



과목명	시스템프로그래밍
담당교수	최종무 교수님
학과	소프트웨어학과
학번	32153180
이름	이상민
제출일자	2018.11.20

실행 화면

```
220.149.236.2 - PuTTY
sys32153180@embedded:~/homework$ ls
hello hello.c mycp mycp.c mycp.txt mysh.c
sys32153180@embedded:~/homework$ gcc -o mysh mysh.c
sys32153180@embedded:~/homework$ ./mysh
[ /home/2-class/sys32153180/homework ]
sangmin $ ./hello
Hello World!
[ /home/2-class/sys32153180/homework ]
sangmin $ help
-----
Start simple shell by. sangmin
=> implement basic logic (parsing, fork, execve)
=> implement background processing and redirection

As follows some examples
-----
cd      : change directory
quit    : quit this shell
help    : show this help
>       : output redirection
<       : input redirection
&       : background processing
[ /home/2-class/sys32153180/homework ]
sangmin $ ls > r.txt
[ /home/2-class/sys32153180/homework ]
sangmin $ ls
hello hello.c mycp mycp.c mycp.txt mysh mysh.c r.txt
[ /home/2-class/sys32153180/homework ]
sangmin $ cat < r.txt
hello
hello.c
mycp
mycp.c
mycp.txt
mysh
mysh.c
r.txt
[ /home/2-class/sys32153180/homework ]
sangmin $ cd
[ /home/2-class/sys32153180 ]
sangmin $ cd homework

[ /home/2-class/sys32153180/homework ]
sangmin $ whoami
sys32153180
[ /home/2-class/sys32153180/homework ]
sangmin $ date
2018. 11. 19. (월) 17:47:35 KST
[ /home/2-class/sys32153180/homework ]
sangmin $ ps
  PID TTY          TIME CMD
 13264 pts/38    00:00:00 bash
 16102 pts/38    00:00:00 mysh
 16486 pts/38    00:00:00 ps
[ /home/2-class/sys32153180/homework ]
sangmin $ quit
sys32153180@embedded:~/homework$ ps
  PID TTY          TIME CMD
 13264 pts/38    00:00:00 bash
```

Discussion

이번 과제는 나만의 shell을 만드는 것이었다. 개인적으로 지금까지의 시스템 프로그래밍 과제 중 가장 어려웠다. 수업시간에 배운 이론은 받아들이기 쉬웠는데, 그것을 토대로 코딩을 하려고 하니 결코 쉽지 않았다. ppt에 있는 예시를 보며 혼자 하다가 계속 막혀서 검색도 많이 해보았다. 가끔 500줄이 넘어가는 엄청난 양의 코드들도 있었는데 가독성이 중요하다는 것을 느꼈다. 물론 프로그램이 복잡해지는 만큼 코드가 길어지는 것은 당연하지만 그래도 최대한 다른 사람이 알아보기 쉽도록 코딩해야겠다 느꼈다.

나는 우선 myShell이라는 구조체를 만들어 여러 함수들이 작동할 수 있도록 만들었다. cd, quit 그리고 help 명령어 함수를 만드는 것은 많이 어렵지 않았다. 그에 반해 redirection과 background processing을 하도록 해주는 것이 너무 어려웠다. 이 두 가지를 해결하기 위해 상당히 많은 자료들과 ppt를 참고했다. 우선 shell의 기본적인 원리를 알아야했다. shell은 fork()를 한 후, 자식 프로세스를 execve()함으로써 명령어를 실행하는 것이다. 따라서 redirection은 데이터를 STDIN/STDOUT 대신 파일에 쓰거나 파일로부터 읽어오는 것이고, background processing은 shell과 명령어가 동시에 수행되게끔 하는 것이다. redirection 함수를 만들 때 복사를 위해서 dup2()라는 시스템 호출을 사용했다. 수업시간에 필기한 내용을 보니 background processing을 하기 위해서는 wait()를 없애야 한다고 되어 있었다. wait()를 사용하면 부모 프로세스가 자식 프로세스가 종료될 때까지 기다리는 것이기 때문이다. 하지만 단순히 wait()를 없앤다고 바로 성공하지 않았다. 그래서 인터넷을 찾아보니 waitpid()를 사용하면 조건을 걸 수 있고, 그 중 WNOHANG이라는 조건을 사용해주면 부모 프로세스가 기다리지 않는다고 하여 사용했다.

이번 shell 과제는 정말 오랫동안 붙잡고 있었다. 부족한 점들을 찾아보고 공부해가며 했기 때문에 시간은 오래 걸렸지만 다 하고 실행된 파일을 보면 정말 뿌듯했다. 지금까지 배운 내용들을 거의 다 활용할 수 있는 좋은 과제였다고 생각한다. 앞으로 공부를 더 열심히 해서 아무 도움 없이 이러한 프로그램을 만들 수 있었으면 좋겠다.