

### Step1-1

```
mjswindells@ubuntu:~$ pstree -p
systemd(1)─ModemManager(851)─{ModemManager}(867)
             │               │{ModemManager}(871)
             └─NetworkManager(737)─{NetworkManager}(823)
                                   │{NetworkManager}(835)
             └─VGAAuthService(705)
             └─accounts-daemon(727)─{accounts-daemon}(738)
                                   │{accounts-daemon}(829)
             └─acpid(728)
             └─agetty(2557)
             └─avahi-daemon(731)─avahi-daemon(790)
             └─bluetoothd(732)
             └─colord(1353)─{colord}(1355)
                           │{colord}(1357)
             └─cron(733)
             └─cups-browsed(836)─{cups-browsed}(864)
                                   │{cups-browsed}(865)
             └─cupsd(821)
             └─dbus-daemon(735)
```

pstree [옵션] : 현재 진행중인 프로세스를 트리 구조로 출력한다

옵션

- 1) -p 프로세스들의 오른쪽에 PID 정보를 함께 출력
- 2) -h 현재 프로세스와 그의 부모프로세스들을 강조하여 출력

### Step1-2

```
─gnome-terminal-(2920)─bash(2953)─pstree(3063)
                        │{gnome-terminal-}(2927)
                        │{gnome-terminal-}(2928)
                        │{gnome-terminal-}(2941)
                        │{gnome-terminal-}(2952)
                        └─goa-daemon(1456)─{goa-daemon}(1464)
```

pstree의 식별번호 : 3063

### Step1-3

```
mjswindells@ubuntu:~$ ps -ef | grep pstree
mjswind+    3076    2953  0 01:25 pts/0    00:00:00 grep --color=auto pstree
mjswindells@ubuntu:~$
```

ps [옵션] : 현재 진행중인 프로세스를 출력한다.

옵션1) -e 제어 터미널을 갖고 있는 모든 프로세스를 보여준다

2) -f 프로세스 간에 부모와 자식 관계를 보여준다

3) -u 프로세스의 소유자를 보여준다

4) -l 자세한 형식으로 보여준다

### Step2

프로세스 : 실행중인 프로그램을 의미

PID는 프로세스 아이디로 프로그램이 실행된 시점에 유니크한 PID를 가지게 된다.

Step1-2에서 실행한 프로세스는 pstree에 대한 PID이고

Step1-3에서 실행한 프로세스는 grep에 대한 PID이다. 따라서 둘의 PID가 다를 수 밖에 없다.

이때 PID의 차가 2인 이유는 프로세스생성마다 하나씩 증가하게 되는데

처음 pstree를 실행했을 때 200이라고 하면 그 다음 ps -ef | grep pstree를 하는 과정에서

ps의 PID는 201 grep의 PID는 202 이렇게 설정된다.

Step2-1 Step2-2

```
mjswindells@ubuntu:~$ sleep 6000 &
[1] 3589
mjswindells@ubuntu:~$ sleep 6000
^Z
[2]+  Stopped                  sleep 6000
mjswindells@ubuntu:~$ jobs
[1]-  Running                  sleep 6000 &
[2]+  Stopped                  sleep 6000
mjswindells@ubuntu:~$
```

sleep [시간(초)] : 가상터미널에서 실행을 중지시킨다.

리눅스는 multitasking이 가능하다. 이때 foreground와 background가 있다.

Foreground : 화면에 보여주면서 실행되는 상태

Background : 화면에 보여주지 않으면서 실행되는 상태 (&를 뒤에 붙여 작동시킬 수 있다)

jobs : background로 실행중인 작업을 확인한다.

Step3

```
mjswindells@ubuntu:~$ ps -l
F S  UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000      2953     2920  0  80   0  -  2816 do_wai pts/0        00:00:00 bash
0 S  1000      3589     2953  0  80   0  -  2021 hrtime pts/0        00:00:00 sleep
0 T  1000      3591     2953  0  80   0  -  2021 do_sig pts/0        00:00:00 sleep
0 R  1000      4055     2953  0  80   0  -  2854 -          pts/0        00:00:00 ps
mjswindells@ubuntu:~$ kill -9 3591
mjswindells@ubuntu:~$ jobs
[1]-  Running                  sleep 6000 &
[2]+  Killed                  sleep 6000
mjswindells@ubuntu:~$
```

빨간 테두리 친 부분의 맨 위의 S는 STAT의 약어이다.

S : sleep상태                      T : 정지된 상태                      R : 실행 상태

( T 상태인 sleep의 PID : 3591)

프로세스 간의 통신수단으로 signal이 있다.

kill [옵션] PID : 프로세스에 시그널 보내기

옵션) -9 : 강제종료

#### Step4

```
mjswindells@ubuntu:~$ jobs
[1]+  Running                  sleep 6000 &
mjswindells@ubuntu:~$ fg 1
sleep 6000
█
```

fg [num] : background로 실행중인 프로세스를 foreground로 돌린다.

<Ctrl+Z> : foreground로 실행중인 작업을 정지상태로 만든다 (재개 가능)

<Ctrl+C> : foreground로 실행중인 작업을 종료한다.