데몬과 프로세스 관리 1

ps 명령어

프로세스를 확인하는 명령어

```
# ps
  PID TTY          TIME CMD
 4151 pts/1    00:00:00 bash
16339 pts/1    00:00:00 ps
```

- 현재 ps 를 실행시킨 사용자는 bash 쉘 사용중에 ps 를 실행시켰다는 것을 알 수 있다.

```
# ps -ef
```

- -e 옵션 : 모든 프로세스를 표시
- -f 옵션 : 전체경로로 프로세스 표시

특정 프로세스 조회

[사용형식] ps –ef | grep <프로세스명>

```
# ps -ef | grep crond
root        9994       1   0 15:26 ?        00:00:04 /usr/sbin/crond -n
root       19365    5173   0 22:48 pts/1    00:00:00 grep --color=auto crond

# ps -ef | grep atd
root       18400       1   0 21:51 ?        00:00:00 /usr/sbin/atd -f
root       19441    5173   0 22:50 pts/1    00:00:00 grep --color=auto atd
```

프로세스들의 가계도를 확인하는 pstree

pstree 는 현재 실행중인 프로세스들을 트리구조로 보여준다. 프로세스의 상호관계를 파악하기 위해 필수적인 명령어이다.

```
# pstree -p
systemd(1)─┬─ModemManager(576)─┬─{ModemManager}(584)
           |                   └─{ModemManager}(598)
           ├─NetworkManager(6548)─┬─{NetworkManager}(6549)
           |                      └─{NetworkManager}(6551)
           ├─abrt-watch-log(607)
           ├─abrt-watch-log(623)
           ├─abrtd(602)
           ├─accounts-daemon(571)─┬─{accounts-daemon}(582)
           |                      └─{accounts-daemon}(597)
           ├─alsactl(575)
           ├─at-spi-bus-laun(2834)─┬─dbus-daemon(2839)───{dbus-daemon}(2840)
           |                       ├─{at-spi-bus-laun}(2835)
           |                       ├─{at-spi-bus-laun}(2836)
           |                       └─{at-spi-bus-laun}(2838)
.....
```

- pstree 의 –p 옵션은 프로세스들의 오른쪽 괄호에 PID 정보를 함께 출력해준다.

```
# pstree -a
systemd --switched-root --system --deserialize 21
  ├─ModemManager
  |    └─2*[{ModemManager}]
  ├─NetworkManager --no-daemon
  |    └─2*[{NetworkManager}]
  ├─abrt-watch-log -F Backtrace /var/log/Xorg.0.log --/usr/bin/abrt-dump-x
  ├─abrt-watch-log -F BUG: WARNING: at WARNING: CPU:INFO: possible recursi
  ├─abrtd -d -s
.....
```

- pstree 에서 –a 옵션을 사용하면 프로세스들이 실행 될 때의 인자나 옵션들을 모두 함께 표시해준다. 즉, 프로세스들이 실행이 될 때 사용하였던 옵션들까지 상세하게 보여준다.

백그라운드모드와 포그라운드모드와 실행모드 전환

- 모든 프로세스들은 백그라운드(back ground) 또는 포그라운드(fore ground)라는 두 가지 중 하나의 모드로 작동한다.
- 우리가 흔히 일반적인 방법으로 실행시키는 거의 모든 프로세스들은 foreground 작업이다.
- bg 명령어와 fg 라는 명령어, 그리고 jobs 라는 명령어는 모두 이러한 프로세스들의 foreground 작업과 background 작업에 관한 설정과 확인하는 명령어들이다.
- bg 명령어 :
  - 실행시킨 프로세스를 백그라운드작업으로 전환시키는 쉘내부명령어이다.
  - 즉, 실행시킨 명령어의 실행이 끝나지 않은 상태에서 다른 작업을 해야 할
  - 필요성이 있을 때, 이미 실행시킨 작업을 백그라운드작업으로 전환시킨 후에 다른
  - 작업을 할 수 있도록 하기 위해서 사용한다.
  - bg 명령어를 사용하지 않고 명령어 실행 시에 명령행의 끝에 "&"를 붙여 실행하면
  - 처음부터 백그라운드로 실행된다.
- fg 명령어 :
  백그라운드로 실행중인 작업을 다시 전면부로 실행시키는 명령어이다.
- jobs 명령어 :
  실행시킨 프로세스들이 어떤 모드로 실행 중인가를 확인하며 그 리스트를 출력한다.

백그라운드모드로 명령실행하기

```
# tar cfz etc.tar.gz /etc >& /dev/null &
[1] 18066
# jobs
[1]+  Running                 tar cfz etc.tar.gz /etc >&/dev/null &
```

실행중인 프로세스를 백그라운드모드로 전환하기

1. tar 로 /usr 디렉토리 전체를 압축하는 명령어 실행 후 실행시간이 너무 많이 소요되므로 "ctrl + z" 로 정지 시킨다.

```
# tar cvfz usr.tar.gz /usr >& /dev/null
ctrl + z
[1]+  Stopped
```

   ➢ "ctrl + z" 를 입력하면 현재 실행중인 작업을 일단 멈추고 쉘 프롬프트에서 대기한다.

2. 현재 실행상황을 확인하기 위하여 jobs 명령 실행

```
# jobs
[1]+  Stopped                 tar cvfz usr.tar.gz /usr >&/dev/null
```

➢ 실행멈춤(stopped)이 된 프로세스가 작업번호 1 에 있음을 확인할 수 있다.

3. 작업을 백그라운드로 전환하기위하여 "bg %1"을 입력한다.

```
# bg %1
[1]+ tar cvfz usr.tar.gz /usr >&/dev/null &
# jobs
[1]+  Running                 tar cvfz usr.tar.gz /usr >&/dev/null &
```

➢ 백그라운드로 전환 후 jobs 명령어로 확인해보면 백그라운드로 실행중이라는 표시인 "&" 가 되어있는 것을 확인할 수 있다.

4. 다시 foreground process 로 전환

```
# fg %1
tar cvfz usr.tar.gz /usr >&/dev/null
```