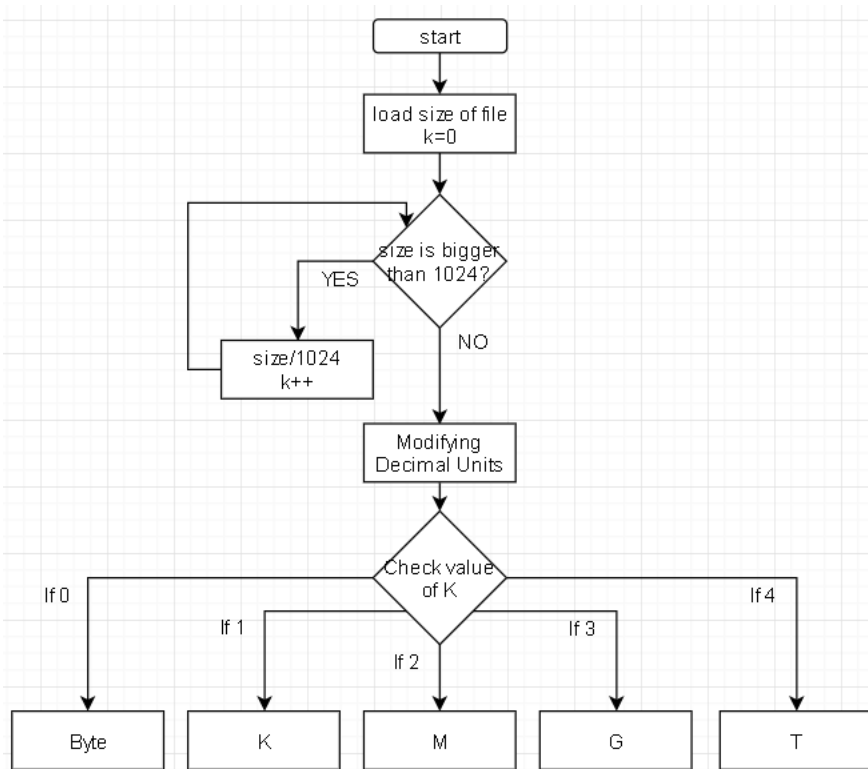
파일의 정보를 받아와 `st_mode`를 통하여 파일의 type을 정의하는 함수다

printPerm



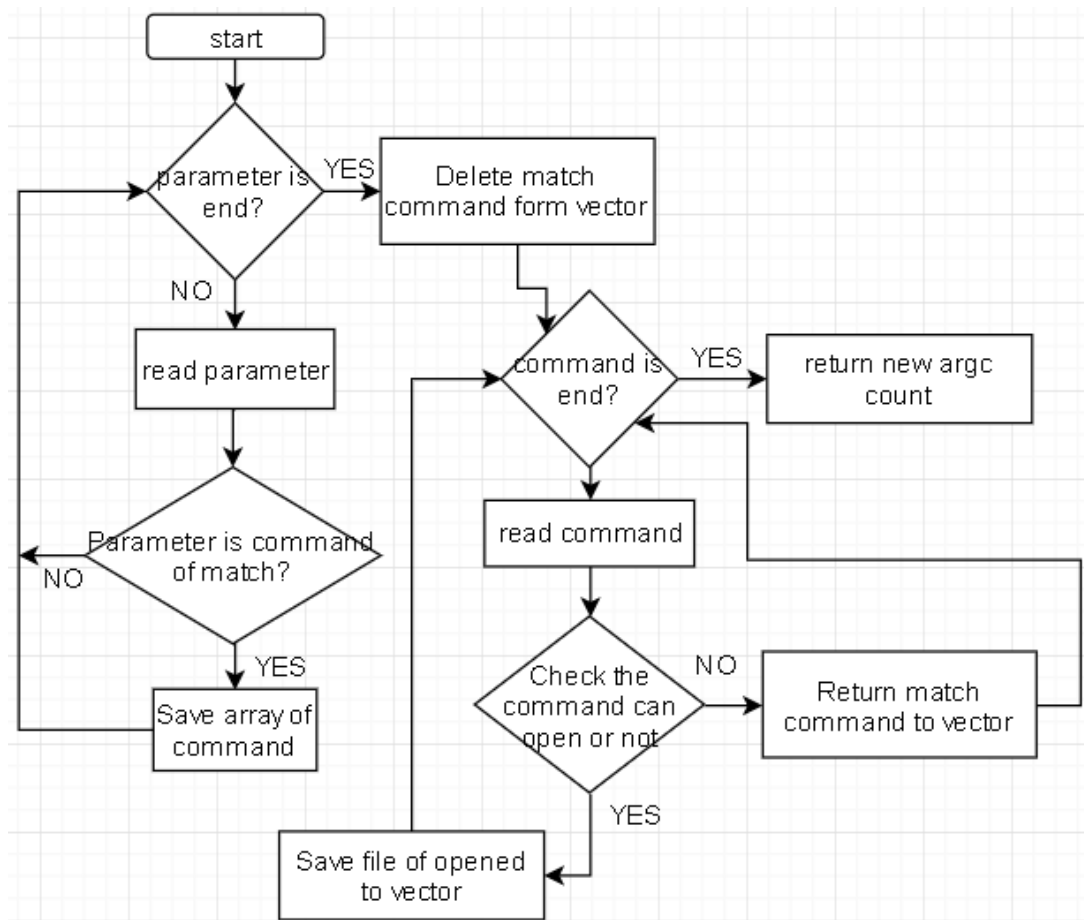
`St_mode`를 받아와 해당 파일의 permission을 확인하여, 최종적인 permission을 출력할 수 있도록 도와주는 함수다.

printOph



h옵션을 구현하기 위한 함수로써 단위를 맞추기 위해 Modifying과정이 필요하며, 그전에는 단순히 1024로 반복하여 나누어주면 된다. 실제 ls와 값이 차이가 나는데 이유는 직접 구현하면 값을 버리게 되고 실제 ls인 경우는 값을 반올림하기에 차이가 난다.

Matchfunction



다음 함수는 match명령이 인자로 들어오는지 판단하며, 해당 명령을 다른 vector에 옮겨 두었다가 해당 명령과 match되는 파일들을 vector로 옮겨 전 과제에서 구현하였던 함수를 그대로 수행할 수 있도록 도와주는 함수다. 경로 없이 단순히 match명령만 들어오게 된다면, 현재 directory를 열어 match시킨후 match되는 파일들을 열어주면 되고 경로와 함께 들어온다면, 경로와 match명령을 따로 읽어 경로에 해당하는 file들을 모두 출력해주면 된다. 단 조건으로 match명령은 인자의 끝에만 들어와야 하고 그 전 ex) test/*/* 이런 식의 명령은 다른 예외로 처리해주어야 한다.

3. Pseudo code

Main

```
int main(int argc, char** argv)
{
    int c = 0;
    int height = 2;
    DIR *dirp;

    struct dirent *dir;
    struct stat buf;
    struct group *gid;
    struct passwd *uid;

    struct tm *time;
    char buff[256];

    int argcount = 0;
    int total=0;
    char swap[256];
    char Subargv[256][256]={'\0'};
    char match[256];
    int count=1;
    char NULLpath[256]={0};
```

```

////////////////////////////////Set option////////////////////////////////
while (repeat function before checking all option)
{
    switch (value of option)
    {
        if option is a
            flagA = 1; //on flag
            break;
        if option is l
            flagL = 1; //on flag
            break;
        if option is h
            flagh=1; //on flag
            break;
        if option is S
            flagS=1; //on flag
            break;
        if option is r
            flagr=1; //on flag
            break;
    }
}

////////////////////////////////

////////////////////////////////move parameter for other working////////////////////////////////
char paname[256][256]={'\0'};
char sortname[256][256]={'\0'};
while(repeat function before all argument vector copies) //move content of parameter
    strcpy(paname[i],argv[i]);
////////////////////////////////

    practice function of matching for checking command of match //find match

////////////////////////////////Sorting argument vector////////////////////////////////
    Move argument vector to other string(sortname)

    check character of '/' from sortname[0]

    char tmp[256]={'\0'};

    To sort of vector that is has each command
    //////////////////////////////////

```

```

////////////////////////////////move parameter////////////////////////////////
while(brfor read all argv)
{
    if(First name of argv is '-')
    {
        strcpy(Subargv[count],argv[i]);
        count++;
    }
}

while(brfor read all argv)
{
    if(File is not read)
    {
        strcpy(Subargv[count],argv[i]);
        count++;
    }
}

while(brfor read all argv)
{
    Read file;
    if(File is not directory)
    {
        strcpy(Subargv[count],argv[i]);
        count++;
    }
}

while(brfor read all argv)
{
    if(File is directory)
    {
        strcpy(Subargv[count],argv[i]);
        count++;
    }
}
////////////////////////////////

```

```

if(option is S) //check option of S(size)
    sortingfile(Subargv,argc); //practice sorting

if (parameter nubmer is 1)
{
    dirp = opendir(current directory)

    do
    {
        dir = readdir(dirp);

        if (File is not opened)
            break;

        else
        {
            lstat(dir->d_name,&buf);
            InsertNode;
        }

    } while (1);

    Print List
    return 0;
}

```

```

while (befor read all argv)
{
    if (File is not opended)
    {
        Print error;
        continue;
    }
    else
    {
        if (File is directory)
        {
            dirp = open directroy

            do
            {
                dir = readdir(dirp);

                if (Directory is not opended)
                    break;
                else
                {
                    lstat(dir->d_name,&buf);
                    InsertNode;
                }
            } while (1);

            Print List
        }
        else
            printfile(flagA,flagL,Subargv[i]);
    }
}

return 0;
}

```

PrintlistHTML

```
void printHTML(FILE* file,int flagA, int flagL, int total, int argc,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;

    cwd=current working directory; //load current working directory

    if (FlagL or argumentcount > 2) //check option if '-l'
    {
        for(;cwd[k]!='\0';k+);
        for(;cwd[k]!='/';k--);
        for(;m<k;m++)
            subpath[m]=cwd[m];

        Make path for link file;

        lstat(checkpath,&buf);
        if(Is Link file)
            strcpy(cwd,checkpath);

        if(flagL is 0 and argument count is bigger than 1 expect option)
            fprintf(file,"<h3>Directory path : %s</h3>\n",cwd);
        else if(flagL is 1 and argument count is bigger than 1 expect option)
        {
            fprintf(file,"<h3>Directory path : %s</h3>\n",cwd);
            fprintf(file,"<h3>total %d</h3>\n",total);
        }
        else if(flagL is 1 and argument count is smaller than 1 expect option)
            fprintf(file,"<h3>total %d</h3>\n",total);
    }

    while(before check all list)
    {
        if(name is html_ls.html)
            continue;

        Make path for hyperlink;

        if (Needed detail format)
        {
            if(createflag==0)
            {
                fprintf(file,"<body>\n");
                fprintf(file,"<table border=1>\n");
                if(flagL==1)
                    fprintf(file,"<tr><th>Name</th><th>Permission</th><th>Link</th><th>Owner</th><th>Group</th><th>Size</th><th>Last Modified</th></tr>\n");
                else
                    fprintf(file,"<tr><th>Name</th></tr>\n");
                createflag=1;
            }
        }
    }
}
```

```

lstat(tmp->filename, &buf); //read file

Declare color of file name;

Complete path of hyperlink;

Check Absolute path;

get user id;
get group id;

load time;

if(file is LNK)
    Make path name that has "->" format

strtime(buff, 255, "%b %d %H:%M", time);
fprintf(file, "<tr %s <td> <a href=%s>%s </a> </td> ", color, hyperlink, tmp->filename);
fprintf(file, "<td> %c%s </td> ", printType(buf.st_mode), printPerm(buf.st_mode));
fprintf(file, "<td> %ld </td> ", buf.st_nlink);
fprintf(file, "<td> %s </td> ", uid->pw_name);
fprintf(file, "<td> %s </td> ", gid->gr_name);

if(flagh==1)
    printOph(buf.st_size, file);

else
    fprintf(file, "<td> %ld\t </td> ", buf.st_size);
fprintf(file, "<td> %s </td> </tr> ", buff);

}

else if (Needed normal format)
{
    if(createflag==0)
    {
        fprintf(file, "<body>\n");
        fprintf(file, "<table border=1>\n");
        if(flagL==1)
            fprintf(file, "<tr> <th> Name </th> <th> Permission </th> <th> Link </th> <th> Owner </th> <th> Group </th> <th> Size </th> <th> Last Modified </th> </tr>\n");
        else
            fprintf(file, "<tr> <th> Name </th> </tr>\n");
        createflag=1;
    }
    Make path of hyperlink;

    fprintf(file, "<tr> <td> <a href=%s>%s </a> </td> </tr> ", hyperlink, tmp->filename);

}

}

fprintf(file, "</table>\n</body>");
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 (;before 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 function

        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

./html_ls . /opt -al

Directory path : /home/sp20157522025

total 3720

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxr-xr-x	26	sp20157522025	sp20157522025	4096	May 02 00:23
..	drwxr-xr-x	3	root	root	4096	Mar 15 19:55
...	drwxrwxr-x	2	sp20157522025	sp20157522025	4096	Apr 03 22:54
A	drwxrwxr-x	3	sp20157522025	sp20157522025	4096	May 02 00:00
A.c	-rw-rw-r--	1	sp20157522025	sp20157522025	629	Apr 28 22:29
abTde	drwxrwxr-x	3	sp20157522025	sp20157522025	4096	Apr 22 21:12
.ACD.swp	-rw-r--r--	1	sp20157522025	sp20157522025	12288	Mar 29 09:14
.bash_history	-rw-----	1	sp20157522025	sp20157522025	35081	May 02 00:01
.bash_logout	-rw-r--r--	1	sp20157522025	sp20157522025	220	Mar 15 19:55
.bashrc	-rw-r--r--	1	sp20157522025	sp20157522025	3771	Mar 15 19:55
.cache	drwx-----	17	sp20157522025	sp20157522025	4096	Apr 30 00:05
cli	-rwxrwxr-x	1	sp20157522025	sp20157522025	9128	Apr 29 21:04
cli.c	-rw-rw-r--	1	sp20157522025	sp20157522025	1070	Apr 29 21:35
.compiz	drwx-----	3	sp20157522025	sp20157522025	4096	Mar 21 01:45
.config	drwx-----	16	sp20157522025	sp20157522025	4096	Apr 30 00:05
Desktop	drwxr-xr-x	2	sp20157522025	sp20157522025	4096	Apr 18 01:09
.dmrc	-rw-r--r--	1	sp20157522025	sp20157522025	25	Mar 15 20:33
Documents	drwxr-xr-x	2	sp20157522025	sp20157522025	4096	Apr 18 01:09
Downloads	drwxr-xr-x	2	sp20157522025	sp20157522025	4096	Apr 29 23:45
ET	drwxrwxr-x	2	sp20157522025	sp20157522025	4096	Apr 18 01:09
.gconf	drwx-----	2	sp20157522025	sp20157522025	4096	Mar 15 22:08
.gnupg	drwx-----	3	sp20157522025	sp20157522025	4096	May 02 00:20
.hello.swo	-rw-----	1	sp20157522025	sp20157522025	4096	Mar 16 08:42
.hello.swp	-rw-----	1	sp20157522025	sp20157522025	12288	Mar 16 08:40
HONG->test	lrwxrwxrwx	1	sp20157522025	sp20157522025	4	Apr 26 00:35

Directory path : /opt

total 8

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxr-xr-x	2	root	root	4096	Mar 15 20:01
..	drwxr-xr-x	24	root	root	4096	Apr 02 17:10

/home/sp20157522025

+

file:///home/sp20157522025/html_ls.html

...

./html_ls -al

total 3728

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxr-xr-x	26	sp20157522025	sp20157522025	4096	May 02 05:30
..	drwxr-xr-x	3	root	root	4096	Mar 15 19:55
...	drwxrwxr-x	2	sp20157522025	sp20157522025	4096	Apr 03 22:54
A	drwxrwxr-x	3	sp20157522025	sp20157522025	4096	May 02 00:00
A.c	-rw-rw-r--	1	sp20157522025	sp20157522025	629	Apr 28 22:29
abTde	drwxrwxr-x	3	sp20157522025	sp20157522025	4096	Apr 22 21:12
.ACD.swp	-rw-r--r--	1	sp20157522025	sp20157522025	12288	Mar 29 09:14
.bash_history	-rw-----	1	sp20157522025	sp20157522025	35198	May 02 01:46
.bash_logout	-rw-r--r--	1	sp20157522025	sp20157522025	220	Mar 15 19:55
.bashrc	-rw-r--r--	1	sp20157522025	sp20157522025	3771	Mar 15 19:55
.cache	drwx-----	17	sp20157522025	sp20157522025	4096	Apr 30 00:05
cli	-rwxrwxr-x	1	sp20157522025	sp20157522025	9128	Apr 29 21:04
cli.c	-rw-rw-r--	1	sp20157522025	sp20157522025	1070	Apr 29 21:35
.compiz	drwx-----	3	sp20157522025	sp20157522025	4096	Mar 21 01:45

Index of file:///home/


+

file:///home/

...

Index of file:///home/

[Up to higher level directory](#)

Name	Size	Last Modified
 sp20157522025		5/2/19 5:30:14 AM PDT

직접 작성한 예시



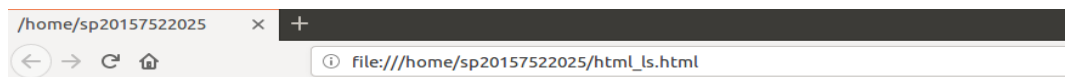
./html_ls Notfile1 Notfile2 Notfile3 A.c test

List of non-existent files or directories
cannot access Notfile1 : No such file or directory
cannot access Notfile2 : No such file or directory
cannot access Notfile3 : No such file or directory

List of file
Name
A.c

Directory path : /home/sp20157522025/test

Name
CC
HH
Makefile
oo
simple_ls
simple_ls.c
splls_advanced
tue_2-1_2015722025 (1).tar.gz
tue_2-1_2015722025.pdf



./html_ls Notfile1 Notfile2 Notfile3 A.c test -alhS

List of non-existent files or directories
cannot access Notfile1 : No such file or directory
cannot access Notfile2 : No such file or directory
cannot access Notfile3 : No such file or directory

List of file						
Name	Permission	Link	Owner	Group	Size	Last Modified
A.c	-rw-rw-r--	1	sp20157522025	sp20157522025	629	Apr 28 22:29

Directory path : /home/sp20157522025/test

total 860

Name	Permission	Link	Owner	Group	Size	Last Modified
tue_2-1_2015722025.pdf	-rwxrw-rw-	1	sp20157522025	sp20157522025	409.9K	Apr 02 03:39
tue_2-1_2015722025 (1).tar.gz	-rwxrw-rw-	1	sp20157522025	sp20157522025	357.3K	Apr 02 05:35
splls_advanced	-rwxrwxr-x	1	sp20157522025	sp20157522025	35.7K	Apr 20 04:17
A.swp	-rw-----	1	sp20157522025	sp20157522025	12K	Mar 27 17:43
simple_ls	-rwxrwxr-x	1	sp20157522025	sp20157522025	9.6K	Apr 18 17:31
.	drwxrwxr-x	5	sp20157522025	sp20157522025	4K	Apr 26 01:02
..	drwxr-xr-x	26	sp20157522025	sp20157522025	4K	May 02 00:51
CC	drwxrwxr-x	2	sp20157522025	sp20157522025	4K	Apr 26 01:02
HH	drwxrwxr-x	2	sp20157522025	sp20157522025	4K	Apr 26 01:02
oo	drwxrwxr-x	2	sp20157522025	sp20157522025	4K	Apr 26 01:02
simple_ls.c	-rw-rw-r--	1	sp20157522025	sp20157522025	365	Apr 18 17:31
Makefile	-rw-rw-r--	1	sp20157522025	sp20157522025	38	Apr 01 04:36

./html_ls file * abTde/AA/*[1-4]

List of non-existent files or directories
cannot access file : No such file or directory

List of file
Name
A.c
abTde/AA/file1
abTde/AA/file2
abTde/AA/file3
abTde/AA/file4
cli
cli.c
html_ls
Makefile
run
simple_ls.txt
spls_final
spls_final.c
srv
srv.c
tue_2-3_2015722025.pdf
tue_2-3_2015722025.tar.gz
txt1

Directory path : /home/sp20157522025/A

Name
A
a
D
dddd
run
spls_advanced
spls_advanced.c
spls_advanced.o
test_ls
X

Directory path : /home/sp20157522025/abTde

Name
AA

Directory path : /home/sp20157522025/Desktop

Directory path : /home/sp20157522025/Documents

Directory path : /home/sp20157522025/Downloads

Name
cli

Directory path : /home/sp20157522025/ET

Directory path : /home/sp20157522025/test

Name
CC
HH

/home/sp20157522025

file:///home/sp20157522025/html_ls.html

./html_ls -al

total 3728

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxr-xr-x	26	sp20157522025	sp20157522025	4096	May 02 05:30
..	drwxr-xr-x	3	root	root	4096	Mar 15 19:55
...	drwxrwxr-x	2	sp20157522025	sp20157522025	4096	Apr 03 22:54
A	drwxrwxr-x	3	sp20157522025	sp20157522025	4096	May 02 00:00
A.c	-rw-rw-r--	1	sp20157522025	sp20157522025	629	Apr 28 22:29
abTde	drwxrwxr-x	3	sp20157522025	sp20157522025	4096	Apr 22 21:12
.ACD.swf	-rw-r--r--	1	sp20157522025	sp20157522025	12288	Mar 29 09:14
.bash_history	-rw-----	1	sp20157522025	sp20157522025	35198	May 02 01:46
.bash_logout	-rw-r--r--	1	sp20157522025	sp20157522025	220	Mar 15 19:55
.bashrc	-rw-r--r--	1	sp20157522025	sp20157522025	3771	Mar 15 19:55
.cache	drwx-----	17	sp20157522025	sp20157522025	4096	Apr 30 00:05
cli	-rwxrwxr-x	1	sp20157522025	sp20157522025	9128	Apr 29 21:04
cli.c	-rw-rw-r--	1	sp20157522025	sp20157522025	1070	Apr 29 21:35
.compiz	drwx-----	3	sp20157522025	sp20157522025	4096	Mar 21 01:45

Index of file:///home/sp

file:///home/sp20157522025/abTde

Index of file:///home/sp20157522025/abTde

Up to higher level directory

Name	Size	Last Modified
AA		4/23/19 1:10:49 AM PDT

/home/sp20157522025
×
+

←
→
↺
🏠
📄
file:///home/sp20157522025/html_ls.html

./html_ls *.*

List of file	
Name	
A.c	
cli.c	
simple_ls.txt	
splls_final.c	
srv.c	
tue_2-3_2015722025.pdf	
tue_2-3_2015722025.tar.gz	

/home/sp20157522025
×
+

←
→
↺
🏠
📄
file:///home/sp20157522025/html_ls.html

./html_ls *.* -alhS

List of file						
Name	Permission	Link	Owner	Group	Size	Last Modified
splls_final.c	-rw-rw-r--	1	sp20157522025	sp20157522025	47.2K	May 02 17:52
simple_ls.txt	-rw-rw-r--	2	sp20157522025	sp20157522025	7.8K	Apr 29 02:33
srv.c	-rw-rw-r--	1	sp20157522025	sp20157522025	1.4K	Apr 29 21:09
cli.c	-rw-rw-r--	1	sp20157522025	sp20157522025	1K	Apr 29 21:35
A.c	-rw-rw-r--	1	sp20157522025	sp20157522025	629	Apr 28 22:29

./html_ls HONG Min test

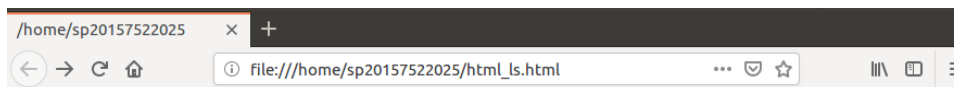
Directory path : /home/sp20157522025/test

Name
CC
HH
Makefile
oo
simple_ls
simple_ls.c
spls_advanced
tue_2-1_2015722025 (1).tar.gz
tue_2-1_2015722025.pdf

Directory path : /home/sp20157522025/test

Name
CC
HH
Makefile
oo
simple_ls
simple_ls.c
spls_advanced
tue_2-1_2015722025 (1).tar.gz
tue_2-1_2015722025.pdf

Directory path : /home/sp20157522025/test



./html_ls HONG Min test -al

List of file						
Name	Permission	Link	Owner	Group	Size	Last Modified
HONG->test	lrwxrwxrwx	1	sp20157522025	sp20157522025	4	Apr 26 00:35
Min->test	lrwxrwxrwx	1	sp20157522025	sp20157522025	4	Apr 13 05:59

Directory path : /home/sp20157522025/test

total 860

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	5	sp20157522025	sp20157522025	4096	Apr 26 01:02
..	drwxr-xr-x	26	sp20157522025	sp20157522025	4096	May 02 05:30
.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	38	Apr 01 04:36
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

5. Conclusion

3-1과제는 비교적 수월한 과제였다. 이미 구현한 `final.js`를 이용하여 출력 format만 HTML로 바꾸어주면 전 과제에서 크게 이상이 없었다면 PDF를 참고하여 구현할 수 있다고 생각된다. 특히 HTML에서 table을 다양하게 다루어본 과제는 처음이었으며, 실습시간에는 배우지 않았지만 명령어를 통하여 표를 병합하는 방법과 배경색을 다양하게 바꿀 수 있다는 것을 알게 되었고 실습 수업내용과 PDF를 참고하여 조건에 맞게 이름에 색 넣기 등, 하이퍼링크를 설정하여 경로를 지정할 수 있다는 것을 배우게 되었다. 배운 내용을 응용하여 콘솔 보다는 출력을 보기 좋고 깔끔하게 정리할 수 있다는 것을 알게 되었다.