

# 셴다운

- 컴퓨터 전원을 끄기 위해서 OS를 완전히 정지시키는 것을 셴다운이라고 함
- 운영체제에는 여러 프로그램이 백그라운드에서 돌아가기 때문에 컴퓨터 전원을 끄기 전에 반드시 운영체제를 정상적으로 종료하는 것이 좋음
- 리눅스를 셴다운하려면 shutdown 명령어를 사용
- 이 명령어를 실행하기 위해서는 슈퍼 유저(root user)로 명령어를 실행해야 함
- 실제 서버에서는 shutdown 시키는 경우가 흔하지 않으며 24시간 365일 구동되는 경우가 많음

## 셴다운 실행

```
sudo shutdown -h now
```

## 재부팅 실행

- 재부팅은 shutdown 명령어 혹은 reboot를 사용하면 재부팅할 수 있음

```
sudo shutdown -r now  
sudo reboot
```

# 파일 및 디렉토리 명령어

## pwd

- CLI환경에서 작업하다보면 현재 위치한 디렉토리 정보가 필요할 때 pwd 명령어를 사용

```
$ pwd
```

## cd

- cd 명령어는 명령어 뒤에 지정한 디렉터리로 이동함
- 이때 명령어 뒤에 스페이스로 구분하여 전달하는 문자열을 커맨드 라인 인자라고 함

## 틸드 확장

- ~(틸드)를 사용해서 홈 디렉터리로 이동

```
$ cd ~
```

- 이러한 셸의 기능을 틸드 확장

## mkdir

- mkdir 명령어를 사용하면 디렉토리를 만들수 있음

```
mkdir [옵션] <작성할 디렉터리 이름>
```

- 중간 경로의 디렉터리가 존재하지 않을 때 에러가 발생하기 때문에 이럴 경우 -p 옵션을 사용
- 아래 예제는 workspace 폴더가 없을 경우 -p 옵션을 사용하여 중간 디렉터리까지 한번에 생성하는 예제

```
$ mkdir -p ./workspace/encore
```

## rm

- 파일을 삭제할 때 rm 명령어를 사용
- 여러 파일 삭제도 가능

```
$ rm [옵션] <삭제할 파일1> <삭제할 파일2> ...
```

## 디렉터리 삭제

- rm 명령어를 사용하여 디렉터를 삭제하기 위해서는 -r 옵션을 사용

```
$ rm -rf ./workspace
```

## cp

- 파일을 복사하려면 cp 명령어를 사용

```
$ cp [옵션] <복사할 파일> <복사할 위치>
```

## 디렉토리 복사

- cp 명령어로 디렉터를 복사하려고 하면 에러가 발생
- -r 옵션을 사용하면 원본 디렉터리 안의 파일과 디렉터리가 모두 재귀적으로 복사

## mv

- 파일 위치를 옮길 때 사용하는 명령어

```
$ mv [옵션] <이동할 파일> <이동할 위치>
```

## 이름 변경하기

- mv 명령어는 파일 및 디렉터리의 위치를 이동할 때뿐만 아니라 파일 및 디렉터리 이름을 변경할 때도 사용
- test 폴더를 test2로 이름을 변경하는 예제

```
$ mv ./test ./test2
```

## torch

- 리눅스에서 파일을 만드는 방법은 여러 가지가 있는데 여기서는 torch라는 명령어를 사용
- torch는 인자로 지정한 이름의 파일을 생성

```
$ torch <생성할 파일1> <생성할 파일2>
```

- 원래 torch 명령어는 파일의 타임스탬프를 갱신하기 위한 명령어
- 이때 대상이 되는 파일이 존재하지 않으면 새롭게 파일을 생성
- 단순히 빈 파일을 만드는 용도로도 많이 사용함
- torch 명령어로 이미 존재하는 파일을 지정해도 내용이 지워지거나 하지는 않음

## ln

- ln 명령어는 링크를 생성

```
$ ln [옵션] <링크할 파일> <링크 이름>
```

## 링크란?

- 리눅스의 파일 시스템에서는 파일에 별명을 붙이는 것이 가능
- 이를 링크라고 부르며 별명을 붙이는 것을 링크를 생성한다라고 함
- 링크에 종류에는 하드 링크와 심볼릭 링크 두 종류가 존재
- 보통 심볼릭 링크가 많이 사용되며 하드 링크가 필요한 경우는 많이 없음

## 하드링크

- 하드 링크는 한 파일 원본에 이름을 여러 개 붙이는 기능
- 이렇게 만들어진 하드 링크 파일은 둘 다 원본 파일이기 때문에 어느 것이 원본이고 어느 것이 복사본인지 구분하지 않아도 됨

## find

- 파일을 찾을 때 사용하는 명령어
- 사용법

```
find / -name "test*" 2>/dev/null
```

- 2>/dev/null은 find가 접근하지 못하여 permission denied가 발생하는 내용은 출력하지 않게 해줌

## cat

- cat 명령어를 사용하면 파일 내용을 출력할 수 있음

```
$ cat [옵션] <파일 이름>
```

- 출력시 행 번호를 같이 출력하고 싶으면 옵션에 -n를 사용

```
hadoop@hadoop:~$ cat -n /etc/crontab
1 # /etc/crontab: system-wide crontab
2 # Unlike any other crontab you don't have to run the `crontab'
3 # command to install the new version when you edit this file
4 # and files in /etc/cron.d. These files also have username fields,
5 # that none of the other crontabs do.
6
7 SHELL=/bin/sh
8 # You can also override PATH, but by default, newer versions inherit
it from the environment
9 #PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
10
11 # Example of job definition:
12 # .----- minute (0 - 59)
13 # | .----- hour (0 - 23)
14 # | | .----- day of month (1 - 31)
15 # | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
16 # | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR
sun,mon,tue,wed,thu,fri,sat
```

```

17 # | | | | |
18 # * * * * * user-name command to be executed
19 17 * * * * root cd / && run-parts --report /etc/cron.hourly
20 25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-
parts --report /etc/cron.daily )
21 47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-
parts --report /etc/cron.weekly )
22 52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-
parts --report /etc/cron.monthly )

```

## 폴더 개수 세기

```
ls -l | grep ^d | wc -l
```

## 파일 개수 세기

```
ls -l | grep ^- | wc -l
```

## 하위 파일 개수 세기

```
find . -type f | wc -l
```

## 디스크 용량 확인

- df 명령어 사용하면 시스템에 마운트 된 디스크의 남은 용량을 확인하는 명령어.
- 아래와 같은 옵션과 함께 사용 가능하다.

옵션	설명
-a	모든 파일 시스템을 출력
-B [SIZE]	지정 크기를 블록 단위로 하여 용량 표시
-h	용량을 가장 보기 편한 단위로 변환하여 보여줌 (ex. 1K, 256M, 4G)
-H	용량을 1024가 아닌 1000단위로 표시
-i	inode의 남은 공간, 사용 공간, 사용 퍼센트를 출력
-k	'-B 1K'와 같은 의미
-l	출력 목록을 로컬 파일 시스템으로 제한
-P	POSIX에서 사용되는 형태로 출력
-t [TYPE]	출력 목록을 파일 시스템의 타입(TYPE)으로 제한

옵션	설명
-T	파일 시스템의 타입을 같이 출력 (ex. ext4)
-x [TYPE]	출력 목록에서 파일 시스템의 타입(TYPE)만 제외

## 디스크 사용량을 확인

- du option file

옵션	설명
-a	현재 디렉토리 아래의 모든 파일과 디렉토리의 사용 정보를 출력
-B [SIZE]	지정 크기를 블록 단위로 하여 용량 표시
-b	바이트 단위 출력
-c	출력된 의 사용량 합계 표시
-D [FILE]	입력한 심볼릭 링크 파일만 역참조(원본 파일의 용량을 출력)
-h	용량을 가장 보기 편한 단위로 변환하여 보여줌
-H	'-D' 옵션과 동일한 기능 수행
-k	'-B 1K' 옵션과 동일한 기능 수행
-l	하드 링크 파일일 경우, 여러번 용량을 카운트(원본과 하드 링크 파일 모두 카운트)
-L	모든 심볼릭 링크 파일들을 역참조
-S	서브디렉토리들은 결과에서 제외
-s	전체 사용량만을 출력
-x	다른 파일 시스템의 디렉토리는 결과에 포함하지 않음

## 예제

- 첫 단계의 하위 디렉토리의 결과를 포함하고 싶으면 다음과 같이 입력합니다.
  - du -h --max-depth=1/data/plexdrive
  - --max-depth의 설정 값에 따라 하위의 하위까지 단계 별로 확인할 수 있습니다.

```
du -h --max-depth=1 /etc
# - --max-depth를 2로 설정하는 경우 /data/plexdrive/cache와
/data/plexdrive/config의 하위 디렉토리까지 표시
```

## cpu 정보 확인

- cat /proc/cpuinfo

## cpu 코어 개수

- `grep -c processor /proc/cpuinfo`

## 물리적인 cpu 개수

- `grep "physical id" /proc/cpuinfo | sort -u | wc -l`

## cpu당 물리 코어 개수

- `grep "cpu cores" /proc/cpuinfo | tail -1`

## os 버전 확인

- `cat /etc/issue`
- `cat /etc/redhat-release`
- `cat /etc/release`

## os bit 확인

- `getconf LONG_BIT`

## 사용 중인 port 확인

- netstat 명령어 사용
- 옵션을 사용하면 현재 사용중인 포트를 어느 프로세스가 사용중인지 확인 가능
- `netstat -ntlp | grep :80`
- `netstat -tulpn | grep 포트번호`

## 프로세스 죽이기

- `kill -9 PID`
- `pkill -f python`
  - python이 들어간 모든 프로세스 죽이기

## path 추가하기

- `PATH=$PATH:대상폴더`

## tar 파일

### 1. tar로 압축하기

- `tar -cvf [파일명.tar] [폴더명]`
- ex) abc라는 폴더를 aaa.tar로 압축하고자 한다면
  - `tar -cvf aaa.tar abc`

### 2. tar 압축 풀기

- `tar -xvf [파일명.tar]`
- ex) aaa.tar라는 tar파일 압축을 풀고자 한다면
  - `tar -xvf aaa.tar`

### 3. tar.gz로 압축하기

- `tar -zcvf [파일명.tar.gz] [폴더명]`
- ex) abc라는 폴더를 aaa.tar.gz로 압축하고자 한다면
  - `tar -zcvf aaa.tar.gz abc`

### 4. tar.gz 압축 풀기

- `tar -zxvf [파일명.tar.gz]`
- ex) aaa.tar.gz라는 tar.gz파일 압축을 풀고자 한다면
  - `tar -zxvf aaa.tar.gz`

## DNS 도메인의 IP 찾기

- dig 명령어를 사용하면 DNS로 도메인 이름에 대응하는 IP를 조회할 수 있음

```
apt update
apt install dnsutils
```

## 방화벽

### 방화벽 설치

```
# centos
yum install firewalld

# ubuntu
```

### 방화벽 시작



```
##### centos #####
systemctl start firewalld
# daemon 시작 등록하면서 바로 시작하기
systemctl enable --now firewalld

# ubuntu
```

## usermod

- usermod 명령어는 user modification(사용자 수정)의 약자로 사용자 계정 정보를 수정하는 명령어입니다.
- 사용법

usermod <계정> -l <바꿀 계정>	계정명을 바꿀 계정명 으로 변경합니다. (사용 비 권장)
usermod -c <"이름"> <계정>	계정의 이름을 변경합니다.
usermod -d <"경로"> <계정>	계정의 홈 디렉터리를 변경합니다.
usermod -s <"셸"> <계정>	계정의 로그인 기본 셸을 변경합니다.
usermod -e <날짜> <계정>	계정이 해당 날짜에 만료되도록 합니다. ex) usermod -e 2018-05-01 myuser
usermod -g <그룹> <계정>	사용자의 기본 소속 그룹을 변경합니다.
usermod -G <그룹> <계정>	계정의 소속 그룹을 변경합니다. 만약 여러 그룹을 지정할 때에는 ,(кома) 로 구분하여 지정합니다.
usermod -a -G <그룹> <계정>	계정의 소속 그룹을 추가(add) 합니다.