

# 데몬과 프로세스 관리 2

## kill

### [사용형식] Kill [옵션] 프로세스 ID

kill 은 특정 프로세스에게 특정 시그널(signal)을 보내는 명령어이다.

즉, 특정 프로세스에게 실행종료시그널, 재시작시그널 등, 명령어의 의미에서도 알 수 있듯이 프로세스를 종료하는 용도로 사용되는 명령어이다.

또한 슈퍼유저(root)는 모든 프로세스에 대하여 kill 명령어를 사용할 수 있으나, 일반사용자들은 자기가 실행시킨 프로세스에 대해서만 kill 명령어를 사용할 수 있다.

## 옵션

| 옵션 | 설명                            |
|----|-------------------------------|
| -s | 특별히 보낼 시그널의 이름이나 번호를 지정한다.    |
| -p | 시그널은 보내지 않고 그 프로세스의 이름만 보여준다. |
| -l | 시그널의 목록을 출력한다.                |

- 아래는 kill 명령어에서 사용할 수 있는 시그널의 종류이다.

|             |               |             |               |
|-------------|---------------|-------------|---------------|
| # 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    |                 |                 |

- 위의 예에서 모두 64 가지의 시그널은 숫자와 시그널 명으로 짝지어져 있다
- kill 명령어 사용시에 옵션으로 숫자를 사용할 수도 있고, 시그널명을 사용할 수도 있다.

```
# kill -9 PID
# kill -SIGNAL PID
```

➤ 위 두 가지는 동일한 결과를 얻는다.

- ✓ 위의 시그널리스트는 모두 운영체제가 특정 프로세스에게 보내는 인터럽트(interrupt)라고 볼 수 있다. 쉬운 예로서 윈도우 시스템에서 [CTRL]+[ALT]+[DEL] 키를 동시에 눌러서, 실행되는 "작업 관리자"에서 실행중인 특정 프로그램(프로세스)를 "프로세스 끝내기"로 강제 종료할 수 있는 것과 같은 것이다.

### kill 명령어로 특정 프로세스 죽이기(숫자 시그널 사용)

가장 흔한 용도로 사용되는 kill 명령어의 사용 방법이다. 특정 프로세스를 종료하기 위해 사용하는 것이다. 시그널번호 9 번은 특정 프로세스를 종료하는 것이다.

```
# ps -ef | grep atd
root      18400      1  0   3 월 08 ?        00:00:00 /usr/sbin/atd -f
root      20873    5173  0 00:11 pts/1    00:00:00 grep --color=auto atd

# kill -9 18400

# ps -ef | grep atd
root      20893    5173  0 00:11 pts/1    00:00:00 grep --color=auto atd
```

## kill 명령어로 특정 프로세스 죽이기(시그널 기호 사용)

---

위에서 kill 시킨 atd 프로세스를 다시 시작하고 시그널 기호를 사용하여 kill 명령을 실행한다.

```
# systemctl start atd.service
# ps -ef | grep atd
root      20934      1  0 00:13 ?          00:00:00 /usr/sbin/atd -f
root      20947    5173  0 00:13 pts/1      00:00:00 grep --color=auto atd
```

```
# kill -SIGKILL 20934
# ps -ef | grep atd
root      20977    5173  0 00:14 pts/1      00:00:00 grep --color=auto atd
# systemctl start atd.service
```

- 각 프로세스마다의 PID 는 변경된다. 즉, 한번 kill 로 서비스를 kill 시킨 이후에는 그 프로세스가 생성하는 PID 는 변경된다. /proc/숫자디렉토리 에서 확인할 수 있다!

## kill 명령어로 특정 프로세스를 재시작하는 시그널 보내기

---

이 옵션은 아파치(apache), proftpd, 삼바(samba),crond 등과 같은 응용 프로그램의 설정을 변경 후에, 프로세스에게 재 시작하는 시그널(1 번 시그널, HUP)을 보내면 특정 프로세스를 재시작하는 것과 동일한 결과를 얻을 수 있다.

```
# ps -ef | grep atd
root      21021      1  0 00:16 ?          00:00:00 /usr/sbin/atd -f
root      21036    5173  0 00:17 pts/1      00:00:00 grep --color=auto atd
# kill -HUP 21021
# ps -ef | grep atd
root      21021      1  0 00:16 ?          00:00:00 /usr/sbin/atd -f
root      21044    5173  0 00:17 pts/1      00:00:00 grep --color=auto atd
```

## kill 명령어로 segmentation fault 된 프로세스 종료하기

---

리눅스를 관리하거나 사용하다보면 아주 가끔씩 "segmentation fault"라는 에러 메시지를 확인할 수 있다. 이 에러 메시지는 메모리의 비정상적인 사용이 그 원인이며 특정 프로세스가

사용가능 영역 이외의 메모리 영역을 사용하거나, 허용되지 않는 값을 0 으로 나누는 등의 비정상적인 연산의 결과로서 메모리의 사용에 문제가 발생하였음을 알리는 메시지 입니다.

이런 경우에 "segmentaion fault"를 유발시킨 특정 프로세스를 종료하기 위해 kill 명령어로 시그널 11 번(SIGSEGV)을 보내서 종료하도록 해야 한다.

```
# ps -ef | grep crond
root      9994      1  0   3 월 08 ?        00:00:04 /usr/sbin/crond -n
root      21235    5173  0 00:22 pts/1    00:00:00 grep --color=auto crond

# kill -SIGSEGV 9994

# ps -ef | grep crond
root      21263    5173  0 00:22 pts/1    00:00:00 grep --color=auto crond

# systemctl start crond.service

# ps -ef | grep crond
root      21281      1 45 00:22 ?        00:00:01 /usr/sbin/crond -n
root      21286    5173  0 00:22 pts/1    00:00:00 grep --color=auto crond
```

## killall

killall 명령어는 명령어의 이름에서도 알 수 있듯이 특정 프로세스를 한번에 모두 종료시킬수 있는 명령어이다. 간단히 특정 프로세스를 한번에 종료시키고자 할 때 유용하게 사용한다.

## killall 명령으로 sshd 프로세스 모두 종료하기

secureCRT 로 여러개의 connection 생성 후 테스트한다.

```
# ps -ef | grep sshd
root      3550      1  0 10:51 ?        00:00:00 /usr/sbin/sshd
root      19494    3550  0 16:59 ?        00:00:00 sshd: root@pts/1
root      21702    3550  0 18:07 ?        00:00:00 sshd: root@pts/3
root      21728    3550  0 18:07 ?        00:00:00 sshd: root@pts/4
root      21806    6293  0 18:07 pts/2    00:00:00 grep sshd

# killall sshd

# ps -ef | grep sshd
root      21891    6293  0 18:10 pts/2    00:00:00 grep sshd

# systemctl start sshd.service
```

## killall 실행으로 프로세스 종료시 종료 여부 확인하기

---

killall 명령어로 프로세스를 종료할 때에 차례대로 종료 확인을 하도록 하는 방법이다.

-i 옵션(interactive)을 사용하면 된다.

secureCRT 로 여러개의 connection 생성 후 테스트한다..

```
# ps -ef | grep sshd
```

```
root      22024      1  0 18:14 ?          00:00:00 /usr/sbin/sshd
```

```
root      22025 22024  0 18:14 ?          00:00:00 sshd: root@pts/1
```

```
root      22051 22024  0 18:14 ?          00:00:00 sshd: root@pts/3
```

```
root      22122  6293  0 18:16 pts/2    00:00:00 grep sshd
```

```
# killall -i sshd
```

```
Kill sshd(22024) ? (y/N) y
```

```
Kill sshd(22025) ? (y/N) y
```

```
Kill sshd(22051) ? (y/N) y
```