2018 시스템 프로그래밍

- lab08 -

제출일자	2018.11.26
분 반	01
이 름	함지희
학 번	201702087

Trace 08

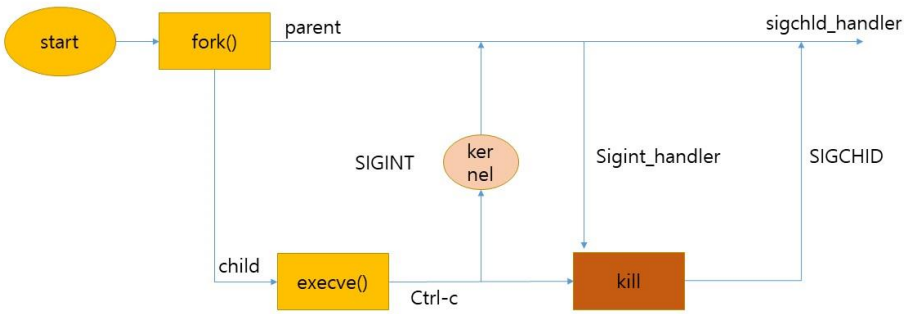
```
b201702087@2018-sp: ~/shlab-handout
b201702087@2018-sp:~/shlab-handout$ ./sdriver -V -t 08 -s ./tsh
Running trace08.txt...
Success: The test and reference outputs for trace08.txt matched!
Test output:
#
# trace08.txt - Send fatal SIGINT to foreground job.
#
tsh> ./myintp
Job [1] (29040) terminated by signal 2
tsh> quit

Reference output:
#
# trace08.txt - Send fatal SIGINT to foreground job.
#
tsh> ./myintp
Job [1] (29048) terminated by signal 2
tsh> quit

b201702087@2018-sp:~/shlab-handout$ ./sdriver -V -t 08 -s ./tshref
Running trace08.txt...
Success: The test and reference outputs for trace08.txt matched!
Test output:
#
# trace08.txt - Send fatal SIGINT to foreground job.
#
tsh> ./myintp
Job [1] (29092) terminated by signal 2
tsh> quit

Reference output:
#
# trace08.txt - Send fatal SIGINT to foreground job.
#
tsh> ./myintp
Job [1] (29100) terminated by signal 2
tsh> quit
```

각 trace 별 프로우 차트



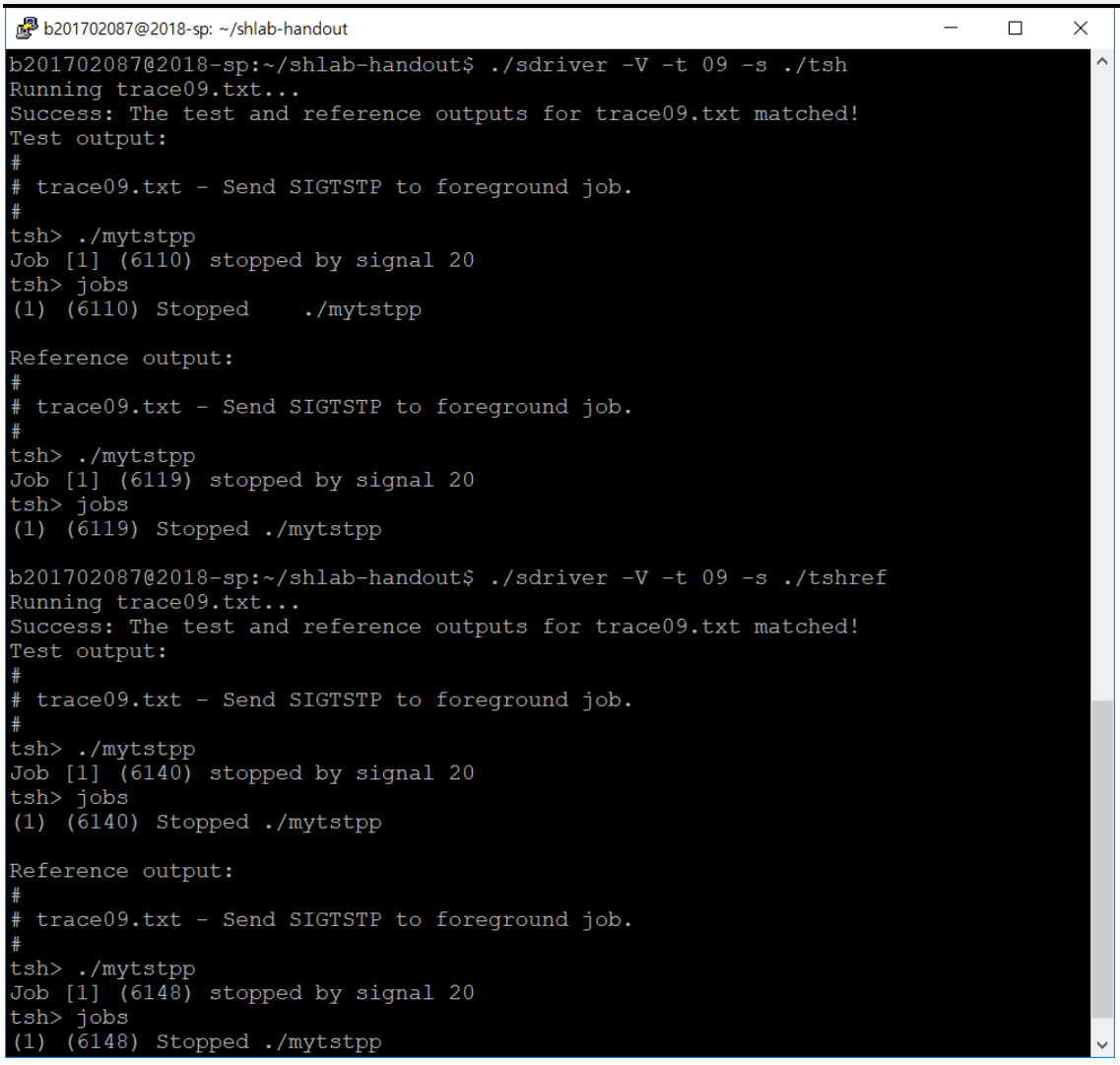
Trace 해결 방법 설명

자식에서 SIGINT가 발생하면 부모로 그 신호를 보내서 sigint_handler를 통해 자식을 kill하게 된다.

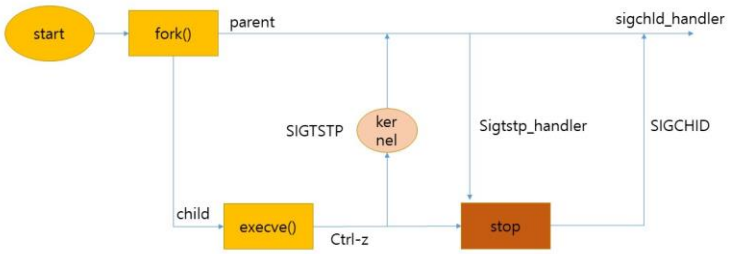
fgpid 함수를 통해 fg의 pid를 알아낸 뒤에, kill을 이용하여 –pid로 해당 그룹의 프로세스들에게 signal을 전달한다. sigchld_handler에서 SIGINT로 인해 종료되었음을 출력하도록한다.

여기서 concurrent한 시그널의 발생으로 Race condition이 생길 수 있기 때문에 eval함수에서 시그널 block, unblock을 통해 시그널의 출입? 을 조절해주어야한다. 먼저 시그널 set을 비운다음에, 시그널들을 시그널 set에 add해준다. 그 다음 시그널 set을 block해준다. fork로 자식을 생성해준 뒤, setpgid(0,0)로 프로세스를 같은 그룹으로 묶고 자식을 실행하기 전에 시그널set을 unblock해준다. 부모 프로세스는 addjob을 잘 수행해야하므로 addjob이후에 unblock을 해준다.

Trace 09



각 trace 별 프로우 차트



Trace 해결 방법 설명

trace 08의 설명과 같다.

sigchld_handler에서 joblist에서의 현재 pid의 상태를 ST로 바꾼 뒤, 상황을 출력한다.

Trace 10

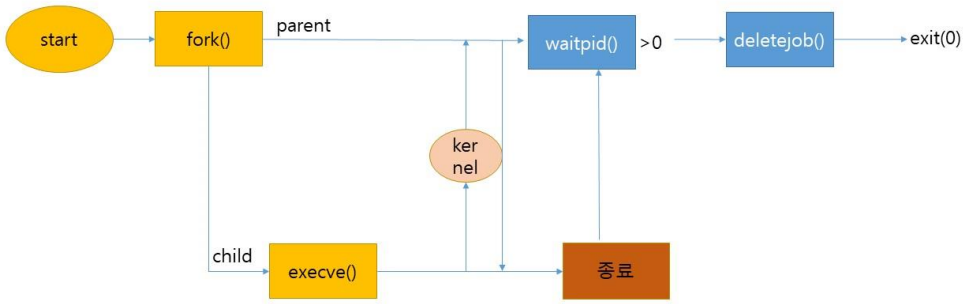
```
b201702087@2018-sp: ~/shlab-handout
b201702087@2018-sp:~/shlab-handout$ ./sdriver -V -t 10 -s ./tsh
Running trace10.txt...
Success: The test and reference outputs for trace10.txt matched!
Test output:
#
# trace10.txt - Send fatal SIGTERM (15) to a background job.
#
tsh> ./myspin1 5 &
(1) (8511) ./myspin1 5 &
tsh> /bin/kill myspin1
kill: failed to parse argument: 'myspin1'
tsh> quit

Reference output:
#
# trace10.txt - Send fatal SIGTERM (15) to a background job.
#
tsh> ./myspin1 5 &
(1) (8522) ./myspin1 5 &
tsh> /bin/kill myspin1
kill: failed to parse argument: 'myspin1'
tsh> quit

b201702087@2018-sp:~/shlab-handout$ ./sdriver -V -t 10 -s ./tshref
Running trace10.txt...
Success: The test and reference outputs for trace10.txt matched!
Test output:
#
# trace10.txt - Send fatal SIGTERM (15) to a background job.
#
tsh> ./myspin1 5 &
(1) (8558) ./myspin1 5 &
tsh> /bin/kill myspin1
kill: failed to parse argument: 'myspin1'
tsh> quit

Reference output:
#
# trace10.txt - Send fatal SIGTERM (15) to a background job.
#
tsh> ./myspin1 5 &
(1) (8569) ./myspin1 5 &
tsh> /bin/kill myspin1
kill: failed to parse argument: 'myspin1'
tsh> quit
```

각 trace 별 프로우 차트



Trace 해결 방법 설명

sigchld_handler에서 waitpid함수를 이용해 자식프로세스가 종료될 때까지 기다린 후 자식이 종료되면 처리해준다.

Trace 11 – 12

원래는 시그널이 들어오면 부모프로세스로 보낸 뒤, 시그널에 맞는 핸들러가 kill()를 통해 자식프로세스에게 시그널을 보내준다. 여기서 바로 자식 프로세스에게 시그널을 보내줘도 달라질 것은 없다. 단지 어디에서 시그널이 보내졌느냐의 차이지 시그널로 인해 발생할 일은 똑같다.