
시스템 종료와 재부팅

시스템을 관리하다 보면 시스템을 종료하거나 재부팅을 해야 하는 경우가 있다.

시스템의 전원을 차단하여 종료하는 방법이 있지만, 데이터의 안전을 위하여 shutdown 이나 reboot 등의 명령어를 이용하여 정상적인 과정으로 종료하거나 재부팅 하는 것이 좋다.

shutdown

shutdown 명령어는 시스템을 종료하거나 재부팅할 때 사용한다.

<사용법>

shutdown [옵션] [시간] [메시지]

<옵션>

-t n : 경고 메시지를 보낸 후 n 초 후에 kill 시그널을 보낸다.

-h : shutdown 시 halt 를 실행하게 한다.

-n : 디스크 동기화 동작의 수행을 금지한다.

-r : 시스템을 재부팅한다.

-f : 다음 부팅 시 파일 시스템 검사를 하지 않는다.

-c : 이미 예약되어 있는 shutdown 를 취소한다. 이 옵션을 준다면 시간 인수는 줄 수 없다. 하지만 메시지는 사용자들에게 줄 수 있다.

-k : 모든 동작을 제대로 수행하지만, 실제로 시스템을 종료하지는 않는다.

shutdown 명령어를 이용하여 시스템을 종료해 보자. 가장 자주 사용되는 방식은 종료하는 명령인 shutdown -h now 이다.

셸 프롬프트에서 shutdown -h now 를 입력하면 현재 접속된 모든 사용자에게 종료된다는 메시지를 보내고 종료하게 된다

```
# shutdown -h now
```

.

✧ 예

```
# shutdown -h 1
```

1 분 후에 시스템을 종료한다.

```
# shutdown -r 22:00
```

시스템을 재부팅한다.

```
# shutdown -c
```

shutdown 명령으로 예약된 시스템 종료 명령을 취소할 수 있다.

halt

halt 역시 시스템을 종료할 때 사용한다. 시스템이 아무 응답이 없는 경우 사용하는 강제 종료 명령어이다. GUI 환경에서 시스템 종료 메뉴를 사용하지 않고 그냥 전원을 내리는 것과 같다.

<사용법>

halt [옵션]

<옵션>

-n : sync 를 하지 않고 종료한다. -d 옵션도 포함한다.

-w : 실제로 종료는 하지 않지만, /var/log/wtmp 파일에 기록을 남긴다.

-d : wtmp 파일에 기록을 남기지 않는다.

-f : 강제로 종료한다.

halt 명령을 이용해 시스템을 종료해 보자.

```
# halt
```

아무 옵션 없이 사용하면 접속되어 있는 모든 사용자에게 메시지를 보내고 /var/log/wtmp 파일에 로그를 기록하고 종료된다.

Runlevel

리눅스 시스템이 가동되는 방법에는 0~6 까지 7 가지가 있다. 이를 런레벨(RUNLEVEL)이라고 한다.

각각의 런레벨은 다음과 같다.

런레벨	영문 모드	의 미	비고
0	Power Off	종료 모드	
1	Rescue	시스템 복구 모드	단일 사용자 모드
2	Multi-User		사용하지 않음
3	Multi-User	텍스트 모드의 다중 사용자 모드	
4	Multi-User		사용하지 않음
5	Graphical	그래픽 모드의 다중 사용자 모드	
6	Reboot	재부팅 모드	

- ✓ 리눅스 설치 시 그래픽 모드로 설치를 했다면 기본 설정은 5 번으로 다중 사용자 그래픽 모드로 부팅이 된다. 텍스트 모드로 설치했다면 3 번으로 다중 사용자 텍스트 모드로 부팅이 된다.

현재 runlevel 확인

1. systemctl 명령어로 확인
systemctl get-default
graphical.target
2. ls -l /lib/systemd/system/default.target 명령으로 default target 에 연결된 파일 확인
(default.target 은 시스템에 기본으로 설정된 런레벨이 지정되어 있다.)
ls -l /lib/systemd/system/default.target
lrwxrwxrwx. 1 root root 16 3 월 3 11:03 /lib/systemd/system/default.target -> graphical.target

전체 런레벨 모드 확인

```
# ls /lib/systemd/system/runlevel*target -l
lrwxrwxrwx. 1 root root 15 3 월 3 11:03 /lib/systemd/system/runlevel0.target -> poweroff.target
lrwxrwxrwx. 1 root root 13 3 월 3 11:03 /lib/systemd/system/runlevel1.target -> rescue.target
lrwxrwxrwx. 1 root root 17 3 월 3 11:03 /lib/systemd/system/runlevel2.target -> multi-user.target
lrwxrwxrwx. 1 root root 17 3 월 3 11:03 /lib/systemd/system/runlevel3.target -> multi-user.target
lrwxrwxrwx. 1 root root 17 3 월 3 11:03 /lib/systemd/system/runlevel4.target -> multi-user.target
lrwxrwxrwx. 1 root root 16 3 월 3 11:03 /lib/systemd/system/runlevel5.target -> graphical.target
```

```
lrwxrwxrwx. 1 root root 13  3 월  3 11:03 /lib/systemd/system/runlevel6.target -> reboot.target
```

- 7 개의 runlevel*target 파일은 링크 파일이다. 각각의 링크 파일은 실제 파일과 연결되어 있다. 예를 들어 runlevel0.target 파일은 실제로는 poweroff.target 파일을 가리킨다.
- 링크 파일(Linked File)은 Windows 의 바로 가기 아이콘과 비슷한 개념이다. 즉, 실제 파일이 아니라 다른 파일을 가리키는 개념이다.

부팅 시에 텍스트 모드로 부팅되도록 런레벨 변경

```
# systemctl set-default multi-user.target
Removed symlink /etc/systemd/system/default.target.
Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/multi-user.target.
# systemctl get-default
multi-user.target
# shutdown -r
```

원복

```
# systemctl set-default graphical.target
Removed symlink /etc/systemd/system/default.target.
Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/graphical.target.
# systemctl get-default
graphical.target
# shutdown -r
```

reboot

reboot 명령은 시스템을 재부팅 할 때 사용한다.

<사용법>

reboot [옵션]

<옵션>

- n : sync 를 하지 않고 종료한다. -d 옵션도 포함한다.
- w : 실제로 재부팅은 하지 않지만, /var/log/wtmp 파일에 기록을 남긴다.
- d : wtmp 파일에 기록을 남기지 않는다.
- f : 강제로 재부팅 한다.

```
# reboot
```

로그아웃

로그인된 상태에서 시스템을 재시작하지 않고 다른 사용자로 로그인 하려면 셸프롬프트에서 `logout` 또는 `exit` 또는 `logout` 의 단축키인 `[CTRL]+D` 키를 입력한다.

로그인 상태에서 `[CTRL]+D` 단축키를 누르면 로그아웃 한다.

일반적으로 서버 작업을 위해서는 원격에서 `ssh` 로 접속하여 일반 사용자로 로그인하고 `su -` 명령어로 슈퍼유저(`root`)로 진입하여 `root` 셸을 얻고 작업을 진행한다.

작업을 완료한 다음 `root` 셸을 빠져나오기 위해서 매번 `logout` 이나 `exit` 명령어를 입력하는 것은 번거롭다. 간단히 `[CTRL]+D` 단축키를 누르면 `root` 셸을 빠져나오고, `su -` 명령을 수행했던 일반유저의 셸로 돌아간다.

다시 한번더 `[CTRL]+D` 를 누르면 `ssh` 원격 접속을 종료하게 된다.

리눅스 시스템을 관리하다보면 `logout` 입력 보다는 `[CTRL]+D` 단축키를 많이 사용하게 될 것이다.