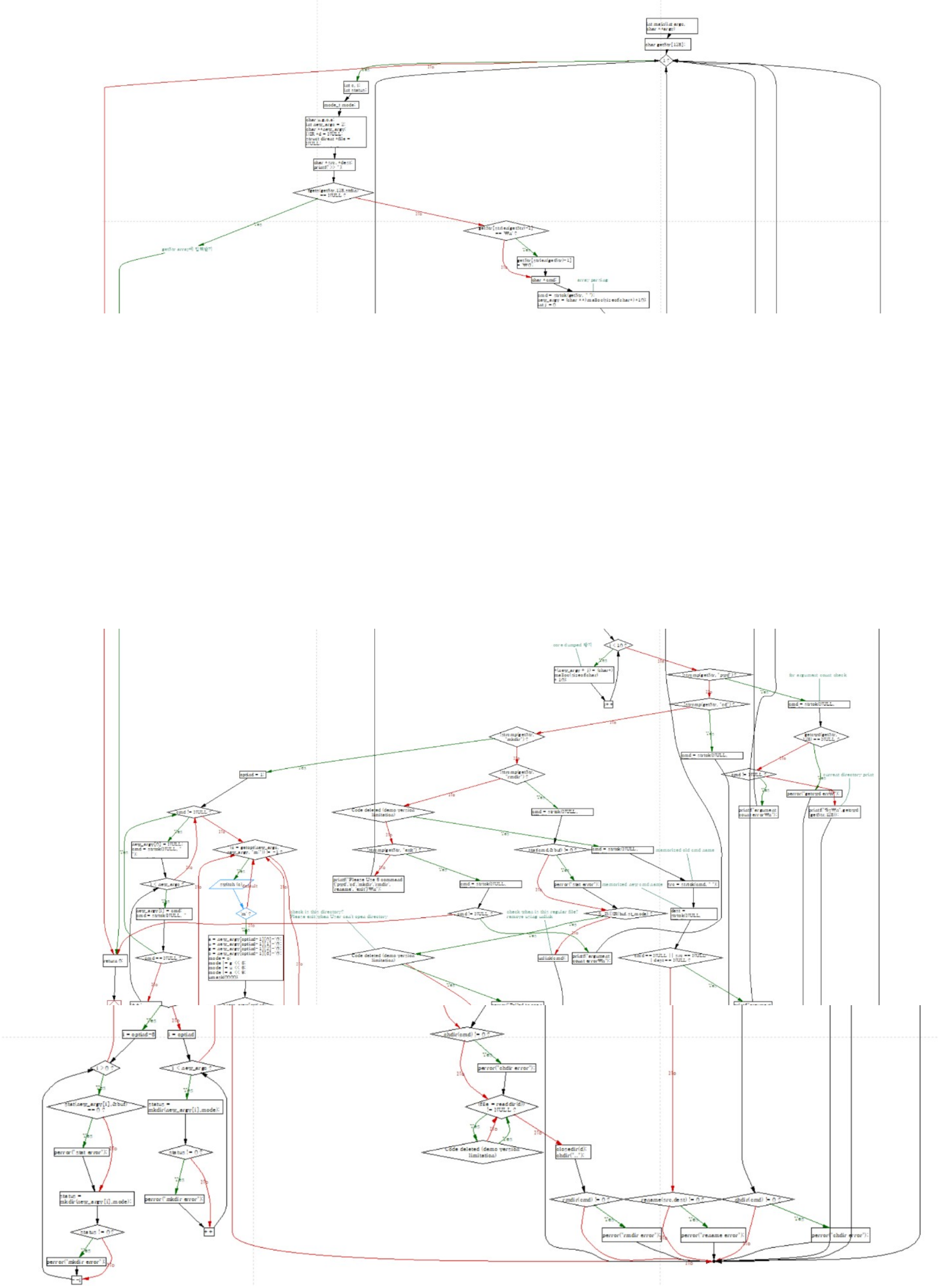
<Assignment2-3>

□□□	Assignment2-2(File system command implementation)			
□□□	□□□□□□□□□(H020-3-0922-01)(□5□6)			
□□	□□□	□□□	□□□	010–8648-7561
□□	□□□□□□□		□□□	tiktaktok116@naver.com
□□	2014707003	□□□□	□□□ □□□	
□□□□	2020.04.28~2020.05.08			
□□ □□	OperatingSystem	Linux (Ubuntu 18.04 LTS)		
	Language	C		
	Development Tools	Visual Studio Code, gcc Compiler		
□□ □□□□ □ □□ □□	- Current working directory □□ (□□□ □□ □□ error □□) - Current working directory □□ (□□□ □□ □□ error □□, ~, - □□□ □□□□ □□) - □□□□ □□ (□□□ □□ □□□ error □□, getopt □□□ □□□ -m □□□ □□ □□ ℗ □□□ □□ □□(rwx) - □□□□ □□ (□□□ □□ □□□ error □□) - □□ □ □□□□ □□ □□ (□□□ □□□ □□ □□□ error □□) - □□ □□□□ □□ (□□□ □□ □□ error □□)			

1. Introduction

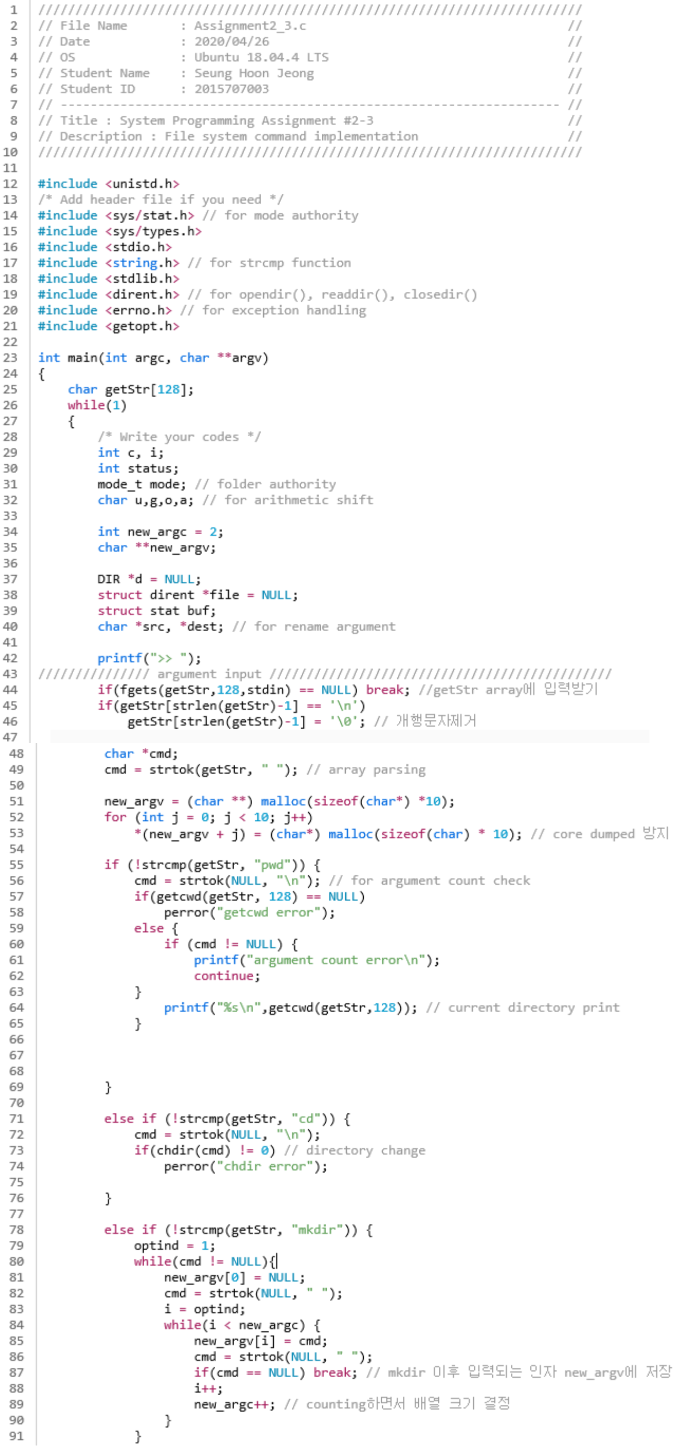
Assignment2-3□ file system command implementation□□ 'pwd', 'mkdir -m', 'rmdir', 'rename', 'exit'□ □□□□□. □□□ □□□ □□□ □□□□ □□□ □□□ □ error □□□ □□. Mkdir □□□□□ □□□ □□ □□ □□□ □□□□ □□ getopt □□□ □□□□□. □□□ □□□ □□□ □□□ □□□ □□□□.

2. Flow Chart

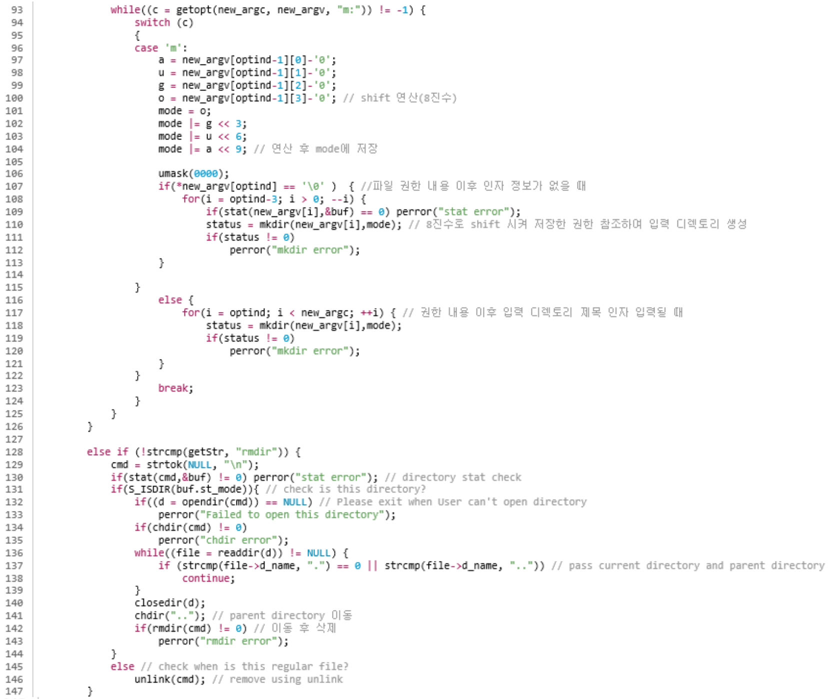


- **Flow chart □□□ □□□ □□□ □□ □□**

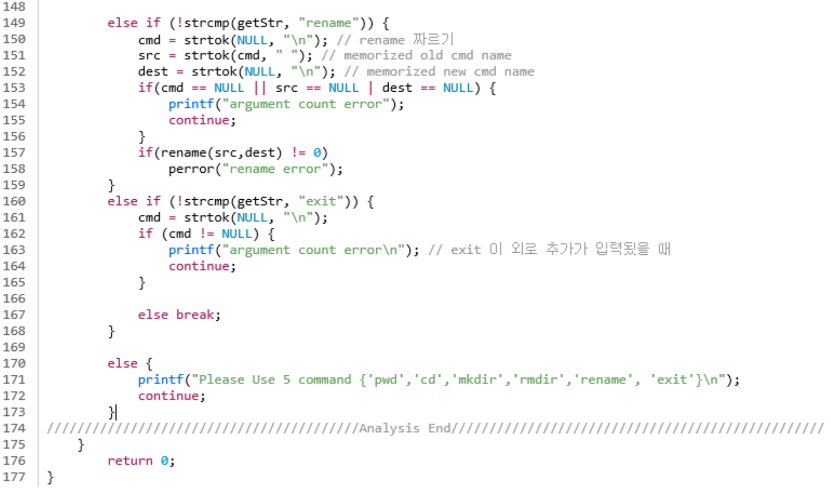
3. Source Code



```c
//////////////////////////////////////////////////////////////////////////
// File Name       : Assignment2_3.c                                    //
// Date            : 2020/04/26                                         //
// OS              : Ubuntu 18.04.4 LTS                                 //
// Student Name    : Seung Hoon Jeong                                   //
// Student ID      : 2015707003                                         //
// ------------------------------------------------------------------- //
// Title : System Programming Assignment #2-3                           //
// Description : File system command implementation                     //
//////////////////////////////////////////////////////////////////////////

#include <unistd.h>
/* Add header file if you need */
#include <sys/stat.h> // for mode authority
#include <sys/types.h>
#include <stdio.h>
#include <string.h> // for strcmp function
#include <stdlib.h>
#include <dirent.h> // for opendir(), readdir(), closedir()
#include <errno.h> // for exception handling
#include <getopt.h>

int main(int argc, char **argv)
{
    char getStr[128];
    while(1)
    {
        /* Write your codes */
        int c, i;
        int status;
        mode_t mode; // folder authority
        char u,g,o,a; // for arithmetic shift

        int new_argc = 2;
        char **new_argv;

        DIR *d = NULL;
        struct dirent *file = NULL;
        struct stat buf;
        char *src, *dest; // for rename argument

        printf(">> ");
/////////////// argument input ////////////////////////////////////////////
        if(fgets(getStr,128,stdin) == NULL) break; //getStr array에 입력받기
        if(getStr[strlen(getStr)-1] == '\n')
            getStr[strlen(getStr)-1] = '\0'; // 개행문자제거

        char *cmd;
        cmd = strtok(getStr, " "); // array parsing

        new_argv = (char **) malloc(sizeof(char*) *10);
        for (int j = 0; j < 10; j++)
            *(new_argv + j) = (char*) malloc(sizeof(char) * 10); // core dumped 방지

        if (!strcmp(getStr, "pwd")) {
            cmd = strtok(NULL, "\n"); // for argument count check
            if(getcwd(getStr, 128) == NULL)
                perror("getcwd error");
            else {
                if (cmd != NULL) {
                    printf("argument count error\n");
                    continue;
            }
                printf("%s\n",getcwd(getStr,128)); // current directory print
            }



        }

        else if (!strcmp(getStr, "cd")) {
            cmd = strtok(NULL, "\n");
            if(chdir(cmd) != 0) // directory change
                perror("chdir error");

        }

        else if (!strcmp(getStr, "mkdir")) {
            optind = 1;
            while(cmd != NULL){
                new_argv[0] = NULL;
                cmd = strtok(NULL, " ");
                i = optind;
                while(i < new_argc) {
                    new_argv[i] = cmd;
                    cmd = strtok(NULL, " ");
                    if(cmd == NULL) break; // mkdir 이후 입력되는 인자 new_argv에 저장
                    i++;
                    new_argc++; // counting하면서 배열 크기 결정
                }
            }
```

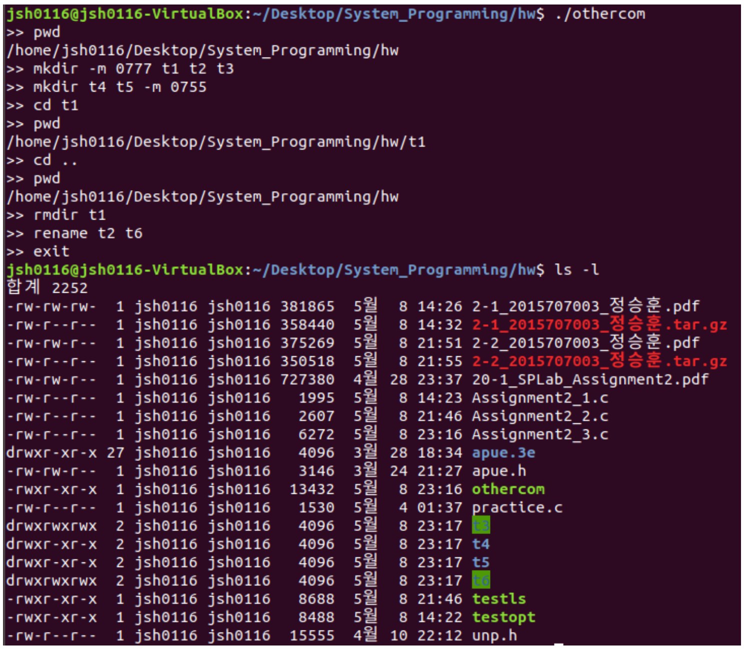


```
            while((c = getopt(new_argc, new_argv, "m:")) != -1) {
                switch (c)
                {
                case 'm':
                    a = new_argv[optind-1][0]-'0';
                    u = new_argv[optind-1][1]-'0';
                    g = new_argv[optind-1][2]-'0';
                    o = new_argv[optind-1][3]-'0'; // shift 연산(8진수)
                    mode = o;
                    mode |= g << 3;
                    mode |= u << 6;
                    mode |= a << 9; // 연산 후 mode에 저장

                    umask(0000);
                    if(*new_argv[optind] == '\0' )  { //파일 권한 내용 이후 인자 정보가 없을 때
                        for(i = optind-3; i > 0; --i) {
                            if(stat(new_argv[i],&buf) == 0) perror("stat error");
                            status = mkdir(new_argv[i],mode); // 8진수로 shift 시켜 저장한 권한 참조하여 입력 디렉토리 생성
                            if(status != 0)
                                perror("mkdir error");
                    }

                }
                    else {
                        for(i = optind; i < new_argc; ++i) { // 권한 내용 이후 입력 디렉토리 제목 인자 입력될 때
                            status = mkdir(new_argv[i],mode);
                            if(status != 0)
                                perror("mkdir error");
                    }
                }
                    break;
                }
            }
        }

        else if (!strcmp(getStr, "rmdir")) {
            cmd = strtok(NULL, "\n");
            if(stat(cmd,&buf) != 0) perror("stat error"); // directory stat check
            if(S_ISDIR(buf.st_mode)){ // check is this directory?
                if((d = opendir(cmd)) == NULL) // Please exit when User can't open directory
                    perror("Failed to open this directory");
                if(chdir(cmd) != 0)
                    perror("chdir error");
                while((file = readdir(d)) != NULL) {
                    if (strcmp(file->d_name, ".") == 0 || strcmp(file->d_name, "..")) // pass current directory and parent directory
                        continue;
                }
                closedir(d);
                chdir(".."); // parent directory 이동
                if(rmdir(cmd) != 0) // 이동 후 삭제
                    perror("rmdir error");
            }
            else // check when is this regular file?
                unlink(cmd); // remove using unlink
        }
```



```

        else if (!strcmp(getStr, "rename")) {
            cmd = strtok(NULL, "\n"); // rename 짜르기
            src = strtok(cmd, " "); // memorized old cmd name
            dest = strtok(NULL, "\n"); // memorized new cmd name
            if(cmd == NULL || src == NULL | dest == NULL) {
                printf("argument count error");
                continue;
            }
            if(rename(src,dest) != 0)
                perror("rename error");
        }
        else if (!strcmp(getStr, "exit")) {
            cmd = strtok(NULL, "\n");
            if (cmd != NULL) {
                printf("argument count error\n"); // exit 이 외로 추가가 입력됐을 때
                continue;
            }

            else break;
        }

        else {
            printf("Please Use 5 command {'pwd','cd','mkdir','rmdir','rename', 'exit'}\n");
            continue;
        }
////////////////////////////////////////Analysis End/////////////////////////////////////////////////
    }

        return 0;
}
```

4. Result



```
jsh0116@jsh0116-VirtualBox:~/Desktop/System_Programming/hw$ ./othercom
>> pwd
/home/jsh0116/Desktop/System_Programming/hw
>> mkdir -m 0777 t1 t2 t3
>> mkdir t4 t5 -m 0755
>> cd t1
>> pwd
/home/jsh0116/Desktop/System_Programming/hw/t1
>> cd ..
>> pwd
/home/jsh0116/Desktop/System_Programming/hw
>> rmdir t1
>> rename t2 t6
>> exit
jsh0116@jsh0116-VirtualBox:~/Desktop/System_Programming/hw$ ls -l
합계 2252
-rw-rw-rw-  1 jsh0116 jsh0116 381865  5월  8 14:26 2-1_2015707003_정승훈.pdf
-rw-r--r--  1 jsh0116 jsh0116 358440  5월  8 14:32 2-1_2015707003_정승훈.tar.gz
-rw-rw-r--  1 jsh0116 jsh0116 375269  5월  8 21:51 2-2_2015707003_정승훈.pdf
-rw-r--r--  1 jsh0116 jsh0116 350518  5월  8 21:55 2-2_2015707003_정승훈.tar.gz
-rw-rw-r--  1 jsh0116 jsh0116 727380  4월 28 23:37 20-1_SPLab_Assignment2.pdf
-rw-r--r--  1 jsh0116 jsh0116   1995  5월  8 14:23 Assignment2_1.c
-rw-r--r--  1 jsh0116 jsh0116   2607  5월  8 21:46 Assignment2_2.c
-rw-r--r--  1 jsh0116 jsh0116   6272  5월  8 23:16 Assignment2_3.c
drwxr-xr-x 27 jsh0116 jsh0116   4096  3월 28 18:34 apue.3e
-rw-rw-r--  1 jsh0116 jsh0116   3146  3월 24 21:27 apue.h
-rwxr-xr-x  1 jsh0116 jsh0116  13432  5월  8 23:16 othercom
-rw-r--r--  1 jsh0116 jsh0116   1530  5월  4 01:37 practice.c
drwxrwxrwx  2 jsh0116 jsh0116   4096  5월  8 23:17 t3
drwxr-xr-x  2 jsh0116 jsh0116   4096  5월  8 23:17 t4
drwxr-xr-x  2 jsh0116 jsh0116   4096  5월  8 23:17 t5
drwxrwxrwx  2 jsh0116 jsh0116   4096  5월  8 23:17 t6
-rwxr-xr-x  1 jsh0116 jsh0116   8688  5월  8 21:46 testls
-rwxr-xr-x  1 jsh0116 jsh0116   8488  5월  8 14:22 testopt
-rw-r--r--  1 jsh0116 jsh0116  15555  4월 10 22:12 unp.h
```