

# SYSProg HW6

## -Simple Shell Programming-

학 과  
학 번  
이 름  
제 출 일

컴퓨터 공학과  
201211704  
김기홍  
2015.11.05

# 목 차

|                                      |   |
|--------------------------------------|---|
| I . 서론 .....                         | 3 |
| 1. 개요 .....                          | 3 |
| II . 본론 .....                        | 4 |
| 1. 프로그램 기능 .....                     | 4 |
| 1) 기본적인 명령어 .....                    | 4 |
| 2) log 기록 .....                      | 5 |
| 2) cd [Path] .....                   | 5 |
| 3) exit .....                        | 6 |
| 2. 프로그램 구현 .....                     | 6 |
| 1) 현재 디렉토리 경로 .....                  | 6 |
| 2) 입력받은 문자열 토큰화 .....                | 6 |
| 3) log 기록 .....                      | 7 |
| 4) 부모 자식 간 동기화 .....                 | 7 |
| 5) Change Directory(cd) [Path] ..... | 8 |
| 6) 현재 시간 얻기 .....                    | 8 |
| III . 결론 .....                       | 9 |
| 1. 느낀점 및 보완해야 할 부분 .....             | 9 |

# I . 서론

## 1. 개요

mysh 프로그램은 아주 간단한 셸 프로그램으로서 bash 셸에서 사용할 수 있는 기본적인 명령어를 프로그램 내에서 명령하고, 화면에 출력할 수 있습니다. 그리고 cd 기능과 exit 기능이 있으며, 저번 과제였던 ls -alR의 기능을 하였던 ll 프로그램을 사용 할 수 있도록 /bin으로 이동시켰습니다.

bash 셸 보다는 아주 미흡하지만 아주 기본적인 셸 프로그램으로서는 손색이 없는 프로그램입니다.

## II. 본론

### 1. 프로그램 기능

#### 1) 기본적인 명령어

mysh 프로그램 내에서는 기본으로 내장되어있는 명령어들(예를들어, ls, rm, mkdir 등)을 사용할 수 있습니다. 그리고 그 결과는 화면으로 출력되는 모습을 확인할 수 있습니다.

```
mysh.[/home/hong/Desktop/1104]-> ls
6-6arg 6-6arg.c ex6-2.c ex6-6.c hello log.txt mysh mysh.c
mysh.[/home/hong/Desktop/1104]-> ls -al
total 60
drwxr-xr-x  3 hong hong 4096 Nov  5 17:13 .
drwxr-xr-x 10 hong hong 4096 Nov  4 23:53 ..
-rwxr-xr-x  1 hong hong 8564 Nov  4 20:48 6-6arg
-rw-r--r--  1 hong hong  353 Nov  4 20:48 6-6arg.c
-rw-r--r--  1 hong hong  618 Nov  4 22:26 ex6-2.c
-rw-r--r--  1 hong hong  424 Nov  4 20:53 ex6-6.c
drwxr-xr-x  2 hong hong 4096 Nov  4 22:27 hello
-rw-r--r--  1 hong hong 1504 Nov  5 17:14 log.txt
-rwxr-xr-x  1 hong hong 13603 Nov  5 14:06 mysh
-rw-r--r--  1 hong hong  1989 Nov  5 14:06 mysh.c
mysh.[/home/hong/Desktop/1104]-> rm hello
rm: cannot remove 'hello': Is a directory
mysh.[/home/hong/Desktop/1104]-> rm -r hello
mysh.[/home/hong/Desktop/1104]-> mkdir SYSProg
mysh.[/home/hong/Desktop/1104]-> ls
6-6arg 6-6arg.c ex6-2.c ex6-6.c log.txt mysh mysh.c SYSProg
mysh.[/home/hong/Desktop/1104]->
mysh.[/home/hong/Desktop/1104]->
mysh.[/home/hong/Desktop/1104]->
mysh.[/home/hong/Desktop/1104]->
```

## 2) log 기록

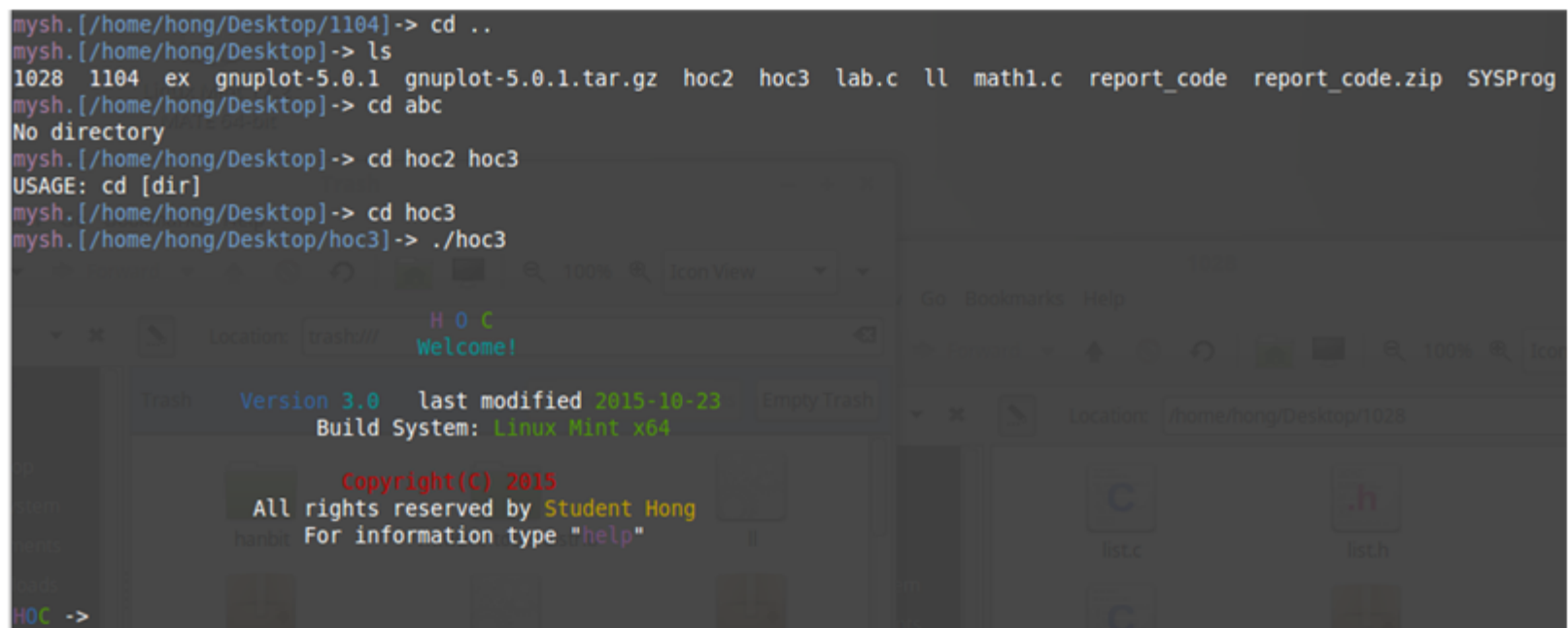
mysh 프로그램은 사용자가 입력했던 명령들을 어느 지정된 경로 (/home/hong/Desktop/1104/log.txt)에 명령을 입력했던 시간과 함께 파일로 저장합니다. 아무 명령없이 엔터만 쳤을 경우에는 log 파일에 저장되지 않습니다.

```
ls Thu Nov 5 17:14:47 2015
ls -al Thu Nov 5 17:14:49 2015
rm hello Thu Nov 5 17:14:59 2015
rm -r hello Thu Nov 5 17:15:02 2015
mkdir SYSProg Thu Nov 5 17:15:10 2015
ls Thu Nov 5 17:15:11 2015
```

## 3) cd [Path]

기본 명령어 프로그램이 아닌 cd(change directory)는 mysh내에 함수의 형태로 구현하였습니다. 자세한 내용은 프로그램 구현부에서 다루도록 하겠습니다.

```
mysh.[/home/hong/Desktop/1104]-> cd ..
mysh.[/home/hong/Desktop]-> ls
1028 1104 ex gnuplot-5.0.1 gnuplot-5.0.1.tar.gz hoc2 hoc3 lab.c ll math1.c report_code report_code.zip SYSProg
mysh.[/home/hong/Desktop]-> cd abc
No directory
mysh.[/home/hong/Desktop]-> cd hoc2 hoc3
USAGE: cd [dir]
mysh.[/home/hong/Desktop]-> cd hoc3
mysh.[/home/hong/Desktop/hoc3]-> ./hoc3
```



### 3) exit

프로그램 내에서 exit는 프로그램을 종료시키는 커맨드입니다.

```
mysh.[/home/hong/Desktop/1104]->  
mysh.[/home/hong/Desktop/1104]-> exit  
hong@hong ~/Desktop/1104 $
```

## 2. 프로그램 구현

### 1) 현재 디렉토리 경로

mysh 프로그램의 프롬프트에 출력되는 현재 위치한 디렉토리를 얻어오는 함수가 필요했는데 Google에서 검색해 본 결과 (char \*) get\_current\_dir\_name(void)라는 함수가 있었습니다. 그 함수를 이용해서 현재 디렉토리의 위치를 출력할 수 있었습니다.

```
printf("\033[95mmysh\033[0m\033[94m.\033[0m\033[94m[%s]\033[0m-> ", get_current_dir_name());
```

### 2) 입력받은 문자열 토큰화

mysh 프로그램은 '\n'(개행 문자)가 입력 될 때 까지 사용자로부터 문자열을 입력받습니다. 그런 다음 그 문자열을 분석해서 명령은 무엇인가, 옵션은 무엇인가, 파라미터는 무엇인가 등을 구별해내는 방법으로 문자열을 공백으로 토큰화 시켜서 \*argv[]에 저장하였습니다. 토큰화는 strtok함수를 이용하였습니다. 토큰으로 나뉘어진 각각의 문자열들은 fork()로 생성된 자식 프로세스에 execvp 함수를 이용하여 명령을 수행하도록 구현하였습니다.

```
argv[m] = strtok(command, token);  
while(argv[m] != NULL)  
{  
    argc++;  
    argv[++m] = strtok(NULL, token);  
}
```

### 3) log 기록

명령어를 입력받을 때마다 log.txt파일에 기록을 남기기 위해 일단 부모프로세서에서 FILE\*형의 변수와 time\_t형의 변수를 선언해두고, 자식 프로세서에서 execvp 함수를 호출하기 전에 a모드로 파일을 열어 명령어와 현재 시간을 파일에 출력한 다음에 파일을 닫아 줌으로써 구현하였습니다.

```
case 0: fp = fopen("/home/hong/Desktop/1104/log.txt", "a");  
        time(&t);  
        fprintf(fp, "%-20s\t%s\n", temp, ctime(&t));  
        fclose(fp);
```

### 4) 부모 자식 간 동기화

만약 자식프로세서에 지시된 명령이 시간이 걸리는 명령이었다면 부모프로세서가 먼저 종료 되거나 mysh 프로그램 같은 경우에는 사용자에게 다음 명령을 입력 받아버리는 문제가 생겼을 것입니다. 이 문제를 해결하기 위해 자식이 종료되기 전까지는 대기하도록 wait() 함수를 사용하였습니다. 이로써 자식 프로세스가 자기 할 일을 다 마치고 종료가 된 후에야 다음 명령어를 입력할 수 있습니다.

```
default: while(wait(&status) != pid)  
        continue;
```

## 5) Change Directory(cd) [Path]

매개변수로 int argc 와 char \*argv[]을 받으며, argc가 1일 경우에는 cd만 입력된 경우  
이므로 ~로 chdir() 해주고, 2일 경우에는 그 뒤의 문자열로 chdir() 해줍니다. 만약 실패했  
다면 존재하지 않는 디렉토리라는 글귀를 출력합니다. 만약 argc가 1과 2도 아니면 cd의  
사용법을 출력해줍니다.

```
void cd(int argc, char* argv[])
{
    if( argc == 1 )
        chdir( getenv( "HOME" ) );

    else if( argc == 2 )
    {
        if( chdir( argv[1] ) )
            printf( "No directory\n" );
    }

    else
        printf( "USAGE: cd [dir]\n" );
}
```

## 6) 현재 시간 얻기

log.txt 파일에 명령어를 입력한 시간을 출력하기 위해 먼저 현재시간을 구하는 방법이 필  
요했습니다. time() 함수를 이용해서 1970년 1월 1일부터 지금까지의 초단위의 값을 받아온  
다음 ctime() 함수를 이용하여 현재 연도, 월, 일, 시간, 요일로 이루어진 문자열을 얻을 수  
있었습니다.

```
time(&t);
fprintf(fp, "%-20s\t%s\n", temp, ctime(&t));
```

(위에서 보이는 temp는 토큰화 되기 전의 전체 명령어입니다.)



### Ⅲ. 결론

#### 1. 느낀점 및 보완해야 할 부분

마냥 어려울 줄만 알았던 셸 프로그램이 직접 구현해보니 생각보다 그렇게 어려운 프로그램은 아니라는 사실을 깨닫고 자신감을 얻었습니다. 물론 아주 많이 미흡하지만 충분히 셸 프로그램의 기능을 구현해내는 mysh의 모습에 한편으로는 뿌듯하기도 합니다.

보완점으로는 ps 명령어의 구현, background, foreground간의 전환, 그리고 조금 더 사용자로 하여금 가독성을 높이기 위한 글꼴 색상 등이 있습니다.