

리눅스 명령어

하둡을 제대로 활용하려면 운영 체제인 리눅스를 잘 다루어야 한다.
이번 장에서는 리눅스의 기본적인 사용법에 대하여 숙지해보도록 한다.

윈도우에서 사용할 수 있는 기본 명령어는 다음과 같은 항목들이 있다.

기본 명령어

rm, cat, find, cp, mv, ls, pwd 등등

프롬프트

prompt란 사용자로부터 명령어를 받을 수 있도록 하는 대기 상태를 말한다.

기호	설명
!	홈 디렉토리(윈도우의 내 문서 같은 역할)를 의미한다.
#	관리자용 prompt를 의미한다.
\$	일반 사용자용 prompt 를 의미한다.
../	상위 디렉토리로 이동하기 위한 명령어이다.
./	현재 디렉토리를 의미한다.

도움말 보기

도움말을 보려면 man이라는 명령어를 사용하면 된다.
q 명령어는 "빠져 나오기" 명령어이다.
도움말에서 pageup/pagedown을 사용하면 위아래 페이지로 이동이 가능하다.

실습 : 도움말 보기

복사를 위한 명령어인 cp 명령어에 대한 도움말을 살펴 보도록 한다.

소스 코드 :

```
man cp
```

mkdir(make directory)

mkdir은 디렉토리를 생성해주는 구문이다.

실습 : 디렉토리 만들기

홈 디렉토리에 다음과 같은 구조의 디렉토리를 만들어 보고자 한다.

디렉토리 구조			
홈디렉토리	happy	fine	
		thank	
	bluesky	how	
		are	
		you	

cd # cd 명령어는 홈 디렉토리로 이동하는 명령어이다.

```
mkdir happy
mkdir happy/fine
mkdir happy/thank

mkdir bluesky
mkdir bluesky/how
mkdir bluesky/are
mkdir bluesky/you
```

실습 : 디렉토리 생성하기(-p 옵션)

-p 옵션을 사용하게 되면 상위 디렉토리가 존재하지 않더라도 바로 생성함과 동시에 하위 디렉토리도 생성해준다.
home 디렉토리 바로 아래에 peace 디렉토리(현재 존재하지 않음)에 hahaha, hohoho 디렉토리를 다음과 같이 생성해보도록 한다.

소스 코드
<pre>mkdir -p peace/hahaha mkdir -p peace/hohoho</pre>

pwd

pwd(print working directory) 명령어는 현재 디렉토리의 위치를 확인하는 명령어이다.

실습 : 현재 디렉토리 확인하기

다음과 같은 명령어를 이용하여 디렉토리를 확인해보도록 한다.

소스 코드 :
<pre>cd / # 최상위 디렉토리로 이동하기 pwd cd # 사용자 홈 디렉토리로 이동하기 pwd</pre>

cd(change directory)

cd 명령어는 디렉토리를 이동하기 위한 명령어이다.

실습 : 디렉토리 변경하기

소스 코드 :

```
cd # cd 엔터 키를 누르면 home 디렉토리로 이동한다.

cd happy # happy 디렉토리로 이동하기

cd fine # fine 디렉토리로 이동하기

cd .. # 상위 디렉토리로 이동하기

# thank 디렉토리로 이동하기
cd happy
cd thank

cd ../../ # 상위 2단계로 이동하기

cd bluesky/how # bluesky/how 디렉토리로 이동하기

cd ../are # are 디렉토리로 이동하기

cd ../you # you 디렉토리로 바로 이동하기

cd ../../happy/fine # happy/fine 디렉토리로 바로 이동하기
```

cp(copy)

파일을 복사하는 명령어이다.

cp 명령어

```
cp [option] source target # source를 target 파일명으로 복사한다.
# [option]은 옵션 사항이다.
```

cp의 여러 가지 옵션

옵션 사항은 다음과 같은 항목들이 있다.

항목	설명
-a	원래 파일의 퍼미션 보존
-b	백업 파일 생성

-f	복사할 파일명이 겹치면 강제로 복사
-R, r	하위 디렉토리까지 복사

실습 : 파일 복사해보기

다음과 같이 여러 가지 방법으로 파일을 복사해보도록 한다.

소스 코드 :

```
# 홈디렉토리 - happy - fine 디렉토리에 파일 file01을 생성한다.
# file01 내용 : vi로 abcdefg 입력 후 저장하기
cat file01

cd # 홈디렉토리로 이동한다.

# file01을 thank 디렉토리에 복사하시오.
cp happy/fine/file01 happy/thank/

# file01을 bluesky/how 디렉토리에 file02라는 이름으로 복사하시오.
cp happy/fine/file01 bluesky/how/file02

# file02를 happy/fine 디렉토리에 file03라는 이름으로 복사하시오.
cp bluesky/how/file02 happy/fine/file03

# file03을 bluesky/are 디렉토리에 file04라는 이름으로 복사하시오.
cp happy/fine/file03 bluesky/are/file04

# 참고 사항
# cp는 디렉토리 복사를 할 수 없다.
# cp /backup/ /home/

# cp는 -r 옵션을 이용하여 디렉토리를 복사한다.
# cp -r /backup/ /home/
```

현재까지의 파일 상황은 다음과 같다.

디렉토리	파일
happy-fine	file01과 file03
happy-thank	file01
bluesky-how	file02
bluesky-are	file04

ls

ls 명령어는 디렉토리 목록을 보여 주는 명령어이다.

ls 명령어

# 파일 이름	설명
# -a, --all	. 을 포함한 경로안의 모든 파일과 디렉토리를 표시해준다.
# -l, --format=long	지정한 디렉토리의 내용을 자세히 출력한다.
# -d, --directory	지정된 디렉토리의 정보를 출력한다.
# -n, --numeric	파일 및 디렉토리 정보 출력시 UID, GID를 사용한다

실습 :

소스 코드 :

```
# 파일이 존재하는 디렉토리로 이동하여 모든 파일 목록을 조회해 보도록 한다.
cd
cd happy/fine
ls
cd ../thank
# ls -al : 숨김 파일까지 모두 보여준다.
ls -al
cd ../../bluesky/how
ls -al
cd ../are
ls

# 현재 디렉토리의 하위 디렉토리까지 보여 주기
cd
cd bluesky/
ls -lR
# ls 명령어 사용시 동시에 여러 디렉토리의 내용을 볼수 있다.
ls /etc /home/hadoop/

# 다음 실습을 위하여 파일을 삭제하세요.
# happy-fine 디렉토리 : file01 파일을 삭제하세요.
cd
cd happy/fine
rm file01
```

mv(move)

mv 명령어는 파일을 이동시키거나, 이름을 변경하기 위한 명령어이다.

사용 형식은 mv [source] [target] -> source파일을 target 파일 이름으로 변경한다.

mv 실습 이전 파일 상황

```
# happy-fine 디렉토리 : file03
# happy-thank 디렉토리 : file01
# bluesky-how 디렉토리 : file02
# bluesky-are 디렉토리 : file04
```

실습 : mv 실습예보기

파일 상황이 다음과 같이 되도록 mv 명령어를 실습하도록 하세요.

소스 코드

```
# happy-fine 디렉토리 : file02
# happy-thank 디렉토리 : file04
# bluesky-you 디렉토리 : file01
# bluesky-are 디렉토리 : file03

mv happy/fine/file03 bluesky/are/file03
mv happy/thank/file01 bluesky/you/file01
mv bluesky/how/file02 happy/fine/file02
mv bluesky/are/file04 happy/thank/file04
```

실습 : 다음 실습을 위한 사전 작업

차후 실습을 수행하기 전에 다음과 같이 실습하도록 한다.

소스 코드

```
# 지울 때 home 디렉토리는 절대로 지우면 안된다.
# 지우기 : rmdir은 비어 있는 디렉토리만 삭제가 가능하다.
# 지우기 실습용 디렉토리 만들기
cd
mkdir -p a/b/c/d
cd rmdir a/b/c/d
rmdir a/b/c
rmdir a/b
rmdir a

mkdir -p a/b/c/d

# rm : remove의 약자
rm -rf a/b/c/d
rm -rf a
```

실습 : 파일 복사를 사전 실습

실습 1, 2, 3 을 풀기 전에 환경 설정을 먼저 실습해 주세요.

환경 설정(관리자 모드에서) :

```
su -  
# 비번 입력하고 ...  
mkdir /backup/  
cp /etc/inittab /backup/  
cp /etc/login.defs /backup/  
cp /boot/grub/grub.conf /backup/  
cd /home/hadoop/  
  
# /backup 디렉토리의 권한을 hadoop에 준다.  
chown -R hadoop:hadoop /backup  
exit  
cd
```

실습 : 파일 복사 실습예보기

실습 1, 2, 3 에 대하여 풀어 보세요.

소스 코드

```
실습 1 '/backup/' 안에 있는 파일들을 '/home/hadoop/' 디렉토리로 복사  
cp /backup/grub.conf /home/hadoop/grub.conf  
cp /backup/inittab ./  
cp /backup/login.defs ./login  
  
실습 2와 실습 3을 상대 경로와 절대 경로로 나누어 각각 테스트 해보세요.  
실습 2 '/home/hadoop/' 안에 있는 파일들을 '/home/hadoop/happy/thank' 디렉토리로 복사  
  
실습 3 '/backup/' 안에 있는 파일들을 '/home/hadoop/bluesky/how' 디렉토리로 복사
```

touch

파일의 시간 정보 변경 및 파일 생성을 위한 명령어이다.

즉, 크기가 0인 새로운 파일을 생성 하거나 파일이 기존에 존재하는 경우 수정 시간을 변경하는 명령어이다.

실습 : touch 명령어

소스 코드 :

```
# 원활한 실습을 수행하기 위하여 home 디렉토리로 이동한다.  
cd  
  
# testfile 파일을 생성해본다.
```

```
touch testfile
```

```
ls -al t*
```

```
# 다음과 같이 빈 파일이 생성되었다.
```

```
# -rw-rw-r--. 1 hadoop hadoop 0 Jan 20 07:53 testfile
```

head

head 명령어는 파일의 내용 중에서 상단의 처음 행부터 아래로 10줄(기본 값)을 출력해준다.

head 명령어의 반대 명령어는 tail이다.

사용 형식 :

```
# head : # 스위치 -n : 여기서 n은 숫자를 의미한다.
```

실습 : head 및 tail 명령어 사용 예시

passwd라는 파일에 대하여 내용을 확인해보도록 한다.

소스 코드 :

```
# head 명령어를 수행하기 위하여 passwd 파일을 복사한다.
```

```
cd
```

```
cp /etc/passwd /home/hadoop/
```

```
head passwd
```

```
head -5 passwd
```

```
tail passwd
```

```
tail -5 passwd
```

cat

파일의 내용을 표준 출력장치로 출력하는 명령어이다.

소스 코드 :

```
# cat은 총 5가지의 역할을 수행한다.
```

```
# 1. 파일의 내용을 출력한다. 기본 값이 (<을 사용한다.)
```

```
# 2. 파일의 내용을 입력한다. (> 을 사용한다.)
```

```
# 3. 파일을 생성한다. (> 을 사용한다.)
```

```
# 4. 파일에 내용을 추가한다. (>> 을 사용한다.)
```

```
cat passwd # 파일의 내용 출력
```


실습 : cat 명령어 실습

소스 코드

```
cd /home/hadoop
cat < /etc/passwd
cat > linux 엔터(# 기호가 사라졌다.)
hello 입력 후 엔터
다음에 Ctrl + z 으로 빠져 나온다.

cat linux

# 다음 문장은 cat linux와 동일한 결과를 출력한다.
cat < linux

# >>는 append 기능이다.
cat >> linux

cd
cd happy/thank

# 파일 병합 시키기
# 우선 파일 a를 만든다.
cat > a
1
2
3

# 파일 b를 만든다.
cat > b
4
5
6

# 2개의 파일을 병합한다.
cat a b > c
cat c
1
2
3
4
5
6
```

find

find 명령어는 파일이나 디렉토리를 검색할 때 사용하는 명령어이다.

단점	파일인지 디렉토리인지 구분이 안된다.
문법	find <경로> -name <찾고자하는이름>

실습 : find 명령어

다음과 같이 find 명령어를 실습해보도록 한다.

소스 코드 :

```
cd
find -name you
# 현 위치에서 bluesky 하단에 존재한다.
# ./bluesky/you

find -name happy
# 현 위치에서 바로 하단에 존재한다.
# ./happy

# 관리자 모드
su -
find / -name you
# /home/hadoop/bluesky/you

find / -name hadoop
# 좀 많이 나옴

# -type 옵션을 사용하는 경우 d이면 디렉토리, f 이면 파일만 찾는다.
# find / -name hadoop -type d
```

more

내용이 많은 파일을 화면 단위로 끊어서 출력한다.

명령어의 특징

스페이스 바를 누르면 page 단위로 내려간다
엔터는 1줄씩 내려 간다.
한번 내려가면 다시 위로 올라갈 수 없다.
그래서 사용 하는 게 less 명령어 이다.

실습 : more 와 less 명령어

소스 코드 :

```
more passwd
```

```
# less : 내용이 많은 파일을 화면 단위로 끊어서 출력한다.  
# 추가 사항      page를 위아래로 왔다 갔다 할 수 있다.  
# j 키를 누르면 아래로 내려 가고, k 키를 누르면 위로 올라간다.  
# 빠져 나가려면 q를 입력한다.
```

```
less passwd
```

```
# nl : 줄번호가 달려서 나온다  
nl passwd
```

```
# 명령어 조합 : ls의 결과를 less 형태로 보겠다.  
# ls -l /etc/ | less
```

```
# 문제 풀기
```

```
/etc/ 디렉토리 목록을 자세히 출력하면서 줄 번호 활성화와 재검색을 기능을 갖도록 출력해 주세요.  
# ls -l /etc/ | nl | less
```

cal

이번 달의 달력을 보여 준다.

소스 코드 :

```
cal
```

```
# 해당 년도의 달력을 보여 준다.  
cal 2014
```

Ownership/Permission

관련된 구성 요소

drwxr-xr-x	2	root	root	4096	3월 11 18:23	Desktop
-rw-----	1	root	root	1655	3월 11 06:02	anaconda-ks.cfg
-rw-r--r--	1	root	root	46643	3월 11 06:01	install.log
-rw-r--r--	1	root	root	6863	3월 11 05:59	install.log.syslog
-rw-r--r--	1	root	root	74625	3월 17 23:17	iptables.txt
-rw-r--r--	1	root	root	199	3월 11 18:06	scsrn.log

Owner/Group/Others

Owner			Group			Other		
r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1
7			7			7		

예시 : 자주 사용하는 명령어

사용 예시

```
#test 파일의 사용자 권한은 rw, 그룹의 권한은 r, 그 외 사용자의 권한은 r 로 설정 (numeric method)
chmod 644 test
```

```
# test 파일의 그룹에 읽기와 실행권한을 허가(symbolic method)
chmod g+rx test
```

```
# test 파일의 그 외 사용자에게 읽기/쓰기/실행 권한 허가
chmod o+rw 2test
```

예시 : 자주 사용하는 명령어

자주 사용되는 명령어 예제(Changing Ownership)

사용 예시

```
# test 파일의 소유자를 testuser1 으로 바꿈
chown testuser1 test
```

```
# test 파일의 소유자와 그룹을 testuser1 으로 바꿈
chown testuser1:testuser1 test
```

```
# test 파일의 소유자는 testuser1 그룹은 root 로 바꿈
chown testuser1:root test
```

```
# Group Ownership
# 자주 사용되는 명령어 예제(Changing Group Ownership)
# test 파일의 소유 그룹을 testgroup1 으로 바꿈
chgrp testgroup1 test
```