**Operating System HW 1**

# Simple Shell

20123496 김현태

1. prompt

코드에 prompt라는 함수를 만들어서 루프 마다 한번씩 호출되도록 만들었습니다. 호출은 다음과 같습니다.

Hyeontae Kim $

2. fork repeatly

fork 를 하게 되면 동일한 프로세스가 복사됩니다. Parent process와 Child pocess 로서 fork 되었을 때 return 되는 pid 값으로 서로를 확인할 수 있습니다.

만약 pid < 0 이라면 에러가 발생한 것으로 확인할 수 있습니다.

Pid = 0 이라면 이는 child process입니다.

Pid > 0 ( child process id ) 라면 이는 parent process입니다.

While 루프에서 fork하고 pid값을 확인하는 방식으로 코드를 작성했습니다.

3. wait

while 루프 안에서 parent process 구문에 wait을 실행시킵니다.

해당 구문을 통해서 child process가 끝날 때 까지 기다리게 했습니다. ( stat\_val 값을 확인하는 로직은 작성하지 않았습니다. )

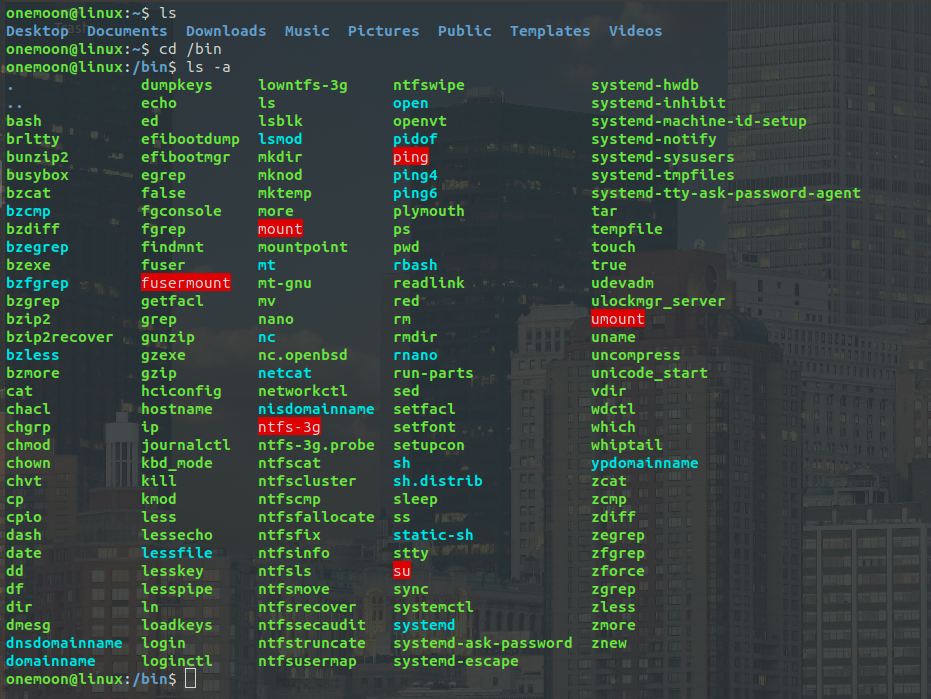
4. exit

while 루프 초기에 값을 입력받는 함수가 있으며 strcmp 를 이용해서 exit이 입력 되었다면 while 루프를 빠져나가는 구문을 작성 했습니다.

5. shell 명령어는 되지만 exec 명령어가 되지 않는 경우

shell 명령어는 내부 명령어( Internal Commands and Builtins ) 로서 기존 리눅스 명령어와의 차이가 있습니다. 예를 들어서 cd 와 같은 경우 경로와 해당 경로의 파일들의 존재 등 내부 쉘의 환경변수를 알아야 하고, 성능상의 이슈 때문에 편의를 위해 따로 만들어준 명령어들 입니다. 하지만 exec에서 실행시킬 수 있는 명령어는 /bin 에 있는 명령어로 한정되어 있고 이에 따라서 차이가 생기는 것입니다.

아래 스크린샷은 exec로 실행할 수 있는 명령어들 입니다.



# 실행 스크린샷

