

리눅스시스템 실습 #5

2021/10/13

2021-2

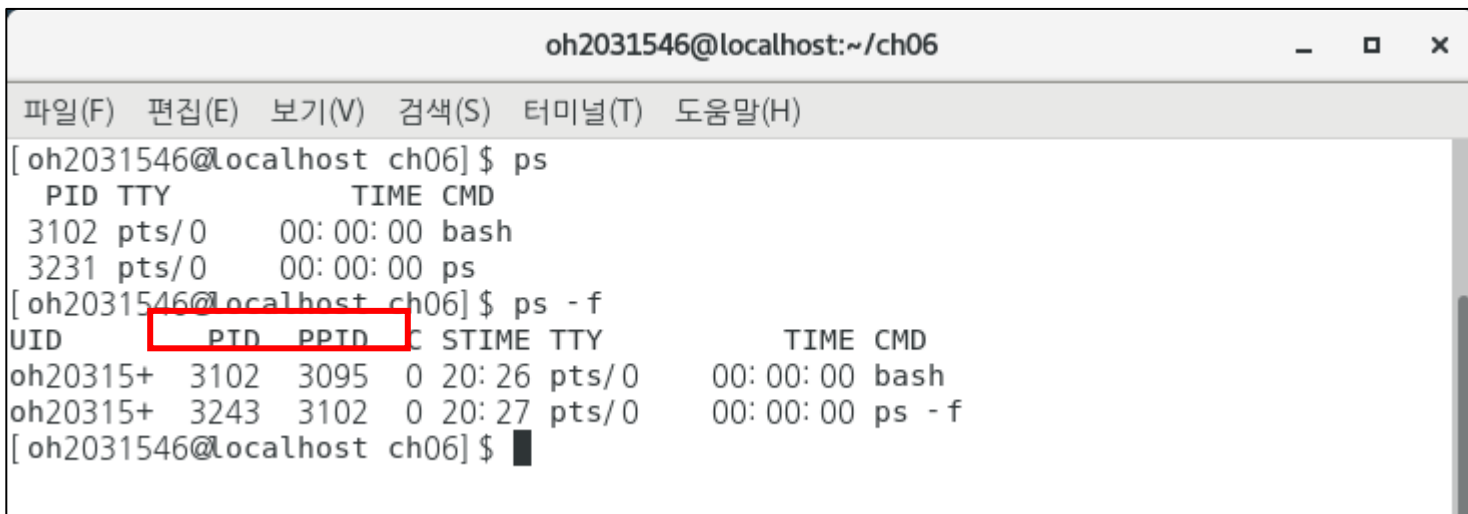
리눅스시스템(창병모 교수님)

TA: 오현주

1. ps 실습

- 다음 명령을 실행하고 출력 결과에 대해 설명하시오.

```
$ ps
$ ps -f
$ ps aux
$ ps -ef
```



```
oh2031546@localhost:~/ch06
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[oh2031546@localhost ch06]$ ps
  PID TTY          TIME CMD
 3102 pts/0    00:00:00 bash
 3231 pts/0    00:00:00 ps
[oh2031546@localhost ch06]$ ps -f
UID      PID  PPID  C  STIME TTY          TIME CMD
oh20315+ 3102  3095  0  20:26 pts/0    00:00:00 bash
oh20315+ 3243  3102  0  20:27 pts/0    00:00:00 ps -f
[oh2031546@localhost ch06]$
```

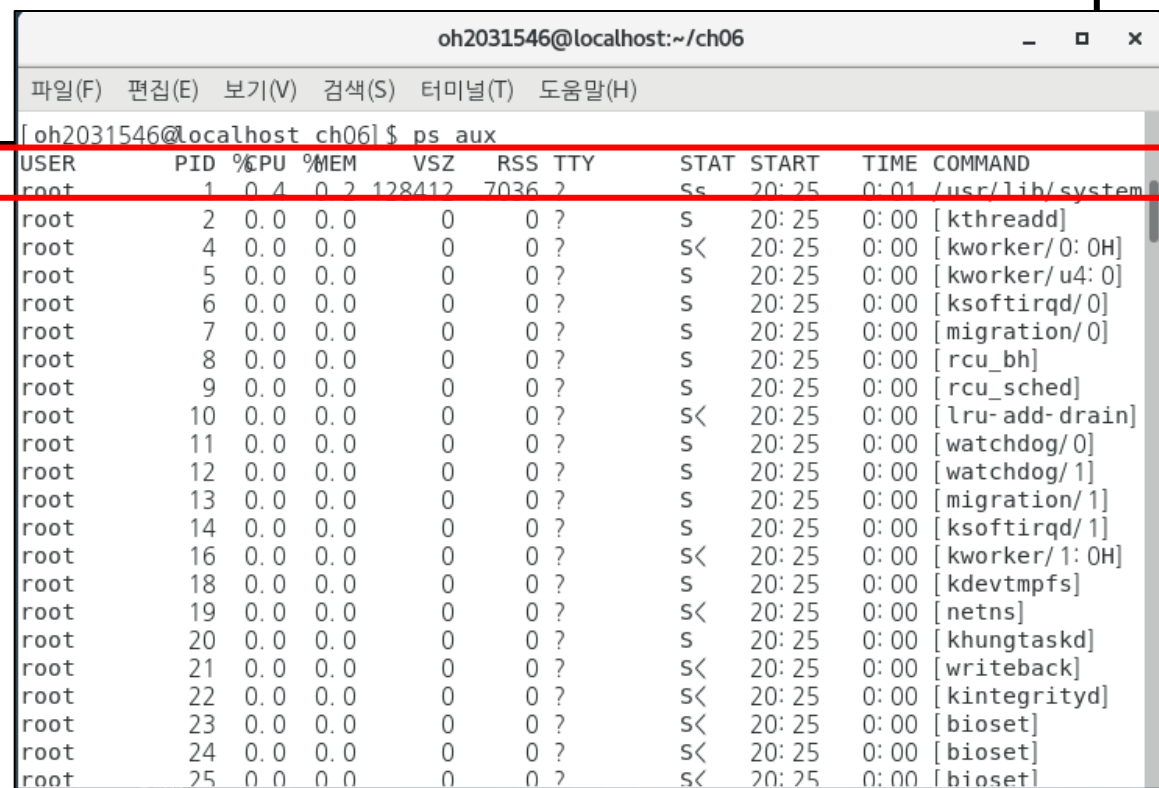
※ ps 명령어가 무엇을 하는 명령어인지, 각 옵션이 어떤 의미를 가지고 있는지 구체적으로 설명을 적어주시기 바랍니다.

1. ps 실습

- 다음 명령을 실행하고 출력 결과에 대해 설명하시오.

```
$ ps
$ ps -f
$ ps aux
$ ps -ef
```

※ ps 명령어가 무엇을 하는 명령어인지, 각 옵션이 어떤 의미를 가지고 있는지 구체적으로 설명을 적어주시기 바랍니다.



oh2031546@localhost:~/ch06

파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

```
oh2031546@localhost ch06$ ps aux
```

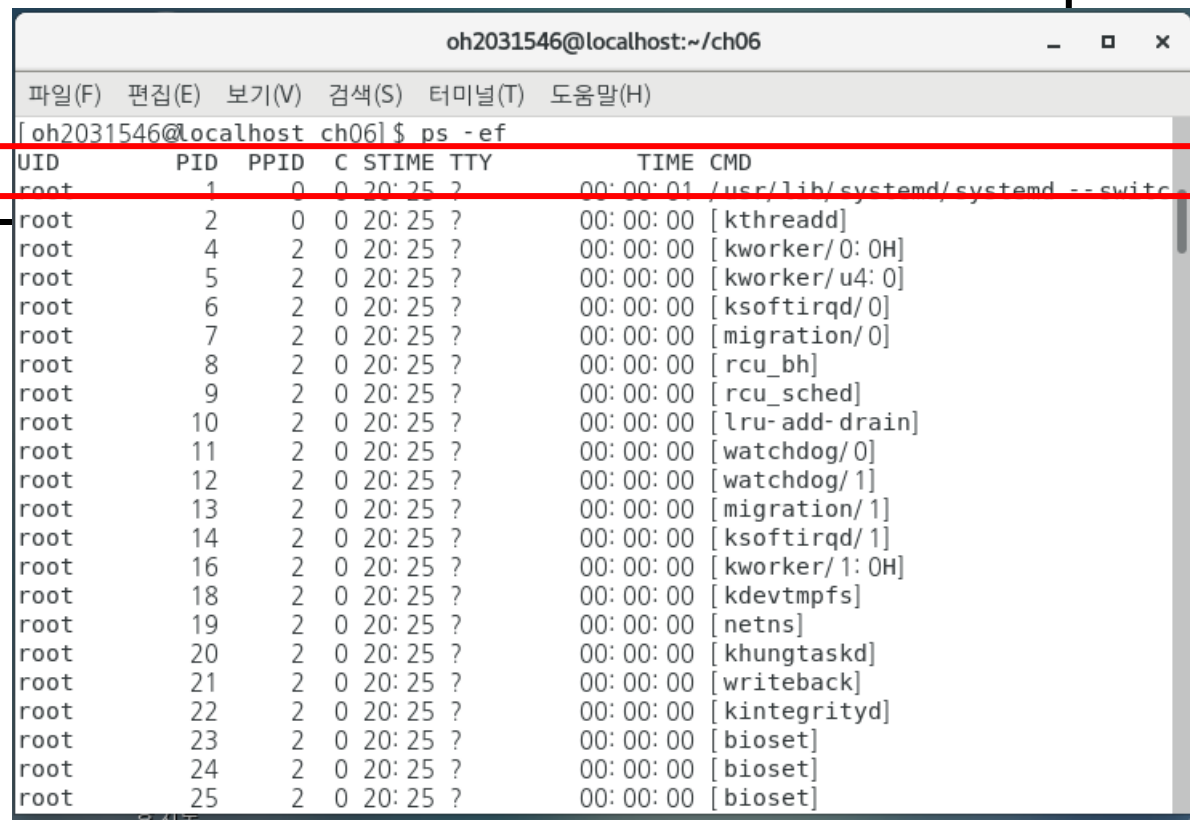
USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.4	0.2	128412	7036	?	Ss	20:25	0:01	/usr/lib/systemd
root	2	0.0	0.0	0	0	?	S	20:25	0:00	[kthreadd]
root	4	0.0	0.0	0	0	?	S<	20:25	0:00	[kworker/0:0H]
root	5	0.0	0.0	0	0	?	S	20:25	0:00	[kworker/u4:0]
root	6	0.0	0.0	0	0	?	S	20:25	0:00	[ksoftirqd/0]
root	7	0.0	0.0	0	0	?	S	20:25	0:00	[migration/0]
root	8	0.0	0.0	0	0	?	S	20:25	0:00	[rcu_bh]
root	9	0.0	0.0	0	0	?	S	20:25	0:00	[rcu_sched]
root	10	0.0	0.0	0	0	?	S<	20:25	0:00	[lru-add-drain]
root	11	0.0	0.0	0	0	?	S	20:25	0:00	[watchdog/0]
root	12	0.0	0.0	0	0	?	S	20:25	0:00	[watchdog/1]
root	13	0.0	0.0	0	0	?	S	20:25	0:00	[migration/1]
root	14	0.0	0.0	0	0	?	S	20:25	0:00	[ksoftirqd/1]
root	16	0.0	0.0	0	0	?	S<	20:25	0:00	[kworker/1:0H]
root	18	0.0	0.0	0	0	?	S	20:25	0:00	[kdevtmpfs]
root	19	0.0	0.0	0	0	?	S<	20:25	0:00	[netns]
root	20	0.0	0.0	0	0	?	S	20:25	0:00	[khungtaskd]
root	21	0.0	0.0	0	0	?	S<	20:25	0:00	[writeback]
root	22	0.0	0.0	0	0	?	S<	20:25	0:00	[kintegrityd]
root	23	0.0	0.0	0	0	?	S<	20:25	0:00	[bioset]
root	24	0.0	0.0	0	0	?	S<	20:25	0:00	[bioset]
root	25	0.0	0.0	0	0	?	S<	20:25	0:00	[bioset]

1. ps 실습

- 다음 명령을 실행하고 출력 결과에 대해 설명하시오.

```
$ ps
$ ps -f
$ ps aux
$ ps -ef
```

※ ps 명령어가 무엇을 하는 명령어인지, 각 옵션이 어떤 의미를 가지고 있는지 구체적으로 설명을 적어주시기 바랍니다.



oh2031546@localhost:~/ch06

파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

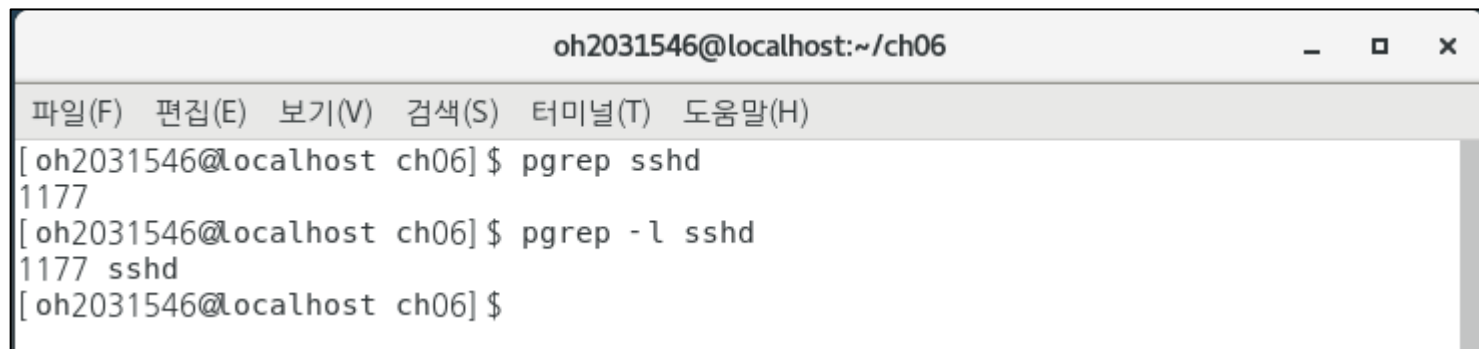
[oh2031546@localhost ch06]\$ ps -ef

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	20:25	?	00:00:01	/usr/lib/systemd/systemd -- switch
root	2	0	0	20:25	?	00:00:00	[kthreadd]
root	4	2	0	20:25	?	00:00:00	[kworker/0:0H]
root	5	2	0	20:25	?	00:00:00	[kworker/u4:0]
root	6	2	0	20:25	?	00:00:00	[ksoftirqd/0]
root	7	2	0	20:25	?	00:00:00	[migration/0]
root	8	2	0	20:25	?	00:00:00	[rcu_bh]
root	9	2	0	20:25	?	00:00:00	[rcu_sched]
root	10	2	0	20:25	?	00:00:00	[lru-add-drain]
root	11	2	0	20:25	?	00:00:00	[watchdog/0]
root	12	2	0	20:25	?	00:00:00	[watchdog/1]
root	13	2	0	20:25	?	00:00:00	[migration/1]
root	14	2	0	20:25	?	00:00:00	[ksoftirqd/1]
root	16	2	0	20:25	?	00:00:00	[kworker/1:0H]
root	18	2	0	20:25	?	00:00:00	[kdevtmpfs]
root	19	2	0	20:25	?	00:00:00	[netns]
root	20	2	0	20:25	?	00:00:00	[khungtaskd]
root	21	2	0	20:25	?	00:00:00	[writeback]
root	22	2	0	20:25	?	00:00:00	[kintegrityd]
root	23	2	0	20:25	?	00:00:00	[bioaset]
root	24	2	0	20:25	?	00:00:00	[bioaset]
root	25	2	0	20:25	?	00:00:00	[bioaset]

2. pgrep 실습

- 다음 명령을 실행하고 출력 결과에 대해 설명하시오.

```
$ pgrep sshd  
$ pgrep -l sshd
```

A terminal window titled 'oh2031546@localhost:~/ch06' with standard window controls. The terminal shows the execution of 'pgrep sshd' which outputs '1177', followed by 'pgrep -l sshd' which outputs '1177 sshd'.

```
oh2031546@localhost:~/ch06  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[oh2031546@localhost ch06]$ pgrep sshd  
1177  
[oh2031546@localhost ch06]$ pgrep -l sshd  
1177 sshd  
[oh2031546@localhost ch06]$
```

※ pgrep 명령어가 무엇을 하는 명령어인지, 각 옵션이 어떤 의미를 가지고 있는지 구체적으로 설명을 적어주시기 바랍니다.

3. 전면처리, 후면처리 실습

※ 위의 6개의 명령들(+ 옵션들)이 **각각** 어떤 의미를 가지고 있는지 혹은 실행 후 어떤 변화가 생겼는지 결과에 대해 구체적으로 설명해주시기 바랍니다.

- 다음 명령을 실행하고 출력 결과에 대해 설명하시오.
- 출력된 프로세스들의 **부모-자식 관계**를 설명하시오.

```
$ sleep 100 &  
$ sleep 100 &  
$ jobs  
$ ps -f  
$ kill %1  
$ ps -f
```

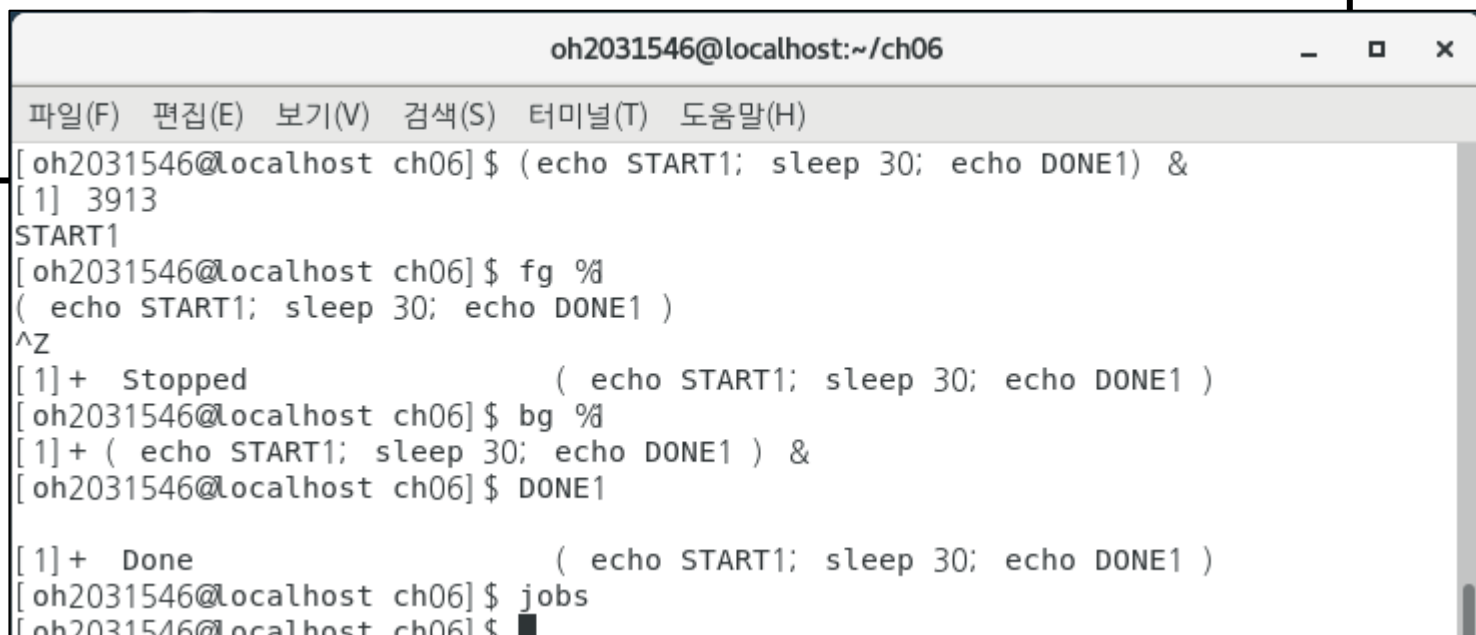
```
oh2031546@localhost:~/ch06  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[oh2031546@localhost ch06]$ sleep 100 &  
[1] 3690  
[oh2031546@localhost ch06]$ sleep 100 &  
[2] 3697  
[oh2031546@localhost ch06]$ jobs  
[1] - Running sleep 100 &  
[2] + Running sleep 100 &  
[oh2031546@localhost ch06]$ ps -f  
UID      PID  PPID  C  STIME TTY          TIME CMD  
oh20315+ 3102 3095  0  20:26 pts/0      00:00:00 bash  
oh20315+ 3690 3102  0  20:58 pts/0      00:00:00 sleep 100  
oh20315+ 3697 3102  0  20:58 pts/0      00:00:00 sleep 100  
oh20315+ 3710 3102  0  20:58 pts/0      00:00:00 ps -f  
[oh2031546@localhost ch06]$ kill %1  
[1] - 종료됨 sleep 100  
[oh2031546@localhost ch06]$ ps -f  
UID      PID  PPID  C  STIME TTY          TIME CMD  
oh20315+ 3102 3095  0  20:26 pts/0      00:00:00 bash  
oh20315+ 3697 3102  0  20:58 pts/0      00:00:00 sleep 100  
oh20315+ 3723 3102  0  20:59 pts/0      00:00:00 ps -f  
[oh2031546@localhost ch06]$
```

4. 작업제어 실습

- 다음 명령을 실행하고 출력 결과에 대해 설명하시오.

```
$ (echo START1; sleep 30; echo DONE1) &  
$ fg %1  
^Z  
$ bg %1  
$ jobs
```

※ 위의 4개의 명령들(+ 옵션들)이 **각각** 어떤 의미를 가지고 있는지 혹은 실행 후 어떤 변화가 생겼는지 결과에 대해 구체적으로 설명해주시기 바랍니다.



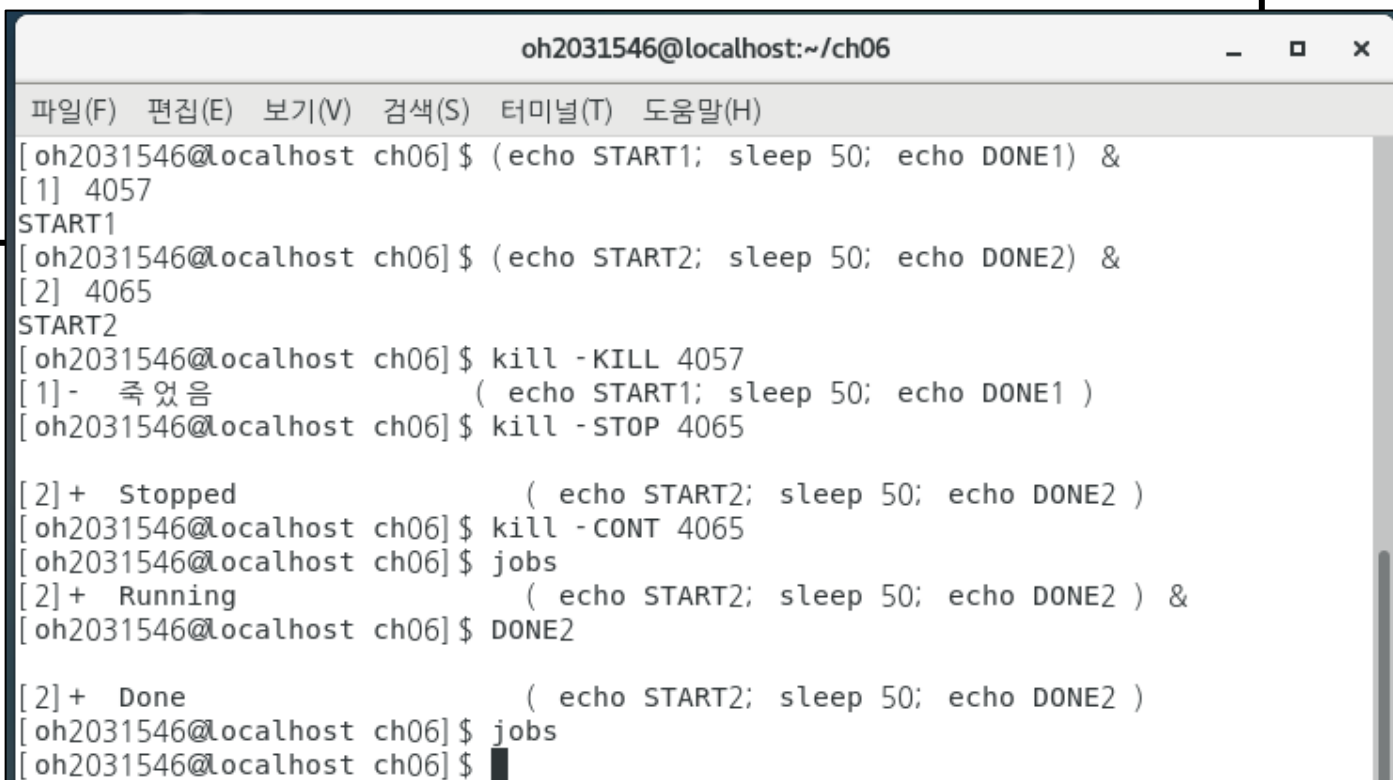
```
oh2031546@localhost:~/ch06  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[oh2031546@localhost ch06]$ (echo START1; sleep 30; echo DONE1) &  
[1] 3913  
START1  
[oh2031546@localhost ch06]$ fg %1  
( echo START1; sleep 30; echo DONE1 )  
^Z  
[1]+  Stopped                  ( echo START1; sleep 30; echo DONE1 )  
[oh2031546@localhost ch06]$ bg %1  
[1]+ ( echo START1; sleep 30; echo DONE1 ) &  
[oh2031546@localhost ch06]$ DONE1  
  
[1]+  Done                      ( echo START1; sleep 30; echo DONE1 )  
[oh2031546@localhost ch06]$ jobs  
[oh2031546@localhost ch06]$
```

5. 작업제어 실습

- 다음 명령을 실행하고 출력 결과에 대해 설명하시오.

```
$ (echo START1; sleep 50; echo DONE1) &  
$ (echo START2; sleep 50; echo DONE2) &  
$ kill -KILL 첫 번째 프로세스  
$ kill -STOP 두 번째 프로세스  
$ kill -CONT 두 번째 프로세스  
$ jobs
```

※ kill 각 옵션들의 의미가 무엇인지에
대한 설명 포함되어 있어야 함



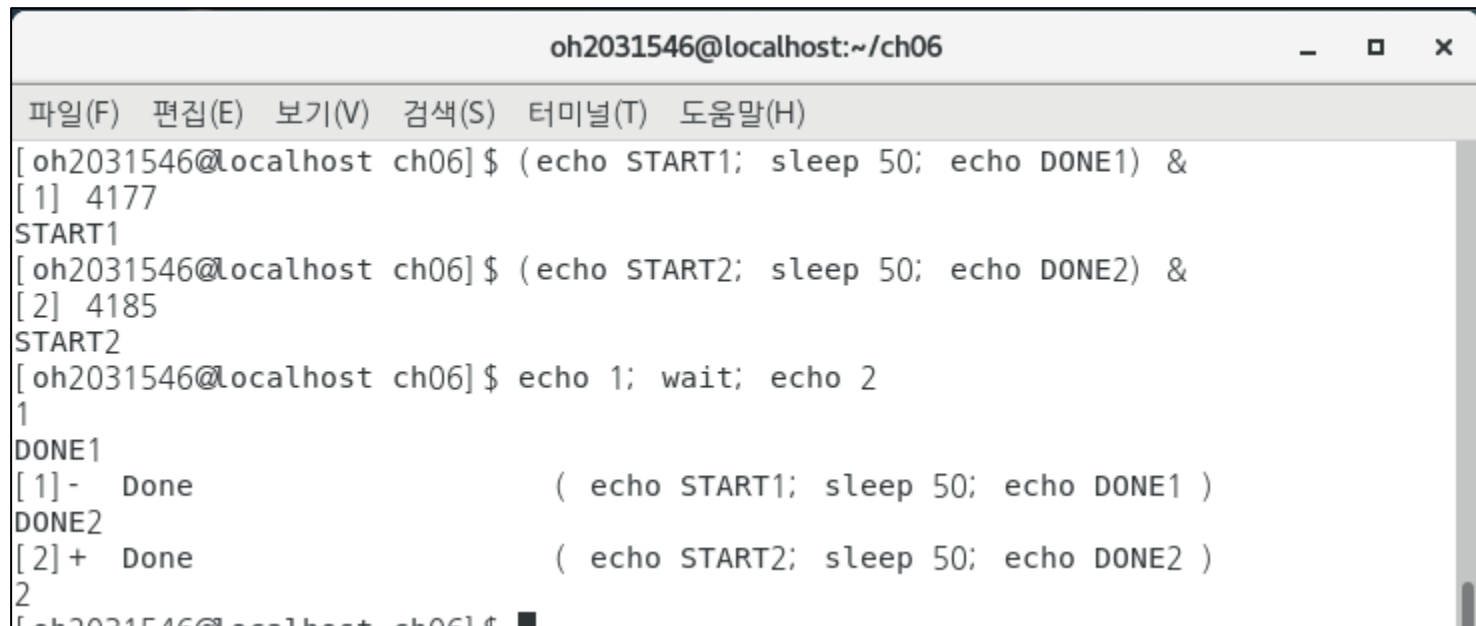
```
oh2031546@localhost:~/ch06  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[oh2031546@localhost ch06]$ (echo START1; sleep 50; echo DONE1) &  
[1] 4057  
START1  
[oh2031546@localhost ch06]$ (echo START2; sleep 50; echo DONE2) &  
[2] 4065  
START2  
[oh2031546@localhost ch06]$ kill -KILL 4057  
[1]- 죽었음 ( echo START1; sleep 50; echo DONE1 )  
[oh2031546@localhost ch06]$ kill -STOP 4065  
[2]+ Stopped ( echo START2; sleep 50; echo DONE2 )  
[oh2031546@localhost ch06]$ kill -CONT 4065  
[oh2031546@localhost ch06]$ jobs  
[2]+ Running ( echo START2; sleep 50; echo DONE2 ) &  
[oh2031546@localhost ch06]$ DONE2  
[2]+ Done ( echo START2; sleep 50; echo DONE2 )  
[oh2031546@localhost ch06]$ jobs  
[oh2031546@localhost ch06]$
```


6. 프로세스 기다리기 실습

- 다음 명령을 실행하고 출력 결과에 대해 설명하시오.

```
$ (echo START1; sleep 50; echo DONE1) &  
$ (echo START2; sleep 50; echo DONE2) &  
$ echo 1; wait; echo 2
```

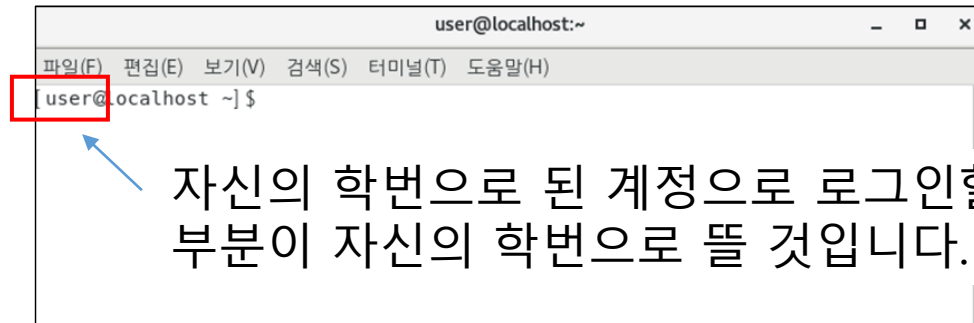
※ 마지막 명령의 출력 결과가 왜 다음과 같이 나왔는지에 대한 설명 포함되어 있어야 함



```
oh2031546@localhost:~/ch06  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[oh2031546@localhost ch06]$ (echo START1; sleep 50; echo DONE1) &  
[1] 4177  
START1  
[oh2031546@localhost ch06]$ (echo START2; sleep 50; echo DONE2) &  
[2] 4185  
START2  
[oh2031546@localhost ch06]$ echo 1; wait; echo 2  
1  
DONE1  
[1] - Done ( echo START1; sleep 50; echo DONE1 )  
DONE2  
[2] + Done ( echo START2; sleep 50; echo DONE2 )  
2
```

Lab05 과제 제출 - 보고서

- 각 문제별 **실행 화면 캡처 및 간략한 설명(2줄 이상)**을 보고서로 작성
 - 보고서에 실습번호, 분반, 과, 학번, 이름을 적을 것
 - 파일명: 실습번호_학번_이름(ex. Lab05_2031546_오현주)
 - 꼭 자신의 학번으로 된 계정으로 로그인하여 실습을 진행한 후 캡처를 하시기 바랍니다.

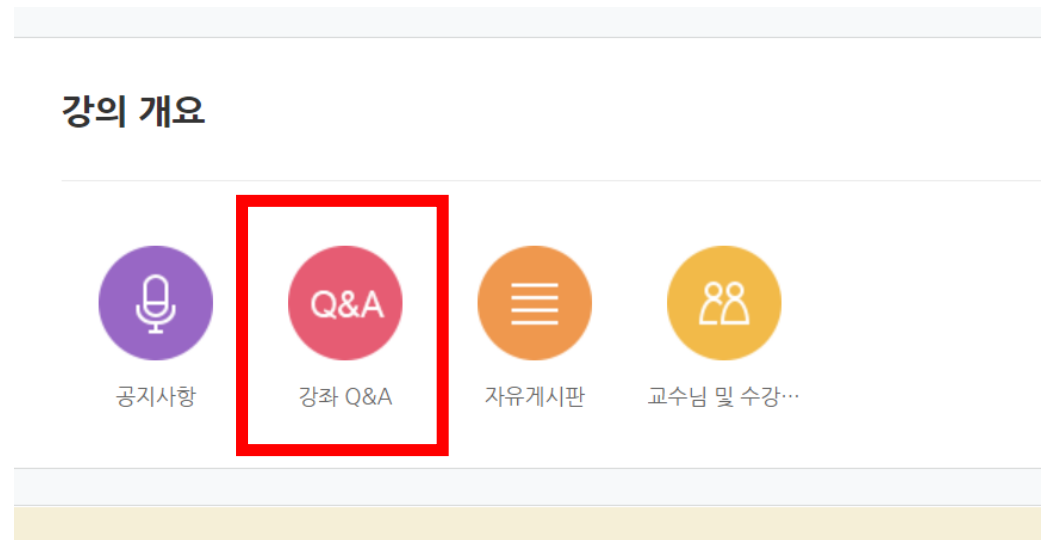


자신의 학번으로 된 계정으로 로그인할 경우 터미널의 이 부분이 자신의 학번으로 뜰 것입니다.

- 스노우보드 **Lab05**에 **PDF 파일**로 제출할 것
 - 워드 혹은 한글 문서로 보고서를 작성하신 뒤 pdf로 변환하시기 바랍니다.
- 위의 사항들을 지키지 않을 경우 **감점 요인이 되므로** 이 점 주의하시기 바랍니다.
- 기한: **10월 17일 일요일, 오후 11시 55분까지**
 - 늦게 제출할 경우(-2), 제출기한 이후 24시간까지만 받음.

실습 관련 문의

- 실습 관련 질문이 있는 경우 스노우보드 강좌 Q&A에 등록해주시기 바랍니다.



- 질문 작성 시, 내용을 구체적으로 명확하게 적어주시기 바랍니다.
- 코드를 첨부하고 싶을 경우, 캡처가 아닌 **텍스트 형식**으로 보내주세요.