우분투 리눅스

시스템 & 네트워크

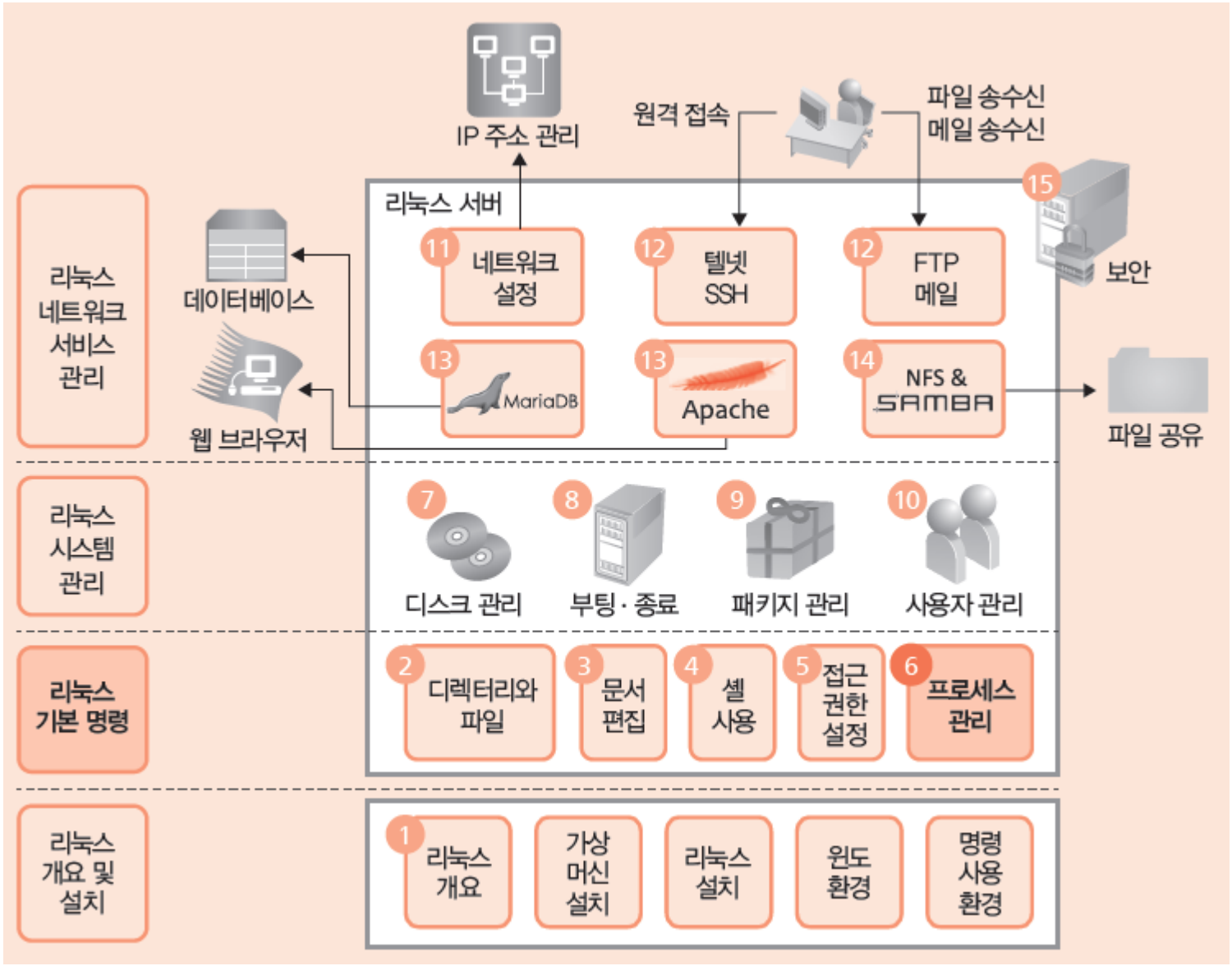
Chapter 06. 프로세스 관리하기

목차

학습목표

- 프로세스에 대해 설명할 수 있다.
- 프로세스 목록을 확인하고 특정 프로세스를 검색할 수 있다.
- 프로세스를 강제로 종료할 수 있다.
- 프로세스 관리 도구로 전체 프로세스의 상태를 확인할 수 있다.
- 포그라운드와 백그라운드 작업의 차이를 설명할 수 있다.
- 백그라운드로 작업을 실행하고 포그라운드로 변환할 수 있다.
- 정해진 시간에 혹은 주기적으로 명령을 실행하도록 설정할 수 있다.

리눅스 실습 스터디 맵


IP 주소 관리
원격 접속
파일 송수신
메일 송수신
리눅스 서버
15 보안
리눅스 네트워크 서비스 관리
데이터베이스
웹 브라우저
11 네트워크 설정
12 텔넷 SSH
12 FTP 메일
13 MariaDB
13 Apache
14 NFS & SAMBA
파일 공유
리눅스 시스템 관리
7 디스크 관리
8 부팅 · 종료
9 패키지 관리
10 사용자 관리
리눅스 기본 명령
2 디렉터리와 파일
3 문서 편집
4 셸 사용
5 접근 권한 설정
6 프로세스 관리
리눅스 개요 및 설치
1 리눅스 개요
가상 머신 설치
리눅스 설치
윈도 환경
명령 사용 환경

00 개요

그림 6-1 6장의 내용 구성

01 프로세스의 개념

- **프로세스: 현재 시스템에서 실행 중인 프로그램**
- **프로세스의 부모-자식 관계**
  - 프로세스는 부모-자식 관계를 가지고 있음
  - 필요에 따라 부모 프로세스(parent process)는 자식 프로세스(child process)를 생성하고, 자식 프로세스는 또 다른 자식 프로세스 생성 가능
  - 부팅할 때 스케줄러가 실행한 프로세스인 systemd와 kthreadd 프로세스를 제외하면 모든 프로세스는 부모 프로세스를 가지고 있음
  - 자식 프로세스는 할 일이 끝나면 부모 프로세스에 결과를 돌려주고 종료
- **프로세스의 번호**
  - 각 프로세스는 고유한 번호를 가지고 있는데 이것이 PID
- **프로세스의 종류**
  - 데몬 프로세스
    - 특정 서비스를 제공하기 위해 존재하며 리눅스 커널에 의해 실행
  - 고아 프로세스
    - 자식 프로세스가 아직 실행 중인데 부모 프로세스가 먼저 종료된 자식 프로세스는 고아(orphan) 프로세스
    - 1번 프로세스가 고아 프로세스의 새로운 부모 프로세스가 되어 고아 프로세스의 작업 종료 지원
  - 좀비 프로세스
    - 자식 프로세스가 실행을 종료했는데도 프로세스 테이블 목록에 남아 있는 경우
    - 좀비 프로세스는 프로세스 목록에 defunct 프로세스라고 나오기도함
    - 좀비 프로세스가 증가하면 프로세스 테이블의 용량이 부족해서 일반 프로세스가 실행되지 않을 수도 있음

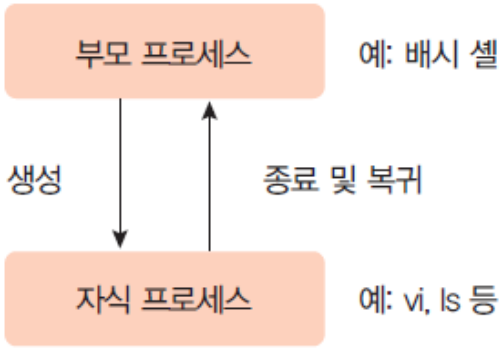


그림 6-2 부모 프로세스와 자식 프로세스의 관계

02 프로세스 관리 명령

프로세스 목록 보기

- 현재 실행 중인 프로세스의 목록을 보는 명령: ps
  - 유닉스(SVR4) 옵션 : 묶어서 사용할 수 있고, 붙임표로 시작한다(예 : -ef).
  - BSD 옵션 : 묶어서 사용할 수 있고, 붙임표로 시작하지 않는다(예 : aux).
  - GNU 옵션 : 붙임표 두 개로 시작한다(예 : --pid).

ps

- 기능 현재 실행 중인 프로세스의 정보를 출력한다.
- 형식 ps [옵션]
- 옵션 〈유닉스 옵션〉
  - -e: 시스템에서 실행 중인 모든 프로세스의 정보를 출력한다.
  - -f: 프로세스의 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 소유자 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 PID의 정보를 출력한다.
- 사용 예 ps ps -ef ps aux

02 프로세스 관리 명령

현재 단말기의 프로세스 목록 출력하기 : ps

- ps 명령을 옵션 없이 사용하면 현재 셸이나 터미널에서 실행한 사용자 프로세스에 대한 정보를 출력

```
user1@myubuntu:~$ ps
   PID TTY          TIME CMD
  4882 pts/2    00:00:00 bash
  4996 pts/2    00:00:00 ps
user1@myubuntu:~$
```

프로세스의 상세 정보 출력하기 : -f 옵션

- 프로세스의 상세한 정보를 출력: PPID와 터미널 번호, 시작 시간 등

```
user1@myubuntu:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
user1       4882    4881  0 13:00 pts/2    00:00:00 -bash
user1       4999    4882  0 14:43 pts/2    00:00:00 ps -f
user1@myubuntu:~$
```

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)

02 프로세스 관리 명령

■ 터미널에서 실행한 프로세스의 정보 출력하기 : a 옵션

- 터미널에서 실행한 프로세스의 정보를 출력

```
user1@myubuntu:~$ ps a
    PID   TTY STAT TIME COMMAND
    600   tty1     Ssl+ 0:00 /usr/lib/gdm3/gdm-wayland-session gnome-session --au
    607   tty1     Sl+  0:00 /usr/lib/gnome-session/gnome-session-binary --autost
    690   tty1     Sl+  0:05 /usr/bin/gnome-shell
(생략)
user1@myubuntu:~$
```

표 6-2 STAT에 사용되는 문자의 의미

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

02 프로세스 관리 명령

■ 터미널에서 실행한 프로세스의 상세 정보 출력하기 : a 옵션과 u 옵션

- a 옵션과 u 옵션을 함께 사용하면 터미널에서 실행한 프로세스의 상세 정보를 출력: CPU와 메모리 사용량 등

```
user1@myubuntu:~$ ps au
USER    PID  %CPU  %MEM     VSZ    RSS  TTY    STAT  START  TIME  COMMAND
gdm     600   0.0   0.2  199116   4780  tty1   Ssl+  10:13  0:00  /usr/lib/gdm3
gdm     607   0.0   0.4  586848   8764  tty1   Sl+   10:13  0:00  /usr/lib/gnom
gdm     690   0.0   4.5 2977116  93052  tty1   Sl+   10:13  0:05  /usr/bin/gnom
gdm     788   0.2   1.0  280596  21456  tty1   S+    10:13  0:39  /usr/bin/Xway
(생략)
user1   2073  0.0   0.6  903548  13616  tty2   Sl+   10:15  0:00  /usr/lib/x86_
user1   2100  0.0   5.9 1000292 119676  tty2   SNl+  10:17  0:08  /usr/bin/pyth
user1   4882  0.0   0.2   31012   5236  pts/2  Ss    13:00  0:00  -bash
user1   5003  0.0   0.1   50328   3400  pts/2  R+    14:50  0:00  ps au
user1@myubuntu:~$
```

표 6-3 ps au의 출력 정보

항목	의미	항목	의미
USER	사용자 계정 이름	VSZ	사용 중인 가상 메모리의 크기(KB)
%CPU	퍼센트로 표시한 CPU 사용량	RSS	사용 중인 물리적 메모리의 크기(KB)
%MEM	퍼센트로 표시한 물리적 메모리 사용량	START	프로세스 시작 시간

02 프로세스 관리 명령

전체 프로세스 목록 출력하기(유닉스 옵션) : -e 옵션

- -e 옵션은 시스템에서 실행 중인 모든 프로세스를 출력
- TTY의 값이 ?인 것은 대부분 데몬으로 시스템이 실행한 프로세스

```
user1@myubuntu:~$ ps -e | more
   PID TTY          TIME CMD
     1 ?        00:00:08 systemd
     2 ?        00:00:00 kthreadd
     4 ?        00:00:00 kworker/0:0H
     6 ?        00:00:00 ksoftirqd/0
     7 ?        00:00:20 rcu_sched
(생략)
    20 ?        00:00:00 khugepaged
    21 ?        00:00:00 crypto
    22 ?        00:00:00 kintegrityd
    23 ?        00:00:00 kblockd
    24 ?        00:00:00 ata_sff
--More--
```

02 프로세스 관리 명령

전체 프로세스 목록 출력하기(유닉스 옵션) : -e 옵션

- -ef 옵션 사용: 전체 프로세스의 더 자세한 정보 출력

```
user1@myubuntu:~$ ps -ef | more
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 10:12 ?        00:00:03 /sbin/init splash
root           2       0  0 10:12 ?        00:00:00 [kthreadd]
root           4       2  0 10:12 ?        00:00:00 [kworker/0:0H]
root           6       2  0 10:12 ?        00:00:00 [mm_percpu_wq]
root           7       2  0 10:12 ?        00:00:00 [ksoftirqd/0]
(생략)
root          20       2  0 10:12 ?        00:00:00 [khugepaged]
root          21       2  0 10:12 ?        00:00:00 [crypto]
root          22       2  0 10:12 ?        00:00:00 [kintegrityd]
root          23       2  0 10:12 ?        00:00:00 [kblockd]
root          24       2  0 10:12 ?        00:00:00 [ata_sff]
--More--
```

02 프로세스 관리 명령

전체 프로세스 목록 출력하기(BSD 옵션) : ax 옵션

- 시스템에서 실행 중인 모든 프로세스를 출력

```
user1@myubuntu:~$ ps ax | more
   PID TTY      STAT   TIME  COMMAND
     1 ?        Ss     0:03  /sbin/init splash
     2 ?        S      0:00  [kthreadd]
     4 ?        S<     0:00  [kworker/0:0H]
     6 ?        S<     0:00  [mm_percpu_wq]
     7 ?        S      0:00  [ksoftirqd/0]
(생략)
    20 ?        SN     0:00  [khugepaged]
    21 ?        S<     0:00  [crypto]
    22 ?        S<     0:00  [kintegrityd]
    23 ?        S<     0:00  [kblockd]
    24 ?        S<     0:00  [ata_sff]
--More--
```

02 프로세스 관리 명령

전체 프로세스 목록 출력하기(BSD 옵션) : ax 옵

- aux 옵션은 -ef처럼 시스템에서 실행 중인 모든 프로세스에 대한 자세한 정보를 출력

```
user1@myubuntu:~$ ps aux | more
USER    PID  %CPU  %MEM     VSZ   RSS  TTY   STAT  START   TIME  COMMAND
root      1   0.0   0.3  154784  7152  ?     Ss    10:12   0:03  /sbin/init splash
root      2   0.0   0.0       0     0  ?     S     10:12   0:00  [kthreadd]
root      4   0.0   0.0       0     0  ?     S<    10:12   0:00  [kworker/0:0H]
root      6   0.0   0.0       0     0  ?     S<    10:12   0:00  [mm_percpu_wq]
root      7   0.0   0.0       0     0  ?     S     10:12   0:00  [ksoftirqd/
(생략)
root     18   0.0   0.0       0     0  ?     S     10:12   0:00  [kcompactd0]
root     19   0.0   0.0       0     0  ?     SN    10:12   0:00  [ksmd]
root     20   0.0   0.0       0     0  ?     SN    10:12   0:00  [khugepaged]
root     21   0.0   0.0       0     0  ?     S<    10:12   0:00  [crypto]
--More--
```

02 프로세스 관리 명령

특정 사용자의 프로세스 목록 출력하기 : -u 옵션

```
user1@myubuntu:~$ ps -u user1
   PID TTY          TIME CMD
  1343 ?        00:00:00 systemd
  1344 ?        00:00:00 (sd-pam)
  1351 ?        00:00:00 gnome-keyring-d
  1355 tty2     00:00:00 gdm-wayland-ses
  1357 ?        00:00:00 dbus-daemon
  1359 tty2     00:00:00 gnome-session-b
  1414 ?        00:00:00 gvfsd
(생략)
```

- 더 상세한 정보를 보고 싶으면 -f 옵션을 함께 사용

```
user1@myubuntu:~$ ps -fu user1
UID          PID    PPID  C STIME TTY          TIME CMD
user1       1343       1  0 10:13 ?        00:00:00 /lib/systemd/systemd --user
user1       1344    1343  0 10:13 ?        00:00:00 (sd-pam)
user1       1351       1  0 10:13 ?        00:00:00 /usr/bin/gnome-keyring-daemon
user1       1355    1319  0 10:13 tty2     00:00:00 /usr/lib/gdm3/gdm-wayland-ses
user1       1357    1343  0 10:13 ?        00:00:00 /usr/bin/dbus-daemon --sessio
user1       1359    1355  0 10:13 tty2     00:00:00 /usr/lib/gnome-session/gnome-
user1       1414    1343  0 10:13 ?        00:00:00 /usr/lib/gvfs/gvfsd
(생략)
```

02 프로세스 관리 명령

특정 프로세스 정보 출력하기 : -p 옵션

- -p 옵션과 함께 특정 PID를 지정하면 해당 프로세스의 정보를 출력

```
user1@myubuntu:~$ ps -p 4882
   PID TTY          TIME CMD
  4882 pts/2    00:00:00 bash
user1@myubuntu:~$ ps -fp 4882
UID          PID    PPID  C STIME TTY          TIME CMD
user1       4882    4881  0 13:00 pts/2    00:00:00 -bash
user1@myubuntu:~$
```

ps 명령을 이용해 특정 프로세스 정보 검색하기

- ps 명령과 grep 명령을 |로 연결하여 특정 프로세스에 대한 정보를 검색

```
user1@myubuntu:~$ ps -ef | grep bash
user1       1807    1797  0 10:14 pts/0    00:00:00 bash
user1       4882    4881  0 13:00 pts/2    00:00:00 -bash
user1       5020    4882  0 14:58 pts/2    00:00:00 grep --color=auto bash
user1@myubuntu:~$
```

02 프로세스 관리 명령

pgrep 명령을 이용해 특정 프로세스 정보 검색하기

pgrep

- 기능 지정한 패턴과 일치하는 프로세스의 정보를 출력한다.
- 형식 pgrep [옵션] 패턴
- 옵션 -x: 패턴과 정확히 일치하는 프로세스의 정보를 출력한다.
  -n: 패턴을 포함하고 있는 가장 최근 프로세스의 정보를 출력한다.
  -u 사용자명: 특정 사용자에 대한 모든 프로세스를 출력한다.
  -l: PID와 프로세스 이름을 출력한다.
  -t term: 특정 단말기와 관련된 프로세스의 정보를 출력한다.
- 사용 예 pgrep bash

- bash 패턴을 지정하여 검색한 예

```
user1@myubuntu:~$ pgrep -x bash
1807
4882
user1@myubuntu:~$
```

02 프로세스 관리 명령

pgrep 명령을 이용해 특정 프로세스 정보 검색하기

- pgrep의 경우 -l 옵션을 지정해도 단지 PID와 명령 이름만 출력

```
user1@myubuntu:~$ pgrep -l bash
1807 bash
4882 bash
user1@myubuntu:~$
```

- 더 자세한 정보를 검색하려면 pgrep 명령을 ps 명령과 연결하여 사용

```
user1@myubuntu:~$ ps -fp $(pgrep -x bash)
UID          PID    PPID  C STIME  TTY      STAT   TIME  CMD
user1       1807    1797  0 10:14  pts/0    Ss+    0:00  bash
user1       4882    4881  0 13:00  pts/2    Ss     0:00  -bash
user1@myubuntu:~$
```

- -u 옵션으로 사용자명을 지정하여 검색

```
user1@myubuntu:~$ ps -fp $(pgrep -u user1 bash)
UID          PID    PPID  C  STIME TTY      STAT   TIME  CMD
user1       1807    1797  0  10:14 pts/0    Ss+    0:00  bash
user1       4882    4881  0  13:00 pts/2    Ss     0:00  -bash
user1@myubuntu:~$
```

02 프로세스 관리 명령

시그널(signal)

- 프로세스에 무언가 발생했음을 알리는 간단한 메시지
- 리눅스에서 지원하는 시그널의 목록은 kill -l 명령으로 알 수 있음

```
user1@myubuntu:~$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
user1@myubuntu:~$
```

02 프로세스 관리 명령

■ 시그널(signal)

- 주로 사용하는 시그널

표 6-4 주요 시그널

시그널	번호	기본 처리	의미
SIGHUP	1	종료	터미널과의 연결이 끊어졌을 때 발생한다.
SIGINT	2	종료	인터럽트로 사용자가 Ctrl+c를 입력하면 발생한다.
SIGQUIT	3	종료, 코어덤프	종료 신호로 사용자가 Ctrl+\을 입력하면 발생한다.
SIGKILL	9	종료	이 시그널을 받은 프로세스는 무시할 수 없으며 강제로 종료된다.
SIGALRM	14	종료	알람에 의해 발생한다.
SIGTERM	15	종료	kill 명령이 보내는 기본 시그널이다.

02 프로세스 관리 명령

kill 명령을 이용해 프로세스 종료하기

kill

- 기능 지정한 시그널을 프로세스에 보낸다.
- 형식 kill [-시그널] PID
- 시그널 2: 인터럽트 시그널을 보낸다(Ctrl+C).
  9: 프로세스를 강제로 종료한다.
  15: 프로세스와 관련된 파일을 정리한 후 종료한다. 종료되지 않는 프로세스가 있을 수 있다.
- 사용 예 kill 1001
  kill -9 1001
  kill -15 1001

- kill 예: man을 실행시킨 프로세스를 찾아서 종료시키기

```
user1@myubuntu:~$ ps -fp $(pgrep -x man)
UID          PID    PPID  C STIME  TTY          TIME  CMD
user1       5117    5105  0 15:09  pts/1    00:00:00  man ps
user1@myubuntu:~$
```

02 프로세스 관리 명령

프로세스 강제로 종료하기

- 단순히 kill 명령으로는 종료되지 않는 경우 강제 종료 시그널인 9번을 보낸다.
- 강제종료 예: kill 명령으로 종료되지 않음

```
user1@myubuntu:~$ ps -fp $(pgrep -x sh)
UID          PID    PPID  C STIME  TTY          TIME CMD
user1       5137    5105  0 15:11  pts/1    00:00:00 sh
user1@myubuntu:~$
user1@myubuntu:~$ kill 5137
user1@myubuntu:~$ ps -fp $(pgrep -x sh)
UID          PID    PPID  C STIME TTY          TIME CMD
user1       5137    5105  0 15:11 pts/1    00:00:00 sh
user1@myubuntu:~$
```

- 강제 종료 시그널인 9번을 보내 강제로 종료

```
user1@myubuntu:~$ kill -9 5137
user1@myubuntu:~$
```

02 프로세스 관리 명령

pkill 명령을 이용해 프로세스 종료하기

- PID가 아니라 프로세스의 명령 이름(CMD)으로 프로세스를 찾아 종료

```
user1@myubuntu:~$ ps -fp $(pgrep -x man)
UID          PID    PPID  C STIME TTY      STAT   TIME CMD
user1       5208    5105  0 15:15 pts/1    S+     0:00 man ps
user1       5225    4882  0 15:15 pts/2    S+     0:00 man ps
user1@myubuntu:~$
user1@user1-pc:~$ pkill man
user1@user1-pc:~$ pgrep -x man
user1@user1-pc:~$
```

02 프로세스 관리 명령

■ 프로세스 관리 도구

- top 명령: 현재 실행 중인 프로세스에 대한 정보를 주기적으로 출력

표 6-5 top 명령의 출력 정보

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

02 프로세스 관리 명령

■ 프로세스 관리 도구

- top 명령: 현재 실행 중인 프로세스에 대한 정보를 주기적으로 출력

표 6-6 top 명령의 내부 명령

내부 명령	기능
Enter, Space Bar	화면을 즉시 다시 출력한다.
h, ?	도움말 화면을 출력한다.
k	프로세스를 종료한다. 종료할 프로세스의 PID를 물어본다.
n	출력하는 프로세스의 개수를 바꾼다.
p	CPU 사용량에 따라 정렬하여 출력한다.
q	top 명령을 종료한다.
M	사용하는 메모리의 크기에 따라 정렬하여 출력한다.
u	사용자에 따라 정렬하여 출력한다.

02 프로세스 관리 명령

- **top 실행 화면**

```
user1@myubuntu:~$ top
top - 15:22:25 up  5:09,  4 users,  load average: 0.00, 0.00, 0.00
Tasks: 251 total,   1 running, 250 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.7 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 s
KiB Mem :  2024232 total,   211636 free,  1349216 used,   463380 buff/cache
KiB Swap:  1214880 total,  1179296 free,    35584 used.   504624 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 5269 user1     20   0   54864   3972   3316 R  1.0  0.2   0:00.18 top
   30 root      20   0       0      0      0 S  0.3  0.0   0:14.20 kworker/0+
  788 gdm       20   0  280596  21456   5880 S  0.3  1.1   0:50.74 Xwayland
 4881 user1     20   0  107696   4300   3256 S  0.3  0.2   0:00.93 sshd
    1 root      20   0  154784   7156   5208 S  0.0  0.4   0:03.41 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.01 kthreadd
    4 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/0+
    6 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 mm_percpu+
    7 root      20   0       0      0      0 S  0.0  0.0   0:00.66 ksoftirqd+
    8 root      20   0       0      0      0 S  0.0  0.0   0:00.54 rcu_sched
    9 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_bh
   10 root      rt   0       0      0      0 S  0.0  0.0   0:00.00 migration+
   11 root      rt   0       0      0      0 S  0.0  0.0   0:00.06 watchdog/0
   12 root      20   0       0      0      0 S  0.0  0.0   0:00.00 cpuhp/0
   13 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kdevtmpfs
   14 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 netns
```

02 프로세스 관리 명령

■ 시스템 정보: GNOME

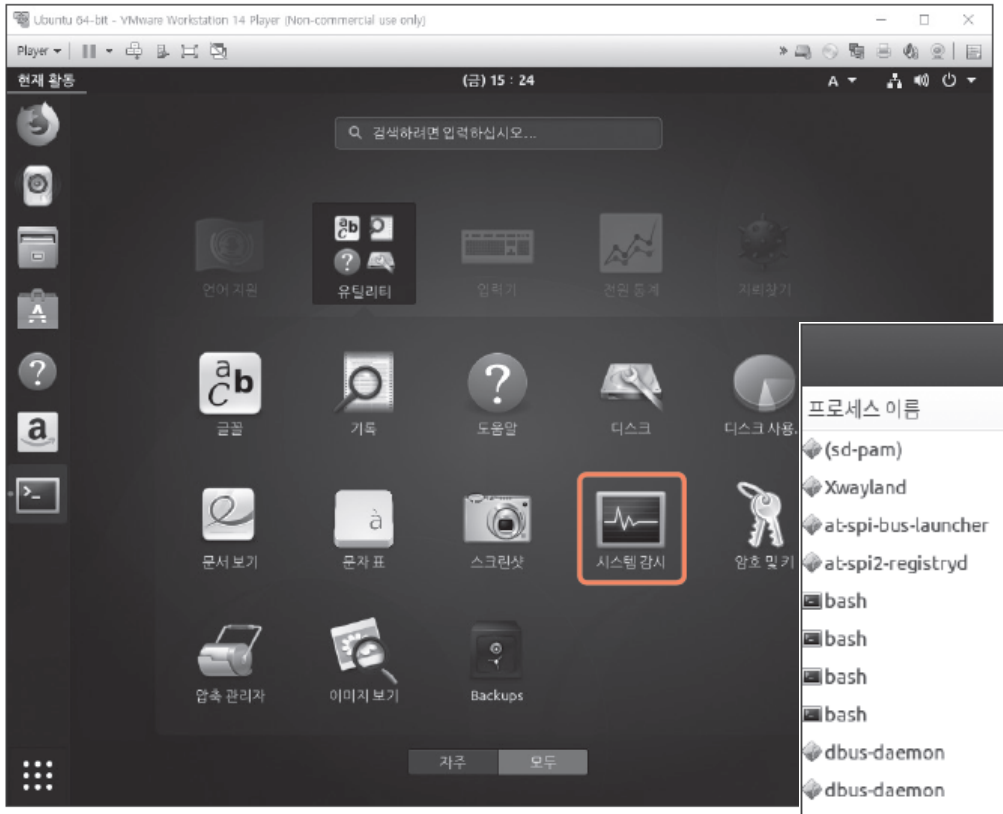
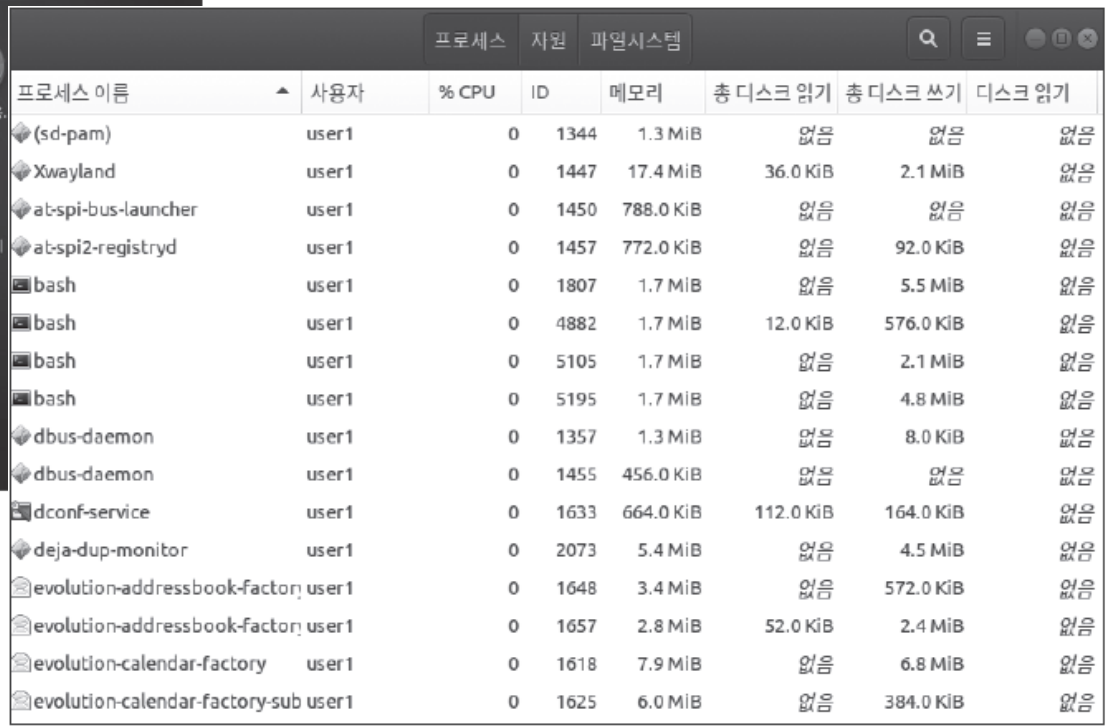


그림 6-3 시스템 감시 동작시키기



프로세스 이름	사용자	% CPU	ID	메모리	총 디스크 읽기	총 디스크 쓰기	디스크 읽기
(sd-pam)	user1	0	1344	1.3 MiB	없음	없음	없음
Xwayland	user1	0	1447	17.4 MiB	36.0 KiB	2.1 MiB	없음
at-spi-bus-launcher	user1	0	1450	788.0 KiB	없음	없음	없음
at-spi2-registryd	user1	0	1457	772.0 KiB	없음	92.0 KiB	없음
bash	user1	0	1807	1.7 MiB	없음	5.5 MiB	없음
bash	user1	0	4882	1.7 MiB	12.0 KiB	576.0 KiB	없음
bash	user1	0	5105	1.7 MiB	없음	2.1 MiB	없음
bash	user1	0	5195	1.7 MiB	없음	4.8 MiB	없음
dbus-daemon	user1	0	1357	1.3 MiB	없음	8.0 KiB	없음
dbus-daemon	user1	0	1455	456.0 KiB	없음	없음	없음
dconf-service	user1	0	1633	664.0 KiB	112.0 KiB	164.0 KiB	없음
deja-dup-monitor	user1	0	2073	5.4 MiB	없음	4.5 MiB	없음
evolution-addressbook-factor	user1	0	1648	3.4 MiB	없음	572.0 KiB	없음
evolution-addressbook-factor	user1	0	1657	2.8 MiB	52.0 KiB	2.4 MiB	없음
evolution-calendar-factory	user1	0	1618	7.9 MiB	없음	6.8 MiB	없음
evolution-calendar-factory-sub	user1	0	1625	6.0 MiB	없음	384.0 KiB	없음

그림 6-4 시스템 감시 화면

03 포그라운드, 백그라운드 프로세스와 작업 제어

포그라운드 작업

- 포그라운드 프로세스: 사용자가 입력한 명령이 실행되어 결과가 출력될 때까지 기다려야 하는 포그라운드 방식으로 처리되는 프로세스
- 이를 작업 제어에서는 포그라운드 작업이라고 함

```
user1@myubuntu:~$ sleep 100      → 포그라운드 작업
                                 → sleep 명령이 끝날 때까지 기다려야 한다.
```

백그라운드 작업

- 백그라운드 프로세스: 명령을 실행하면 명령의 처리가 끝나는 것과 관계없이 곧바로 프롬프트가 출력되어 사용자가 다른 작업을 계속할 수 있음
- 작업 제어에서는 백그라운드 작업이라고 함

```
user1@myubuntu:~$ sleep 100 &    → 백그라운드 작업
user1@myubuntu:~$                → 프롬프트가 바로 나와 다른 명령을 실행할 수 있다.
```

백그라운드 작업과 출력 방향 전환하기

- 백그라운드로 처리할 때는 주로 출력과 오류 방향 전환을 하여 실행 결과와 오류 메시지를 파일로 저장

```
user1@myubuntu:~$ find / -name passwd > pw.dat 2>&1 &   → pw.dat에 결과와 오류를 저장한다.
[1] 52524
user1@myubuntu:~$
```

03 포그라운드, 백그라운드 프로세스와 작업 제어

작업 제어

- 작업 제어는 작업 전환과 작업 일시 중지, 작업 종료를 의미
- 작업 전환: 포그라운드 작업->백그라운드 작업, 백그라운드 작업->포그라운드 작업으로 전환
- 작업 일시 중지: 작업을 잠시 중단
- 작업 종료: 프로세스를 종료하는 것처럼 작업을 종료

작업 목록 보기 : jobs

jobs

- 기능 백그라운드 작업을 모두 보여준다. 특정 작업 번호를 지정하면 해당 작업의 정보만 보여준다.
- 형식 jobs [%작업 번호]
- %작업 번호 %번호: 해당 번호의 작업 정보를 출력한다.
  %+ 또는 %%: 작업 순서가 +인 작업 정보를 출력한다.
  %-: 작업 순서가 -인 작업 정보를 출력한다.
- 사용 예 jobs jobs %1

03 포그라운드, 백그라운드 프로세스와 작업 제어

■ jobs 명령 예

```
user1@myubuntu:~$ jobs
[1]-  실행중                  sleep 100 &
[2]+  실행중                  find / -name passwd > pw.dat 2>&1 &
user1@myubuntu:~$
```

표 6-7 jobs 명령의 출력 정보

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	실행중	작업 상태를 표시한다. • 실행중: 현재 실행 중이다. • 완료: 작업이 정상적으로 종료되었다. • 종료됨: 작업이 비정상적으로 종료되었다. • 정지됨: 작업이 잠시 중단되었다.
명령	sleep 100 &	백그라운드로 실행 중인 명령이다.

03 포그라운드, 백그라운드 프로세스와 작업 제어

■ 작업 전환하기

표 6-8 작업 전환 명령

명령	기능
Ctrl+z 또는 stop [%작업 번호]	포그라운드 작업을 정지한다(종료하는 것이 아니라 잠시 중단하는 것이다).
bg [%작업 번호]	작업 번호가 지시하는 작업을 백그라운드 작업으로 전환한다.
fg [%작업 번호]	작업 번호가 지시하는 작업을 포그라운드 작업으로 전환한다.

- 작업전환 예: 포그라운드 -> 백그라운드

```
user1@myubuntu:~$ jobs                          → 백그라운드 작업이 없다.
user1@myubuntu:~$ sleep 100                     → 포그라운드로 실행한다.
^Z                                              → Ctrl+z로 일시 정지한다.
[1]+  정지됨                 sleep 100          → 일시 정지된 상태이다.
user1@myubuntu:~$ bg %1                         → 백그라운드로 전환한다.
[1]+ sleep 100 &
user1@myubuntu:~$ jobs
[1]+  실행중                 sleep 100 &        → 백그라운드로 실행 중이다.
user1@myubuntu:~$
```

03 포그라운드, 백그라운드 프로세스와 작업 제어

작업 전환하기

- 작업전환 예: 백그라운드 -> 포그라운드

```
user1@myubuntu:~$ jobs
[1]+  실행중                  sleep 100 &
user1@myubuntu:~$ fg                              → 포그라운드로 전환한다.
sleep 100                                         → 포그라운드로 실행 중이다.
```

작업 종료하기 : Ctrl+c

- 포그라운드 작업은 Ctrl+c를 입력하면 대부분 종료

```
user1@myubuntu:~$ sleep 100                       → 포그라운드로 실행 중이다.
^C                                                → 강제 종료한다.
user1@myubuntu:~$
```

- 백그라운드 작업은 kill 명령으로 강제 종료: PID 또는 '%작업 번호'

```
user1@myubuntu:~$ sleep 100&                      → 백그라운드로 실행 중이다.
[1] 52583
user1@myubuntu:~$ kill %1                         → 강제 종료한다.
user1@myubuntu:~$
[1]+  종료됨                sleep 100             → 종료 메시지가 출력된다.
user1@myubuntu:~$
```

03 포그라운드, 백그라운드 프로세스와 작업 제어

■ 로그아웃 후에도 백그라운드 작업 계속 실행하기 : nohup

- 로그아웃한 다음에도 작업이 완료될 때까지 백그라운드 작업을 실행해야 할 경우가 있다. 이때 nohup 명령을 사용

nohup

- 기능 로그아웃한 후에도 백그라운드 작업을 계속 실행한다.
- 형식 nohup 명령&

- nohup 명령 사용 예

```
user1@user1-pc:~$ nohup find / -name passwd &
[1] 52588
user1@user1-pc:~$ nohup: 입력 무시 및 'nohup.out' 에 출력 추가
exit
```

- 다시 로그인하여 파일 내용 확인

```
user1@myubuntu:~$ more nohup.out
/usr/bin/passwd
/usr/share/lintian/overrides/passwd
/usr/share/bash-completion/completions/passwd
/usr/share/doc/passwd
find: '/tmp/systemd-private-1a79653352314b6b9781ba63a4d33c3f-fwupd.service-IfHHA
D': 허가 거부
(생략)
```

03 포그라운드, 백그라운드 프로세스와 작업 제어

로그아웃 후에도 백그라운드 작업 계속 실행하기 : nohup

- 명령 실행 시 다음 예와 같이 출력 방향 전환을 하면 nohup.out 파일을 생성하지 않고 지정한 파일에 결과와 오류 메시지를 출력

```
user1@user1-pc:~$ nohup find / -name passwd > pw.dat 2>&1 &
[1] 52600
user1@user1-pc:~$ exit
```

- 다시 로그인하여 파일 내용 확인

```
user1@myubuntu:~$ more pw.dat
nohup: 입력 무시
/usr/bin/passwd
/usr/share/lintian/overrides/passwd
/usr/share/bash-completion/completions/passwd
/usr/share/doc/passwd
find: '/tmp/systemd-private-1a79653352314b6b9781ba63a4d33c3f-fwupd.service-IfHHA
D': 허가 거부
find: '/tmp/systemd-private-1a79653352314b6b9781ba63a4d33c3f-colord.service-6yAi
Rq': 허가 거부
(생략)
```

04 작업 예약

특정한 시간에 작업을 수행하도록 예약할 수 있는 두 가지 방법

- 정해진 시간에 한 번만 수행
- 정해진 시간에 반복 수행

정해진 시간에 한 번 실행

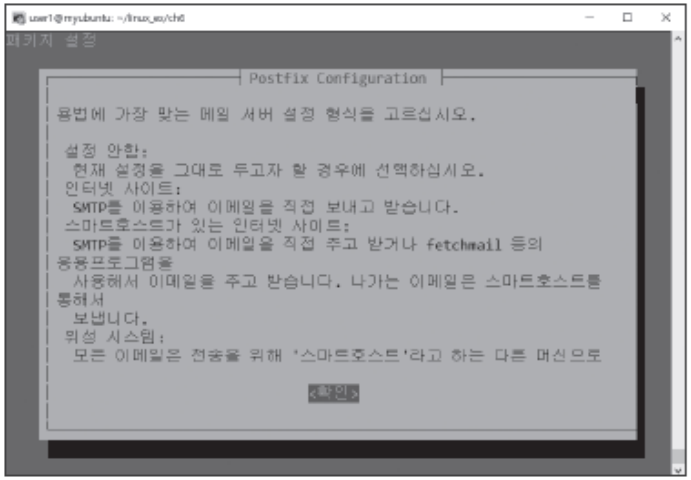
at

- **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 명령 중 해당 작업 번호를 삭제한다(atrm과 동일).
  -m: 출력 결과가 없더라고 작업이 완료되면 사용자에게 메일로 알려준다.
  -f 파일: 표준 입력 대신 실행할 명령을 파일로 지정한다.
- **사용 예**

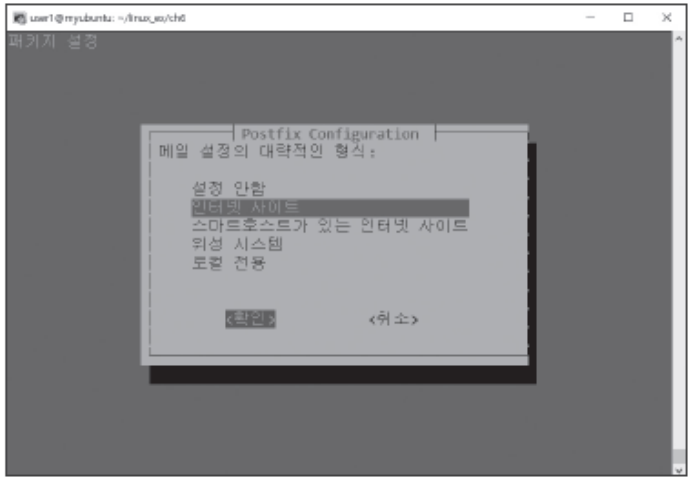
```
at 10:00 pm
at 8:15 am May 30
at -m 0730 tomorrow
```

04 작업 예약

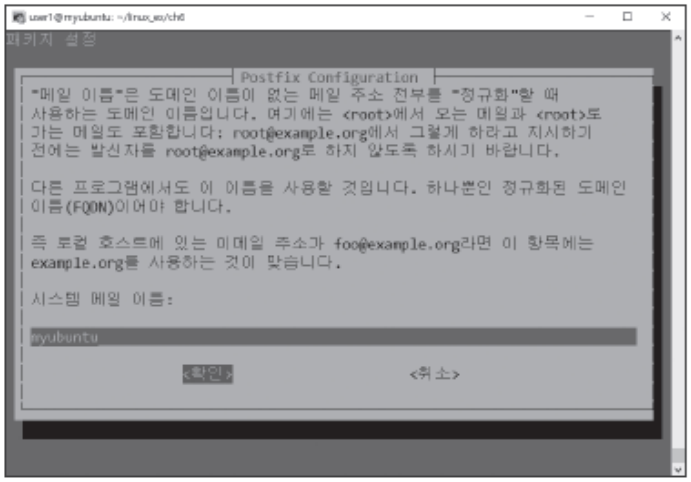
- **at 명령 설치 : sudo apt-get install at, sudo apt-get install mailutils**



(a) 메일 서버 설정 형식 선택



(b) 메일 설정의 대략적인 형식 선택



(c) 시스템 메일 이름 설정

그림 6-5 메일 설정

04 작업 예약

at 명령 설정하기

- at 명령을 사용하여 정해진 시간에 명령을 실행하도록 예약하려면 at 명령 뒤에 시간을 명시

```
user1@myubuntu:~/linux_ex/ch6$ at 07:00 pm
at>
```

- 시간을 지정하는 형식
  - at 4pm + 3 days : 지금부터 3일 후 오후 4시에 작업을 수행한다.
  - at 10am Jul 31 : 7월 31일 오전 10시에 작업을 수행한다.
  - at 1am tomorrow : 내일 오전 1시에 작업을 수행한다.
  - at 10:00am today : 오늘 오전 10시에 작업을 수행한다.
- at로 실행할 명령은 기본적으로 표준 입력으로 지정: 명령의 입력을 마치려면 ctrl+d 입력

```
user1@myubuntu:~/linux_ex/ch6$ at 07:00 pm          → 시간을 지정한다.
warning: commands will be executed using /bin/sh
at> /bin/ls  -l ~user1 > ~user1/at.out              → 실행할 명령을 지정한다.
at> <EOT>                                           → Ctrl+d로 종료한다.
job 1 at Fri Nov 17 19:00:00 2017                   → 작업 예약을 완료한다.
user1@myubuntu:~/linux_ex/ch6$
```

04 작업 예약

at 작업 파일 확인하기

- at로 생성된 작업 파일은 /var/spool/at 디렉터리에 저장

```
user1@myubuntu:~/linux_ex/ch6$ sudo ls -l /var/spool/cron/atjobs
[sudo] user1의 암호:
합계 4
-rwx------ 1 user1 daemon 2806 11월 17 16:48 a0000101803eb8
user1@myubuntu:~/linux_ex/ch6$
```

- daemon 그룹의 사용자만 /var/spool/cron/atjobs 디렉터리 내용 확인 가능

04 작업 예약

at 작업 목록 확인하기 : -l 옵션, atq

- at 명령으로 설정된 작업의 목록은 -l 옵션으로 확인

```
user1@myubuntu:~/linux_ex/ch6$ at -l
1       Fri Nov 17 19:00:00 2017 a user1
user1@myubuntu:~/linux_ex/ch6$
```

- atq 명령으로도 확인 가능

atq

- 기능 현재 사용자의 등록된 작업 목록을 보여준다. 슈퍼유저일 경우 모든 사용자의 작업 목록을 보여준다.
- 형식 atq

```
user1@myubuntu:~/linux_ex/ch6$ atq
1       Fri Nov 17 19:00:00 2017 a user1
user1@myubuntu:~/linux_ex/ch6$
```

04 작업 예약

at 작업 삭제하기 : -d 옵션, atrm

- at 명령으로 설정한 작업이 실행되기 전에 삭제하려면 -d 옵션을 사용하고 삭제할 작업 번호를 지정

atrm

- 기능 지정된 작업 번호의 작업을 삭제한다.
- 형식 atrm 작업 번호

① 작업예약

```
user1@myubuntu:~/linux_ex/ch6$ at 1am tomorrow
warning: commands will be executed using /bin/sh
at> /bin/ls > ~user1/at1.out
at> <EOT>
job 2 at Sat Nov 18 01:00:00 2017
user1@myubuntu:~/linux_ex/ch6$ at 10pm today
warning: commands will be executed using /bin/sh
at> /bin/ls /tmp > ~user1/at2.out
at> <EOT>
job 3 at Fri Nov 17 22:00:00 2017
user1@myubuntu:~/linux_ex/ch6
```

04 작업 예약

at 작업 삭제하기 : -d 옵션, atrm

- at 명령으로 설정한 작업이 실행되기 전에 삭제하려면 -d 옵션을 사용하고 삭제할 작업 번호를 지정

② 설정된 작업 확인

```
user1@myubuntu:~/linux_ex/ch6$ atq
1       Fri Nov 17 19:00:00 2017 a user1
3       Fri Nov 17 22:00:00 2017 a user1
2       Sat Nov 18 01:00:00 2017 a user1
user1@myubuntu:~/linux_ex/ch6$
```

③ 작업 삭제

```
user1@myubuntu:~/linux_ex/ch6$ at -d 2
user1@myubuntu:~/linux_ex/ch6$ atrm 3
user1@myubuntu:~/linux_ex/ch6$ atq
1       Fri Nov 17 19:00:00 2017 a user1
user1@myubuntu:~/linux_ex/ch6$
```

04 작업 예약

at 명령 사용 제한하기

- 관련된 파일: /etc/at.allow와 /etc/at.deny
- /etc/at.allow 파일과 /etc/at.deny 파일에는 한 줄에 사용자 이름을 하나씩만 기록
- /etc/at.allow 파일이 있으면 이 파일에 있는 사용자만 at 명령을 사용할 수 있다. 이 경우에 /etc/at.deny 파일은 무시된다.
- /etc/at.allow 파일이 없으면 /etc/at.deny 파일에 지정된 사용자를 제외한 모든 사용자가 at 명령을 사용할 수 있다.
- 만약 두 파일이 모두 없다면 root만 at 명령을 사용할 수 있다.
- 한 사용자가 두 파일 모두에 속해 있다면 그 사용자는 at 명령을 사용할 수 있다. /etc/at.allow 파일이 적용되기 때문이다.
- /etc/at.deny를 빈 파일로 두면 모든 사용자가 at 명령을 사용할 수 있는데, 이것이 초기 설정이다.

- at.deny 파일에 user1 사용자가 기록되어 있다면 at 명령을 실행했을 때 사용 권한이 없다는 메시지가 출력

```
user1@myubuntu:~/linux_ex/ch6$ at
You do not have permission to use at.
user1@myubuntu:~/linux_ex/ch6$
```

04 작업 예약

■ 정해진 시간에 반복 실행

crontab

- 기능 사용자의 crontab 파일을 관리한다.
- 형식 crontab [-u 사용자 ID] [옵션] [파일]
- 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

분(0~59)	시(0~23)	일(1~31)	월(1~12)	요일(0~6)	작업 내용

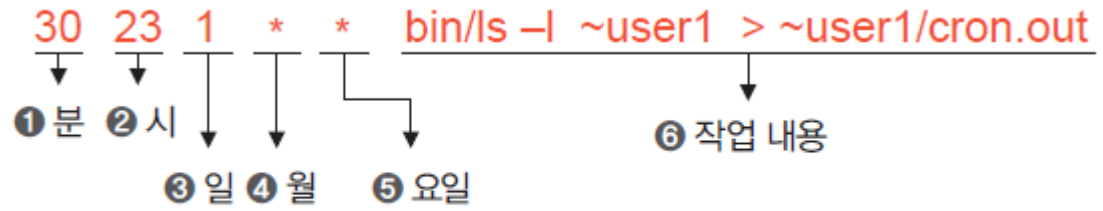


그림 6-6 crontab 파일의 형식

04 작업 예약

crontab 파일 생성하고 편집하기 : crontab –e

- crontab 편집기는 기본적으로 VISUAL 또는 EDITOR 환경 변수에 지정된 편집기를 사용

```
user1@myubuntu:~/linux_ex/ch6$ EDITOR=vi;export EDITOR
user1@myubuntu:~/linux_ex/ch6$
```

- crontab -e 명령으로 편집한 파일을 저장하면 자동적으로 /var/spool/cron/crontabs 디렉터리에 사용자 이름으로 생성

```
user1@myubuntu:~/linux_ex/ch6$ sudo ls -l /var/spool/cron/crontabs
[sudo] user1의 암호:
합계 4
-rw------- 1 user1 crontab 1137 11월 17 17:08 user1
user1@myubuntu:~/linux_ex/ch6$
```

crontab 파일 내용 확인하기 : crontab –l

```
user1@myubuntu:~/linux_ex/ch6$ crontab -l
30 23 1 * * /bin/ls -l ~user1 > ~user1/cron.out
user1@myubuntu:~/linux_ex/ch6$
```

crontab 파일 삭제하기 : crontab -r

```
user1@myubuntu:~/linux_ex/ch6$ crontab -r
user1@myubuntu:~/linux_ex/ch6$ crontab -l
no crontab for user1
user1@myubuntu:~/linux_ex/ch6$
```

04 작업 예약

crontab 명령 사용 제한하기

- /etc/cron.allow, /etc/cron.deny 파일
- cron.deny 파일은 기본적으로 있지만 cron.allow 파일은 관리자가 만들어야 함
- 두 파일이 적용되는 기준
  - /etc/cron.allow 파일이 있으면 이 파일 안에 있는 사용자만 crontab 명령을 사용할 수 있다.
  - /etc/cron.allow 파일이 없고 /etc/cron.deny 파일이 있으면 이 파일에 사용자 계정이 없어야 crontab 명령을 사용할 수 있다.
  - /etc/cron.allow 파일과 /etc/cron.deny 파일이 모두 없다면 시스템 관리자만 crontab 명령을 사용할 수 있다.
- 두 파일이 모두 없는데 일반 사용자가 crontab 명령을 사용하려고 하면 다음과 같은 메시지가 출력

```
user1@myubuntu:~/linux_ex/ch6$ crontab -e
You (user1) are not allowed to use this program (crontab)
See crontab(1) for more information
user1@myubuntu:~/linux_ex/ch6$
```