

System Programming

시스템 프로그래밍

(화5, 목6)

Assignment #2-3

김 태 석 교수님

컴퓨터정보공학부

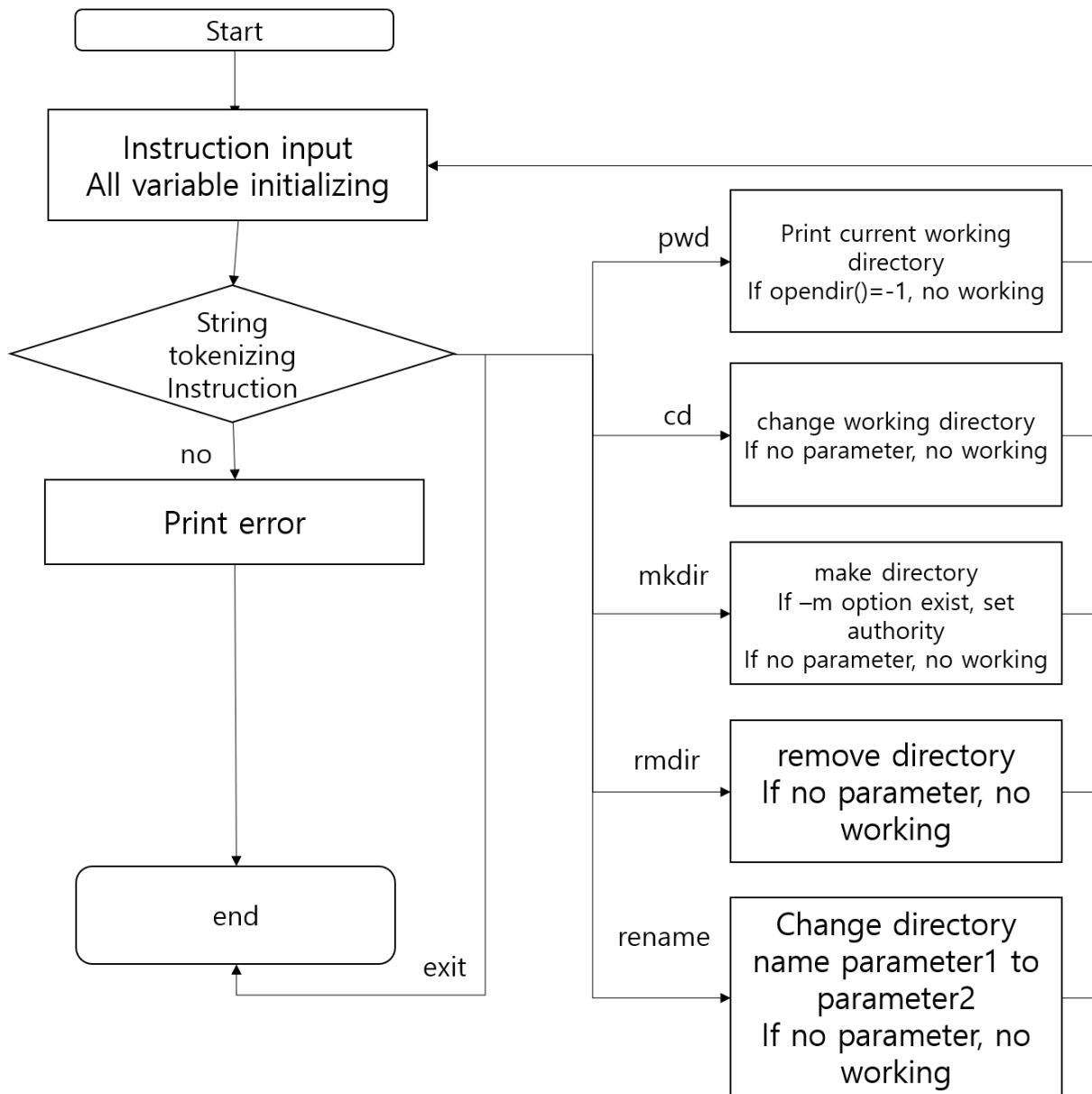
2017202037

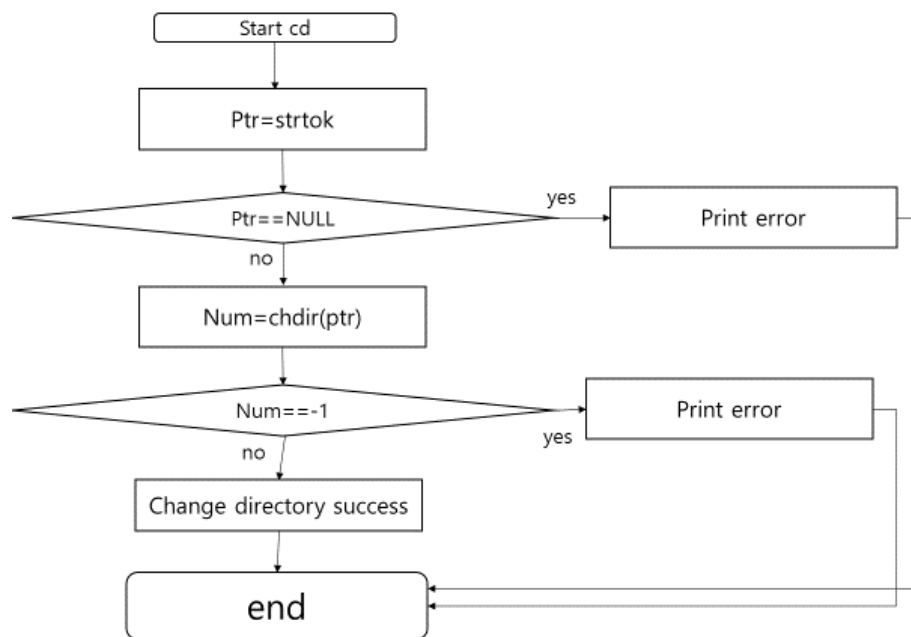
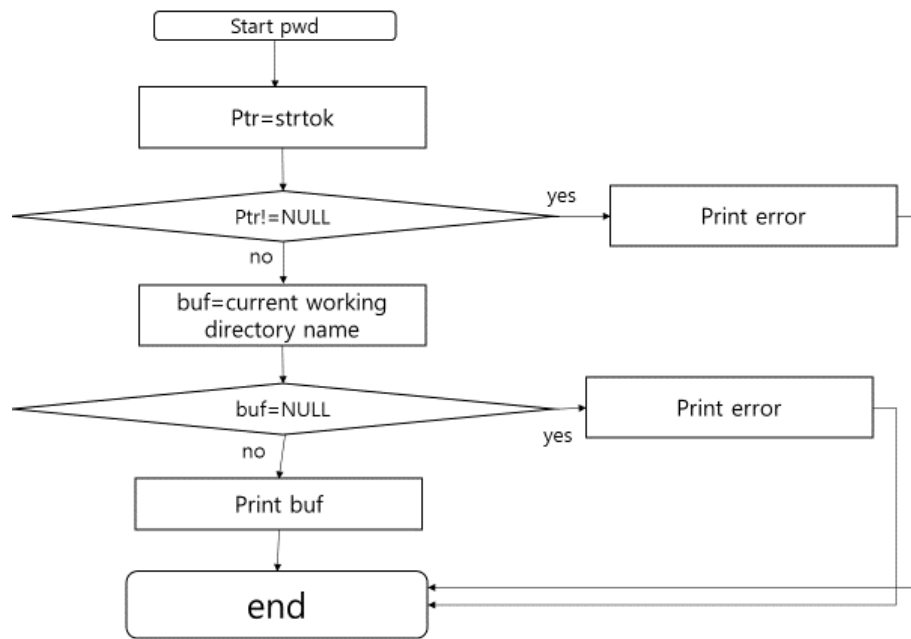
오 민 혁

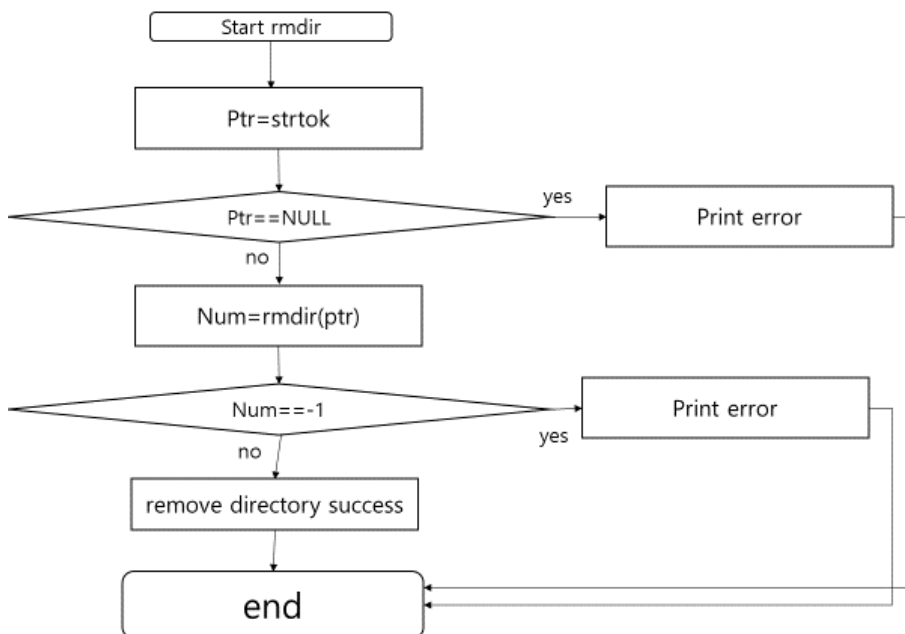
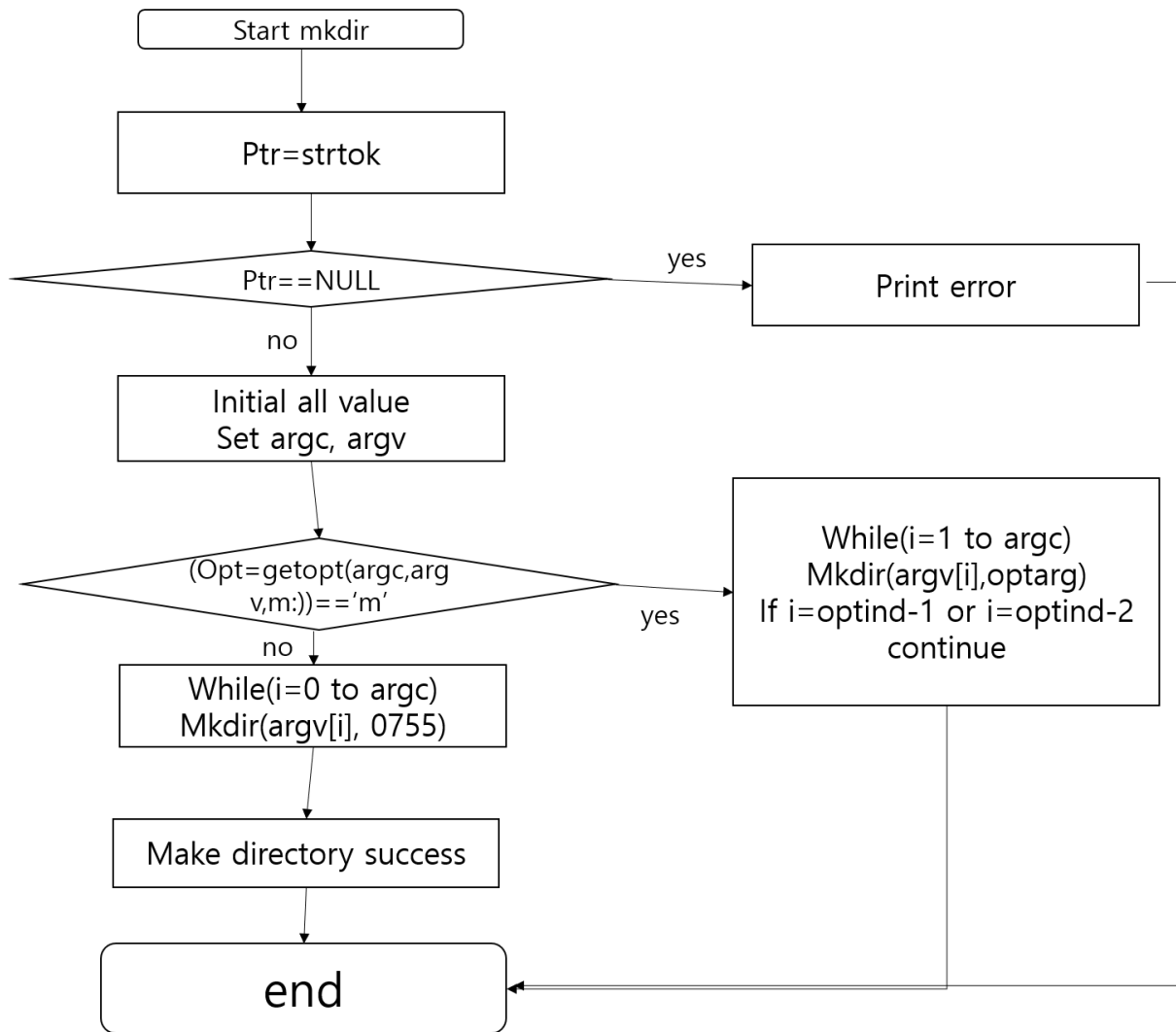
<Introduction>

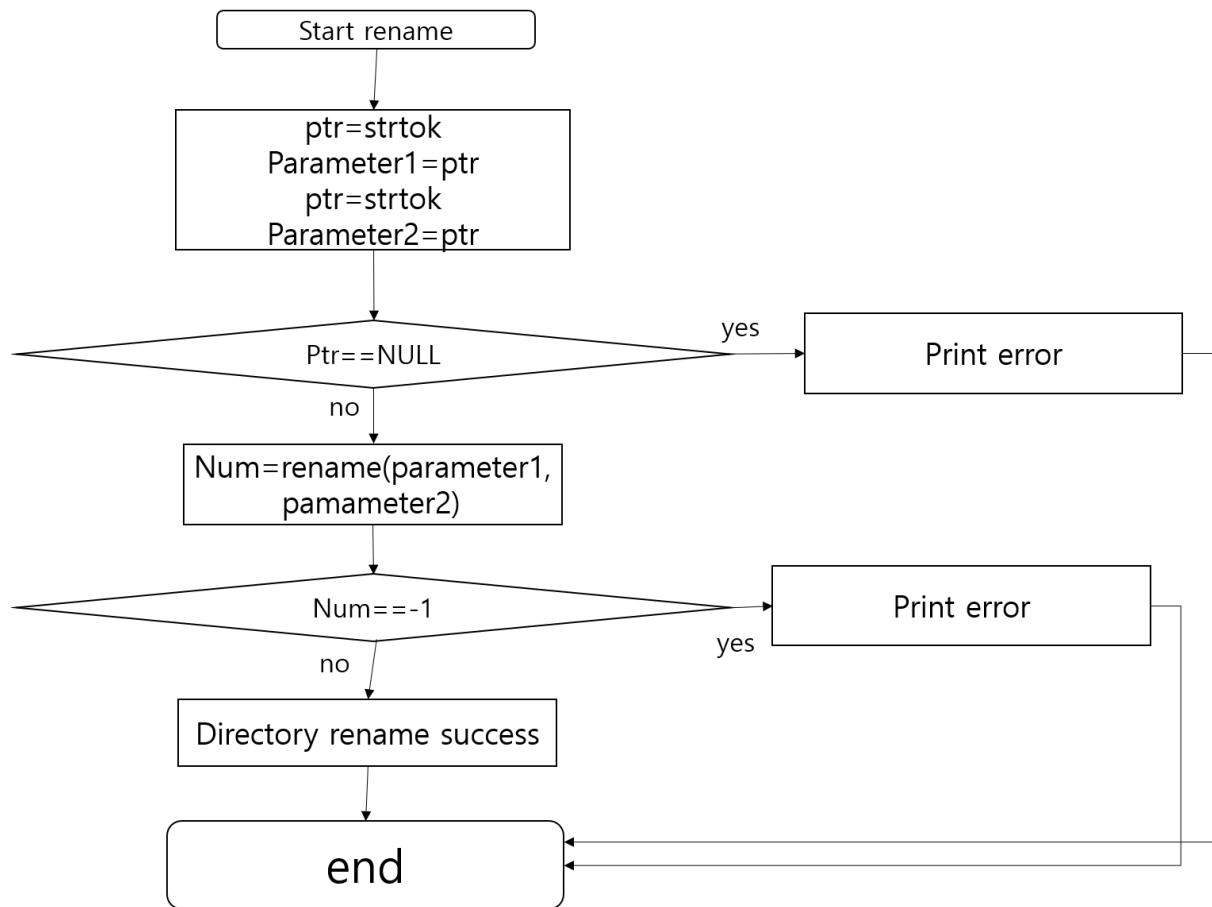
이번 과제는 current working directory를 출력하는 pwd, current working directory를 변경하는 cd, 디렉토리를 생성하는 mkdir, 디렉토리를 삭제하는 rmdir, 파일 및 디렉토리의 이름을 변경하는 rename, 해당 프로세스를 종료하는 exit의 명령어들에 대해 이해하고 사용법을 익혀 직접 구현해보는 것이다.

<Flow Chart>









<Source Code>

```

////////////////////////////////////
// File Name : Assignment2-3.c //
// Date : 2020/05/05 //
// Os : Ubuntu 16.04.5 LTS 64bits //
// Author : Oh Min Hyeok //
// Student ID : 2017202037 //
// ----- //
// Title : System Programming Assignment #2-3 ( practice ) //

```

```

// Description : ... //

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

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/types.h>

#include <sys/stat.h>


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

    umask(0000); // authority


    char getStr[128]={'\0', };

    int opt=0;

    int num=0;

    int c=0;

    char* ptr;

    int count;

    char *parameter[256] = {'NULL', }; // argument vector

    char *token; // for string tokenizing

    while ( 1 )

    {

        ptr = NULL; // initialize

```

```
token=NULL;
```

```
gets(getStr);
```

```
char tmp[256] = {'NULL', };
```

```
strcpy(tmp,getStr);
```

```
ptr=strtok(getStr," "); // instruction set
```

```
if(strcmp(ptr,"pwd")==0){ // instruction pwd
```

```
    ptr=strtok(NULL," ");
```

```
    if(ptr!=NULL){ // exception handling
```

```
        printf("error\n");
```

```
    }
```

```
    char buf[255];
```

```
    getcwd(buf,255);
```

```
    if(buf==NULL){
```

```
        printf("error\n");
```

```
    }
```

```
    printf("%s\n", buf); // print current working directory
```

```
}
```

```
else if(strcmp(ptr,"cd")==0){ // instruction cd
```

```
    ptr=strtok(NULL," ");
```

```

        if(ptr==NULL){ // exception handling

            printf("error\n");

            continue;

        }

        int num=chdir(ptr); // change directory

        if(num==-1){ // exception handling

            printf("error\n");

            continue;

        }

    }

    else if(strcmp(ptr,"mkdir")==0){ // instruction mkdir


        opt=0;

        num=0;

        c=0;

        ptr=strtok(NULL, " ");

        if(ptr==NULL){

            printf("error\n");

            continue;

        }


        while(tmp[num]!=NULL){ // space number

            if(tmp[num]==' ')

```



```

        c++;

        num++;

    }

    count=c+1; // argument counter

    for(int i = 0; parameter[i] != NULL; i++) {

        parameter[i] = NULL;

    }

    token = strtok(tmp, " ");

    for(int i=0; i<c+1; i++){

        parameter[i]=token;

        token=strtok(NULL, " ");

    }

    //initialize

    optind = 0;

    optarg = NULL;

    opterr = 0;

    optopt = 0;

    opt=getopt(count,parameter,"m:"); // determine m exist or do not

```

exist

```

        if(opt=='m'){
            for(int i=1; i<c+1; i++){
                if((optind-1)==i||(optind-2)==i){
                    continue;
                }
                mkdir(parameter[i],(mode_t)strtol(optarg, NULL, 8));
            }
        }
        else{
            for(int i=0; i<c; i++){
                if(mkdir(parameter[i+1],0755)==-1){
                    printf("error\n");
                    break;
                }
            }
            continue;
        }
    }

    else if(strcmp(ptr,"rmdir")==0){ // instruction rmdir

        ptr=strtok(NULL, " ");
    }
}

```

```

        if(ptr==NULL){

            printf("error\n");

            continue;

        }

        int num=rmdir(ptr); // remove directory

        if(num==-1){

            printf("error\n");

            continue;

        }

    }

    else if(strcmp(ptr,"rename")==0){ // instruction rename

        ptr=strtok(NULL, " ");

        char *par1=ptr;

        ptr=strtok(NULL, " ");

        if(ptr==NULL){

            printf("error\n");

            continue;

        }

        char *par2=ptr;

        int num=rename(par1,par2); // rename

        if(num==-1){

            printf("error\n");

            continue;

        }

```

```

    }

    else if(strcmp(ptr,"exit")==0){ // instructoin exit

        ptr=strtok(NULL," ");

        if(ptr!=NULL){

            printf("error\n");

            continue;

        }

        break;

    }

    else{

        printf("error\n");

        continue;

    }

    memset(getStr, '\0', sizeof(char) * 128); // initialize

}

return 0;

}

```

<Result Screen>

```
omh@ubuntu:~/Assignment2-3$ ./othercom
pwd
/home/omh/Assignment2-3
mkdir -m 0777 t1 t2 t3
mkdir t4 t5 -m 0755
cd t1
pwd
/home/omh/Assignment2-3/t1
cd ..
pwd
/home/omh/Assignment2-3
rmdir t1
rename t2 t6
exit
omh@ubuntu:~/Assignment2-3$ ls -l
total 48
-rw-rw-r-- 1 omh omh 3565 May  4 09:10 Assignment2-3.c
-rw-rw-r-- 1 omh omh 5576 May  4 09:10 Assignment2-3.o
-rw-rw-r-- 1 omh omh  123 May  3 15:48 Makefile
-rwxrwxr-x 1 omh omh 13616 May  4 09:10 othercom
drwxrwxrwx 2 omh omh 4096 May  4 09:11 t3
drwxr-xr-x 2 omh omh 4096 May  4 09:11 t4
drwxr-xr-x 2 omh omh 4096 May  4 09:11 t5
drwxrwxrwx 2 omh omh 4096 May  4 10:50 t6
```