

 kim hanwoong 순서수정

4986e06 on 3 Jan

1 contributor

313 lines (234 sloc) | 7.75 KB

리눅스 명령어

리눅스 환경에서 프로그래밍을 하기전에 리눅스 명령어를 알고 있으면 코드를 이해하는데 많은 도움이 됩니다. 우리가 배우는 리눅스 명령어도 C로 짜여진 프로그램이기 때문에 명령어 이름과 용어가 라이브러리의 System관련 함수 이름들과 비슷하기 때문입니다.

cd

경로를 이동하는 명령어입니다.

- 현재 경로로 이동하는 명령은 아래와 같습니다.

```
$ cd .
```

- 상위 경로로 이동하는 명령어입니다.

```
$ cd ..
```

- 홈디렉토리로 이동하는 명령어입니다. 사용자는 홈디렉토리에서 작업을 많이 하기 때문에 자주 사용됩니다.

```
$ cd ~
```

pwd

현재 위치한 경로를 출력합니다.

```
$ pwd
```

ls

현재경로의 파일을 출력하는 명령어.

- 모든 파일(숨김파일포함)을 출력하며 list 형태로 출력합니다.

```
$ ls -al
```

- ls 결과가 한줄로 출력됩니다.(다른 스크립트와 연동하기 좋은 구조로 출력됩니다.)

```
ls -l
```

- 현재경로의 exr 파일만 출력합니다.

```
$ ls *.exr
```

- 현재경로에서 1000 프레임의 시퀀스만 검색합니다.

```
$ ls filename.1???.exr
```

touch

보통은 다음 두가지 경우에 자주 사용하는 명령어 입니다.

- 0 바이트 짜리 파일을 생성할 때 사용합니다.
- 존재하는 파일의 경우 touch 명령을 통해서 생성시간을 현재 시간으로 변경할 때 사용합니다.

```
$ touch ~/test.txt
$ ls -al ~/test.txt
```

몇초 있다가...

```
$ touch ~/test.txt
$ ls -al ~/test.txt # 파일 생성에 대한 시간이 갱신된것을 확인할 수 있습니다.
```

./ ..

`.` 문자는 현재경로라는 뜻을 가진 문자입니다. `..` 문자는 상위경로라는 뜻을 가진 문자입니다. 리눅스의 모든 경로는 언제나 현재경로와 상위 경로를 가지기 때문에 `ls -al` 명령어를 타이핑하면 항상 이 두개의 문자가 먼저 출력됩니다.

```
$ ls -al
```

```
[woong@localhost ~]$ ls -al
합계 48
drwx-----. 20 woong woong 4096 10월 27 21:11 .
drwxr-xr-x.  3 root  root   19 10월 26 02:03 ..
-rw-----.  1 woong woong  624 10월 26 20:00 .ICEauthority
-rw-----.  1 woong woong 2226 10월 27 21:15 .bash_history
-rw-r--r--.  1 woong woong   18  4월 11  2018 .bash_logout
-rw-r--r--.  1 woong woong  193  4월 11  2018 .bash_profile
-rw-r--r--.  1 woong woong  231  4월 11  2018 .bashrc
```

/

경로를 구분하는 문자입니다. `/home/username` 형태로 리눅스에서는 경로를 표기합니다. 유닉스, 리눅스는 `/` 문자로 폴더를 구분합니다. 윈도우즈는 `C:\Users\woong` 처럼 `\` 문자를 경로의 문자로 사용합니다.

연속해서 쓰일 수 있습니다. 응용 : 상위의 상위 경로로 이동하는 명령어는 아래와 같습니다.

```
$ cd ../../..
```

~

홈디렉토리 경로를 의미하는 문자입니다.

```
$ echo ~
```

cp

파일을 복사할 때 사용하는 명령어입니다. 자주 사용하는 옵션은 `-f` 이며 강제로 덮어쓰기 할 때 사용합니다.

- 홈디렉토리에 있는 test.mov 파일을 /project/review 폴더에 복사하는 예제

```
$ cp ~/test.mov /project/review
$ cp -f ~/test.mov /project/review # 강제로 덮어쓰기
```

mv

파일 또는 폴더를 이동할 때 사용합니다.

- /project/foo경로를 /backup 경로로 이동하는 예제입니다.

```
$ mv /project/foo /backup
```

mkdir

디렉토리를 만드는 명령어입니다. `-p` 옵션을 붙이면 중간에 경로가 존재하지 않더라도 중간 경로까지 만들어줍니다.

```
$ mkdir ~/project
$ mkdir -p ~/project/foo/bar
```

숨김폴더, 숨김파일 만들기

- . 문자를 파일명 또는 디렉토리명 앞에 붙이면 숨김 파일 또는 숨김 폴더가 됩니다.

- 홈 디렉토리에 숨겨진 secretdir 폴더를 생성하는 예제입니다.

```
$ mkdir ~/.secretdir
$ ls
$ ls -al ~ # ls -al 명령을 이용해야 숨김폴더가 보입니다.
```

rm

파일을 지울 때 사용합니다. 파일을 지우는 명령어는 프로그래밍 영역을 포함하여 신중히 다루어야 합니다. `-r` 옵션은 하위의 모든 경로, `-f` 는 강제의 의미를 가집니다. `-rf` 옵션을 붙이면 하위 모든 경로를 제거합니다.

```
$ rm foo
$ rm -rf ~/project/foo
```

rmdir

디렉토리를 삭제하는 명령어입니다. 내부에 파일이 존재하면 폴더가 삭제되지 않습니다. 보통 노련한 서버관리자는 실수로 생기는 책임을 위해 디렉토리를 제거할 때 rm 명령어보다는 rmdir을 사용합니다.

```
$ rmdir /project/foo
```

df

디스크의 용량을 체크하는 명령어입니다. 하드디스크의 용량이 얼마나 남아있는지 확인할 때 사용합니다.

```
$ df
$ df -h # 사람이 읽기 쉽게 단위를 변환해 줍니다.
```

du

디렉토리 경로의 용량을 체크합니다. `-h` 옵션을 붙이면 사람이 읽기 쉬운 단위로 변환해서 출력해줍니다.

- 현재 경로의 용량을 체크하는 예제입니다.

```
$ du .
$ du -h .
```

- 홈디렉토리에 있는 project 디렉토리 경로의 용량을 체크하는 예제입니다.

```
$ du -h ~/project
```

cat

파일내용을 터미널에 출력시킬 때 사용합니다.

- 홈디렉토리에 .bashrc 파일의 내용을 출력하는 예제입니다.

```
$ cat ~/.bashrc
```

more

출력 내용이 많을 때 페이지로 묶어서 출력시켜주는 명령어입니다. `|` 문자는 리눅스에서 파이프 문자입니다. `|` 문자를 기준으로 이전 명령어 결과를 다음 명령어의 input에 넣어주는 특징이 있습니다.

```
$ cat test.txt | more
```

head

파일의 앞의 10줄만 출력합니다. 많은 코드들을 상단에 코드에 대한 설명이 있기 때문에 활용하기 head 명령을 활용하기 좋습니다.

```
$ head text.txt
$ cat test.txt | head
```

tail

파일의 뒤쪽 10줄만 출력합니다. 특정파일에 log가 시간순서대로 쌓인다면 보통 맨 마지막 로그를 많이 확인하게 될 때 활용하기 좋습니다.

```
$ tail test.txt
$ cat test.txt | tail
```

which

명령어가 어느 경로에 있는지 출력해주는 툴입니다.

```
$ which ls
/usr/bin/ls
```

whereis

명령어의 위치, 맨페이지 위치를 같이 출력해주는 툴

```
$ whereis ls
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz /usr/share/man/man1p/ls.1p.gz
```

whatis

입력받은 명령어가 어떤 명령어인지 설명하는 명령어입니다.

```
$ whatis ls
ls (1)          - 경로의 내용을 나열한다.
ls (1p)         - list directory contents
```

echo

입력받은 문자를 출력(STDOUT)하는 명령어입니다.

```
$ echo "hello linux"
```

adduser

다른 사람들을 위해서 guest 계정을 만들어봅시다.

id, pw 모두 `guest` 입니다.

```
$ su
# adduser guest
# passwd guest
# exit
$ exit
```

su

리눅스는 보통 관리자가 아닌 일반 사용자로 로그인하여 사용합니다. 관리자가 꼭 필요한 경우에만 관리자로 로그인하여 사용하는 습관을 들이면 좋습니다. 관리자로 로그인하기 위해서는 su 명령어를 실행해야 합니다.

```
$ <- 사용자 커서
$ su
# <- 관리자 커서
# exit
$ <- 사용자로 돌아옵니다.
```

또한 su 명령어는 사용자를 바꿀때도 사용됩니다.

```
$ su - guest
```

테스트 시퀀스 데이터 생성

`{startnum..endnum}` 형태를 입력하면 선언된 번호 범위만큼 명령어가 실행됩니다.

```
$ mkdir -p S{1..10}/C{100..110}

or

$ touch test.{1001..1200}.jpg
```

응용 : 알파벳, 숫자 자릿수등을 설정할 수 있습니다.

```
$ mkdir test{a..z}
$ mkdir test{A..Z}
$ mkdir -p project/circle/shot/S{A..Z}{01..10}/{0010..0100}
$ touch F00_0010.{1001..1100}.jpg
```

reboot

컴퓨터를 재부팅합니다.

```
$ su
# reboot
```

shutdown now

컴퓨터를 끕니다.

```
$ su
# shutdown now
```