

2019 시스템 프로그래밍

- week07 -

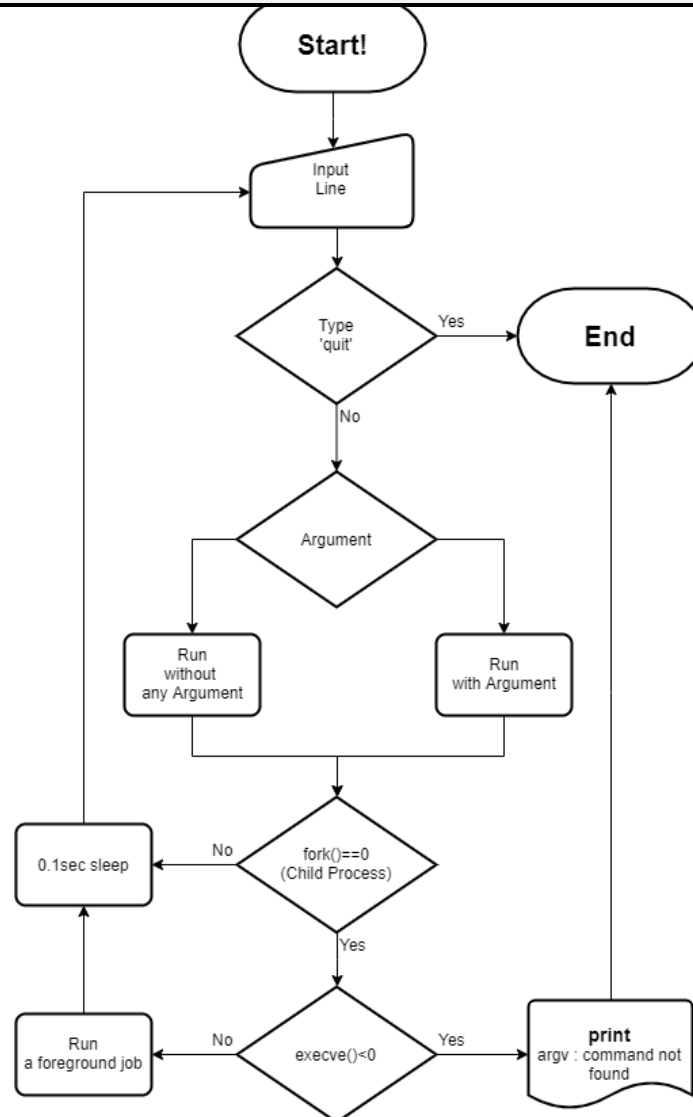
제출일자	2019.11.13
분 반	01
이 름	임준규
학 번	201602057

Trace 번호 (03)

```
b20160205/02019sp:~/shlab-handout$ ./sdriver -V -t 03 -s ./tsh
Running trace03.txt...
Success: The test and reference outputs for trace03.txt matched!
Test output:
#
# trace03.txt - Run a synchronizing foreground job without any arguments.
#
Reference output:
#
# trace03.txt - Run a synchronizing foreground job without any arguments.
#
```

Ref와 비교

각 trace 별 플로우 차트



Trace 해결 방법 설명

2번과 3번, 그리고 4번은 같은 문제라고 봐도 무방하다.

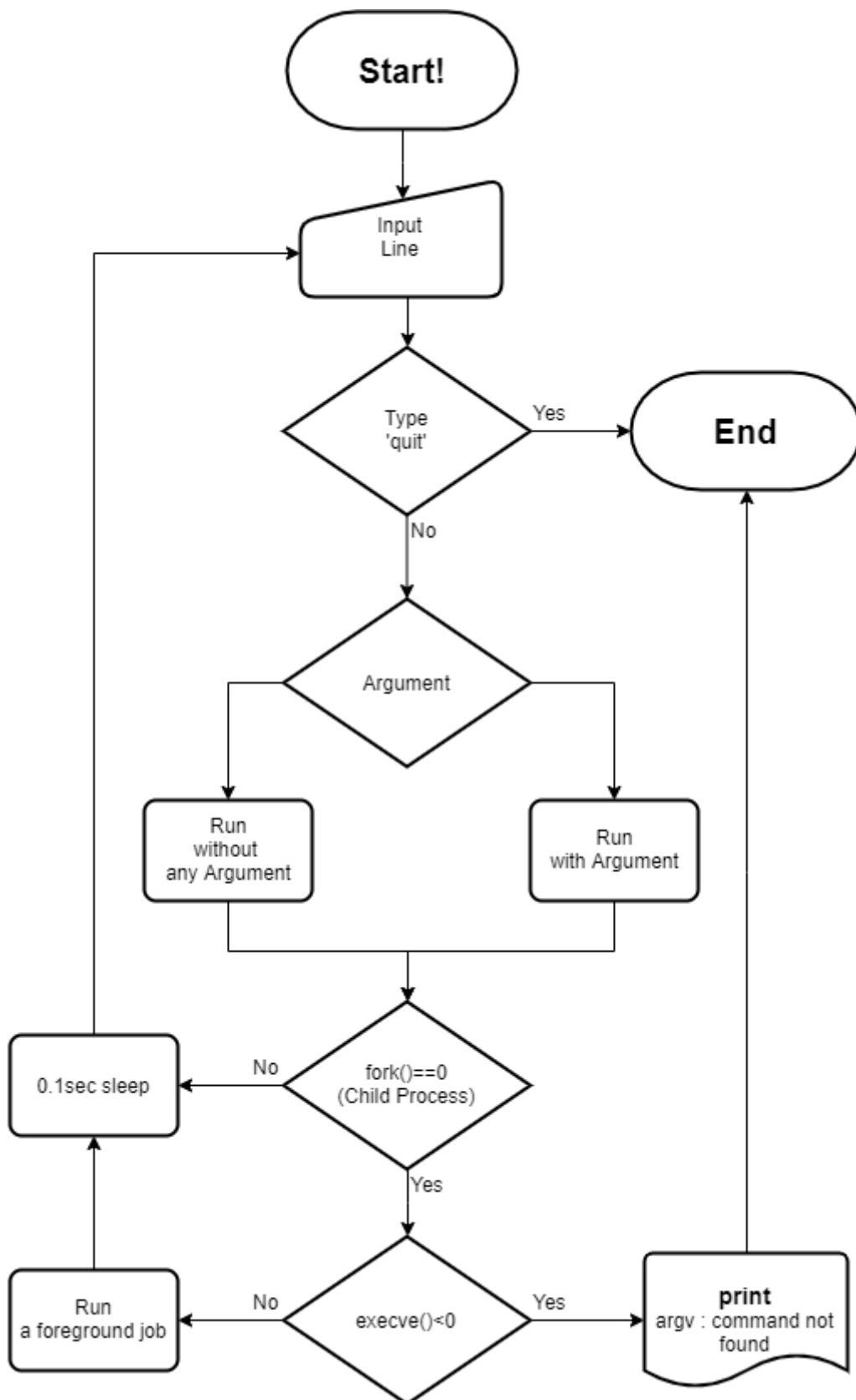
2번과 똑같이 입력을 읽고 하는 과정은 동일하나 이제 매개변수라는 존재가 생겼으므로, 매개변수를 확인해주고, 있다면 매개변수와 함께, 없다면 매개변수가 없이 실행하거나 프로그램의 default값으로 실행한다.

Trace 번호 (04)

```
b201602057@2019sp:~/shlab-handout$ ./sdriver -V -t 04 -s ./tsh
Running trace04.txt...
Success: The test and reference outputs for trace04.txt matched!
Test output:
#
# trace04.txt - Run a foreground job with arguments.
#
tsh> quit

Reference output:
#
# trace04.txt - Run a foreground job with arguments.
#
tsh> quit
```

Ref와 비교



Trace 해결 방법 설명

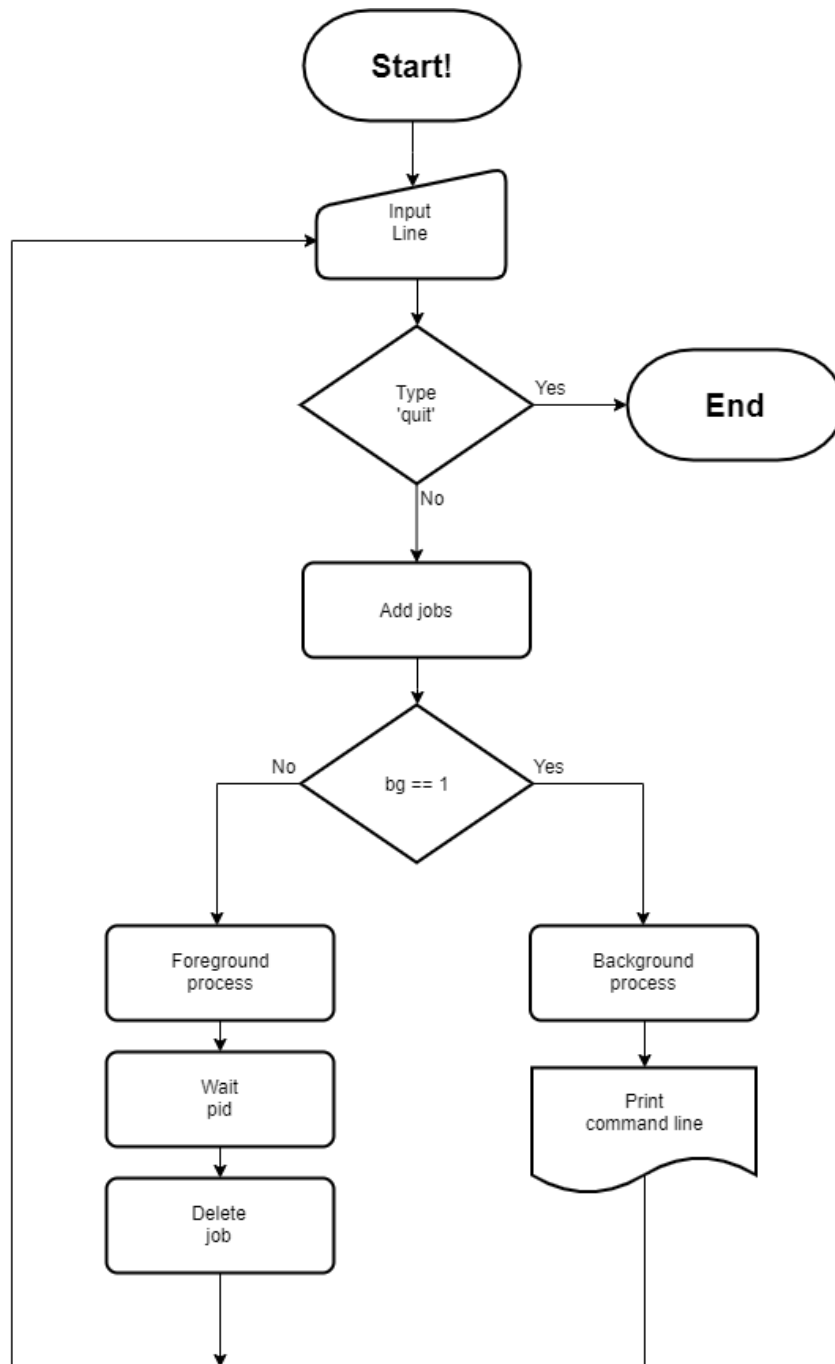
2번과 똑같이 입력을 읽고 하는 과정은 동일하나 이제 매개변수라는 존재가 생겼으므로, 매개변수를 확인해주고, 있다면 매개변수와 함께, 없다면 매개변수가 없이 실행하거나 프로그램의 default값으로 실행한다.

Trace 번호 (05)

```
b201602057@2019sp:~/shlab-handout$ ./sdriver -V -t 05 -s ./tsh
Running trace05.txt...
Success: The test and reference outputs for trace05.txt matched!
Test output:
#
# trace05.txt - Run a background job.
#
tsh> ./myspin1 &
(1) (31822) ./myspin1 &
tsh> quit

Reference output:
#
# trace05.txt - Run a background job.
#
tsh> ./myspin1 &
(1) (31831) ./myspin1 &
tsh> quit
```

Ref와 비교하여 일치함.



5와 6도 사실상 같은 문제이다.

foreground실행과 background실행을 만든 후에, background실행명령을 입력에서 bg에 저장해주고, background 실행이라면 background실행을 해주고, 이럴 경우에 돌아가는 지 알 수 없으므로 cmd라인을 프린트 해주도록한다.

Trace 번호 (06)

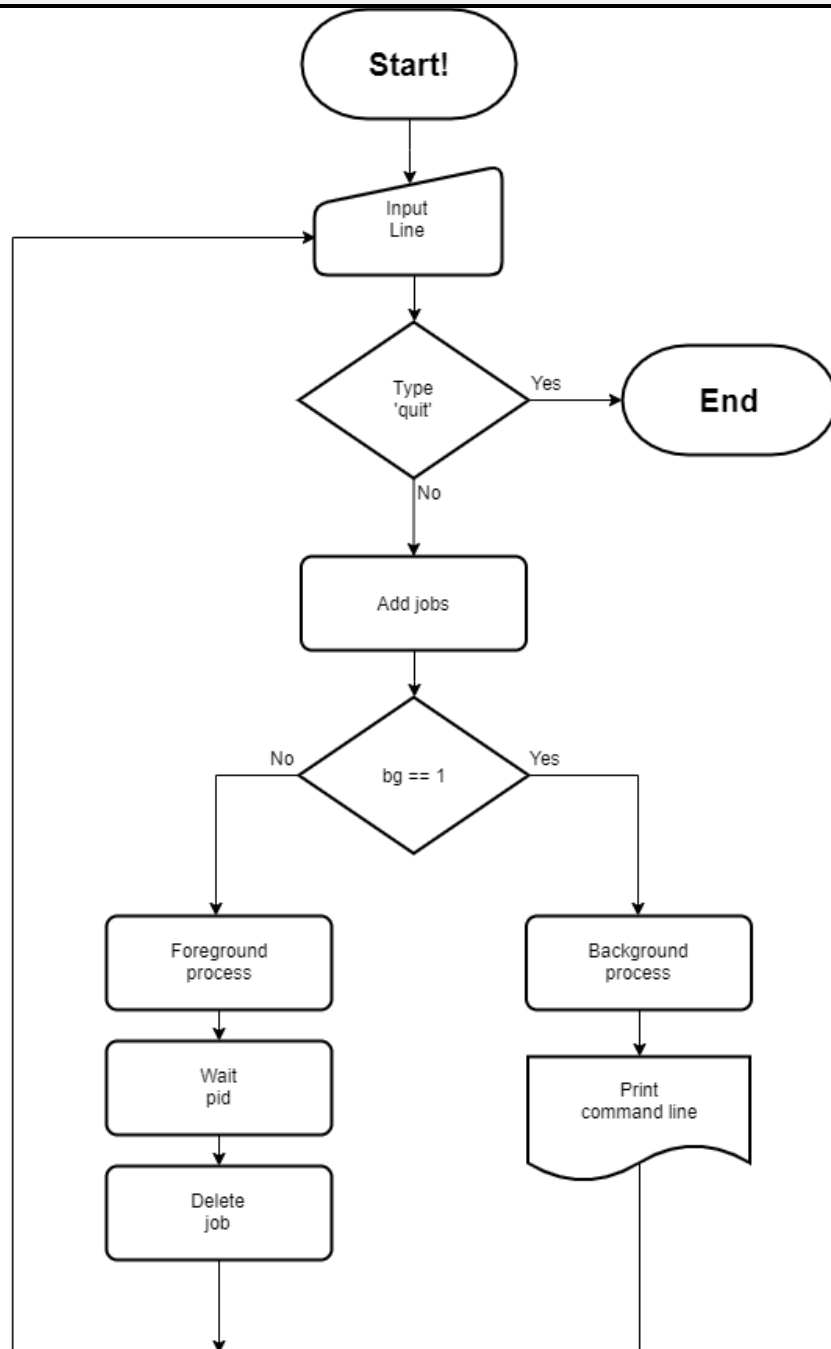
```
b201602057@2019sp:~/shlab-handout$ ./sdriver -V -t 06 -s ./tsh
Running trace06.txt...
Success: The test and reference outputs for trace06.txt matched!
Test output:
#
# trace06.txt - Run a foreground job and a background job.
#
tsh> ./myspin1 &
(1) (32255) ./myspin1 &
tsh> ./myspin2 1

Reference output:
#
# trace06.txt - Run a foreground job and a background job.
#
tsh> ./myspin1 &
(1) (32264) ./myspin1 &
tsh> ./myspin2 1
```

Ref와 비교

]

각 trace 별 플로우 차트



Trace 해결 방법 설명

지금까지 우리는 back/foreground 프로그램을 실행하는 것을 설계해왔다.

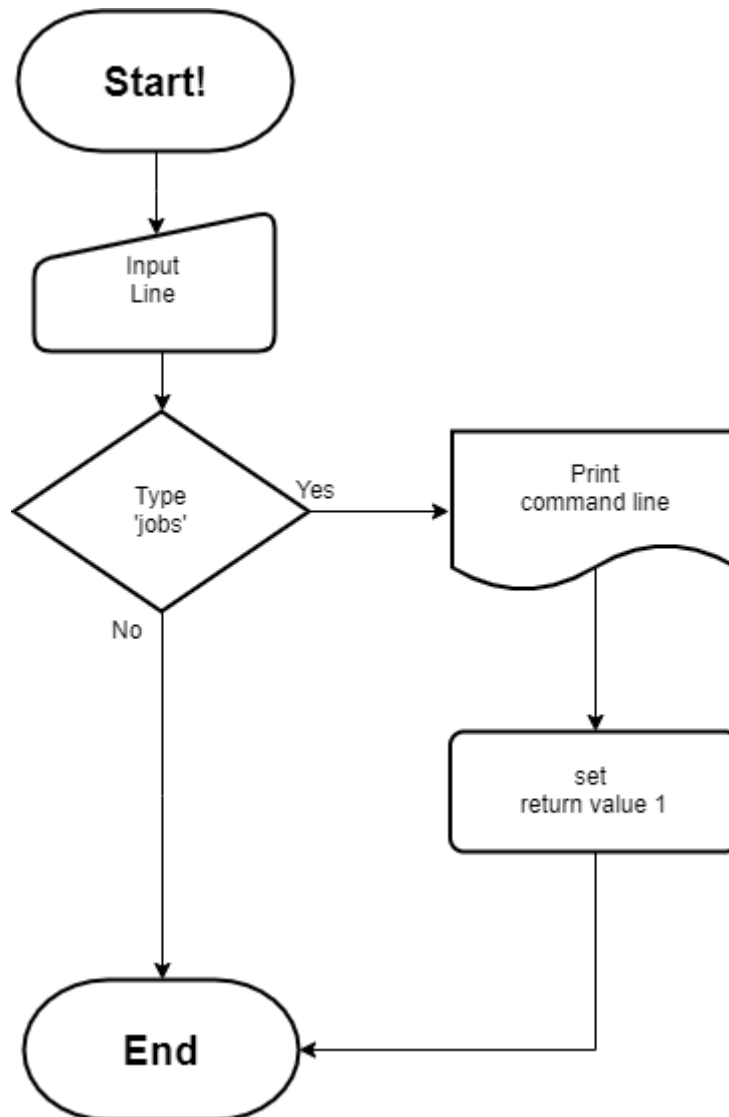
이렇게 모두 설계하였으므로 6번 또한 당연히 됨을 확인할 수 있다.

Trace 번호 (07)

```
b201602057@2019sp:~/shlab-handout$ ./sdriver -V -t 07 -s ./tsh
Running trace07.txt...
Success: The test and reference outputs for trace07.txt matched!
Test output:
#
# trace07.txt - Use the jobs builtin command.
#
tsh> ./myspin1 10 &
(1) (16126) ./myspin1 10 &
tsh> ./myspin2 10 &
(2) (16128) ./myspin2 10 &
tsh> jobs
(1) (16126) Running    ./myspin1 10 &
(2) (16128) Running    ./myspin2 10 &

Reference output:
#
# trace07.txt - Use the jobs builtin command.
#
tsh> ./myspin1 10 &
(1) (16136) ./myspin1 10 &
tsh> ./myspin2 10 &
(2) (16139) ./myspin2 10 &
tsh> jobs
(1) (16136) Running    ./myspin1 10 &
(2) (16139) Running    ./myspin2 10 &
```

Ref와 비교



Listjob의 함수를 이해하는 것이 정말 중요한 문제였다.

listjob함수를 확인 시에 jobs를 받아서 리스트식으로 출력해주는 함수임을 알 수 있었다. 뒤에 매개변수가 하나 더 있는데 이는 유효를 검사한다고 알려주셨고, 정상적으로 출력하려면 1를 투입하면 된다고 배웠다. 위에 맞춰서 완성시켜준 후에, 리턴값을 1로 지정해주고 함수를 종료한다. 1로 지정해주는 이유는 종료가 아닌 어떠한 일이 일어났음을 이를 통해 확인하기 때문이다.