9장

Linux 프로세스 관리

전체 내용

프로세스의 개념

프로세스 관리 명령어

Foreground 및 Background Job 제어

Job 예약하기

1 – Process 개념

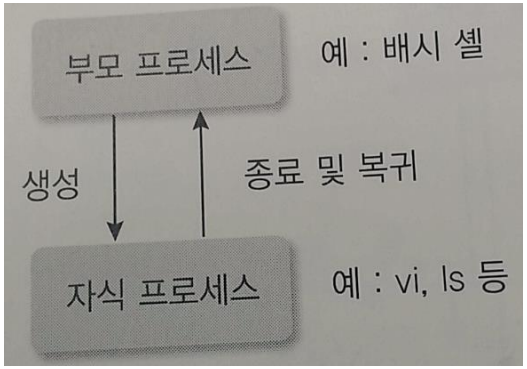
- Process 개념 개요
- 프로세스의 부모 자식 관계
- 프로세스 번호
- 프로세스 종류

1 – 프로세스 개념

- Process 개념 이해
  - 디스크에 저장된 파일을 Program이라고 한다
  - 현재 Memory상에서 실행중인 프로그램을 Process라고 한다
  - Process는 사용자가 직접 실행한 것(사용자 프로세스)과 시스템이 자동으로 실행하는 것(시스템 프로세스)으로 구분된다
    - 시스템 프로세스: 사용자 관리, 메모리 관리, 네트워크 접속 관리 등등
  - Process는 짧게 실행된 후 종료되는 경우가 있어서 생성되었다가 곧장 사라지기도 한다
  - Process를 강제 종료할 수 있으며, Background에서도 실행할 수 있다
  - 예약을 설정하여 Process를 정해진 시간에 자동으로 실행할 수 있고, 또는 주기적으로 일정한 시간마다 실행할 수도 있다

1 – 프로세스 개념

- 프로세스의 부모 자식 관계
  - 리눅스의 모드 프로세스는 부모-자식 관계를 유지하고 있다
  - 자식 프로세스는 부모 프로세스에 의해 생성된 것이다
  - 자식 프로세스 아래에 또 자식이 있을 수 있다
  - Systemd(=/sbin/init), kthreadd 프로세스를 제외한 모든 프로세스는 부모 프로세스를 가지고 있다




```
[adminuser@centos ~]$ pstree
init─┬─NetworkManager─┬─dhclient
     │                └─{NetworkManager}
     ├─abrtd
     ├─acpid
     ├─atd
     ├─auditd───{auditd}
     ├─automount───4*[{automount}]
     ├─bonobo-activati───{bonobo-activat}
```

1 – 프로세스 개념

- 프로세스 번호
  - 프로세스 번호가 할당된다. 이것을 PID라고 한다
  - PID는 1번부터 증가해 가는데, 시스템을 부팅하면 1번 /sbin/init, 2번 kthreadd으로 차례대로 실행된다
  - 1번 프로세스는 모든 시스템 프로세스의 부모 프로세스이다
  - 2번 프로세스는 모든 쓰레드의 부모 프로세스이다

```
[adminuser@centos ~]$ ps aux
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          1  0.0  0.0  19364  1536 ?        Ss   19:19   0:00 /sbin/init
root          2  0.0  0.0      0     0 ?        S    19:19   0:00 [kthreadd]
root          3  0.0  0.0      0     0 ?        S    19:19   0:00 [migration/0]
```

1 – 프로세스 개념

- 프로세스 종류
  - Daemon Process
    - 특정한 서비스를 제공하며 리눅스 커널에 의해 실행된다
    - 평소에 대기 상태로 있다가 서비스 요청이 들어 오면 해당 서비스를 제공
    - 원격 접속 서비스를 제공하는 sshd(ssh server daemon) 등이 있다
  - Orphan Process
    - 자식 프로세스가 종료되면 부모 프로세스로 돌아간다. 그런데 자식이 아직 실행중인데 부모가 먼저 종료되면 자식 프로세스는 고아 프로세스가 된다
    - 이 경우 1번 프로세스가 고아 프로세스의 새로운 부모 프로세스가 되어, 고아 프로세스가 작업을 마치고 종료될 수 있도록 도와준다
  - Zombie Process
    - 자식 프로세스가 종료될 때 부모에게 종료 정보(exit status)를 보내고 부모가 이 정보를 받으면 자식은 프로세스 테이블 목록에서 삭제된다
    - 그런데 자식이 실행을 종료했는데, 프로세스 테이블 목록에 남아 있는 것을 좀비 프로세스라고 한다. 이것은 부모가 자식이 보낸 종료 정보를 제대로 처하지 않았기 때문이다

1 – 프로세스 개념

- 프로세스 종류
  - Zombie Process
    - 프로세스 목록에 defunct 프로세스라고 나오는 것이 좀비 프로세스이다
      - Defunct: 지금은 존재하지 않는 /defunct aunt: 작고한 숙모/ defunct company: 말소 회사
    - 좀비는 실제로 실행되지는 않지만 동작중인 프로세스 테이블 목록을 차지하고 있어서, 잘못 작성된 프로그램 때문에 좀비 프로세스가 증가하면 프로세스 테이블의 용량이 부족해서 정상적인 프로세스가 실행되지 않을 수 있다
    - 좀비 프로세스 확인하기
      - **top**
      - **ps -ef | grep defunct | grep -v grep**
    - 좀비 프로세스 죽이기
      - **ps -ef | grep defunct | awk '{print $3}' | xargs kill -9**

2 – 프로세스 관리 명령어

- 프로세스 목록 보기
- 특정 프로세스 정보 검색하기
- 프로세스 종료하기
- 프로세스 관리도구
- Linux 기본 디렉터리 이해하기

2 – 프로세스 관리 명령어

- 프로세스 목록 보기
  - 개요
    - ps라는 명령어를 사용하여 프로세스 목록을 보며 프로세스의 부모-자식 관계도 확인할 수 있다

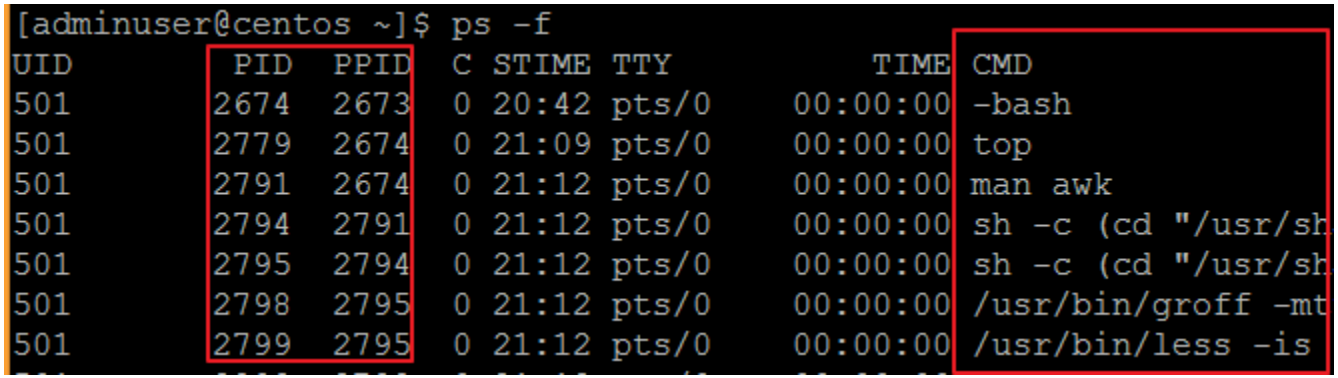
기능	현재 실행 중인 프로세스에 대한 정보를 출력한다.	
형식	ps 옵션	
옵션	〈유닉스 옵션〉	-e : 시스템에서 실행 중인 모든 프로세스의 정보를 출력한다.
		-f : 프로세스에 대한 자세한 정보를 출력한다.
		-u uid : 특정 사용자에 대한 모든 프로세스의 정보를 출력한다.
		-p pid : pid로 지정한 특정 프로세스의 정보를 출력한다.
	〈BSD 옵션〉	a : 터미널에서 실행한 프로세스의 정보를 출력한다.
		u : 프로세스 소유자의 이름, CPU 사용량, 메모리 사용량 등 상세 정보를 출력한다.
		x : 시스템에서 실행 중인 모든 프로세스의 정보를 출력한다.
	〈GNU 옵션〉	--pid PID 목록 : 목록으로 지정한 특정 PID 정보를 출력한다.
사용 예	ps	ps -ef ps aux

  - 쉘이나 터미널에서 명령어를 실행한 **사용자 프로세스** 정보 확인: **ps**

```
[adminuser@centos ~]$ ps
  PID TTY          TIME CMD
 2674 pts/0    00:00:00 bash
 2779 pts/0    00:00:00 top
 2791 pts/0    00:00:00 man
 2794 pts/0    00:00:00 sh
```

2 – 프로세스 관리 명령어

- 프로세스 목록 보기
  - 프로세스의 상세 정보 출력하기: **-f 옵션**



```
[adminuser@centos ~]$ ps -f
UID          PID  PPID  C STIME TTY          TIME CMD
501         2674  2673  0 20:42 pts/0    00:00:00 -bash
501         2779  2674  0 21:09 pts/0    00:00:00 top
501         2791  2674  0 21:12 pts/0    00:00:00 man awk
501         2794  2791  0 21:12 pts/0    00:00:00 sh -c (cd "/usr/sh
501         2795  2794  0 21:12 pts/0    00:00:00 sh -c (cd "/usr/sh
501         2798  2795  0 21:12 pts/0    00:00:00 /usr/bin/groff -mt
501         2799  2795  0 21:12 pts/0    00:00:00 /usr/bin/less -is
```

[표 6-1] ps -f의 출력 정보

항목	의미	항목	의미
UID	프로세스를 실행한 사용자 ID	STIME	프로세스의 시작 날짜나 시간
PID	프로세스 번호	TTY	프로세스가 실행된 터미널의 종류와 번호
PPID	부모 프로세스 번호	TIME	프로세스 실행 시간
C	CPU 사용량(% 값)	CMD	실행되고 있는 프로그램 이름(명령)

2 – 프로세스 관리 명령어

- 프로세스 목록 보기
  - 터미널에서 실행한 프로세스의 정보 출력하기: **a 옵션**

```
[adminuser@centos ~]$ ps a
  PID TTY      STAT   TIME COMMAND
 1989 tty3     Ss+    0:00 /sbin/mingetty /dev/tty3
 1993 tty4     Ss+    0:00 /sbin/mingetty /dev/tty4
 1995 tty5     Ss+    0:00 /sbin/mingetty /dev/tty5
 1998 tty6     Ss+    0:00 /sbin/mingetty /dev/tty6
 2015 tty1     Ss+    0:01 /usr/bin/Xorg :0 -br -verbose
 2674 pts/0    Ss     0:00 -bash
 2779 pts/0    T      0:00 top
 3311 tty2     Ss+    0:00 -bash
 3369 pts/1    S+     0:00 nano 123.txt
 3374 pts/0    T      0:00 man ps
```

[표 6-2] STAT에 사용되는 문자의 의미

문자	의미	비고	문자	의미	비고
R	실행 중(running)		STIME	프로세스의 시작 날짜나 시간	
S	인터럽트가 가능한 대기(sleep) 상태		s	세션 리더 프로세스	BSD 형식
T	작업 제어에 의해 정지된(stopped) 상태		+	포그라운드 프로세스 그룹	
Z	좀비 프로세스(defunct)		l(소문자 L)	멀티 스레드	

2 – 프로세스 관리 명령어

- 프로세스 목록 보기
  - 터미널에서 실행한 프로세스의 **상세 정보** 출력하기: **au 옵션**

```
[adminuser@centos ~]$ ps au
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root       1989  0.0  0.0   4064   544 tty3     Ss+  19:19   0:00 /sbin/mingetty /dev/tty3
root       1993  0.0  0.0   4064   540 tty4     Ss+  19:19   0:00 /sbin/mingetty /dev/tty4
root       1995  0.0  0.0   4064   540 tty5     Ss+  19:19   0:00 /sbin/mingetty /dev/tty5
root       1998  0.0  0.0   4064   544 tty6     Ss+  19:19   0:00 /sbin/mingetty /dev/tty6
root       2015  0.0  2.2 190384 43096 tty1     Ss+  19:19   0:01 /usr/bin/Xorg :0 -br -verb
501        2674  0.0  0.0 108336  1808 pts/0    Ss   20:42   0:00 -bash
501        2779  0.0  0.0  15036  1248 pts/0    T    21:09   0:00 top
501        2791  0.0  0.0 101124   984 pts/0    T    21:12   0:00 man awk
501        2794  0.0  0.0 106092  1152 pts/0    T    21:12   0:00 sh -c (cd "/usr/share/man"
501        2795  0.0  0.0 106092   564 pts/0    T    21:12   0:00 sh -c (cd "/usr/share/man"
```

[표 6-3] ps au의 출력 정보

항목	의미	항목	의미
USER	사용자 계정 이름	VSZ	사용하고 있는 가상 메모리의 크기(KB)
%CPU	CPU 사용량을 퍼센트로 표시	RSS	사용하고 있는 물리적 메모리의 크기(KB)
%MEM	물리적 메모리 사용량을 퍼센트로 표시	START	프로세스 시작 시간

2 – 프로세스 관리 명령어

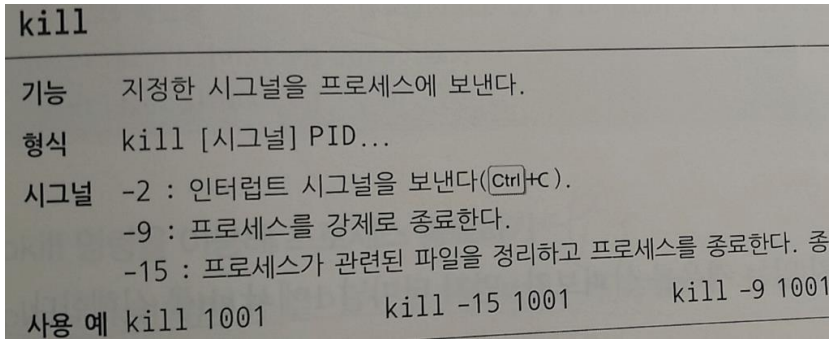
- 프로세스 목록 보기
  - **전체** 프로세스 **목록** 출력하기: **ax 옵션**
    - **ps ax**
  - **전체** 프로세스 **상세 목록** 출력하기: **aux 옵션**
    - **ps aux**
  - 특정한 사용자의 프로세스 목록 출력하기: **-u 옵션**
    - **ps –u adminuser**
  - 특정 프로세스 정보 출력하기: **-fp 옵션**
    - **ps aux | grep nano**
      ps –fp 3369
      ps –p 3369

```
[adminuser@centos ~]$ ps aux | grep nano
501        3369  0.0  0.0 107828  1468 pts/1    S+   21:33   0:00 nano 123.txt
501        3461  0.0  0.0 107408   920 pts/0    S+   21:55   0:00 grep nano
[adminuser@centos ~]$ ps -fp 3369
UID         PID  PPID  C STIME TTY          TIME CMD
501        3369  3283  0 21:33 pts/1    00:00:00 nano 123.txt
[adminuser@centos ~]$ ps -p 3369
  PID TTY          TIME CMD
 3369 pts/1    00:00:00 nano
[adminuser@centos ~]$
```

1 – 프로세스 개념

- 특정 프로세스 정보 검색
  - Ps와 pipe 그리고 grep를 이용하여 검색한다
    - **ps aux | grep nano**

- 프로세스 종료하기: kill 명령어
  - ps aux나 ps –ef로 프로세스의 정보를 확인하면 pid, ppid 번호를 확인하여 **kill -9 pid번호**로 프로세스를 종료시킨다
    - **ps aux**
    - **kill 1234**
    - **kill -9 2345**



kill

기능	지정한 시그널을 프로세스에 보낸다.
형식	kill [시그널] PID...
시그널	-2 : 인터럽트 시그널을 보낸다(Ctrl+C).
	-9 : 프로세스를 강제로 종료한다.
	-15 : 프로세스가 관련된 파일을 정리하고 프로세스를 종료한다. 종
사용 예	kill 1001 kill -15 1001 kill -9 1001

1 – 프로세스 개념

- 프로세스 이름으로 종료하기: **pkill** 명령어
  - Pid 번호로 프로세스 종료하지 않고, 명령어 이름(CMD)으로 프로세스를 종료한다
  - Pkill을 사용하면 시스템에 있는 동일한 이름을 갖는 모든 프로세스가 한꺼번에 종료된다
    - **pkill -x nano**
  - 특정한 사용자가 실행한 것만도 종료할 수 있다
    - **pkill –u 501 nano**

```
[root@centos adminuser]# ps aux | grep nano
501        3577  0.0  0.0 107828  1468 pts/1    S+   22:07   0:00 nano
root       3578  0.0  0.0 107828  1468 pts/2    S+   22:07   0:00 nano
root       3581  0.0  0.0 107408   920 pts/0    S+   22:08   0:00 grep nano
[root@centos adminuser]# pkill -x nano
[root@centos adminuser]# ps aux | grep nano
root       3586  0.0  0.0 107408   924 pts/0    S+   22:08   0:00 grep nano
[root@centos adminuser]#
```

1 – 프로세스 개념

- 프로세스 관리도구
  - top 명령어
    - 현재 프로세스 목록을 주기적으로 출력한다
    - 프로세스의 자세한 요약 정보를 상단에 출력하고 각 프로세스의 정보를 하단에 출력한다

```
top - 22:25:55 up  3:06,  5 users,  load average: 0.00, 0.00, 0.00
Tasks: 213 total,   1 running, 187 sleeping,  25 stopped,   0 zombie
Cpu(s):  0.3%us,  0.0%sy,  0.0%ni, 99.0%id,  0.0%wa,  0.3%hi,  0.3%si,  0.0%st
Mem:   1915524k total,   963192k used,   952332k free,    46700k buffers
Swap:  4095996k total,        0k used,  4095996k free,   525384k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 2933 adminuse  20   0  495m  10m 8292 S  0.3  0.6   0:00.39 gnome-settings-
 3677 root      20   0 15144 1300  928 R  0.3  0.1   0:00.15 top
    1 root      20   0 19364 1536 1224 S  0.0  0.1   0:00.86 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.01 kthreadd
    3 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
```

[표 6-4] top의 내부 명령

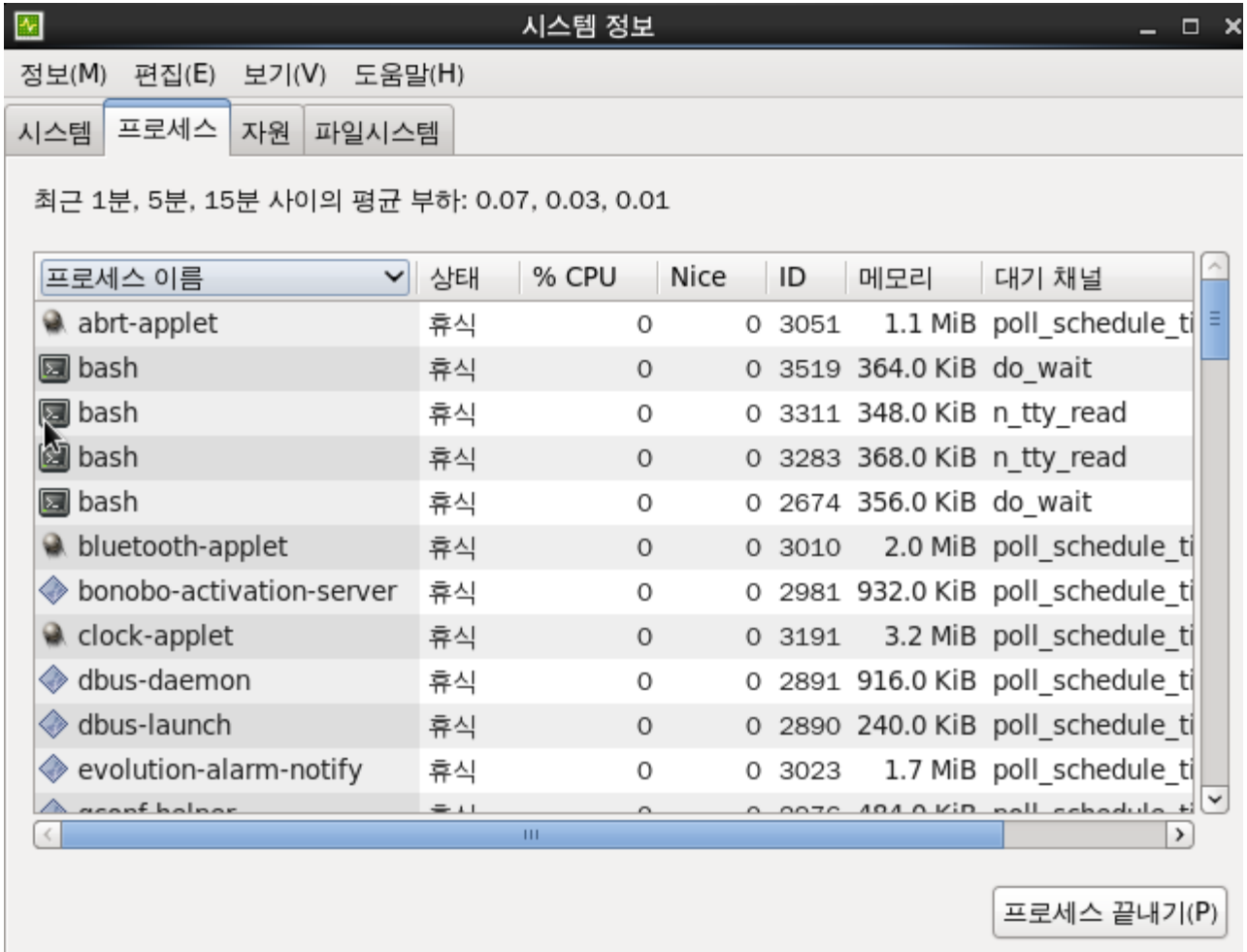
항목	의미	항목	의미
PID	프로세스 ID	SHR	프로세스가 사용하는 공유 메모리 크기
USER	사용자 계정	%CPU	CPU 사용량
PR	우선순위	%MEM	메모리 사용량(%)
NI	Nice 값	TIME+	CPU 누적 이용 시간
VIRT	프로세스가 사용하는 가상 메모리 크기	COMMAND	명령 이름
RES	프로세스가 사용하는 메모리 크기		

[표 6-5] top의 출력 정보

항목	의미	항목	의미
Enter, Space Bar	화면을 즉시 다시 출력한다.	u	사용자에 따라 정렬하여 출력한다.
h, ?	도움말 화면을 출력한다.	M	사용하는 메모리 크기에 따라 정렬하여 출력
k	프로세스를 종료한다. 종료할 프로세스의 PID를 물어본다.	p	CPU 사용량에 따라 정렬하여 출력한다.
n	출력한 프로세스의 개수를 바꾼다.	q	top 명령을 종료한다.

1 – 프로세스 개념

- 프로세스 관리도구
  - 시스템 정보
    - 프로그램 – 시스템 도구 – 시스템 정보



3 - Foreground 및 Background Job 제어

Foreground Job vs Background Job

Job 제어하기

3 - Foreground 및 Background Job 제어

- Foreground Job vs Background Job
  - Foreground Job
    - 터미널에서 작업할 때 일반적으로 사용자가 명령을 입력하면 쉘이 사용자가 입력한 명령어를 해석하여 실행하고 그 결과를 화면에 출력한다
    - 이것을 Foreground Job이라고 한다. 즉, 명령의 결과가 출력된 후에 그 다음 명령어 작업을 할 수 있다
    - 명령어에 대한 출력이 화면에 나오지 않으면 나올 때까지 기다려야 한다
      - **sleep 100**
      - ctrl+z를 입력하여 작업을 강제로 종료한다
  - Background Job
    - Foreground Job은 한꺼번에 여러 개를 동시에 실행할 수 없다
    - Background Job으로 실행하면 명령어의 처리가 끝나는 것을 기다리지 않고 곧바로 프롬프트가 출력되어 사용자가 다른 작업을 추가적으로 할 수 있다
    - Background Job은 명령 실행이 오래 걸리는 작업을 할 때 주로 사용한다
      - **sleep 100 &**

3 - Foreground 및 Background Job 제어

- Foreground Job vs Background Job
  - Background Job과 출력 방향 전환하기
    - Background Job의 실행이 끝나면 현재 터미널에서 결과가 출력되는데, 작업한 순서대로 나타나는 것이 아니라 작업이 끝나는 대로 화면에 보이므로 확인이 어렵다
    - Background로 처리할 때는 출력과 오류 방향 전환을 하여 정상적인 실행 결과와 오류 메시지를 동시에 하나의 파일에 저장하는 방법을 사용하면 좋다
      - **find / -name passwd > pw.dat 2>&1 &**
      - 첫 번째인 **">"**은 find / -name passwd의 정상적인 결과를 pw.dat 파일로 저장하라는 뜻이다
      - 두 번째인 **"2>"**은 find / -name passwd의 표준 오류 메시지를 파일에 저장하라는 뜻이다
      - 세 번째인 **"&1"**은 첫 번째 파일(pw.dat)이라는 것이다. 즉, 오류 메시지가 나오면 pw.dat 파일에 저장하라는 것이다.
      - ls . > ls.good (## 정상적인 출력 메시지를 저장할 때는 **>**을 사용한다)
      - ls NotPresent.txt 2> ls.bad (## 표준 오류 메시지를 저장할 때는 **2>**을 사용한다)

3 - Foreground 및 Background Job 제어

- Job 제어하기
  - Job 목록 보기: **jobs**
    - 현재 실행중인 Background Job을 보는 것이다

```
[root@centos adminuser]# jobs
[1]   Stopped                 man pkill
[2]   Stopped                 man pkill
[3]-  Stopped                 top
[4]+  Stopped                 sleep 5
[5]   Running                 sleep 100 &
[root@centos adminuser]# jobs %5
[5]   Running                 sleep 100 &
[root@centos adminuser]# jobs %+
[4]+  Stopped                 sleep 5
[root@centos adminuser]# jobs %-
[3]-  Stopped                 top
```

[표 6-6] jobs 명령의 출력 항목

항목	출력 예	의미
작업 번호	[1]	작업 번호로서 백그라운드로 실행할 때마다 순차적으로 증가한다([1] [2] [3]...)
작업 순서	+	작업 순서를 표시한다. • +: 가장 최근에 접근한 작업 • −: + 작업보다 바로 전에 접근한 작업 • 공백: 그 외의 작업
상태	실행 중	작업의 상태를 표시한다. • 실행 중(Running): 현재 실행 중이다. • 완료됨(Done): 작업이 정상적으로 종료된다. • 종료됨(Terminated): 작업이 비정상적으로 종료된다. • 정지(Stopped): 작업이 잠시 중단된다.
명령	sleep 100 &	백그라운드로 실행 중인 명령

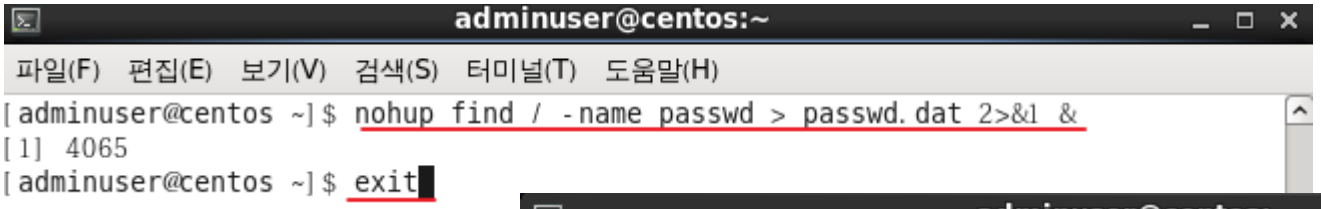
3 - Foreground 및 Background Job 제어

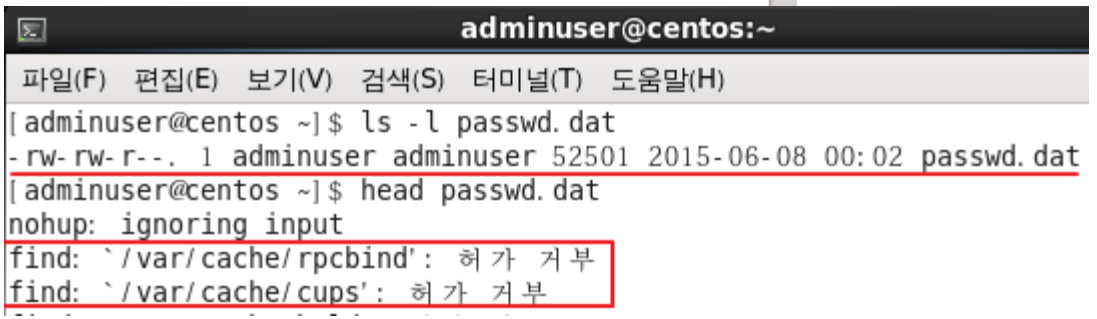
- Job 제어하기
  - Job 전환하기
    - Foreground로 실행 중인 작업을 background로 전환하기 위해서는 **ctrl+z** 로 중지한 후 **bg %5**
    - Background로 실행중인 작업을 Foreground Job으로 전환하려면 그냥 **fg %5**을 사용하면 된다
    - 작업 종료하기
      - **ctrl+c**

```
[root@centos adminuser]# jobs
[1]   Stopped                 man pkill
[2]   Stopped                 man pkill
[3]-  Stopped                 top
[4]+  Stopped                 sleep 5
[root@centos adminuser]# sleep 100
^Z
[5]+  Stopped                 sleep 100
[root@centos adminuser]# bg %5
[5]+ sleep 100 &
[root@centos adminuser]# jobs
[1]   Stopped                 man pkill
[2]   Stopped                 man pkill
[3]-  Stopped                 top
[4]+  Stopped                 sleep 5
[5]   Running                 sleep 100 &
[root@centos adminuser]# fg %5
sleep 100
```

3 - Foreground 및 Background Job 제어

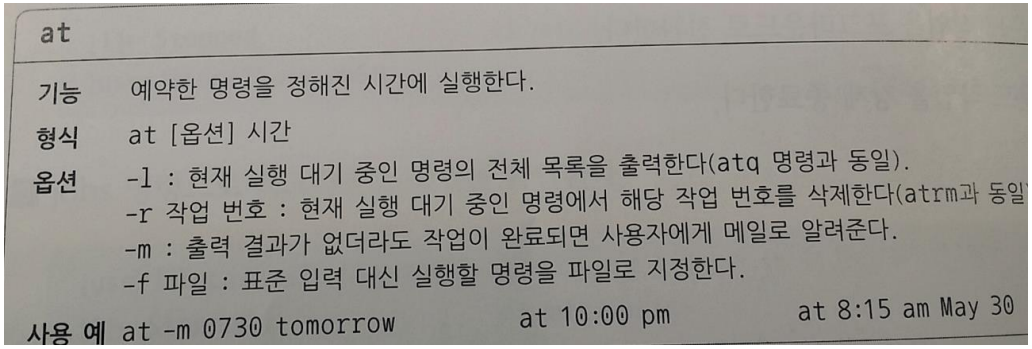
- Job 제어하기
  - Logout 후에도 background job을 계속 실행하기: **nohup 명령 &**
    - Background Job을 실행한 Terminal을 종료하거나 사용자가 Logout하면 실행중이던 Background Job이 함께 종료된다
    - 로그아웃한 후에도 계속 작업을 하기 위해서는 nohup을 사용하고 이 명령어는 반드시 background job으로 해야 한다
    - 별도로 출력 방향 전환을 하지 않으면 nohup 명령의 실행 결과와 오류 메시지가 현재 디렉터리에 nohup.out 파일로 자동 저장된다
    - 출력 방향을 전환하면 원하는 파일에 저장할 수 있다
      - **nohup find / -name passwd > passwd.dat 2>&1 &**





4 – Job 예약하기

- 정해진 시간에 한 번만 실행하기
  - 매주 백업을 하거나 매일 아침 시스템의 상태를 자동으로 확인하여 정해진 시간에 관리자에게 메일을 보내고자 할 때는 Job을 예약하면 된다
  - **at 명령어 설정하기**
    - At 명령을 정해진 시간에 자동으로 실행하려면 [at 시간] 형식을 사용한다
      - at 4 pm + 3days
      - at 10 am Jul 31
      - at 1 am tomorrow
      - at 10:00 am today



at

기능	예약한 명령을 정해진 시간에 실행한다.
형식	at [옵션] 시간
옵션	-l : 현재 실행 대기 중인 명령의 전체 목록을 출력한다(atq 명령과 동일). -r 작업 번호 : 현재 실행 대기 중인 명령에서 해당 작업 번호를 삭제한다(atrm과 동일 -m : 출력 결과가 없더라도 작업이 완료되면 사용자에게 메일로 알려준다. -f 파일 : 표준 입력 대신 실행할 명령을 파일로 지정한다.
사용 예	at -m 0730 tomorrow　　at 10:00 pm　　at 8:15 am May 30

4 – Job 예약하기

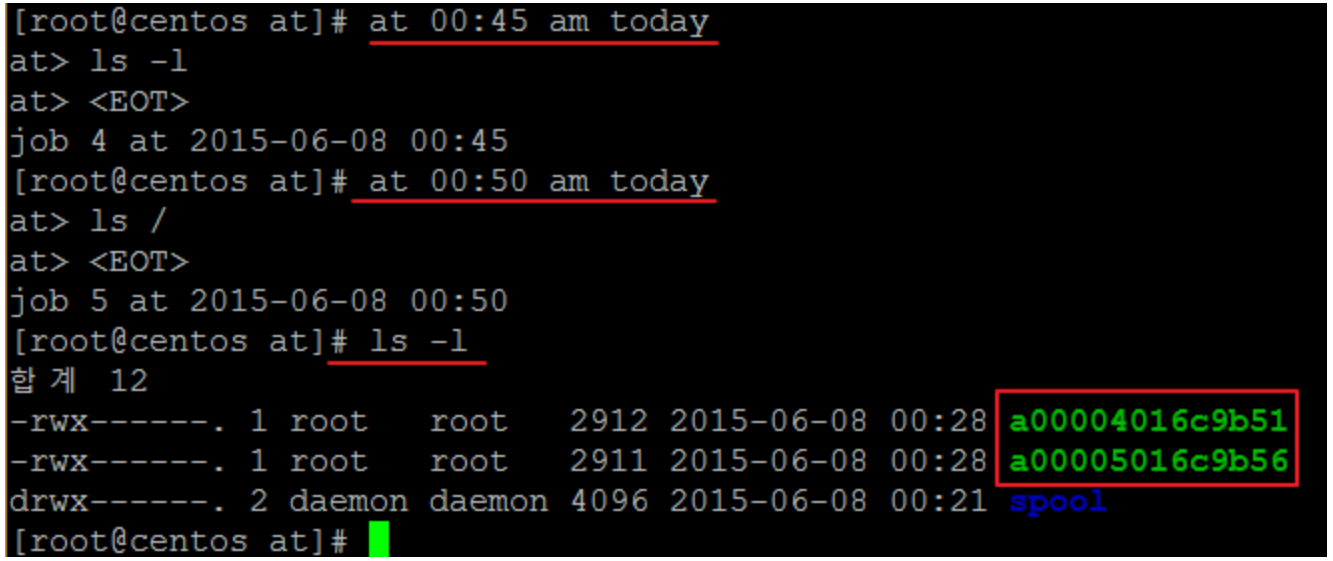
- 정해진 시간에 한 번만 실행하기
  - **at 명령어 설정하기**
    - Job이 실행되면 mail이 온다. 이것을 확인하기 위해 mail을 입력하고 Enter를 누른다

```
[root@centos adminuser]# date
2015. 06. 08. (월) 00:19:45 KST
[root@centos adminuser]# at 00:21 am today
at> ls -l ~adminuser1
at> <EOT>
job 3 at 2015-06-08 00:21
You have new mail in /var/spool/mail/adminuser
[root@centos adminuser]# mail
Heirloom Mail version 12.4 7/29/08.  Type ? for help.
"/var/spool/mail/adminuser": 1 message 1 new
>N  1 root                  Mon Jun  8 00:19  38/2166  "Output from your job        1"
&
Message  1:
From root@centos.webtime.local  Mon Jun  8 00:19:01 2015
Return-Path: <root@centos.webtime.local>
X-Original-To: adminuser
Delivered-To: adminuser@centos.webtime.local
Subject: Output from your job        1
To: adminuser@centos.webtime.local
Date: Mon,  8 Jun 2015 00:19:01 +0900 (KST)
From: root@centos.webtime.local (root)
Status: R

합계 56448
-rw-r--r--. 1 root      root           189 2015-06-07 23:32 1.txt
-rw-rw-r--. 1 adminuser adminuser        0 2015-06-07 21:33 123.txt
-rw-r--r--. 1 adminuser adminuser 26214400 2015-06-06 12:47 25m.img
-rw-r--r--. 1 adminuser adminuser 31457280 2015-06-06 12:46 30m.img
```

4 – Job 예약하기

- 정해진 시간에 한 번만 실행하기
  - **at 작업 파일 확인하기**
    - At으로 생성된 작업 파일은 /var/spool/at 디렉터리에 저장된다
    - su root를 하여 이 디렉터리의 파일의 목록을 확인한다
      - **ls –l /var/spool/at**
    - Job이 실행되면 이 파일들은 자동으로 삭제된다



4 – Job 예약하기

- 정해진 시간에 한 번만 실행하기
  - **at 작업 목록 확인하기: -l 옵션**
    - **at –l**

```
[root@centos at]# at -l
4       2015-06-08 00:45 a root
5       2015-06-08 00:50 a root
[root@centos at]#
```

  - **at 작업 제거하기: -d 옵션**
    - **at –d 작업번호**

```
[root@centos at]# at -l
4       2015-06-08 00:45 a root
5       2015-06-08 00:50 a root
[root@centos at]# atq
4       2015-06-08 00:45 a root
5       2015-06-08 00:50 a root
[root@centos at]# at -d 5
[root@centos at]# at -l
4       2015-06-08 00:45 a root
[root@centos at]#
```

4 – Job 예약하기

- 정해진 시간에 한 번만 실행하기
  - **at 명령어 사용 제한하기**
    - /etc/at.allow와 /etc/at.deny 파일이 있다
    - 기본적으로 /etc/at.deny 파일은 있고 /etc/at.allow 파일은 없다
    - 이 상태인 경우에는 모든 사용자가 Job을 예약할 수 있다
    - 특정한 사용자만 예약 작업을 하도록 하면 /etc/at.allow에 사용자를 추가
    - 특정한 사용자만 예약 작업을 못하도록 하면 /etc/at.deny에 사용자를 추가
    - /etc/at.allow와 /etc/at.deny 파일이 없으면 root만 Job 예약을 할 수 있다

```
[root@centos etc]# cd /etc/
[root@centos etc]# ls -l at.deny
-rw-r--r--. 1 root root 0 2015-06-08 00:42 at.deny
[root@centos etc]# cat > at.deny
adminuser
[root@centos etc]# cat at.deny
adminuser
[root@centos etc]# date
2015. 06. 08. (월) 00:43:12 KST
[root@centos etc]# at 00:55 am today
at> ls -l /
at> <EOT>
job 6 at 2015-06-08 00:55
[root@centos etc]# su adminuser
[adminuser@centos etc]$ at 00:58 am today
You do not have permission to use at.
[adminuser@centos etc]$
```

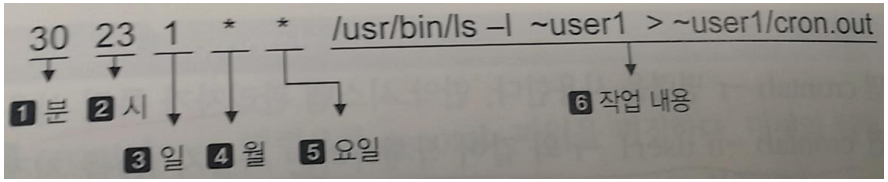
4 – Job 예약하기

- 정해진 시간에 반복적으로 실행하기
  - crontab 명령어 사용

crontab

기능	사용자의 crontab 파일을 관리한다.
형식	crontab [-u 사용자 ID] [옵션] [파일 이름]
옵션	-e : 사용자의 crontab 파일을 편집한다. -l : crontab 파일의 목록을 출력한다. -r : crontab 파일을 삭제한다.
사용 예	crontab -l　　crontab -u user1 -e　　crontab -r

  - crontab 파일 형식 이해하기



    - 분(0~59), 시(0~23),일(1~31),월(1~12),요일(0~6)
    - 각 항목간에는 한 칸씩 띄워서 작업한다

4 – Job 예약하기

- 정해진 시간에 반복적으로 실행하기
  - crontab 파일 생성하고 편집하기: **crontab –e** 또는 **crontab**
    - 저장된 경로는 /var/spool/cron 디렉터리
    - Root 계정으로만 작업한 파일을 확인

```
[adminuser@centos etc]$ crontab
10 13 * * 0 /usr/bin/ls -l ~adminuser > ~adminuser/cron.out
[adminuser@centos etc]$ ls /var/spool/cron
ls: cannot open directory /var/spool/cron: 허가 거부
[adminuser@centos etc]$ su root
암호:
[root@centos etc]# ls /var/spool/cron
adminuser
[root@centos etc]#
```

  - Crontab 파일 내용 확인하기: **crontab –l**

```
[root@centos etc]# crontab -l
no crontab for root
[root@centos etc]# su adminuser
[adminuser@centos etc]$ crontab -l
10 13 * * 0 /usr/bin/ls -l ~adminuser > ~adminuser/cron.out
[adminuser@centos etc]$
```

4 – Job 예약하기

- 정해진 시간에 반복적으로 실행하기
  - crontab 파일 삭제하기: **crontab –r**
    - 사용자가 자기가 한 작업을 삭제하기
      - **crontab –r**
      - **crontab –l** (##삭제되었는지 목록 확인하기)
    - 다른 사람이 만든 작업을 관리자가 삭제하기
      - **crontab –u adminuser –r**
      - **crontab –u adminuser –l** (##삭제되었는지 목록 확인하기)

4 – Job 예약하기

- 정해진 시간에 한 번만 실행하기
  - **crontab 명령어 사용 제한하기**
    - /etc/cron.allow와 /etc/cron.deny 파일이 있다
    - 기본적으로 /etc/cron.deny 파일은 있고 /etc/cron.allow 파일은 없다
    - 이 상태인 경우에는 모든 사용자가 Job을 예약할 수 있다
    - 특정한 사용자만 예약 작업을 하도록 하면 /etc/cron.allow에 사용자를 추가
    - 특정한 사용자만 예약 작업을 못하도록 하면 /etc/cron.deny에 사용자를 추가한다
    - /etc/cron.allow와 /etc/cron.deny 파일이 없으면 root만 Job 예약을 할 수 있다
    - /etc/cron.deny 파일조차도 없으면 일반 사용자는 crontab 작업을 할 권한이 없다

```
[root@centos etc]# ls -l cron.de*
-rw-------. 1 root root 0 2013-11-23 21:43 cron.deny_old
[root@centos etc]# su adminuser
[adminuser@centos etc]$ crontab -e
You (adminuser) are not allowed to use this program (crontab)
See crontab(1) for more information
[adminuser@centos etc]$
```