REPORT

HW2

simple linux shell

Subject : System programming

Major : Computer Science & Enginerring

Student ID : 2010036433

Name : 조재호

Due Date :2014.12.7. Mon

1. source Code

간단한 코드

```
main()
{
    while(1)
    {
        if(exit를 입력했다면)
            .......
        if(enter를 입력했다면)
            ......
        if(cd로 시작하는 명령어를 입력했다면)
        {
            if(cd만 입력했다면)
                .........
            else ---cd 뒤에 path를 입력함
            {
                if(cd 이후 절대경로가 입력됐다면)
                    .....
                else ---상대경로
            {
        }
        else ---cd 이외의 명령어 입력
        {
            if(history를 입력했다면)
                .....
            while( ; 으로 parsing)

            for( 위에서 자른 ; 만큼 for loop를 돌린다)
            {
                if(명령어에 &가 포함되어있다면)
                    ....
                if(vfork()==0)   ---명령어 실행을 위한 child process
                {
                    명령어 redirection check(>,>!,>>,<)
                    각명령어마다 적절하게 file descriptor를 open 하고 close한다
                    if(명령어가 redirection을 포함한다면)
                        execvp
                    명령어 pipe check
                    if(명령어에 pipe개수가 2개 이상이라면)
                    {
                        for(파이프개수만큼 loop를 돌린다)
                        {
                            if(vfork()==0)
                            {
                                적절하게 file descriptor를 open 하고 close
                                execvp;
                            }
                            else
                            {
                                ......
                                적절하게 file descriptor를 open 하고 close
                            }
                        }
                        execvp
                    }
                    else --- 명령어에 pipe개수 1개
                    {
                        execvp
                    }
                }
                else
                {
                    if(amper==0)
                        wait(&state);
                }
            }
        }
    }
}
```

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/wait.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<unistd.h>
```

```
#include<dirent.h>
#include<string.h>
#include<fcntl.h>
#include<pwd.h>
#include<signal.h>

#define BUFSIZE 1024

void main(){

        char *history[30];
        int h=0;
        int history_cnt=0;
        int *ptr_h;
        ptr_h=&h;
        int h_loop=0;

        for(h_loop;h_loop<30;h_loop++){
                history[h_loop]=(char*)malloc(sizeof(char)*128);
        }



        while(1)
        {
                char buf[BUFSIZE]={0};
                char buf_forCD[BUFSIZE]={0};
                char *ptr=NULL;
                char *strptr[10]={NULL};
                int strptr_cnt=0;

                char buf_exec_cnt[BUFSIZE]={0};
                char *temp_exec_cnt=NULL;
                int exec_cnt=0;

                char buf_check_PandR[BUFSIZE];



                char *command[10]={NULL};
                char *filename;
                char *argv;
                int i=1;//for command option
                int state;

                int t;//for initialize of command pointer array
                int Loop_p=0;//for loop of check of command has pipe;
                int Loop_r=0;
                char *ptr_p;
                char *ptr_r1;
                char *strptr_p[10];
                char *strptr_r[10];

                char for_cd[64];
                char *ptr_cd;
                char *ptr_CD;
                char *cd_path=NULL;
                char *temp_name1;
                char *temp_name2;

                int r=0;//for count of command has redirection
                int p=0;//for count of command has pipe



                printf("NEW$[%s]$  ",getcwd(NULL,0));
                gets(buf);

                if(strcmp(buf,"exit")==0){
                        kill(getpid(),SIGKILL);
                        kill(getppid(),SIGKILL);
                //      exit(0);
                }if(strcmp(buf,"")==0){
                        continue;
                }



                strcpy(buf_forCD,buf);
                //line 90 has segmentation about I type the enter
                if(strcmp("cd",strtok_r(buf_forCD," ",&ptr_CD))==0){//cd command
                        char* final_path=(char*)malloc(sizeof(char)*256);
```

```
int path_cnt=0;

strcpy(for_cd,buf);

cd_path=strtok_r(for_cd," ",&ptr_cd);
cd_path=strtok_r(NULL," ",&ptr_cd);



/*history*/
if(history_cnt>=30){
        history_cnt=history_cnt%30;
        free(history[history_cnt]);
        history[history_cnt]=(char*)malloc(sizeof(char)*128);
}
strcpy(history[history_cnt],buf);


history_cnt++;

/*history end*/

if(cd_path==NULL){// execution the 'cd' command
        char *temptemp;
        char *ptr_temptemp;
        int ii=0;
        uid_t user_id;
        struct passwd *user_pw;


        temp_name1=(char*)malloc(sizeof(char)*64);


        user_id=getuid();
        user_pw = getpwuid(user_id);

        strcpy(temp_name1 ,user_pw->pw_dir);
        chdir(temp_name1);
```

```
    execl(temp_name1,temp_name1,NULL);


}else{//execution 'cd some directory'->2 case
    //1)absolute path 2)relative path
    char absol[50]={0};
    char* ptr_ab=NULL;
    strcpy(absol,cd_path);

    if(strncmp(absol,"/",1)==0){//absolute path command 'cd /some1/some2...'

        chdir(cd_path);
        execl(cd_path,cd_path,NULL);

    }else{//relative path


        temp_name1=getcwd(NULL,0);
        temp_name2="/";




        while(*temp_name1!='₩0'){
            *final_path=*temp_name1++;
            final_path++;
            path_cnt++;
        }
        while(*temp_name2!='₩0'){
            *final_path=*temp_name2++;
            final_path++;
            path_cnt++;
        }
        while(*cd_path!='₩0'){
            *final_path=*cd_path++;
            final_path++;
            path_cnt++;
        }
        *final_path = '₩0';
        final_path = final_path-path_cnt;
```

```
                        chdir(final_path);
                        execl(temp_name1,temp_name1,NULL);
                }
        }
}
else{
        char temp_buf_h[128]={0};

        int his_exec_num=0;
        char *temp_c=NULL;
        /* parsing ";" to check how many runs to program */

        strcpy(buf_exec_cnt,buf);//copy for check number of ";"

        strptr[strptr_cnt]=strtok_r(buf_exec_cnt,";",&ptr);
        /*for history*/
        if(history_cnt>=30){
                history_cnt=history_cnt%30;
                free(history[history_cnt]);
                history[history_cnt]=(char*)malloc(sizeof(char)*128);

        }
        strcpy(history[history_cnt],strptr[strptr_cnt]);



        /*for history end*/

        if((strptr[strptr_cnt])!=NULL){
                exec_cnt++;
                /*history*/
                history_cnt++;
                /*history*/
        }



        /*history*/
        if(strcmp(strptr[strptr_cnt],"history")==0){
```

```
        int a=0;
        for(a;a<30;a++){
            if(strcmp(history[a],"")==0)
                break;
            printf("history[%d]=%s₩n",a,history[a]);
        }
    }
/* I can't success !num command

    strcpy(temp_buf_h,strptr[strptr_cnt]);
    if(strncmp(temp_buf_h,"!",1)==0){
        printf("temp_buf_h!!=%s₩n",temp_buf_h);
        temp_c=strtok(temp_buf_h,"!");//temp_c=num
        //strcpy(temp_num,temp_c);
        //printf("temp_num=%s₩n",temp_num);
        his_exec_num=atoi(temp_c);
        printf("his_exec_num=%d₩n",his_exec_num);

        strcpy(strptr[strptr_cnt],"");
        printf("strptr[%d]=%s₩n",strptr_cnt,strptr[strptr_cnt]);
        if(his_exec_num>=5){
            his_exec_num=his_exec_num%5;
        }
        strcpy(strptr[strptr_cnt],history[his_exec_num]);
        printf("strptr[%d]=%s₩n",strptr_cnt,strptr[strptr_cnt]);

        //exit(0);
    }
*/
    /*history end*/

    strptr_cnt++;

    while((strptr[strptr_cnt]=strtok_r(NULL,";",&ptr))!=NULL){
        /*for history*/
        char temp_buf_hh[128];

        //segment fault!!!
        strcpy(history[history_cnt],strptr[strptr_cnt]);
```

```
		history_cnt++;


		/*for history*/
		exec_cnt++;


		strptr_cnt++;
	}




	/*we check if command has & or not*/

	for(exec_cnt;exec_cnt>0;exec_cnt--){


		int amper=0;
		if(strstr(strptr[Loop_p],"&")!=NULL){

			strtok(strptr[Loop_p],"&");
			amper=1;

		}




		if(vfork()==0){//child for execution of command
			int command_cnt=0;
			char *last1;
			char *last0;
			int k=0;
			int loop_cnt=0;
			r=0;
			p=0;
			int rr=0;
			ptr_p=NULL;
```

```
ptr_r1=NULL;
char *ptr_temp=NULL;
i=1;
int temp;
char *for_pipe[10]={NULL};
char *for_redir[10]={NULL};


int for_redir_check=0;

int redir_check1=0;
int redir_check2=0;
int redir_check3=0;
int redir_check4=0;

int rerecheck=1;


////////////////////for redirection check/////////////////////////////////////////


for_redir[Loop_p]=(char*)malloc(strlen(strptr[Loop_p])+1);
strcpy(for_redir[Loop_p],strptr[Loop_p]);




//first, parsing by ">" and check command option and adjust the redirection


strptr_r[r]=strtok_r(for_redir[Loop_p],">",&ptr_r1);
if(strptr_r[r]!=NULL){
        r++;
}
while((strptr_r[r]=strtok_r(NULL,">",&ptr_r1))!=NULL){


        r++;
        ptr_temp=ptr_r1;
}
if(r>=2){
        if(strstr(strptr[Loop_p],">!")!=NULL){
                r=0;
                ptr_temp=(char*)malloc(strlen(strptr[Loop_p])+1);

                strcpy(ptr_temp,strptr[Loop_p]);

                if(strtok(ptr_temp,"!")!=NULL){
                        r++;
                }
                while((strptr_r[r]=strtok(NULL,"!"))!=NULL){
                        r++;
                }
                redir_check4=1;
                rerecheck=0;
        }
        if(strstr(strptr[Loop_p],">>")!=NULL){
                redir_check2=1;
                rerecheck=0;

        }if(rerecheck==1){
                redir_check1=1;
        }
        /*else{
                redir_check1=1;


        }*/
}
if(r==1){


        strcpy(for_redir[Loop_p],strptr[Loop_p]);


        r=0;
        ptr_r1=NULL;
        strptr_r[r]=strtok_r(for_redir[Loop_p],"<",&ptr_r1);
        if(strptr_r[r]!=NULL){
                r++;
        }
        while((strptr_r[r]=strtok_r(NULL,"<",&ptr_r1))!=NULL){
                r++;
            }
            if(r>=2)
                redir_check3=1;
        }
        if(redir_check1==1||redir_check2==1||redir_check3==1||redir_check4==1){



            if((filename=strtok_r(/*buf*/strptr_r[/*command_cnt*/0]," ",&last0))!=NULL){
                command[0]=(char*)malloc(strlen(filename)+1);
                strcpy(command[0],filename);//command[0] is filename
                while((argv=strtok_r(NULL," ",&last0))!=NULL){
                    command[i]=(char*)malloc(strlen(argv)+1);
                    strcpy(command[i],argv);
                    i++;
                }
                command_cnt=command_cnt+1;
            }



            if(redir_check1==1){// ">"
                int fd;
                fd=open(strptr_r[1], O_RDWR | O_CREAT | O_TRUNC, 0644) ;

                close(1);//close stdout
                dup(fd);
                close(fd);
            }else if(redir_check2==1){//">>"
                int fd;
                fd=open(strptr_r[1], O_RDWR | O_APPEND, 0644);
                if(fd<0){
                    fd=open(strptr_r[1], O_RDWR | O_CREAT | O_TRUNC, 0644) ;

                }
                close(1);
                dup(fd);
                close(fd);
```

```
    }else if(redir_check3==1){//"<<"
        int fd;
        fd=open(strptr_r[1], O_RDONLY);
        close(0);
        dup(fd);
        close(fd);
    }
    //
    else if(redir_check4==1){//">!"
        int fd;
        fd=open(strptr_r[1], O_RDWR | O_CREAT , 0644) ;
        close(1);//close stdout
        dup(fd);
        close(fd);
    }
    Loop_p++;
    execvp(command[0],command);

}

//for pipe check
for_pipe[Loop_p]=(char*)malloc(strlen(strptr[Loop_p])+1);
strcpy(for_pipe[Loop_p],strptr[Loop_p]);



// command line is parsing by ";". So we check whether each command has |, >,>>....


//check whether each command has "|"
strptr_p[p]=strtok_r(for_pipe[Loop_p],"|",&ptr_p);



if((strptr_p[p])!=NULL){
    p++;
    loop_cnt++;
```

```
}




while((strptr_p[p]=strtok_r(NULL,"|",&ptr_p))!=NULL){


        p++;
        loop_cnt++;

}
Loop_p++;


// parse the command and store
for(t=0;t<10;t++){//initialize
        command[t]=NULL;
}

if((filename=strtok_r(/*buf*/strptr_p[/*command_cnt*/0]," ",&last1))!=NULL){

        command[0]=(char*)malloc(strlen(filename)+1);
        strcpy(command[0],filename);//command[0] is filename
        while((argv=strtok_r(NULL," ",&last1))!=NULL){
                command[i]=(char*)malloc(strlen(argv)+1);
                strcpy(command[i],argv);
                i++;
        }
        command_cnt=command_cnt+1;
}


if(p>=2){
        for(k;k<p-1;k++){

                int fildes[2]={0};
                if(pipe(fildes)==-1){
```

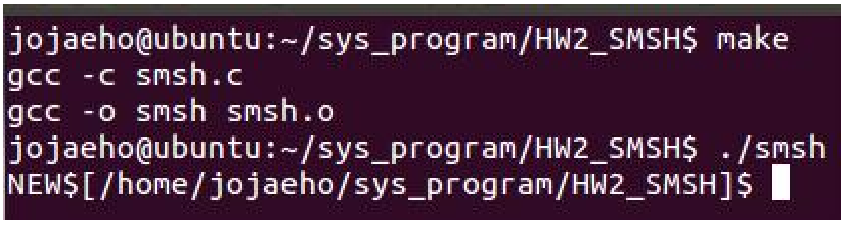
```
                                        perror("fail to make pipe");
                                        exit(1);
                            }

                            if(vfork()==0){//first component of command
                                        close(1);
                                        dup(fildes[1]);
                                        close(fildes[1]);
                                        close(fildes[0]);
                                        execvp(command[0],command);

                            }else{
                                        //char *command1[10]={NULL};
                                        char *filename1=NULL;
                                        //char *argv1;
                                        wait(&state);
                                        for(t=0;t<10;t++){//initialize
                                                    command[t]=NULL;
                                        }
                                        i=1;



                                        // parse the command and store

if((filename1=strtok(/*buf*/strptr_p[command_cnt]," "))!=NULL){



command[0]=(char*)malloc(strlen(filename1)+1);

strcpy(command[0],filename1);//command[0] is filename
                                                    while((argv=strtok(NULL," "))!=NULL){

command[i]=(char*)malloc(strlen(argv)+1);

                                                                strcpy(command[i],argv);
                                                                i++;
                                                    }
                                                    command_cnt=command_cnt+1;
                                        }
```

```
                                close(0);
                                dup(fildes[0]);
                                close(fildes[0]);
                                close(fildes[1]);
                            }
                        }
                        execvp(command[0],command);
                    }else{
                        execvp(command[0],command);

                    }

                }else{//parent*/
                    if(amper==0)
                        wait(&state);
                }
            }
        }
    }

}
```

---

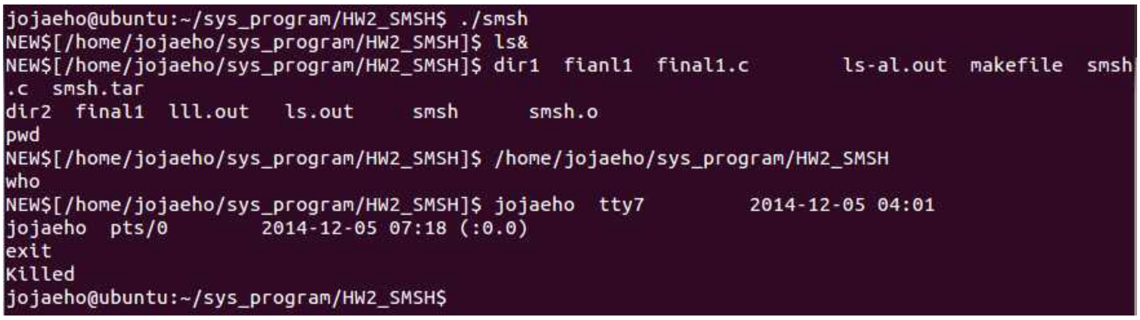
2. Screen dump of running program

make로 컴파일 후 실행 모습



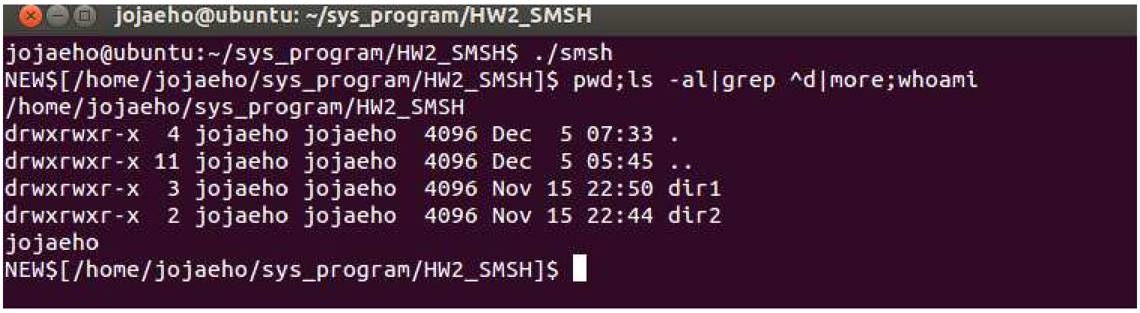
```
jojaeho@ubuntu:~/sys_program/HW2_SMSH$ make
gcc -c smsh.c
gcc -o smsh smsh.o
jojaeho@ubuntu:~/sys_program/HW2_SMSH$ ./smsh
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ 
```

foreground and background execution (&)



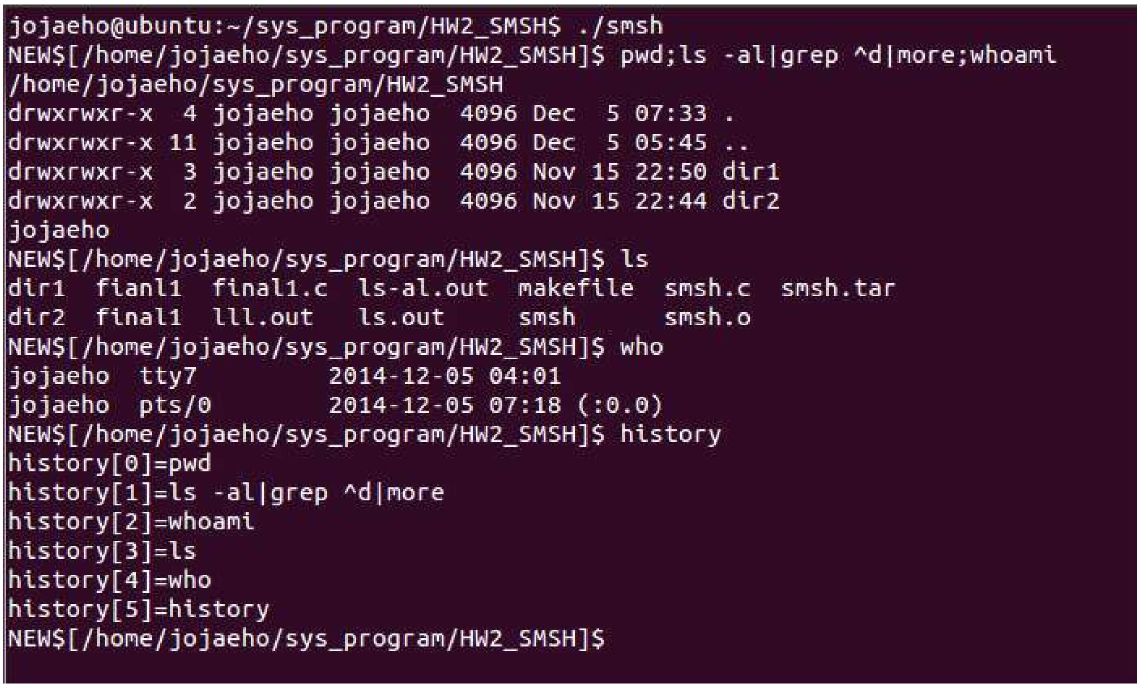
```
jojaeho@ubuntu:~/sys_program/HW2_SMSH$ ./smsh
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ ls&
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ dir1  fianl1  final1.c       ls-al.out  makefile  smsh
.c  smsh.tar
dir2  final1  lll.out   ls.out     smsh      smsh.o
pwd
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ /home/jojaeho/sys_program/HW2_SMSH
who
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ jojaeho  tty7         2014-12-05 04:01
jojaeho  pts/0        2014-12-05 07:18 (:0.0)
exit
Killed
jojaeho@ubuntu:~/sys_program/HW2_SMSH$
```

multiple commands separated by semicolons



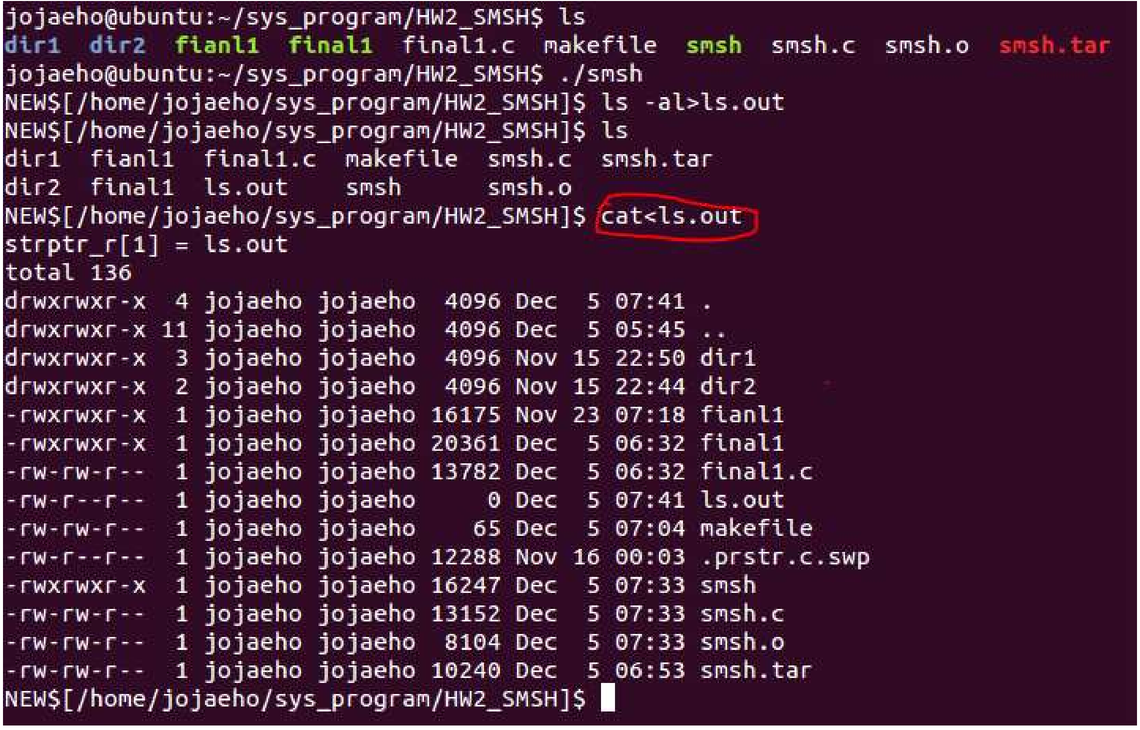
```
jojaeho@ubuntu:~/sys_program/HW2_SMSH$ ./smsh
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ pwd;ls -al|grep ^d|more;whoami
/home/jojaeho/sys_program/HW2_SMSH
drwxrwxr-x  4 jojaeho jojaeho  4096 Dec  5 07:33 .
drwxrwxr-x 11 jojaeho jojaeho  4096 Dec  5 05:45 ..
drwxrwxr-x  3 jojaeho jojaeho  4096 Nov 15 22:50 dir1
drwxrwxr-x  2 jojaeho jojaeho  4096 Nov 15 22:44 dir2
jojaeho
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$
```

history command



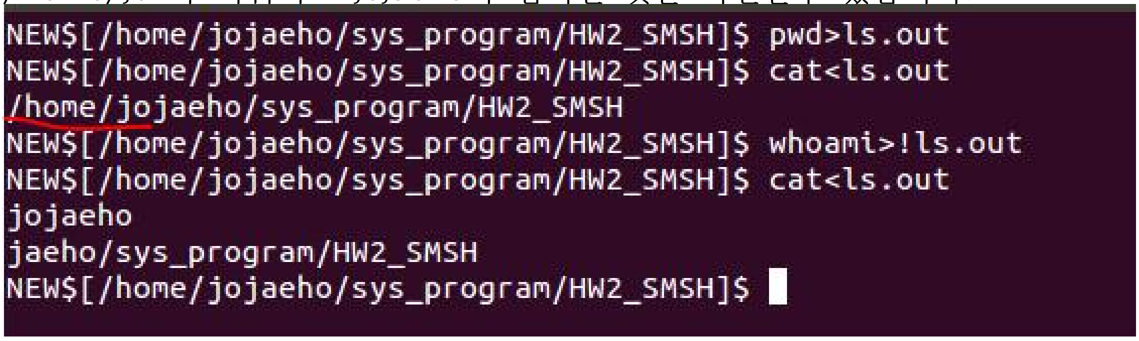
```
jojaeho@ubuntu:~/sys_program/HW2_SMSH$ ./smsh
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ pwd;ls -al|grep ^d|more;whoami
/home/jojaeho/sys_program/HW2_SMSH
drwxrwxr-x  4 jojaeho jojaeho  4096 Dec  5 07:33 .
drwxrwxr-x 11 jojaeho jojaeho  4096 Dec  5 05:45 ..
drwxrwxr-x  3 jojaeho jojaeho  4096 Nov 15 22:50 dir1
drwxrwxr-x  2 jojaeho jojaeho  4096 Nov 15 22:44 dir2
jojaeho
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ ls
dir1  fianl1  final1.c  ls-al.out  makefile  smsh.c  smsh.tar
dir2  final1  lll.out   ls.out     smsh      smsh.o
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ who
jojaeho  tty7         2014-12-05 04:01
jojaeho  pts/0        2014-12-05 07:18 (:0.0)
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ history
history[0]=pwd
history[1]=ls -al|grep ^d|more
history[2]=whoami
history[3]=ls
history[4]=who
history[5]=history
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$
```

shell redirection >, <



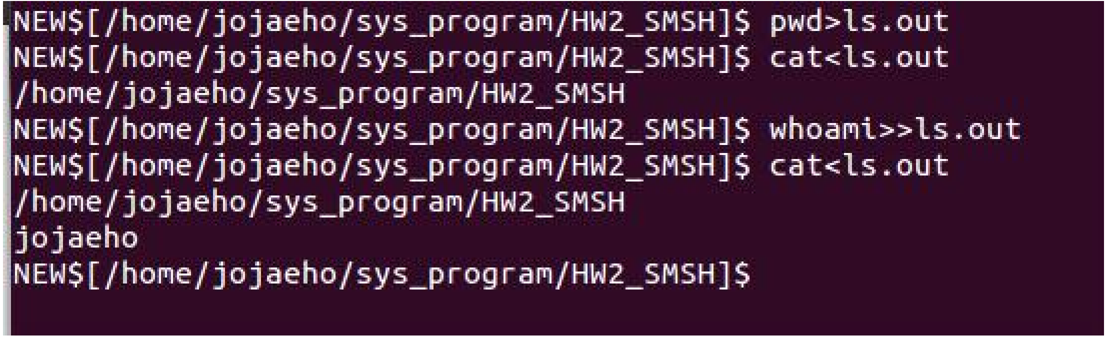
```
jojaeho@ubuntu:~/sys_program/HW2_SMSH$ ls
dir1  dir2  fianl1  final1  final1.c  makefile  smsh  smsh.c  smsh.o  smsh.tar
jojaeho@ubuntu:~/sys_program/HW2_SMSH$ ./smsh
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ ls -al>ls.out
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ ls
dir1  fianl1  final1.c  makefile  smsh.c  smsh.tar
dir2  final1  ls.out    smsh      smsh.o
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ cat<ls.out
strptr_r[1] = ls.out
total 136
drwxrwxr-x  4 jojaeho jojaeho  4096 Dec  5 07:41 .
drwxrwxr-x 11 jojaeho jojaeho  4096 Dec  5 05:45 ..
drwxrwxr-x  3 jojaeho jojaeho  4096 Nov 15 22:50 dir1
drwxrwxr-x  2 jojaeho jojaeho  4096 Nov 15 22:44 dir2
-rwxrwxr-x  1 jojaeho jojaeho 16175 Nov 23 07:18 fianl1
-rwxrwxr-x  1 jojaeho jojaeho 20361 Dec  5 06:32 final1
-rw-rw-r--  1 jojaeho jojaeho 13782 Dec  5 06:32 final1.c
-rw-r--r--  1 jojaeho jojaeho     0 Dec  5 07:41 ls.out
-rw-rw-r--  1 jojaeho jojaeho    65 Dec  5 07:04 makefile
-rw-r--r--  1 jojaeho jojaeho 12288 Nov 16 00:03 .prstr.c.swp
-rwxrwxr-x  1 jojaeho jojaeho 16247 Dec  5 07:33 smsh
-rw-rw-r--  1 jojaeho jojaeho 13152 Dec  5 07:33 smsh.c
-rw-rw-r--  1 jojaeho jojaeho  8104 Dec  5 07:33 smsh.o
-rw-rw-r--  1 jojaeho jojaeho 10240 Dec  5 06:53 smsh.tar
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$
```

shell redirection >!, <
pwd명령후 /home/jojaeho/sys_program/HW2_SMSH가 ls.out파일에 입력이 됐습니다.
그후 whoami라는 명령을 >!을 통해 ls.out에 입력했습니다.
/home/jo 가 지워지고 jojaeho가 입력된 것을 확인할수 있습니다.



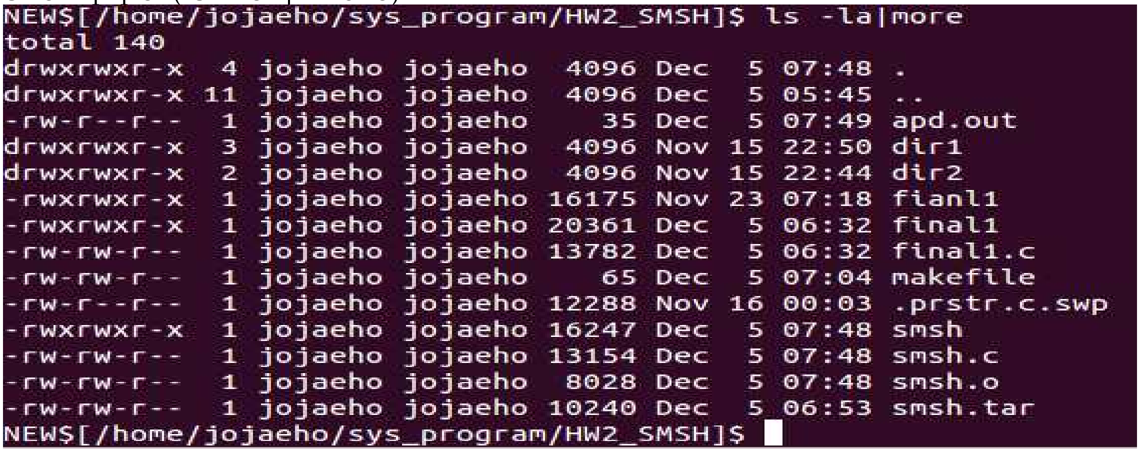
```
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ pwd>ls.out
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ cat<ls.out
/home/jojaeho/sys_program/HW2_SMSH
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ whoami>!ls.out
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ cat<ls.out
jojaeho
jaeho/sys_program/HW2_SMSH
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$
```

shell redirection >>, <



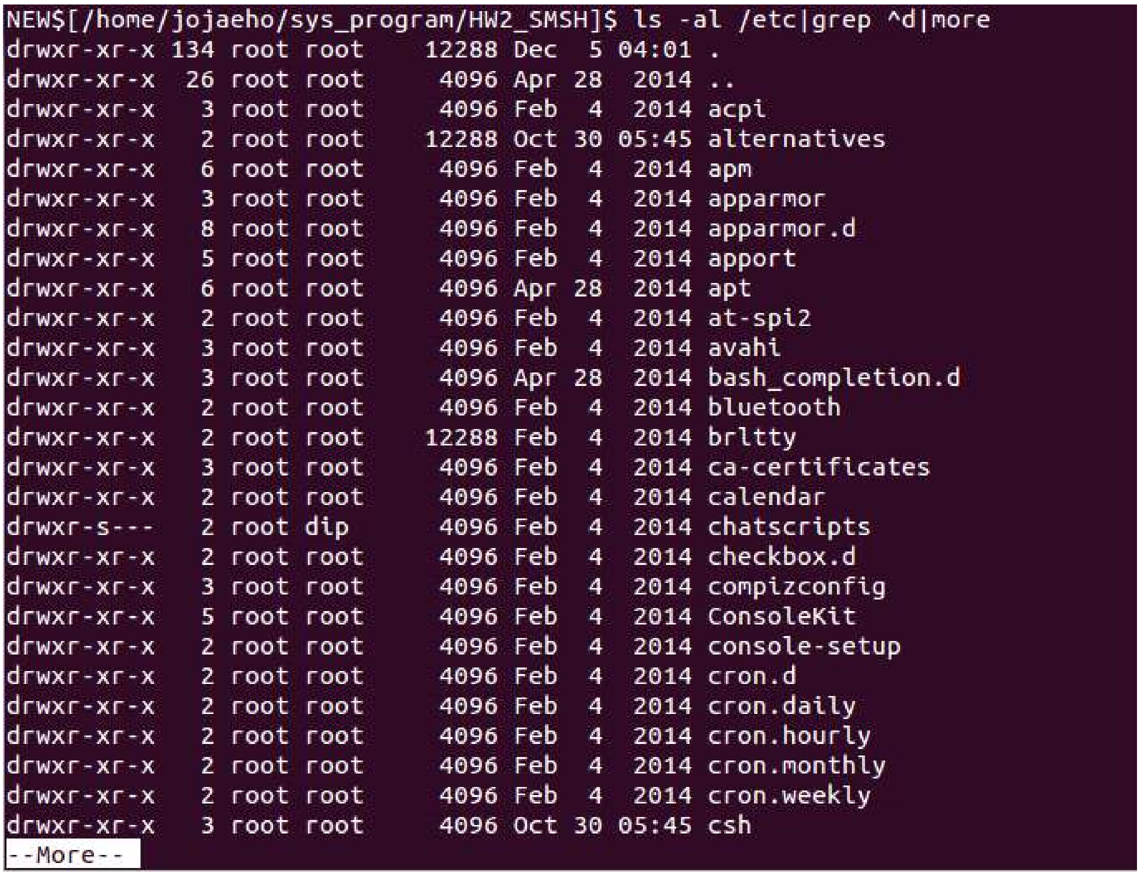
```
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ pwd>ls.out
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ cat<ls.out
/home/jojaeho/sys_program/HW2_SMSH
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ whoami>>ls.out
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ cat<ls.out
/home/jojaeho/sys_program/HW2_SMSH
jojaeho
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$
```

shell pipe (ls -la | more)



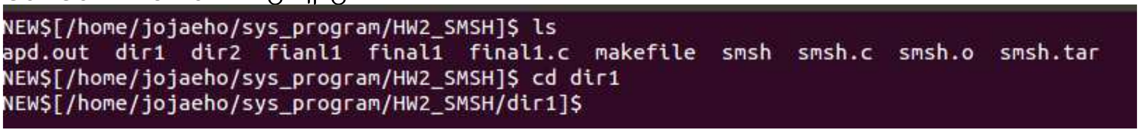
```
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ ls -la|more
total 140
drwxrwxr-x  4 jojaeho jojaeho  4096 Dec  5 07:48 .
drwxrwxr-x 11 jojaeho jojaeho  4096 Dec  5 05:45 ..
-rw-r--r--  1 jojaeho jojaeho    35 Dec  5 07:49 apd.out
drwxrwxr-x  3 jojaeho jojaeho  4096 Nov 15 22:50 dir1
drwxrwxr-x  2 jojaeho jojaeho  4096 Nov 15 22:44 dir2
-rwxrwxr-x  1 jojaeho jojaeho 16175 Nov 23 07:18 fianl1
-rwxrwxr-x  1 jojaeho jojaeho 20361 Dec  5 06:32 final1
-rw-rw-r--  1 jojaeho jojaeho 13782 Dec  5 06:32 final1.c
-rw-rw-r--  1 jojaeho jojaeho    65 Dec  5 07:04 makefile
-rw-r--r--  1 jojaeho jojaeho 12288 Nov 16 00:03 .prstr.c.swp
-rwxrwxr-x  1 jojaeho jojaeho 16247 Dec  5 07:48 smsh
-rw-rw-r--  1 jojaeho jojaeho 13154 Dec  5 07:48 smsh.c
-rw-rw-r--  1 jojaeho jojaeho  8028 Dec  5 07:48 smsh.o
-rw-rw-r--  1 jojaeho jojaeho 10240 Dec  5 06:53 smsh.tar
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$
```

Multiple pipe (ls -al /etc|grep ^d|more)



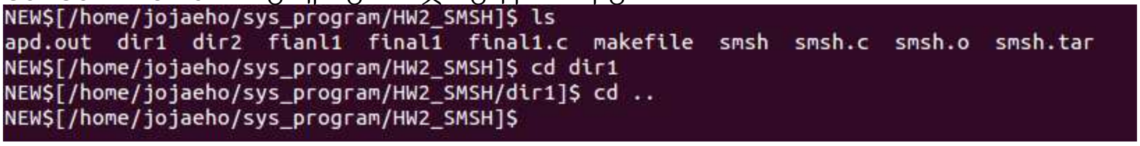
```
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ ls -al /etc|grep ^d|more
drwxr-xr-x 134 root root    12288 Dec  5 04:01 .
drwxr-xr-x  26 root root     4096 Apr 28  2014 ..
drwxr-xr-x   3 root root     4096 Feb  4  2014 acpi
drwxr-xr-x   2 root root    12288 Oct 30 05:45 alternatives
drwxr-xr-x   6 root root     4096 Feb  4  2014 apm
drwxr-xr-x   3 root root     4096 Feb  4  2014 apparmor
drwxr-xr-x   8 root root     4096 Feb  4  2014 apparmor.d
drwxr-xr-x   5 root root     4096 Feb  4  2014 apport
drwxr-xr-x   6 root root     4096 Apr 28  2014 apt
drwxr-xr-x   2 root root     4096 Feb  4  2014 at-spi2
drwxr-xr-x   3 root root     4096 Feb  4  2014 avahi
drwxr-xr-x   3 root root     4096 Apr 28  2014 bash_completion.d
drwxr-xr-x   2 root root     4096 Feb  4  2014 bluetooth
drwxr-xr-x   2 root root    12288 Feb  4  2014 brltty
drwxr-xr-x   3 root root     4096 Feb  4  2014 ca-certificates
drwxr-xr-x   2 root root     4096 Feb  4  2014 calendar
drwxr-s---   2 root dip      4096 Feb  4  2014 chatscripts
drwxr-xr-x   2 root root     4096 Feb  4  2014 checkbox.d
drwxr-xr-x   3 root root     4096 Feb  4  2014 compizconfig
drwxr-xr-x   5 root root     4096 Feb  4  2014 ConsoleKit
drwxr-xr-x   2 root root     4096 Feb  4  2014 console-setup
drwxr-xr-x   2 root root     4096 Feb  4  2014 cron.d
drwxr-xr-x   2 root root     4096 Feb  4  2014 cron.daily
drwxr-xr-x   2 root root     4096 Feb  4  2014 cron.hourly
drwxr-xr-x   2 root root     4096 Feb  4  2014 cron.monthly
drwxr-xr-x   2 root root     4096 Feb  4  2014 cron.weekly
drwxr-xr-x   3 root root     4096 Oct 30 05:45 csh
--More--
```

cd command - 상대경로



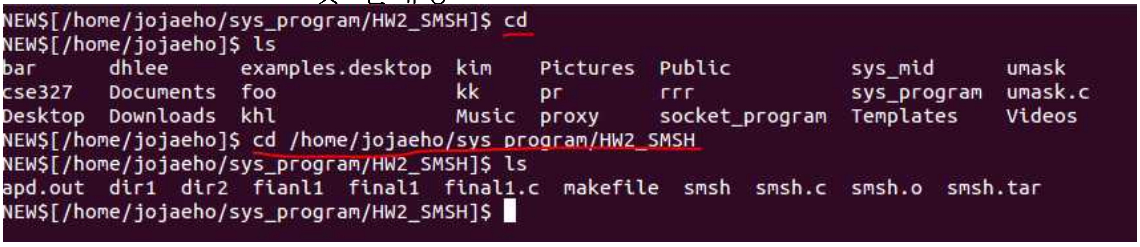
```
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ ls
apd.out  dir1  dir2  fianl1  final1  final1.c  makefile  smsh  smsh.c  smsh.o  smsh.tar
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ cd dir1
NEW$[/home/jojaeho/sys_program/HW2_SMSH/dir1]$
```

cd command - 상대 경로 및 상위로 이동



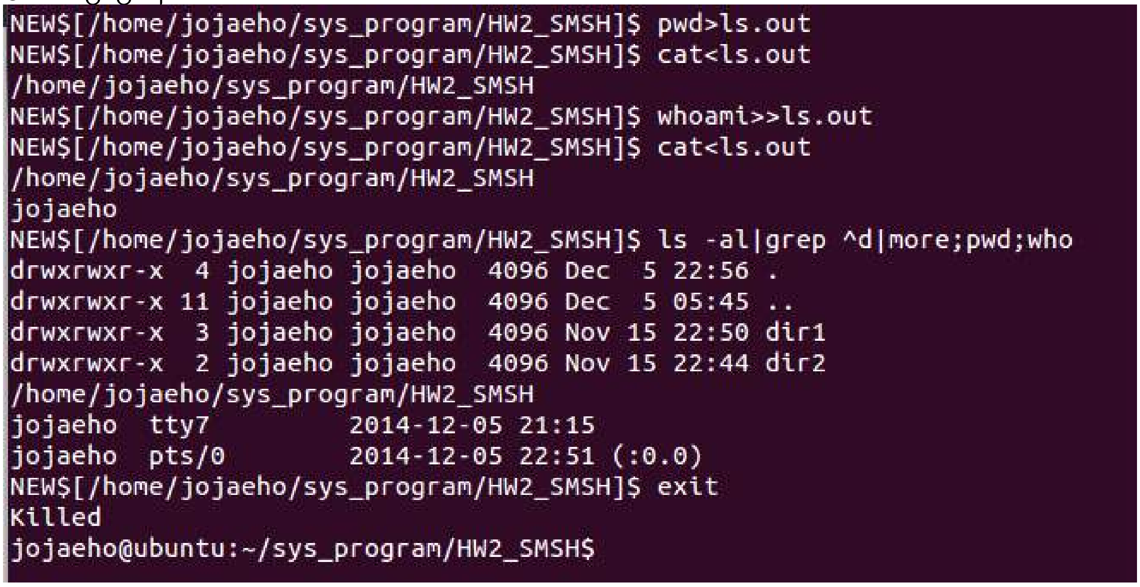
```
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ ls
apd.out  dir1  dir2  fianl1  final1  final1.c  makefile  smsh  smsh.c  smsh.o  smsh.tar
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ cd dir1
NEW$[/home/jojaeho/sys_program/HW2_SMSH/dir1]$ cd ..
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$
```

cd command - cd 및 절대경로



```
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ cd
NEW$[/home/jojaeho]$ ls
bar      dhlee      examples.desktop  kim    Pictures  Public          sys_mid      umask
cse327   Documents  foo               kk     pr        rrr             sys_program  umask.c
Desktop  Downloads  khl               Music  proxy     socket_program  Templates    Videos
NEW$[/home/jojaeho]$ cd /home/jojaeho/sys_program/HW2_SMSH
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ ls
apd.out  dir1  dir2  fianl1  final1  final1.c  makefile  smsh  smsh.c  smsh.o  smsh.tar
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ 
```

exit 명령어



```
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ pwd>ls.out
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ cat<ls.out
/home/jojaeho/sys_program/HW2_SMSH
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ whoami>>ls.out
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ cat<ls.out
/home/jojaeho/sys_program/HW2_SMSH
jojaeho
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ ls -al|grep ^d|more;pwd;who
drwxrwxr-x  4 jojaeho jojaeho  4096 Dec  5 22:56 .
drwxrwxr-x 11 jojaeho jojaeho  4096 Dec  5 05:45 ..
drwxrwxr-x  3 jojaeho jojaeho  4096 Nov 15 22:50 dir1
drwxrwxr-x  2 jojaeho jojaeho  4096 Nov 15 22:44 dir2
/home/jojaeho/sys_program/HW2_SMSH
jojaeho  tty7         2014-12-05 21:15
jojaeho  pts/0        2014-12-05 22:51 (:0.0)
NEW$[/home/jojaeho/sys_program/HW2_SMSH]$ exit
Killed
jojaeho@ubuntu:~/sys_program/HW2_SMSH$
```

----

history명령어 관련 !3 을했을 때 세 번째로 실행했던 명령어가 다시 실행되게하는 것은 구현하지 못했습니다.