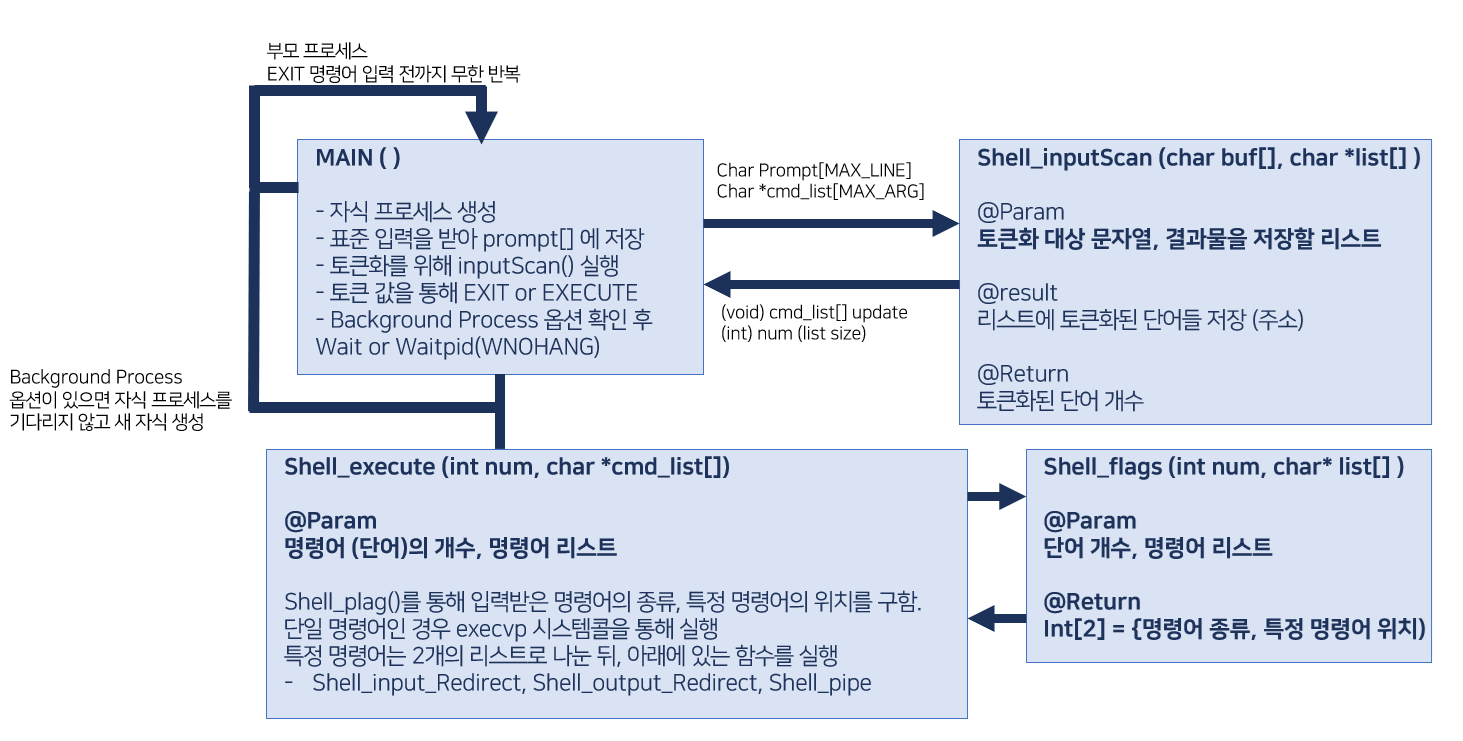
Operating Systems

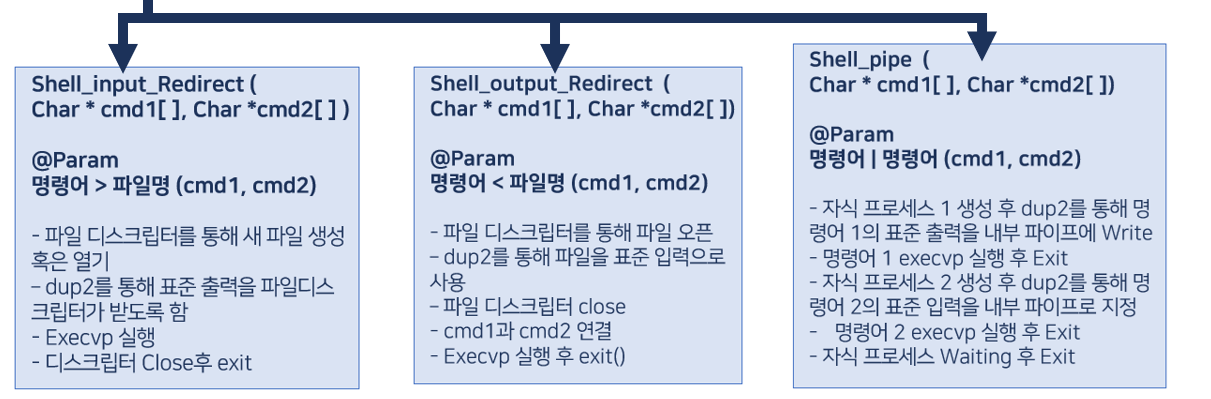
Project #1

COM2005 – 24081

소프트웨어학부 컴퓨터전공

2018045214 최준희

0. Simple Shell Algorithm

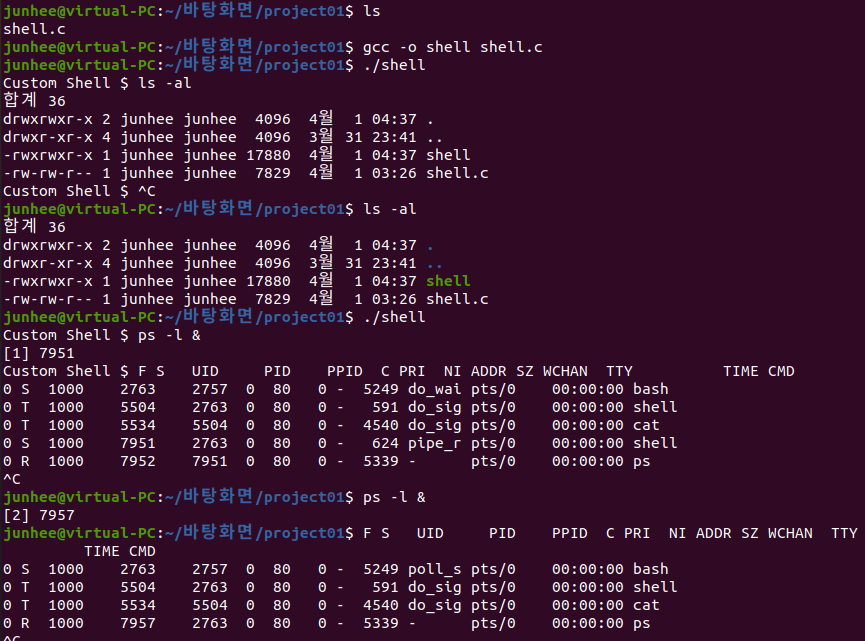


프로그램의 흐름은 위 사진과 같으며 세세한 부분은 2. Code의 주석에서 확인할 수 있다. 전반적으로 부모 프로세스를 통해 프로그램이 반복되며, 한 줄의 명령어를 위해 자식 프로세스가 생성, inputScan과 execute를 맡게 된다. 이 후 입력 받은 명령어의 특성에 따라 바로 execvp()가 실행되는 경우도 있고, 약간의 전처리를 거쳐 input, output\_Redirect(), shell\_pipe() 를 실행하는 경우도 있다.

여기서 유일하게 pipe()는 자식 프로세스의 자식 프로세스 (손자 프로세스) 를 생성, 명령어 1의 출력을 명령어 2에 넣고 나온 값을 프로세스가 받아 출력하는 구조이다.

1. Compile & Execute Result

* 1. 단일 명령어 조합

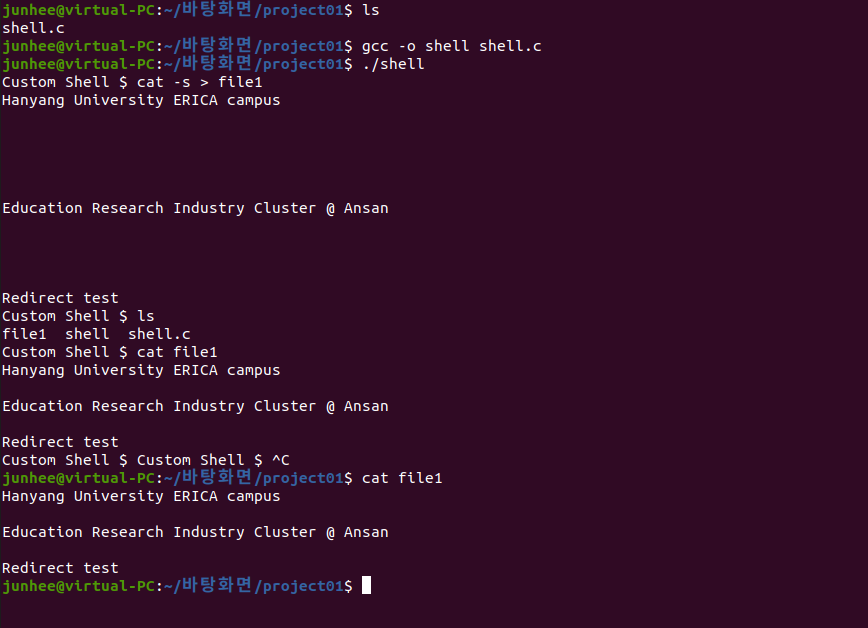


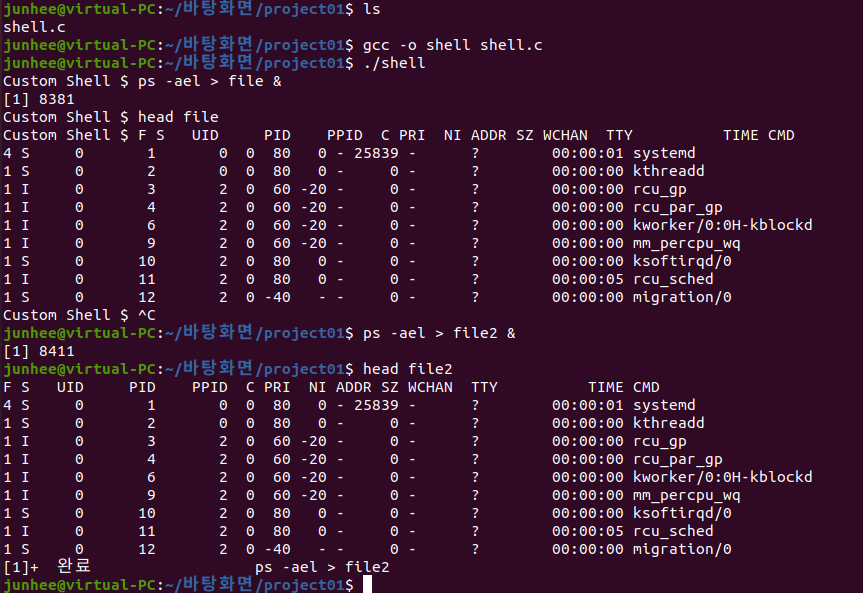
사용한 명령어

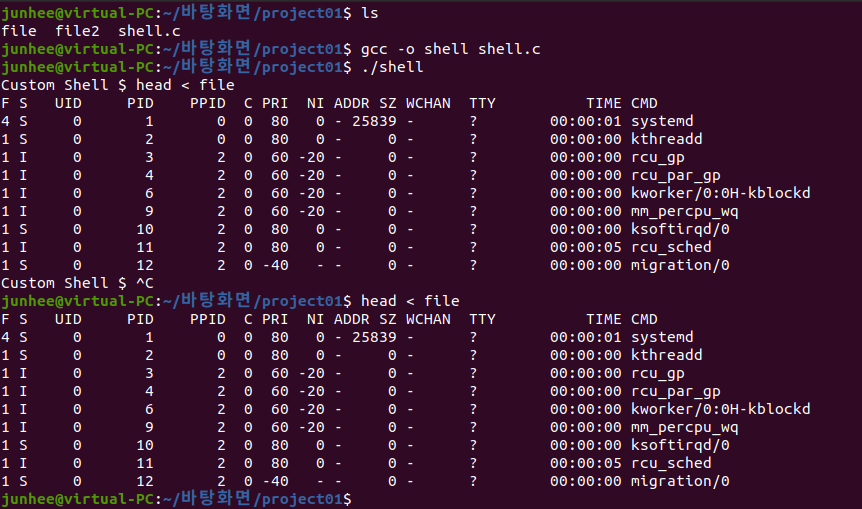
**ls -al**

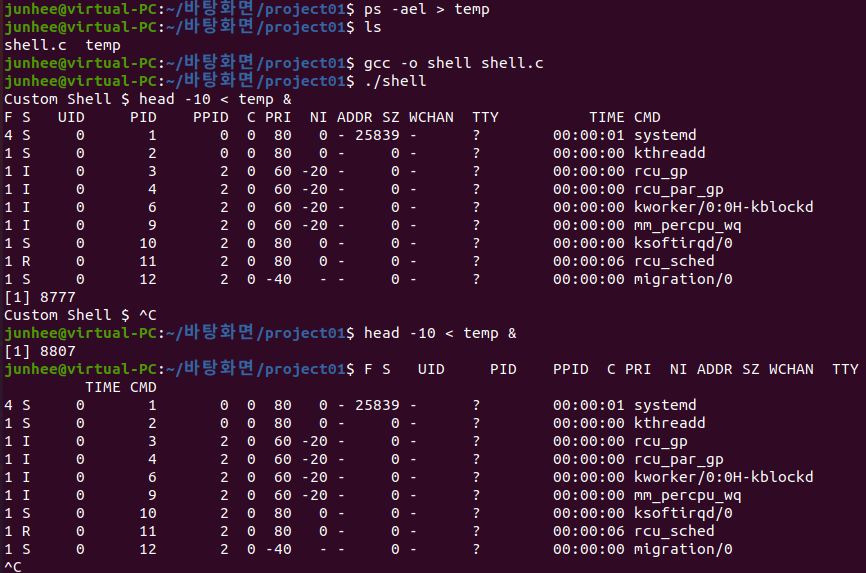
**ps -l &**

* 1. 리다이렉트 명령어 조합







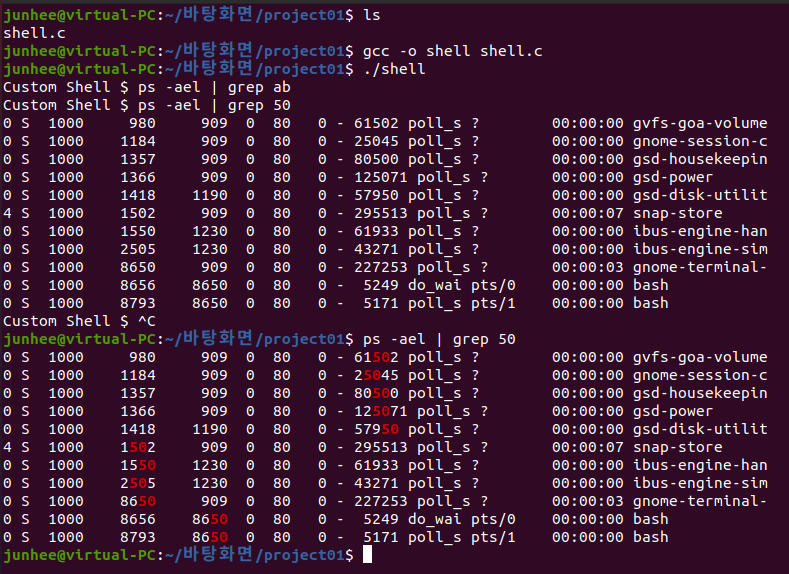


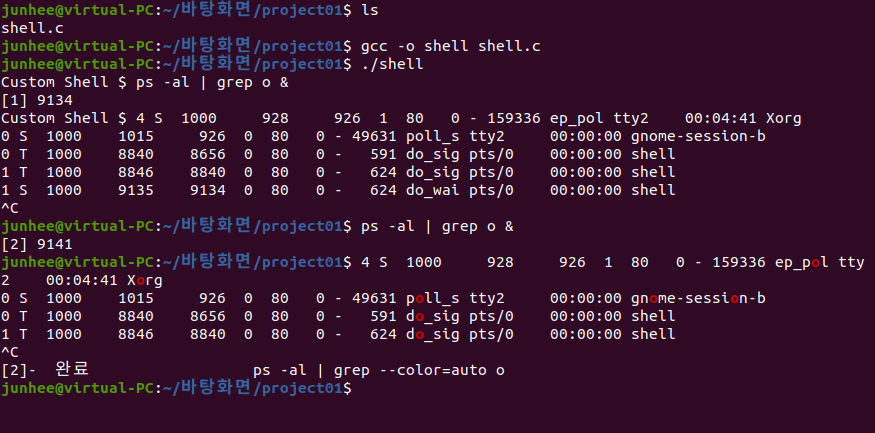
사용한 명령어

**cat -s > file1**, cat file1, **ps -ael > file &**

**head < file**, ps -ael > temp, **head -10 < temp &**

* 1. 파이프 명령어 조합

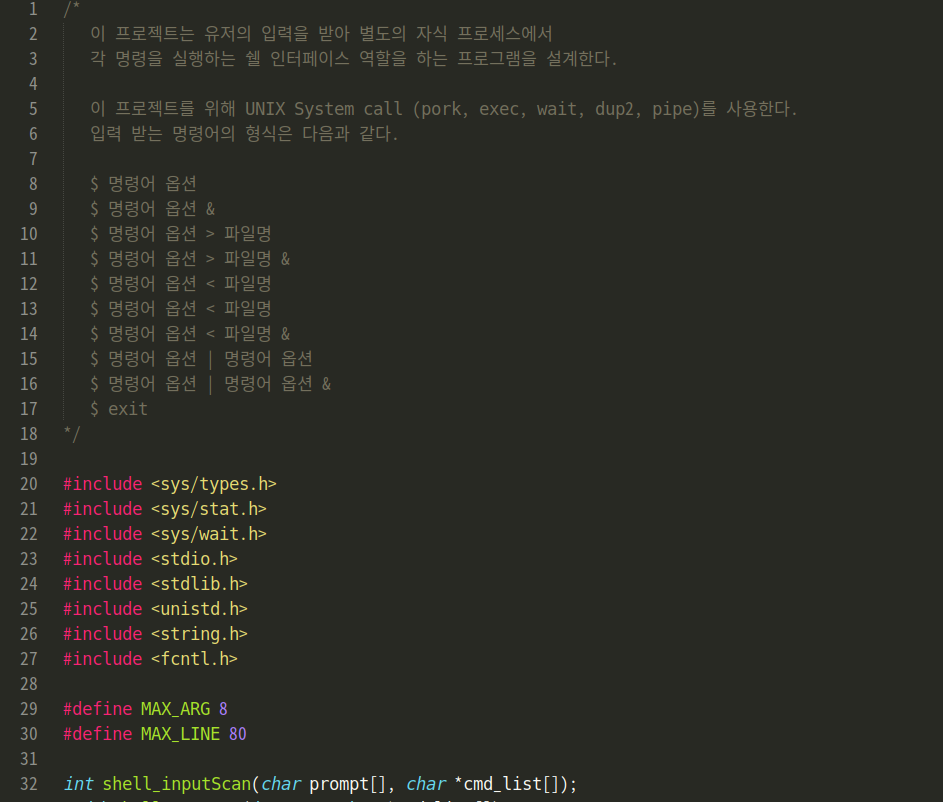


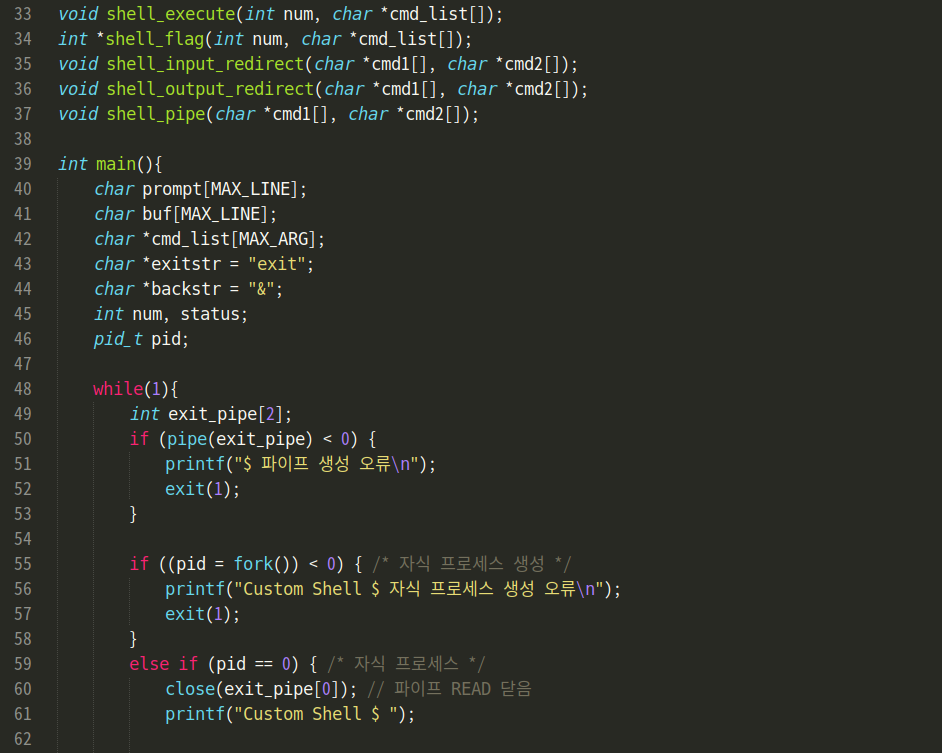


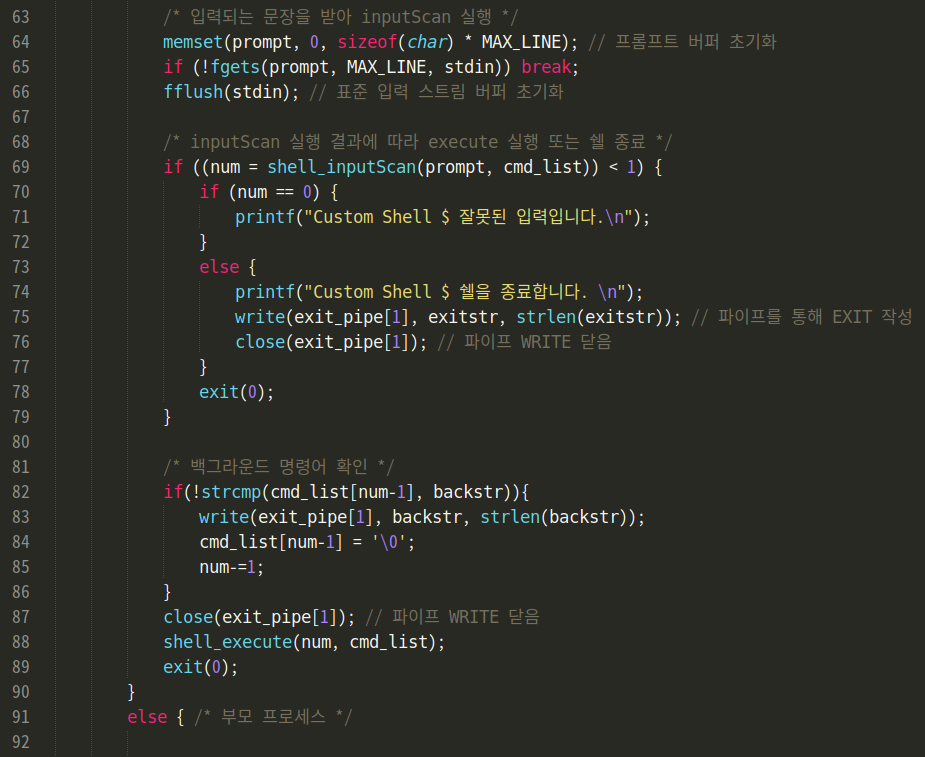
사용한 명령어

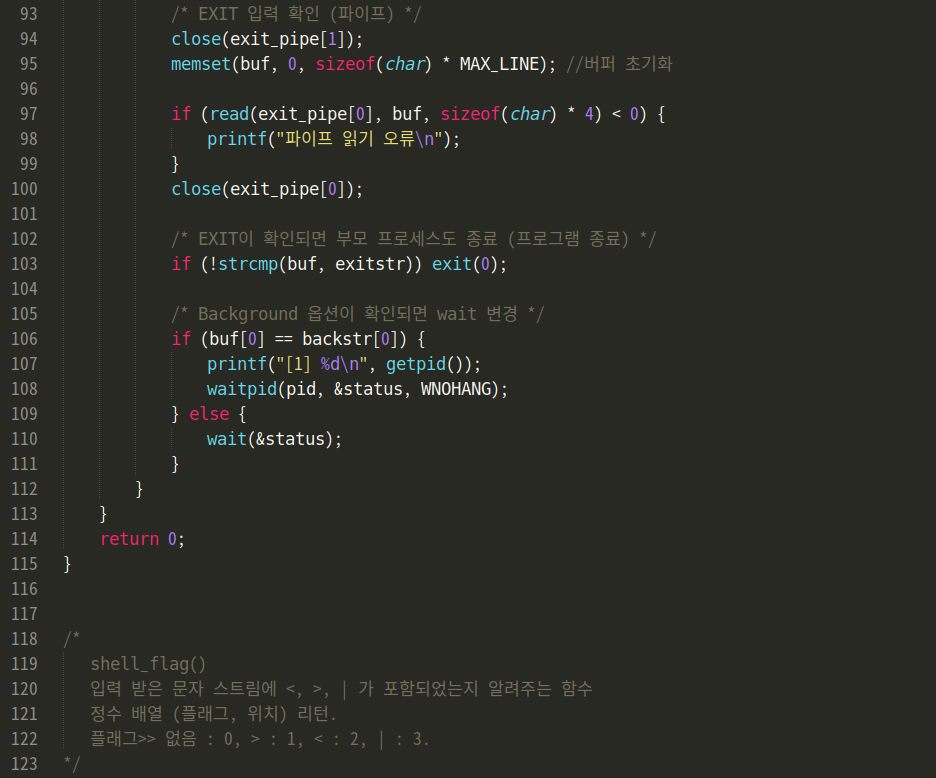
**ps -ael | grep 50**

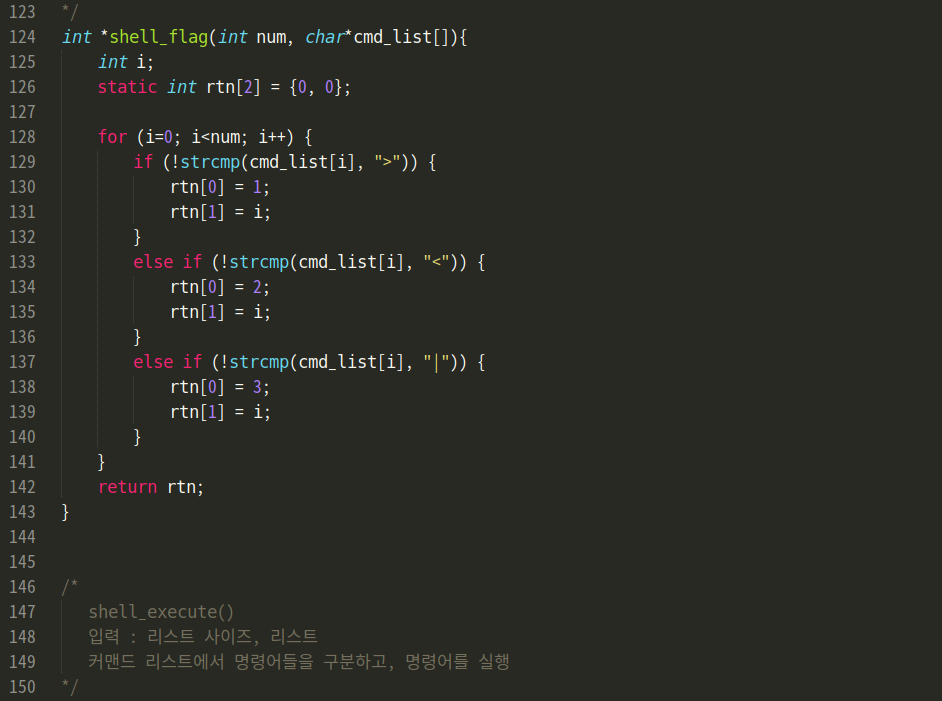
**ps -al | grep o &**

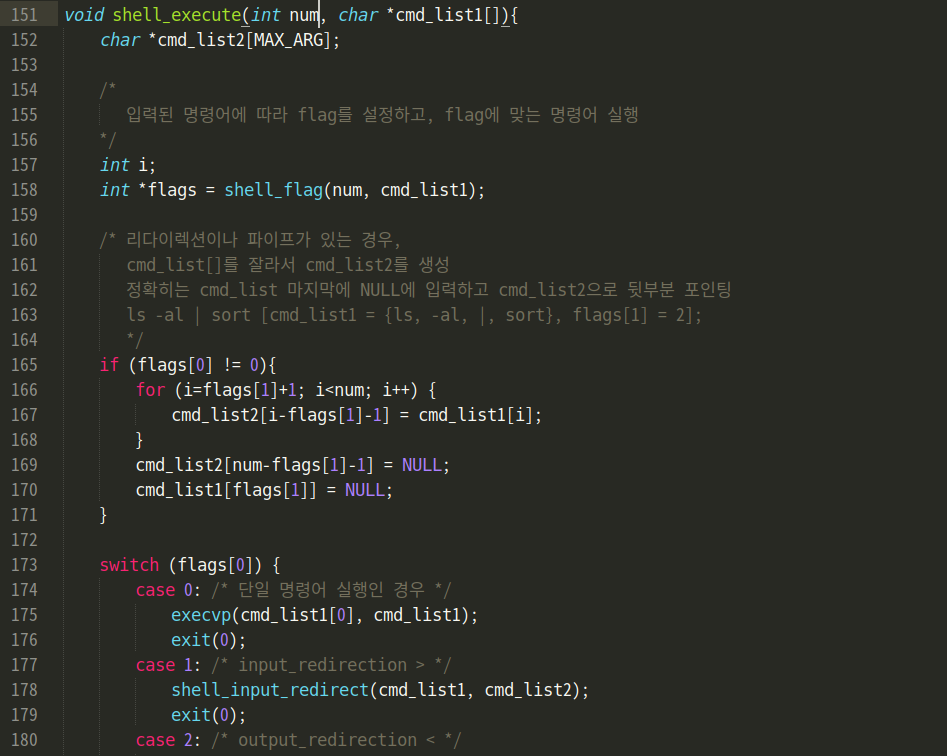
2. Code

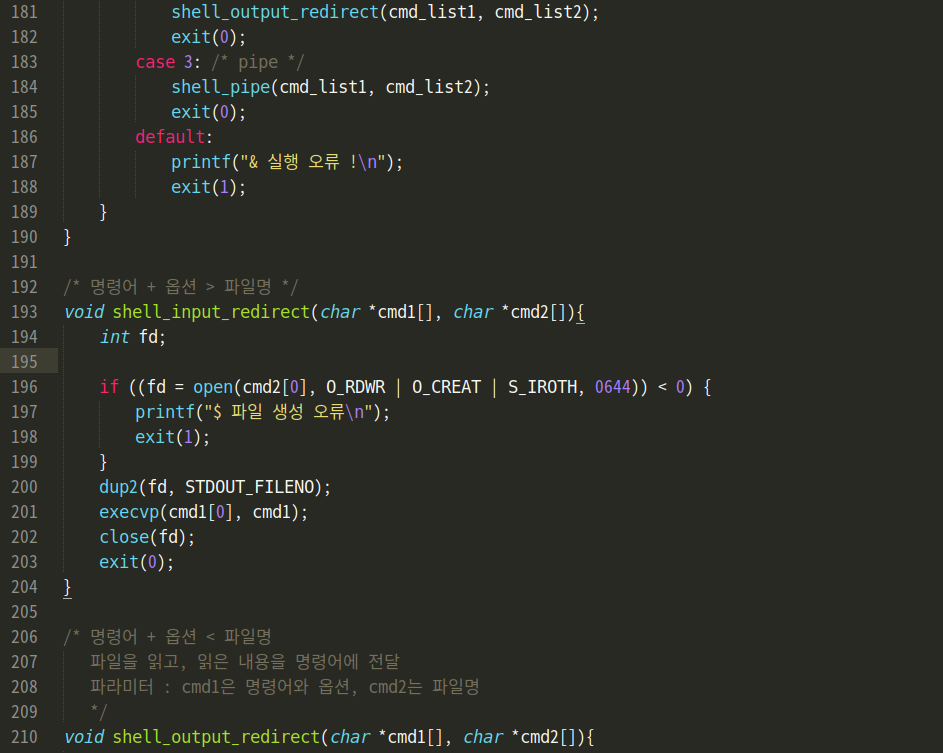
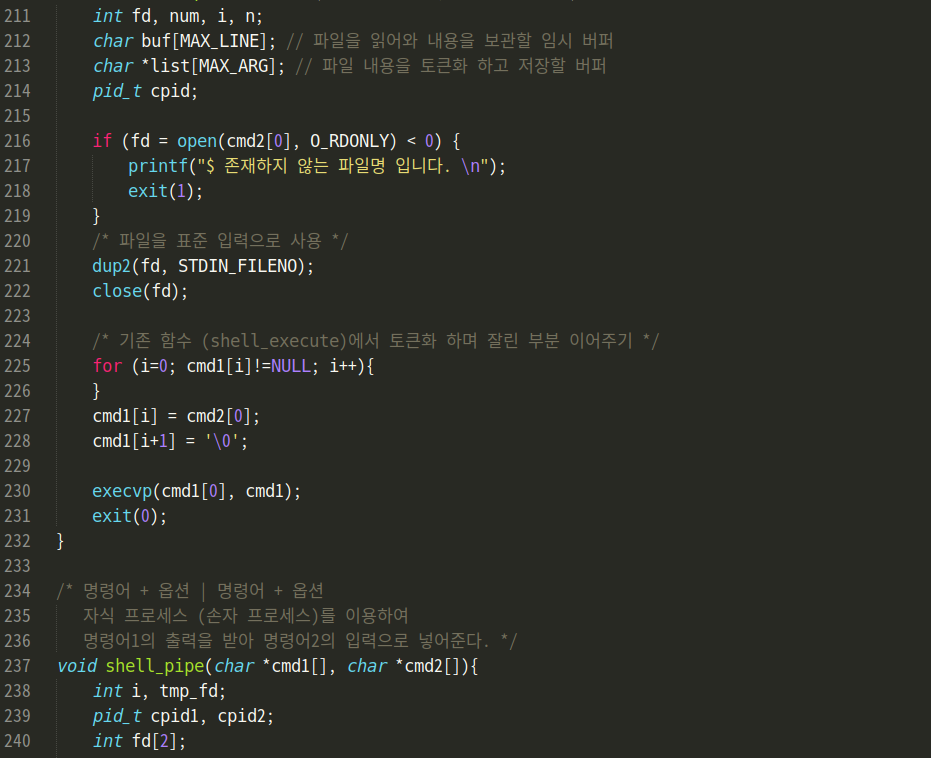


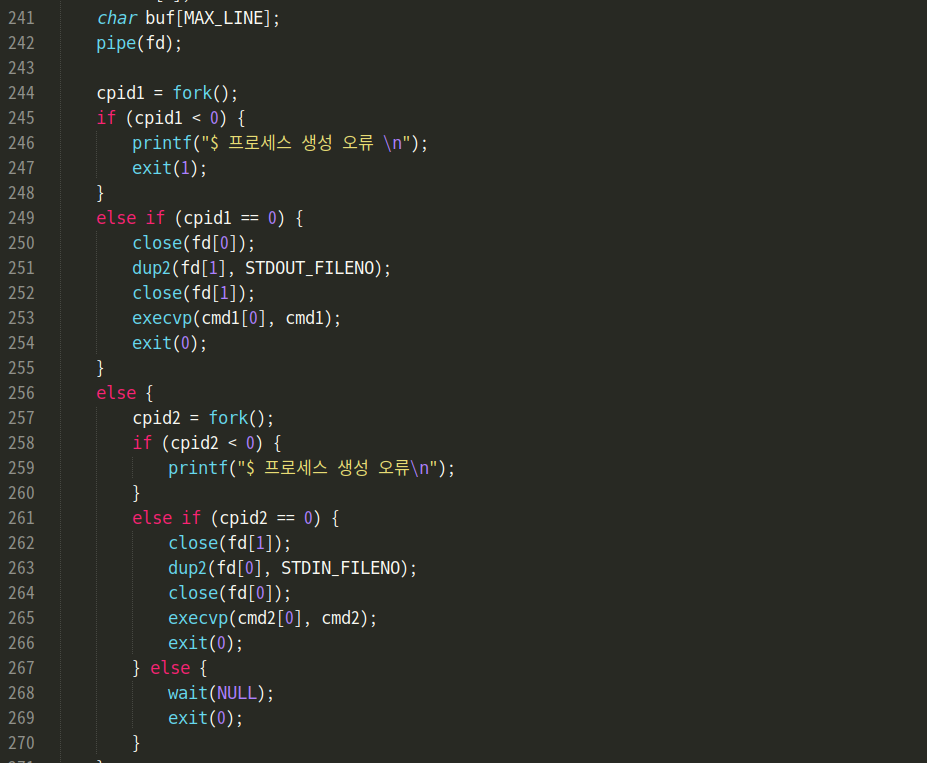


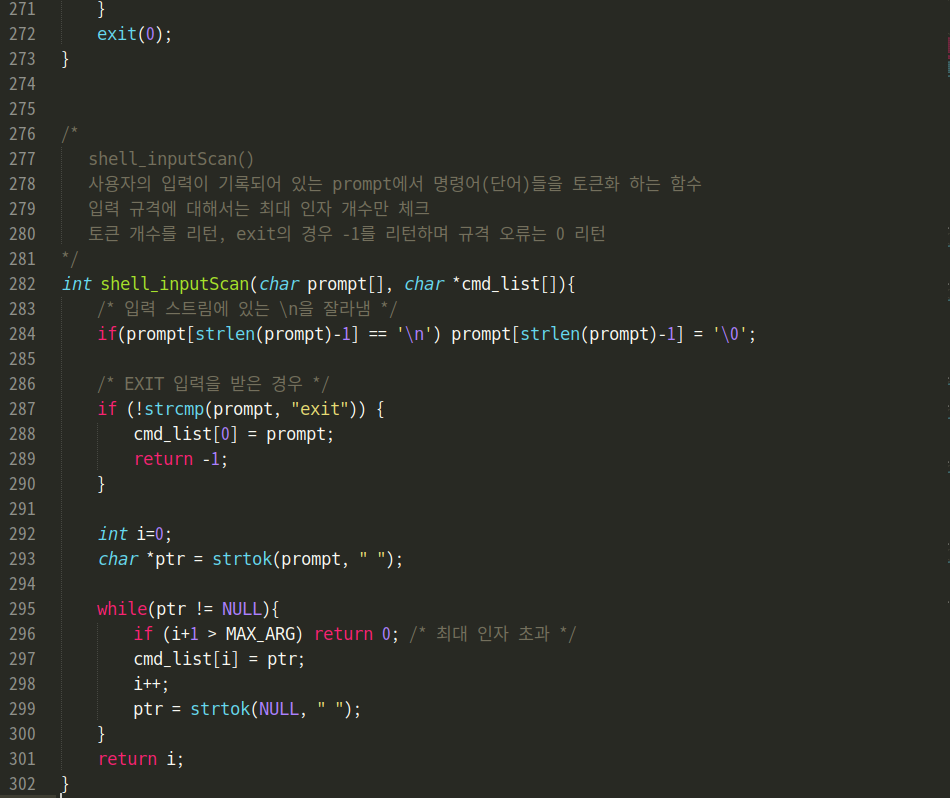












감사합니다.