

운영체제 프로그래밍 과제 I

2016.9

< 간단한 Command Executor 만들기 >

이번 프로그래밍 과제에서는 Unix에서 수행되는 명령어들을 입력 받아 차례로 수행하는 프로그램입니다. 수행할 명령어를 차례로 입력하면 일단 받아들인 명령어들을 수행의 반대순서로 출력해 줍니다. 그 후에는 입력된 명령어를 차례대로 수행하게 됩니다. 그리고 어떤 프로세스가 해당 명령어를 수행하는지 또 명령어 수행 후에 exit status 는 어떤지 출력합니다. 명령어 수행을 실패하였을 경우 에러 메시지를 출력합니다. 참고로 명령어는 절대경로를 다 써주어야 합니다.

0) 수행 예

다음은 /bin/ls, /bin/ps -l, ok 세 명령어를 수행한 예입니다.

```
>./cmdor
/bin/ls
/bin/ps -l
ok
Command line : ok
Command line : /bin/ps -l
Command line : /bin/ls
*** Process 2814 Executing Command /bin/ls
CBP ECL cmdor commands shell shell.c shell.o
*** Process 2813 Child process 2814 DONE with status 0

*** Process 2813 Executing Command /bin/ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  2696  2695  0  80   0 -  5789 wait  pts/18  00:00:00 bash
0 S  1000  2811  2696  0  80   0 -  1084 wait  pts/18  00:00:00 cmdor
1 S  1000  2812  2811  0  80   0 -  1084 wait  pts/18  00:00:00 cmdor
0 R  1000  2813  2812  0  80   0 -  1785 -      pts/18  00:00:00 ps
*** Process 2812 Child process 2813 DONE with status 0

*** Process 2812 Executing Command ok
ERROR - not executing command "ok"
*** Process 2811 Child process 2812 DONE with status 100
```

1) 출력문

출력문에서는 명령어를 수행하는 프로세스의 id를 출력하고 있습니다. 또 child 프로

세스의 수행 status 도 출력하고 명령어를 수행 못하는 경우 이또한 출력합니다. 출력문은 다음 문장을 사용하여 바꾸지 말고 동일하게 출력 할 수 있도록 합니다. 출력문에서 사용하는 변수는 바꾸어도 됩니다.

```
printf("*** Process %d Child process %d DONE with status %x\n\n",getpid(),child_pid,status);
printf("*** Process %d Executing Command %s",getpid(),cmd_line);
printf("ERROR - not executing command \"%s\"\n",parameters[0]);
```

2) main() 함수는 다음과 같습니다.

```
int main() {
    readCommandLines();
    printCommands();
    executeCommands();
    return(0);
}
```

readCommandLines(), printCommands(), executeCommands() 세 함수를 구현하면 됩니다. 특히 executeCommands() 함수는 다음과 같은 조건을 만족하여 구현되어야 합니다.

executeCommands() 함수 구현 조건

- 1) 본 함수에서 child 프로세스를 만들어 명령어를 수행 합니다.
- 2) 각각의 명령어는 서로 다른 프로세스가 수행하여야 합니다.
- 3) 수행해야 될 명령어가 남아 있는 경우 child 프로세스가 해당 프로세스의 child (손자) 프로세스를 만들어 수행합니다.
- 4) 3)을 위해 executeCommands()에서 여러 명령어 처리를 위하여 loop 구조를 사용하지 말고 executeCommands() 함수는 **recursive 함수로** 구현해야 합니다.

그 외 주의 및 참고 사항 :

- 모든 프로그래밍 과제는 C 언어를 사용합니다. C++ 사용하지 마십시오.
- exec() 시스템콜은 execve()를 사용하고 execve()의 세 번째 인자는 NULL로 처리한다.
- 제출하는 .c 파일에는 이름과 학번을 첫 주석에 만들어 넣어야 합니다.