



---

# 시스템프로그래밍

## 과제 #4

“ Make MyShell ”

---

담당교수	최종무 교수님
학번	32164959
이름	허전진

## 1. Snapshot including bonus implements (redirection)

```
sys164959@embedded: ~
sys164959@embedded:~$ ./myshell
sys164959@/home/1-class/sys164959 >> help
-----HELLO, It's MyShell by HJJ at DKU-----
| These shell commands are defined internally. |
-----
CMD      FUNCTION
-----
help     Show this help again
quit     Quit MyShell
cd       Change directory
tty      Show connected terminal's filename
-----
KEY      FUNCTION
-----
>        REDIRECTION: OUTPUT
<        REDIRECTION: INPUT
&        Run task on background
sys164959@/home/1-class/sys164959 >> tty
current tty: /dev/pts/14
sys164959@/home/1-class/sys164959 >> cd dir01
sys164959@/home/1-class/sys164959 >> pwd
/home/1-class/sys164959/dir01
sys164959@/home/1-class/sys164959 >> cd ..
sys164959@/home/1-class/sys164959 >> ls
apple_new.txt apple.txt dir01 examples.desktop myshell myshell.c
sys164959@/home/1-class/sys164959 >> cat apple.txt > apple_copy.txt
sys164959@/home/1-class/sys164959 >> cat apple_copy.txt
I like apple :)
sys164959@/home/1-class/sys164959 >> ps
  PID TTY          TIME CMD
 17628 pts/14    00:00:00 bash
 21798 pts/14    00:00:00 myshell
 23880 pts/14    00:00:00 ps
sys164959@/home/1-class/sys164959 >> ls &
apple_copy.txt apple.txt examples.desktop myshell.c
apple_new.txt  dir01          myshell
sys164959@/home/1-class/sys164959 >> ps
  PID TTY          TIME CMD
 17628 pts/14    00:00:00 bash
 21798 pts/14    00:00:00 myshell
 23925 pts/14    00:00:00 ls <defunct> ←
 24110 pts/14    00:00:00 ps
sys164959@/home/1-class/sys164959 >> quit
sys164959@embedded:~$ whoami
sys164959
sys164959@embedded:~$ date
2019. 11. 05. (화) 02:02:32 KST
sys164959@embedded:~$
```

## 2. Discussion

### - Structure of myshell

<pre> shell: struct {     chk_cmd(char *line[]): int     cmd_cd(int argc, char *argv[]): bool     cmd_help(int argc, char *argv[]): bool     cmd_pipe(int argc, char *argv[]): void     cmd_quit(int argc, char *argv[]): bool     cmd_red_in(int argc, char *argv[]): void     cmd_red_out(int argc, char *argv[]): void     cmd_tty(int argc, char *argv[]): bool     main0: int     run(char *line): bool     tokenize(char *buf, char *delims, char *tokens[], int max): int     cmd[]: shell </pre>	구현한 내부 명령어	구현한 옵션
	help tty cd quit	redirection – STDOUT redirection – STDIN background processing pipe

정말 우여곡절이 많았던 과제였다. 아직 2학년이긴 하지만, 가장 오래 걸렸고 오류도 많아서 일주일을 내내 고민했다. 완벽한 셸을 만들어보고 싶어서 구글링을 100페이지는 넘게 한 것 같고 myshell을 putty에 썼다가 지웠다가를 최소 100번은 반복했다. 가장 힘들었던 문제 중 3가지를 뽑자면,

**첫 번째,** `execvp` 수행 시 같은 내용이 2번 출력되고 myshell을 빠져나가는 문제가 있었다.

```

sys164959@embedded:~$ ./err
sys164959@/home/1-class/sys164959 >> ls
apple_copy.txt  apple.txt  err  examples.desktop  myshell.c
apple_new.txt   dir01      err.c  myshell
apple_copy.txt  apple.txt  err  examples.desktop  myshell.c
apple_new.txt   dir01      err.c  myshell
sys164959@embedded:~$

```

이런 식인데, 이 문제 때문에 이를 동안 코드를 들여다보았다. 생각보다 어이없는 실수였는데, `if((pid=fork()) < 0)`로 블록처리를 해서 연산해야 하는데 `if(pid=fork()) < 0)`로 써버려서 정상적인 연산이 안 되었던 것이었다. 기본적인 C언어 문법을 숙지했어야 했는데 막상 해결하고 나니 정말 허무했던 문제였다.

**두 번째,** 백그라운드 프로세싱이 기능 구현 중에서 가장 골치 아팠다. WNOHANG, WUNTRACE 등의 정보들을 모두 찾아보고 우여곡절 끝에 백그라운드 태스크로 실행한 상태까지는 갔는데 셸에서 이상하게 보이는 문제가 있었다.

```

sys164959@/home/1-class/sys164959 >> ls &
sys164959@/home/1-class/sys164959 >> apple_copy.txt  apple.txt  err  err2.c
examples.desktop  myshell.c
apple_new.txt  dir01      err2  err.c  myshell

sys164959@/home/1-class/sys164959 >> ps
PID TTY          TIME CMD
 926 pts/14      00:00:00 err2
 930 pts/14      00:00:00 ls <defunct>
 944 pts/14      00:00:00 ps
17628 pts/14     00:00:00 bash

```

ls 명령어를 `defunct`, 즉 백그라운드 프로세싱하는 결과는 문제가 없었는데, ls의 출력내용이 셸의

타이핑하는 영역에 들어간 것이었다. 아무리 생각해도 run 함수에서 타이밍이 꼬여서 출력 결과를 잘못된 위치에 출력하는 것 같아서 고민한 결과 waitpid(pid, &stat, WNOHANG) 뒤에 sleep(1)을 추가해 주니 정상적으로 작동되었다.

**세 번째,** redirection 시 파일은 정상적으로 생성되지만 끝에 에러가 출력되는 문제였다.

```
sys164959@/home/1-class/sys164959 >> cat apple.txt > apple_new.txt
cat: '>': No such file or directory
cat: apple_new.txt: input file is output file
sys164959@/home/1-class/sys164959 >> cat apple_new.txt
I like apple :)
```

redirection을 포함하여 cat을 실행하면 오류가 뜨면서 안되는 것처럼 보이지만 실제 apple\_new.txt는 정상적으로 생성되었다. cat이 '>'을 파일로 인식하는 것 같아서 > symbol이 필요한 역할을 다 했다면 재 초기화를 해야 할 것 같아서 '>'을 null로 초기화하니 해결되었다.

이 문제들 말고도 정말 많은 에러가 있었다. 과제 기간이 일주일이었는데, 정말 일주일을 쉼을 위해 다 쓴 것 같다. 그래도 내 손으로 200줄가량의 프로그램을 만들고 나니 그 뿌듯함은 말로 설명할 수 없었다. 교수님의 말씀처럼 지금까지 해왔던 과제 중 가장 길고 난이도 있던 과제였는데 shell을 직접 만들어보면서 task의 동작 원리와 명령어들을 확실하게 공부할 수 있는 계기가 되었다.