

1 프로세스의 개념

2 프로세스 관리 명령

3 포 그라운드, 백 그라운드

4 작업 예약

## 1. 프로세스의 개념

- ❖프로세스의 부모-자<u>식 관계</u>
  - 프로세스는 현재 시스템에서 실행 중인 프로그램을 뜻한다.
  - 리눅스는 기본적으로 다중 프로세스 시스템으로 여러 개의 프로세스가 동시에 실행될 수 있다.
  - 프로세스의 부모-자식 관계는 부모 프로세스가 자식 프로세스를 실행하고 자식 프로세스는 또 다른 자식 프로세스를 실행할 수 있다.
  - 리눅스 시스템을 부팅할 때 스케줄러가 실행한 프로세스인 systemd와 kthreadd 프로세스를 제외하면 모든 프로세스는 부모 프로세스를 가지고 있다.
  - 자식 프로세스는 할 일이 끝나면 부모 프로세스에 결과를 돌려주며 종료 된다.

## ❖프로세스의 번호

- 각 프로세스는 고유한 번로를 가지고 있는데 이를 PID 라고 한다. PID는 1 부터 시작하고 프로세스가 실행되면서 하나씩 증가하여 부여 된다. 리눅 스는 부팅될 때 PID 1번 systemd 프로세스와 2번 kthreadd 프로세스가 차례 로 실행된다.
- 이때 1번 프로세스는 나머지 모든 시스템 프로세스의 부모 프로세스가 되고, 2번 프로세스는 모든 스레드의 부모 프로세스가 된다.

user1@Server1:~/바탕화면\$ ls -l /sbin/init

lrwxrwxrwx 1 root root 20 8월 17 16:28 /sbin/init -> /lib/systemd/systemd



## ❖프로세스의 종류

- 데몬 프로세스
  - ▶특정 서비스를 제공하기 위해 존재하며 리눅스 커널에 의해 실행된다.
  - ▶ 평소에는 대기 상태로 있다가 서비스 요청이 들어오면 서비스를 제공한다.
- 고아 프로세스
  - ▶ 자식 프로세스는 종료되면 부모 프로세스로 돌아가는데 부모 프로세스가 먼저 종료 되면 자식 프로세스는 고아 프로세스가 된다.
  - ▶이 경우 1번 프로세스가 고아 프로세스의 새로운 부모가 되어 고아 프로세스를 종료 시킨다.
- 좀비 프로세스
  - ▶ 자식 프로세스가 종료되면 부모 프로세스는 프로세스 테이블 목록에서 삭제 하게 되는데 이때 자식 프로세스가 종료되었는데 프로세스 테이블 목록에 남아 있게 되 면 좀비 프로세스라고 한다.
  - ▶이는 부모 프로세스가 자식 프로세스의 종료 정보를 제대로 처리 하지 않아서 생긴 다.

# 2. 프로세스 관리 명령

## ❖프로세스 목록 확인

	ps
기능	현재 실행 중인 프로세스에 대한 정보를 출력한다.
형식	ps [옵션]
옵션	-e:시스템에서 실행 중인 모든 프로세스의 정보를 출력한다f: 프로세스에 대한 자세한 정보를 출력한다u [uid]: 특정 사용자에 대한 모든 프로세스의 정보를 출력한다p [pid]: pid로 지정한 특정 프로세스의 정보를 출력한다. a: 터미널에서 실행 시킨 프로세스의 정보를 출력한다. u: 프로세스 소유자 이름, CPU 사용량, 메모리 사용량 등 상세 정보를 출력한다. x:시스템에서 실행 중인 모든 프로세스의 정보를 출력한다pid [PID 목록]: 목록으로 지정한 특정 PID 정보를 출력한다.

user1@Server1:~/바탕화면\$ ps PID TTY TIME CMD 2325 pts/0 00:00:00 bash 4646 pts/0 00:00:00 ps

user1@Server1:~/바탕화면\$ ps -fUIDPIDPPIDC STIME TTYTIME CMDuser1232522990 8월23 pts/0 00:00:00 bashuser1464723250 10:55 pts/0 00:00:00 ps -f

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
С	CPU 사용량(%)	CMD	실행 되고 있는 프로그램 이름

user1@Server1:~/	/바탕화면	\$ ps a
PID TTY	STAT	TIME COMMAND
1627 tty2	Ssl+	0:00 /usr/libexec/gdm-wayland-session env GNOME_SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-sessionsession=ubuntu
1631 tty2	Sl+	0:00 /usr/libexec/gnome-session-binarysession=ubuntu
2325 pts/0	Ss	0:00 bash
4649 pts/0	R+	0:00 ps a

문자	의미	문자	의미
R	실행 중	S	인터럽트가 가능한 대기 상태
Т	작업 제어에 의해 정지된 상태	Z	좀비 프로세스
STIME	프로세스의 시작 날짜나 시간	S	세션 리더 프로세스
+	포 그라운드 프로세스 그룹	1	멀티스레드



us	user1@Server1:~/바탕화면\$ ps au									
US	ER P	ID !	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME COMMAND
us	er1 16	27	0.0	0.3	172540	6272	tty2	Ssl+	8월23	0:00 /usr/libexec/gdm-wayland-session env GNOME_SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session
us	er1 16	31	0.0	0.7	233188	13440	tty2	Sl+	8월23	0:00 /usr/libexec/gnome-session-binarysession=ubuntu
us	er1 23	25	0.0	0.2	21100	4992	pts/0	Ss	8월23	0:00 bash
us	er1 46	51	0.0	0.1	22716	3456	pts/0	R+	11:02	0:00 ps au

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



user1@Se	erver1:~/	바탕화면\$	ps -e   more
PID	TTY	TIME	CMD
1	?	00:00:01	systemd
2	?	00:00:00	kthreadd
3	?	00:00:00	rcu_gp
4	?	00:00:00	rcu_par_gp
5	?	00:00:00	slub_flushwq
6	?	00:00:00	netns
8	?	00:00:00	kworker/0:0H-events_highpri
10	?	00:00:00	mm_percpu_wq
11	?	00:00:00	rcu_tasks_kthread
12	?	00:00:00	rcu_tasks_rude_kthread
13	?	00:00:00	rcu_tasks_trace_kthread

user1@Se	rver1:~/바탕	당화면\$	ps	-ef   more	
UID	PID	PPID	C	STIME TTY	TIME CMD
root	1	0	0	8월23 ?	00:00:01 /sbin/init splash
root	2	0	0	8월23 ?	00:00:00 [kthreadd]
root	3	2	0	8월23 ?	00:00:00 [rcu_gp]
root	4	2	0	8월23 ?	00:00:00 [rcu_par_gp]
root	5	2	0	8월23 ?	00:00:00 [slub_flushwq]
root	6	2	0	8월23 ?	00:00:00 [netns]
root	8	2	0	8월23 ?	00:00:00 [kworker/0:0H-events_highpri]
root	10	2	0	8월23 ?	00:00:00 [mm_percpu_wq]
root	11	2	0	8월23 ?	00:00:00 [rcu_tasks_kthread]
root	12	2	0	8월23 ?	00:00:00 [rcu_tasks_rude_kthread]
root	13	2	0	8월23 ?	00:00:00 [rcu_tasks_trace_kthread]



user1@Se	erver1:~/	바탕화면	\$ ps	ax   more
PID	TTY	STAT	TIME	COMMAND
1	?	Ss	0:01	/sbin/init splash
2	?	S	0:00	[kthreadd]
3	?	I<	0:00	[rcu_gp]
4	?	I<	0:00	[rcu_par_gp]
5	?	I<	0:00	[slub_flushwq]
6	?	I<	0:00	[netns]
8	?	I<	0:00	[kworker/0:0H-events_highpri]
10	?	I<	0:00	[mm_percpu_wq]

user1@Serve	er1:~/l	바탕회	면\$ [	s aux	тоге				
USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME COMMAND
root	1	0.0	0.6	168088	12624	?	Ss	8월23	0:01 /sbin/init splash
root	2	0.0	0.0	0	0	?	S	8월23	0:00 [kthreadd]
root	3	0.0	0.0	0	0	?	I<	8월23	0:00 [rcu_gp]
root	4	0.0	0.0	0	0	?	I<	8월23	0:00 [rcu_par_gp]
root	5	0.0	0.0	0	0	?	I<	8월23	0:00 [slub_flushwq]
root	6	0.0	0.0	0	0	?	I<	8월23	0:00 [netns]
root	8	0.0	0.0	0	0	?	I<	8월23	0:00 [kworker/0:0H-events_highpri]
root	10	0.0	0.0	0	0	?	I<	8월23	0:00 [mm_percpu_wq]
root	11	0.0	0.0	0	0	?	I	8월23	0:00 [rcu_tasks_kthread]
root	12	0.0	0.0	0	0	?	I	8월23	0:00 [rcu_tasks_rude_kthread]
root	13	0.0	0.0	0	0	?	I	8월23	0:00 [rcu_tasks_trace_kthread]
root	14	0.0	0.0	0	0	?	S	8월23	0:00 [ksoftirqd/0]
root	15	0.0	0.0	0	0	?	I	8월23	0:02 [rcu_preempt]



```
user1@Server1:~/바탕화면$ ps -u user1
   PID TTY
                    TIME CMD
  1543 ?
                00:00:00 systemd
  1544 ?
                00:00:00 (sd-pam)
  1550 ?
                00:00:00 pipewire
  1551 ?
                00:00:00 pipewire-media-
  1552 ?
                00:00:00 pulseaudio
  1556 ?
                00:00:00 gnome-keyring-d
  1565 ?
                00:00:00 dbus-daemon
  1571 ?
                00:00:00 gvfsd
  1576 ?
                00:00:00 gvfsd-fuse
```

```
user1@Server1:~/바탕화면$ ps -fu user1
                                             TIME CMD
UID
            PID
                   PPID C STIME TTY
           1543
                      1 0 8월23 ?
                                        00:00:00 /lib/systemd/systemd --user
user1
                   1543 0 8월23 ?
           1544
                                        00:00:00 (sd-pam)
user1
                   1543 0 8월23 ?
           1550
                                        00:00:00 /usr/bin/pipewire
user1
                   1543 0 8월23 ?
           1551
                                        00:00:00 /usr/bin/pipewire-media-session
user1
                   1543 0 8월23 ?
                                        00:00:00 /usr/bin/pulseaudio --daemonize=no --log-target=journal
           1552
user1
                     1 0 8월23 ?
           1556
                                        00:00:00 /usr/bin/gnome-keyring-daemon --daemonize --login
user1
                   1543 0 8월23 ?
           1565
                                        00:00:00 /usr/bin/dbus-daemon --session --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
user1
                   1543 0 8월23 ?
                                        00:00:00 /usr/libexec/gvfsd
           1571
user1
```



```
user1@Server1:~/바탕화면$ ps -p 2325

PID TTY TIME CMD

2325 pts/0 00:00:00 bash

user1@Server1:~/바탕화면$ ps -fp 2325

UID PID PPID C STIME TTY TIME CMD

user1 2325 2299 0 8월23 pts/0 00:00:00 bash
```

## ❖특정 프로세스 정보 검색

```
user1@Server1:~/바탕화면$ ps -ef | grep bash
user1 2325 2299 0 8월23 pts/0 00:00:00 bash
user1 4699 2325 0 11:17 pts/0 00:00:00 grep --color=auto bash
```

	pgrep
기능	지정한 패턴과 일치하는 프로세스의 정보를 출력한다
형식	pgrep [옵션] [패턴]
옵션	-x: 패턴과 정확히 일치하는 프로세스의 정보를 출력한다n: 패턴을 포함하고 있는 가장 최근 프로세스의 정보를 출력한다u [사용자이름]: 특정 사용자에 대한 모든 프로세스를 출력한다l: PID와 프로세스 이름을 출력한다t term: 특정 단말기와 관련된 프로세스의 정보를 출력한다.



```
user1@Server1:~/바탕화면$ pgrep -x bash
2325
user1@Server1:~/바탕화면$ pgrep -l bash
2325 bash
user1@Server1:~/바탕화면$ ps -fp $(pgrep -x bash)
UID
          PID
               PPID C STIME TTY
                                       TIME CMD
          2325 2299 0 8월23 pts/0 00:00:00 bash
user1
user1@Server1:~/바탕화면$ ps -fp $(pgrep -u user1 bash)
UID
          PID
               PPID C STIME TTY
                                       TIME CMD
          2325 2299 0 8월23 pts/0 00:00:00 bash
user1
```

## ❖프로세스 종료

```
user1@Server1:~/바탕화면$ 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
                                             14) SIGALRM
                              13) SIGPIPE
                                                            15) SIGTERM
16) SIGSTKFLT
               17) SIGCHLD
                              18) SIGCONT
                                             19) SIGSTOP
                                                            20) SIGTSTP
21) SIGTTIN
               22) SIGTTOU
                              23) SIGURG
                                             24) SIGXCPU
                                                            25) SIGXFSZ
26) SIGVTALRM
                              28) SIGWINCH
               27) SIGPROF
                                             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
```

시그널	번호	기본 처리	의미
SIGHUP	1	종료	터미널과 연결이 끊겼을 때 발생한다.
SIGINT	2	종료	인터럽트로 사용자가 CTRL + c 를 입력하면 발생한다.
SIGQUIT	3	종료, 코어 덤프	종료 신호로 사용자가 CTRL + \ 을 입력하면 발생한다.
SIGKILL	9	종료	이 시그널을 받은 프로세스는 무시 할 수 없으며 강제로 종료한다.
SIGALRM	14	종료	알람에 의해 발생한다.
SIGTERM	15	종료	kill 명령이 보내는 기본 시그널이다.

	kill
기능	지정한 시그널을 프로세스에게 보낸다.
형식	kill [~시그널] [PID]
시그널	2 : 인터럽트 시그널을 보낸다. (CTRL + c) 9 : 프로세스를 강제로 종료한다. 15 : 프로세스와 관련된 파일들을 정리하고 종료한다.

user1@Server1:~/바탕화면\$ man ps

다른 터미널을 하나 더 열고 프로세스 번호를 확인한다.

```
user1@Server1:~/바탕화면$ ps -fp $(pgrep -x man)
UID PID PPID C STIME TTY TIME CMD
user1 4793 2325 0 11:29 pts/0 00:00:00 man ps
```

```
user1@Server1:~/바탕화면$ kill 4793
```

user1@Server1:~/바탕화면\$ ps -fp \$(pgrep -x man)

오류: 프로세스 ID 지정시 -p 옵션을 지정해야 합니다

사용법:

ps [<옵션>]



### 2개의 터미널에서 ps man 실행

```
user1@Server1:~/바탕화면$ ps -fp $(pgrep -x man)
          PID
              PPID C STIME TTY
                                   STAT
UID
                                         TIME CMD
          4821 4781 0 11:31 pts/1 S+
user1
                                         0:00 man ps
          4848 2325 0 11:31 pts/0 S+
                                         0:00 man ps
user1
user1@Server1:~/바탕화면$ pkill man
pkill: PID 108번 강제로 끝내기 실패: 명령을 허용하지 않음
user1@Server1:~/바탕화면$ pgrep -x man
user1@Server1:~/바탕화면$
```

```
user1@Server1:~/바탕화면$ ps -f
UID
            PID
                   PPID C STIME TTY
                                             TIME CMD
           2325 2299 0 8월23 pts/0 00:00:00 bash
user1
           4898 2325 0 11:37 pts/0
user1
                                       00:00:00 ps -f
user1@Server1:~/바탕화면$ sh
$ more /etc/services
# Network services, Internet style
# Updated from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml
# New ports will be added on request if they have been officially assigned
# by IANA and used in the real-world or are needed by a debian package.
# If you need a huge list of used numbers please install the nmap package.
```



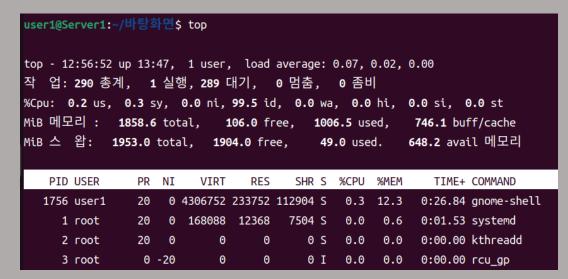
```
user1@Server1:~/바탕화면$ ps -ft pts/0
                                            TIME CMD
UID
            PID
                  PPID C STIME TTY
                  2299 0 8월23 pts/0
           2325
                                       00:00:00 bash
user1
user1
           4885
                  2325 0 11:34 pts/0
                                        00:00:00 sh
           4892
                                        00:00:00 more /etc/services
user1
                  4885 0 11:35 pts/0
user1@Server1:~/바탕화면$ ps a
   PID TTY
               STAT
                     TIME COMMAND
                      0:00 /usr/libexec/gdm-wayland-session env GNOME_SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --session=ubunt
  1627 tty2
               Ssl+
  1631 tty2
               Sl+
                      0:00 /usr/libexec/gnome-session-binary --session=ubuntu
  2325 pts/0
               Ss
                      0:00 bash
  4781 pts/1
               Ss+
                      0:00 bash
  4829 pts/1
                      0:00 pager
  4869 pts/2
               Ss
                      0:00 bash
  4885 pts/0
                      0:00 sh
  4892 pts/0
                      0:00 more /etc/services
  4895 pts/2
                      0:00 ps a
user1@Server1:~/바탕화면$ kill 4892
user1@Server1:~/바탕화면$ kill -9 4885
```

# Quiz

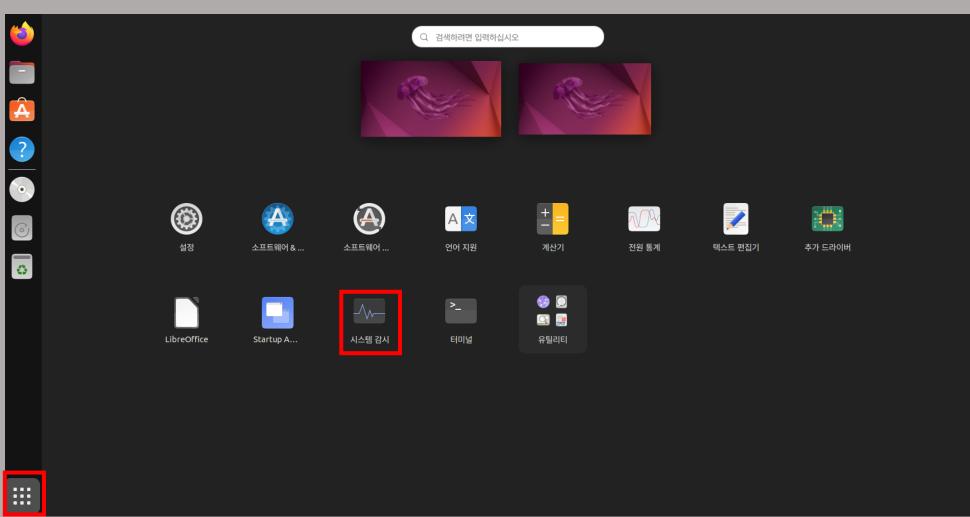
- ❖아래와 같이 실습 해본다.
  - 터미널1에서 sh 를 실행한다.
  - 터미널1에서 more /etc/services 명령을 실행한다.
  - 터미널1에서 실행한 프로세스를 터미널 0에서 찾는다.
  - 터미널1에서 실행한 more 명령을 종료한다.
  - 터미널1에서 실행한 sh 명령을 종료한다.

## ❖프로세스 관리 도구

항목	의미	항목	의미
PR	우선순위	RES	프로세스가 사용하는 메모리 크기
NI	Nice 값	SHR	프로세스가 사용하는 공유 메모리의 크기
TIME+	CPU 누적 이용 시간	VIRT	프로세스가 사용하는 가상 메모리의 크기
COMMAND	명령이름		









	<b>듣 프로세스</b>	① 자원	⊚ 파일시스템						Q = - • ×
프로세스 이름		· 사용자	% CPU	ID	메모리	총 디스크 읽기	총 디스크 쓰기	디스크 읽기	디스크 쓰기 우선 순위
(sd-pam)		user1	0.00	1544	2.0 MB	없음	없음	없음	없음 보통
		user1	0.00	2070	25.2 MB	1.8 MB	28.7 kB	없음	없음 보통
☐ at-spi-bus-launcher		user1	0.00	1746	524.3 kB	24.6 kB	없음	없음	없음 보통
☑ at-spi2-registryd		user1	0.00	1854	786.4 kB	없음	없음	없음	없음 보통
O dbus-daemon		user1	0.00	1565	2.4 MB	53.2 kB	없음	없음	없음 보통
O dbus-daemon		user1	0.00	1757	393.2 kB	8.2 kB	없음	없음	없음 보통
@ dconf-service		user1	0.00	1832	524.3 kB	81.9 kB	20.5 kB	없음	없음 보통
□ evolution-addressbook-factory		user1	0.00	1835	3.8 MB	1.9 MB	36.9 kB	없음	없음 보통
□ evolution-alarm-notify		user1	0.00	1916	17.0 MB	2.7 MB	없음	없음	없음 보통
□ evolution-calendar-factory		user1	0.00	1823	4.8 MB	5.4 MB	없음	없음	없음 보통
□ evolution-source-registry		user1	0.00	1816	4.2 MB	3.4 MB	없음	없음	없음 보통
gdm-wayland-session		user1	0.00	1627	524.3 kB	없음	없음	없음	없음 보통
gjs		user1	0.00	1855	5.0 MB	없음	없음	없음	없음 보통
gjs		user1	0.00	2112	5.1 MB	761.9 kB	없음	없음	없음 보통
gjs		user1	0.00	2206	18.4 MB	21.1 MB	없음	없음	없음 보통
gnome-keyring-daemon		user1	0.00	1556	946.2 kB	454.7 kB	4.1 kB	없음	없음 보통
gnome-session-binary		user1	0.00	1631	1.8 MB	2.0 MB	없음	없음	없음 보통
gnome-session-binary		user1	0.00	1702	1.4 MB	6.2 MB	4.1 kB	없음	없음 보통
gnome-session-ctl		user1	0.00	1685	393.2 kB	20.5 kB	없음	없음	없음 보통
<b>□</b> gnome-shell		user1	0.67	1756	126.3 MB	26.1 MB	12.3 kB	없음	없음 보통
gnome-shell-calendar-server		user1	0.00	1810	2.9 MB	4.5 MB	없음	없음	없음 보통
gnome-system-monitor		user1	0.84	4962	26.1 MB	11.2 MB	없음	없음	없음 보통
<b>⊚</b> goa-daemon		user1	0.00	1700	4.3 MB	323.6 kB	없음	없음	없음 보통
goa-identity-service		user1	0.00	1712	2.1 MB	없음	없음	없음	없음 보통
gsd-a11y-settings		user1	0.00	1873	655.4 kB	없음	없음	없음	없음 보통
☑ gsd-color		user1	0.00	1876	6.0 MB	12.3 kB	없음	없음	없음 보통
gsd-datetime		user1	0.00	1883	2.1 MB	없음	없음	없음	없음 보통
gsd-disk-utility-notify		user1	0.00	1955	1.6 MB	24.6 kB	없음	없음	없음 보통

# 3. 포 그라운드, 백 그라운드

## ❖포 그라운드 작업

■ 터미널에서 작업할 때 일반적으로 사용자가 명령을 입력하면 셸이 그 명령을 해석하여 실행하고 결과를 화면에 출력한다. 그러면 사용자는 화면에 출력된 결과를 보고 다시 명령을 입력하는 대화식으로 작업을 수행한다. 이렇게 결과가 출력될 때까지 기다리는 방식으로 처리되는 프로세스를 포 그라운드 프로세스라고 하고 작업 제어에서는 이를 포그라운드 작업이라고 한다.

## ❖백 그라운드 작업

- 포 그라운드 작업은 동시에 여러 개의 프로세스를 실행 할 수 없다.
- 백 그라운드 방식은 터미널에서 여러 개의 프로세스를 동시에 실행 할 수 있고 명령의 처리가 끝나는 것과 관계없이 다른 작업을 할 수 있도록 해준다.



■ 포그라운드 작업

user1@Server1:~/바탕화면\$ sleep 100

■ 백그라운드 작업

user1@Server1:~/바탕화면\$ sleep 100&

[1] 5084

## ■ 작업 제어

jobs				
기능	백그라운드 작업을 모두 또는 특정 작업 정보를 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.			

```
      user1@Server1:~/바탕화면$ jobs

      [1] 실행중
      find / -name passwd > pw.dat 2>&1 &

      [2]- 실행중
      sleep 100 &

      [3]+ 실행중
      sleep 100 &
```

항목	출력 예	의미
작업 번호	[1]	작업 번호로서 백그라운드로 실행할 때 마다 순차적으로 증가한다.
작업 순서	+	작업 순서를 표시한다. +: 가장 최근에 접근한 작업 -:+ 작업 바로 전에 접근한 작업 공백: 그 외의 작업
상태	실행중	작업 상태를 표시한다. 실행중: 현재 실행되고 있다. 완료: 작업이 정상적으로 종료되었다. 종료됨: 작업이 비정상적으로 종료되었다. 멈춤: 작업이 잠시 중단되었다.
명령	sleep 100 &	백그라운드로 실행 중인 명령이다.

### ■ 작업 전환 명령

명령	기능
CTRL + z 또는 stop % 작업 번호	포그라운드 작업을 중지한다. (일시정지)
bg %작업 번호	작업 번호가 지시하는 작업을 백그라운드 작업으로 전환한다.
fg %작업 번호	작업 번호가 지시하는 작업을 포그라운드 작업으로 전환한다.

```
      user1@Server1:~/바탕화면$ jobs

      user1@Server1:~/바탕화면$ sleep 100

      ^Z

      [1]+ 멈춤 sleep 100

      user1@Server1:~/바탕화면$ jobs

      [1]+ 멈춤 sleep 100

      user1@Server1:~/바탕화면$ bg %1
```

```
[1]+ sleep 100 &
user1@Server1:~/바탕화면$ jobs
[1]+ 실행중 sleep 100 &
user1@Server1:~/바탕화면$ fg %1
sleep 100
^Z
[1]+ 멈춤 sleep 100
```



■ 작업 종료

```
user1@Server1:~/바탕화면$ jobs
user1@Server1:~/바탕화면$ sleep 100
^C
user1@Server1:~/바탕화면$ sleep 100 &
[1] 5151
user1@Server1:~/바탕화면$ kill %1
user1@Server1:~/바탕화면$ jobs
[1]+ 종료됨 sleep 100
```

### ■ 로그 아웃 후에도 백그라운드 작업 계속 실행

```
        nohup

        기능
        로그아웃한 후에도 백그라운드 작업을 계속 실행한다.

        형식
        nohup 명령 &
```

```
user1@Server1:~/바탕화면$ nohup find / -name passwd > pw.dat 2>&1 & [1] 5278
user1@Server1:~/바탕화면$ exit
```

```
user1@Server1:~/바탕화면$ more pw.dat
nohup: 입력 무시
/usr/bin/passwd
/usr/share/doc/passwd
/usr/share/lintian/overrides/passwd
/usr/share/bash-completion/completions/passwd
find: '/boot/lost+found': 허가 거부
find: '/etc/cups/ssl': 허가 거부
/etc/pam.d/passwd
find: '/etc/ssl/private': 허가 거부
/etc/passwd
find: '/etc/polkit-1/localauthority': 허가 거부
find: '/proc/tty/driver': 허가 거부
```

# Quiz

- ❖/home/user1/linux\_ex/ch12 디렉터리를 만들고 아래와 같이 실습한다.
  - /etc/hosts 파일을 복사하여 현재 디렉터리에 붙여 넣기 한다.
  - vi 를 이용하여 포 그라운드로 실행한다.
  - CTRL + Z 를 이용하여 중지한다.
  - 백 그라운드 작업을 확인한다.
  - 정지 중인 백 그라운드 작업을 포그라운드로 전환한다.
  - 파일을 강제 종료 한다.

# 4. 작업 예약

## ❖정해진 시간에 한 번 실행

	at
기능	예약한 명령을 정해진 시간에 실행한다.
형식	at [옵션] [시간]
옵션	-I: 현재 실행 대기 중인 명령의 전체 목록을 출력한다.(atq 명령) -r [작업 번호]: 현재 실행 대기 중인 명령 중 해당 작업 번호를 삭제한다. (atrm 명령) -m: 출력 결과가 없더라도 작업이 완료되면 사용자에게 메일로 알려준다f [파일]: 표준 입력 대신 실행할 명령을 파일로 지정한다.
사용 예	at -m 0730 tomorrow at 12:00 pm at 6:00 am Jan 1

## user1@Server1:~/바탕화면\$ at 명령어 'at' 을(를) 찾을 수 없습니다. 그러나 다음을 통해 설치할 수 있습니다: sudo apt install at user1@Server1:~/바탕화면\$ sudo apt-get install -y at [sudo] user1 암호: 패키지 목록을 읽는 중입니다... 완료 의존성 트리를 만드는 중입니다... 완료 상태 정보를 읽는 중입니다... 완료 다음 패키지가 자동으로 설치되었지만 더 이상 필요하지 않습니다: libutempter0 'sudo apt autoremove'를 이용하여 제거하십시오.



user1@Server1:~/바탕화면\$ sudo apt-get install -y mailutils 패키지 목록을 읽는 중입니다... 완료 의존성 트리를 만드는 중입니다... 완료 상태 정보를 읽는 중입니다... 완료 다음 패키지가 자동으로 설치되었지만 더 이상 필요하지 않습니다: libutempter0 'sudo apt autoremove'를 이용하여 제거하십시오. 다음의 추가 패키지가 설치될 것입니다 : gsasl-common guile-3.0-libs libgc1 libgsasl7 libmailutils8 libmysqlclient21 libntlm0 libpg5 mailutils-common mysql-common postfix 제안하는 패키지: mailutils-mh mailutils-doc procmail postfix-mysql postfix-pgsql postfix-ldap postfix-pcre postfix-lmdb postfix-sqlite sasl2-bin | dovecot-common resolvconf postfix-cdb postfix-mta-sts-resolver postfix-doc 다음 새 패키지를 설치할 것입니다: gsasl-common guile-3.0-libs libgc1 libgsasl7 libmailutils8 libmysqlclient21 libntlm0 libpq5 mailutils mailutils-common mysql-common postfix 0개 업그레이드, 12개 새로 설치, 0개 제거 및 15개 업그레이드 안 함.



## 패키지 설정

## Postfix Configuration

Please select the mail server configuration type that best meets your needs.

No configuration:

Should be chosen to leave the current configuration unchanged.

Internet site:

Mail is sent and received directly using SMTP.

Internet with smarthost:

Mail is received directly using SMTP or by running a utility such as fetchmail. Outgoing mail is sent using a smarthost.

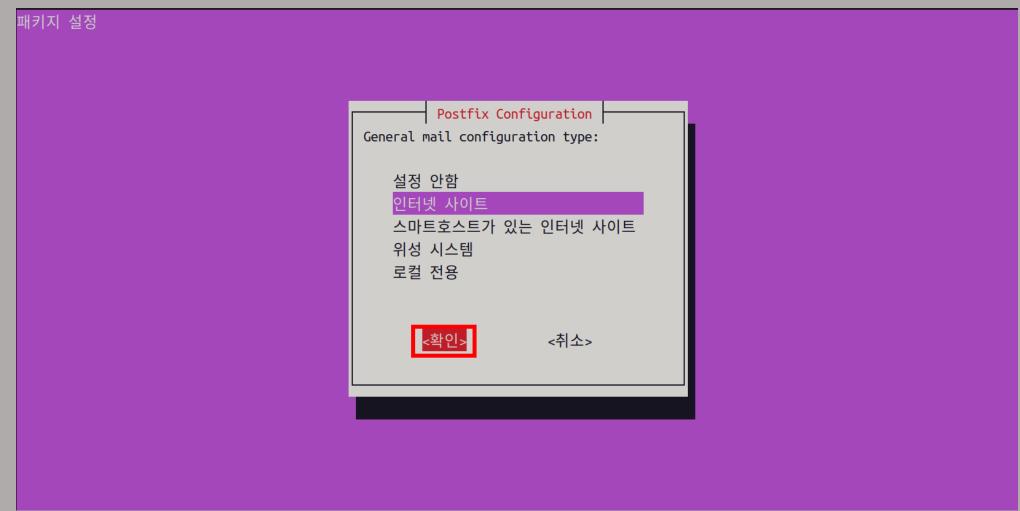
Satellite system:

All mail is sent to another machine, called a 'smarthost', for delivery.

TAB 키 이동









## 패키지 설정

## Postfix Configuration

The 'mail name' is the domain name used to 'qualify' \_ALL\_ mail addresses without a domain name. This includes mail to and from <root>: please do not make your machine send out mail from root@example.org unless root@example.org has told you to.

This name will also be used by other programs. It should be the single, fully qualified domain name (FQDN).

Thus, if a mail address on the local host is foo@example.org, the correct value for this option would be example.org.

System mail name:

Server1



<취소>

# ■ at 명령 설정하기

```
user1@Server1:~/바탕호면$ at 17:00
warning: commands will be executed using /bin/sh
at Thu Aug 24 17:00:00 2023
at> <EOT>
job 1 at Thu Aug 24 17:00:00 2023
```

at 4pm + 3 days : 현재 부터 3일 후 오후 4시 작업 수행 at 10am Jul 31 : 7월 31일 오전 10시 작업 수행 at 1am tomorrow : 내일 오전 1시 작업 수행 at 10:00am today : 오늘 오전 10시 작업 수행 저장은 CTRL + d 를 눌러 작업 내용을 저장 및 at 종료 한다.

# user1@Server1:~/바탕화면\$ at 04:30pm warning: commands will be executed using /bin/sh at Fri Aug 25 16:30:00 2023 at> ls -l Ouser1 > ~user1/at.out at> <EOT> job 2 at Fri Aug 25 16:30:00 2023

## ■ at 작업 파일 확인하기

```
user1@Server1:~/바탕화면$ ls -l /var/spool/cron/atjobs
ls: '/var/spool/cron/atjobs' 디렉터리를 열 수 없습니다: 허가 거부
user1@Server1:~/바탕화면$ sudo ls -l /var/spool/cron/atjobs
합계 16
-rwx----- 1 user1 daemon 4117 8월 24 14:34 a0000101ae8480
-rwx----- 1 user1 daemon 4146 8월 24 16:43 a0000201ae8a02
```

# ■ at 작업 목록 확인하기

```
user1@Server1:~/바탕화면$ at -l
2 Fri Aug 25 16:30:00 2023 a user1
1 Thu Aug 24 17:00:00 2023 a user1
user1@Server1:~/바탕화면$ atq
2 Fri Aug 25 16:30:00 2023 a user1
1 Thu Aug 24 17:00:00 2023 a user1
```

# ■ at 작업 삭제하기

```
user1@Server1:~/바탕화면$ at -l
2 Fri Aug 25 16:30:00 2023 a user1
1 Thu Aug 24 17:00:00 2023 a user1
user1@Server1:~/바탕화면$ at -d 2
user1@Server1:~/바탕화면$ atrm 1
user1@Server1:~/바탕화면$ at -l
```

# ❖정해진 시간에 반복 작업

crontab	
기능	사용자의 crontab 파일을 관리한다.
형식	crontab [-u 사용자ID] [옵션] [파일명]
옵션	-e : 사용자의 crontab 파일을 편집한다. -l : crontab 파일의 목록을 출력한다. -r : crontab 파일을 삭제한다.

■ crontab 파일 형식

```
# For more information see the manual pages of crontab(5) and cron(8)
# m h dom mon dow command
0 4 * * /usr/bin/ls -l ~user1 > ~user1/cron.out
                     요일(0~6)
            월(1~12)
                                            작업 내용
          일(1~31)
      시(0~23)
```

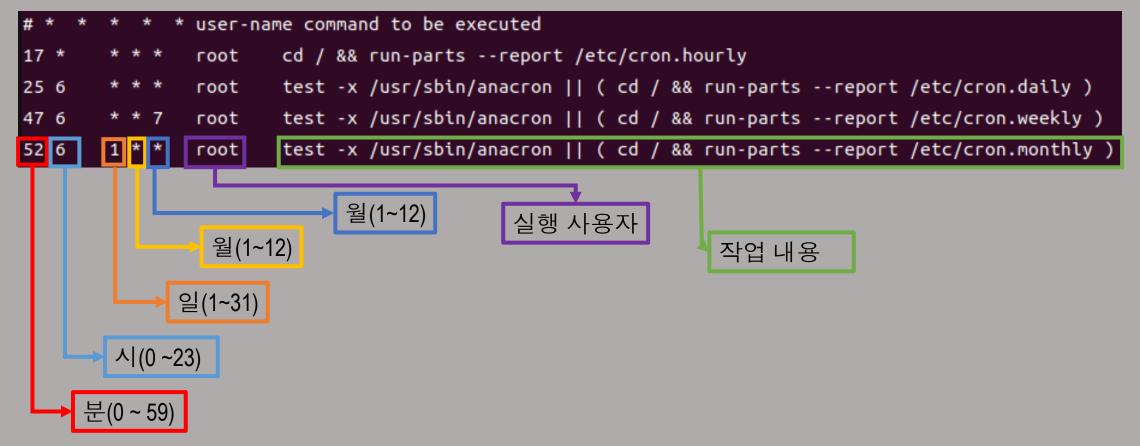
# ■ crontab 파일 생성하고 편집하기

```
user1@Server1:~/바탕화면$ EDITOR=vi;export EDITOR
user1@Server1:~/바탕화면$ crontab -e
# For more information see the manual pages of crontab(5) and cron(8)
# m h dom mon dow command
30 23 1 * * /usr/bin/ls -l -user1 > ~user1/cron.out
:wq
user1@Server1:~/바탕화면$ sudo ls -l /var/spool/cron/crontabs
[sudo] user1 암호:
합계 4
-rw----- 1 user1 crontab 1142 8월 24 17:41 user1
user1@Server1:~/바탕화면$ crontab -l
30 23 1 * * /usr/bin/ls -l -user1 > ~user1/cron.out
```

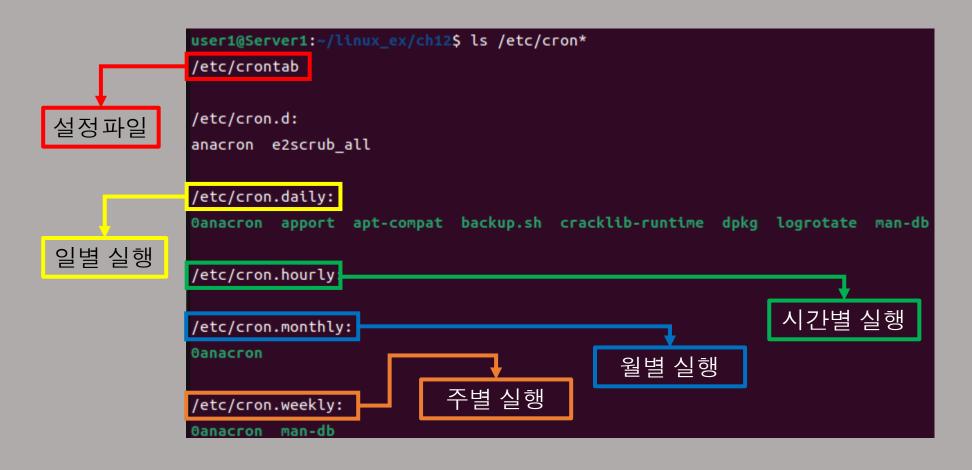
■ crontab 파일 삭제하기

```
user1@Server1:~/바탕화면$ crontab -r
user1@Server1:~/바탕화면$ crontab -l
no crontab for user1
```

# ❖관리자가 cron 사용 하기









```
user1@Server1:~$ cd /etc/cron.daily/
user1@Server1:/etc/cron.daily$ sudo mkdir /backup
user1@Server1:/etc/cron.daily$ sudo touch backup.sh
user1@Server1:/etc/cron.daily$ sudo chmod 777 backup.sh
user1@Server1:/etc/cron.daily$ sudo vi backup.sh
```

```
#! /bin/sh
set $(date)
fname="backup-$2$3tar.xz"
sudo tar cfJ /backup/$fname /etc
```

user1@Server1:/etc/cron.daily\$ sudo vi /etc/crontab

```
#
0 4 * * * root /etc/cron.daily/backup.sh
날짜와 시간은 두 칸씩 띄워야 한다.
```



```
user1@Server1:/etc/cron.daily$ sudo timedatectl set-ntp false
user1@Server1:/etc/cron.daily$ sudo date -s "2023-09-02 03:59:00"
2023. 09. 02. (토) 03:59:00 KST
user1@Server1:/etc/cron.daily$ date
2023. 09. 02. (토) 03:59:02 KST
user1@Server1:/etc/cron.daily$ sudo systemctl restart cron.service
user1@Server1:/etc/cron.daily$ ls /backup
user1@Server1:/etc/cron.daily$ date
2023. 09. 02. (토) 03:59:55 KST
user1@Server1:/etc/cron.daily$ ls /backup
user1@Server1:/etc/cron.daily$ sudo timedatectl set-ntp true
user1@Server1:/etc/cron.daily$ date
2023. 09. 02. (토) 13:29:11 KST
```

# Quiz

- ❖/home/user1/linux\_ex/ch12 디렉터리에서 아래와 같이 실습한다.
  - crontab 을 이용하여 매일 3시에 /home 디렉터리를 /backup 디렉터리에 bzip2 로 압축 저장해 보세요
  - at 을 이용하여 5분 후 /var 의 목록을 var.out 으로 저장해 보세요

