리눅스 기본명령어 I

● 쉘 프롬프트의 모양이 의미하는 것

[root @ localhost root]#

접속한 사용자명 호스트명 현재디렉토리위치

→ 루트권한

$ →일반사용자 권한

● 현재 작업 디렉토리는?→pwd

유닉스계의 시스템에서 여기저기 다니면서 작업을 하다보면 현재 내가 어떤 디렉토리에 있는가를 확인해야 할 경우가 있다.

이때 사용하는 명령어가 pwd 이며 현재 디렉토리의 절대 경로를 나타내 준다.

● 작업 디렉토리 변경→cd

형식 : cd [디렉토리]

"change directory"의 약어로 디렉토리를 이동하기 위한 명령어로 원하는 디렉토리로 이동하고자 할 때는"cd 디렉토리명"을 하면된다. 또한 한번에 이동하기 위해 절대경로로 지정하여 이동할수도 있다.

그냥"cd"라고 하면 자기의 홈디렉토리로 이동한다.

현재 어디에 있든지 처음에 로그인 했을 때의 위치로 자신의 홈디렉토리로 가게된다.

참고

기호	설명
. (single point)	현재 경로 의미
.. (double point)	현재 경로의 상위 경로 의미
~	현재 유저의 홈디렉토리 의미(cd)
~user 명	특정사용자 홈디렉토리 가기
-	바로 이전 디렉토리로 가기

● ls

- DOS 의 dir 과 비슷한 명령어인데, 디렉토리명 등을 출력시키며 옵션에 따라 다양한 정보와 함께 출력된다.

옵션

-a: 디렉토리 내의 모든 파일 출력

-i : 파일의 inode 와 함께 출력한다.

-l : 파일 허용 여부, 소유자,그룹,크기,날짜 등을 출력한다.

-m: 파일을 쉼표로 구분하여 가로로 출력한다.

-r : 정렬 옵션이 선택되었을 때,그 역순으로 출력한다.

-s : KB 단위의 파일 크기를 출력한다.

-t : 최근에 만들어진 파일 순서대로 출력한다.

-x : 파일 순서를 세로로 출력한다.

-F : 파일의 형태와 함께 출력한다.

출력되는 파일의 형태는 '*', '@', '|', '=' 등이며, 이것은 각각 실행 파일, 심볼릭 링크, FIFO 소켓을 나타낸다.

-R : 서브 디렉토리의 내용을 포함하여 출력한다.

-S :파일 크기가 큰 순서로 출력한다.

-U : 정렬하여 출력한다.

-1 : 라인당 한 파일씩 출력한다.

--help : 도움말을 화면상에 나타낸다.

--version : 'ls'의 파일 버전과 함께 출력한다

● 나의 홈디렉토리는?

로그인을 했을 때 처음으로 보이는 곳이 자기의 홈디렉토리이다.

리눅스에서 사용자 홈디렉토리는 /home 밑에 ID 와 같은 디렉토리명이 만들어 진다.

작업중 "cd"라고만 하면 로그인했을 때의 홈디렉토리로 이동한다.

※ 접속한 시스템에서 나 자신이 누구인가를 확인해 보는 방법이 몇가지 있다.

ID 를 확인하거나 속해있는 Group, 홈 디렉토리등에 대한 간단한 정보를 조회해 볼 수 있다.

● whoami

"시스템에 접속해 있는 나 자신이 누구인가?"를 시스템에서 확인해 보는 것으로 가장 간단한 명령이다.

● who am i

whoami 보다 좀 더 자세한 정보를 보여준다.

● id

id 는 주로 시스템에 속해있는 자신에 대한 uid 나 gid 에 대한 정보를 보여준다.

● groups

groups 는 자기 자신이 속해있는 그룹이 어떤것인가를 보여준다.

리눅스에서는 계정이 생성될 때 두가지가 자동으로 만들어 진다.

홈디렉토리(/home/peter)와 자기자신의 그룹(peter)이 그것이다.

물론 한 계정에 대한 다수의 그룹이 존재할 수 있다.

- ✧ /etc/group file 에 기술되어 있는 내용과 동일

  #cat /etc/group

● 패스워드 변경 : passwd

자기의 패스워드를 변경해 보도록 하자.

로그인 후에 어디서나 "passwd"라고만 하면 기존의 패스워드를 입력하고 새로운 패스워드를 2 회 반복입력하면 패스워드가 변경된다.

또한 root 권한을 가진 시스템 관리자는 모든 사용자의 패스워드를 변경할 수 있으며 이때는 기존 패스워드 입력없이 새로운 패스워드만 2 회 반복입력해 줌으로써 모든 사용자의 패스워드 변경이가능하다.

root 로 작업을 할 때는 기존 패스워드 입력없이 새로운 패스워드만 입력을 해주면 된다.

●날짜 확인하기→ date

시스템의 날짜를 확인하고 변경할 수 있으며, 단순히 날짜확인은 일반계정에서도 얼마든지 가능 하지만 시스템 날짜를 변경하는 것은 root 권한일 때만 가능하다.

● man 페이지 활용하기

리눅스나 유닉스시스템에는 외우기 힘들 정도로 많은 명령어가 있다.

이들명령어를 모두다 알고 사용할 수 있으면 좋겠지만 어떤 명령어가 있다는 것과 어떤 상황에서 무엇을 알고자 할 때는 어떤 명령어를 사용해야 한다는 것 정도만 알아도 시스템관리를 훌륭하게 할 수 있다.

시스템관리를 잘하는 관리자는 명령어 도움말 즉, 메뉴얼 페이지를 잘 활용한다.

명령어에 대한 도움말이나 사용법등을 확인하려면 "man 명령어"를 사용하라.

✧ 예

#man ls

● touch

파일의 정보를 변경하거나 파일을 생성한다 : 0byte .

✧ 예

#touch linux.txt

● 디렉토리 만들기 → mkdir

디렉토리를 만들기 위한 명령어이다.

형식 : mkdir [options] 디렉토리명

옵션

-m(mode) : 권한까지 설정해서 디렉토리를 만든다.

-p(parents) : 상위 디렉토리도 함께 만든다.

✧ 예

1. 최상위 디렉토리 밑에 linux 디렉토리 생성

#mkdir /linux

2. /linux 디렉토리로 이동

#cd /linux

3. /linux 디렉토리 밑에 centos/test 디렉토리를 한번에 생성

mkdir -p centos/test

4. 확인

ls -al

합계 16

drwxr-xr-x 3 root root 4096 4 월 8 00:52 .

drwxr-xr-x 29 root root 4096 4 월 8 00:51 ..

drwxr-xr-x 3 root root 4096 4 월 8 00:52 centos

ls centos/

test

5. 644 권한을 부여하여 test2 디렉토리 생성

mkdir -m 644 test2

6. 확인

ls -al

합계 20

drwxr-xr-x 4 root root 4096 4 월 8 00:55 .

drwxr-xr-x 29 root root 4096 4 월 8 00:51 ..

drwxr-xr-x 3 root root 4096 4 월 8 00:52 centos

drw-r--r-- 2 root root 4096 4 월 8 00:54 test2

● 디렉토리 지우기 → rmdir

디렉토리를 삭제하기 위한 명령어이다

✧ 예

1. 현재 경로 확인

   # pwd

   /linux → 이 경로로 이동

2. 하위 디렉토리 확인

   # ls

   centos test2

3. test2 디렉토리 삭제

   # rmdir test2

4. centos 디렉토리 삭제 시 다음 경고 메시지 확인

   # rmdir centos

   rmdir: centos: 디렉토리가 비어있지 않음

5. centos 의 하위 디렉토리 확인 (test 가 하위에 있음을 확인)

   # ls centos

   test

6. centos 하위 디렉토리인 test 디렉토리를 삭제하면서 상위 디렉토리인 centos 디렉토리도 함께 삭제

   # rmdir -p centos/test

●파일이나 디렉토리 복사하기→cp

파일을 복사해 주는 명령어로 DOS 의 copy 명령어와 같으며 사용하는 형식은 다음과 같다.

형식 : cp [options] <복사할 파일명> <복사되어 생성될 파일명>

옵션

-a : 가능한 한 원래 파일의 구조, 속성을 그대로 복사한다.

-b : 덮어쓰거나 지울 때 백업 파일을 만든다.

-d : 심볼릭 링크 파일 그대로 복사한다.(디폴트는 연결된 원래 파일을 복사함).

-f : 같은 파일명을 갖는 파일이 있을 경우,지운 후 복사한다.

-i : 같은 파일명을 갖는 파일이 있을 경우,사용자 확인후 복사한다.

-p : 원시 파일의 소유자,그룹, 허용 여부,시간 등을 그대로 복사한다.

-r : 서브 디렉토리를 포함한 모든 파일 복사한다.

-u : 복사할 파일이 구 버전일 경우만 복사한다.

-v : 복사하기 전에 각각의 파일명을 출력한다.

-x : 파일 시스템이 같을 경우만 복사한다.

-P : 원시 파일이 존재하는 디렉토리까지 포함하여 복사한다.

-R : 디렉토리를 포함하여 복사한다.

-S : 환경 변수 SIMPLE_BACKUP_SUFFIX 에 의해 지정된 백업 꼬리말로 백업 파일 생성한다

✧ 예

1. /var 디렉토리를 cp_test1 이름으로 복사 시 하위 디렉토리 때문에 복사할 수 없다는 경고 메시지 확인

```
#cd  /
# cp  /var  cp_test1
cp: omitting directory `/var'
```

2. 옵션을 사용하여 하위 디렉토리 및 파일 모두 복사

cp -r /var cp_test1

3. –a 옵션을 사용하여 /var 디렉토리를 cp_test2 이름으로 복사

cp -ar /var cp_test2

4. ls –al 명령으로 cp_test1 과 cp_test2 파일 속성 비교

#ls -al var

drwxrwxrwt. 7 root root 4096 4 월 4 17:27 tmp

ls –al cp_test1

drwxr-xr-t. 7 root root 4096 4 월 4 17:27 tmp

#ls -al cp_test2

drwxrwxrwt. 7 root root 4096 4 월 4 17:27 tmp

●파일삭제하기→rm

불필요한 파일을 삭제하기위한 명령어로"rm"이라는 것을 사용한다.

DOS 의"del"명령어와 같은 것으로 remove 의 약어이다.

형식 : rm [options] <삭제 할 파일명> →지정한 파일만을 지운다.

형식 : rm *.html → html 로 끝나는 모든 파일을 지운다.

✧ 예

1. rmdir 명령은 디렉토리를 삭제할 수 있는 명령어지만 하위 디렉토리나 파일이 있을 경우 사용할 수 없다.

rmdir cp_test1

rmdir: cp_test1: 디렉토리가 비어있지 않음

2. rm 명령의 옵션을 사용하여 디렉토리 및 그 하위 디렉토리까지 삭제한다.

rm -rf cp_test1

옵션

-f : 강제로 파일을 지울 수 있다.

-i : 지우기 전에 확인한다.

-r : 서브 디렉토리 및 파일까지 지운다.

-v : 파일을 지우기 전에 지울 파일의 이름을 나타낸다.

-R : -r 과 같다.

●파일과 디렉토리의 이동과 변경→mv

형식 : mv [options] <옮길 파일 또는 디렉토리 명> <옮길 파일 또는 디렉토리 명>

파일을 다른 파일 또는 디렉토리로 옮길 때 사용.

이 명령은 복사와 같으나 원본이 지워진다.

파일의 이름을 바꿀 때도 사용할 수 있다.

옵션

-b : 지워지기 전에 백업본을 만든다.

-f : 옮겨질 디렉토리에 존재하는 파일이 있으면 덮어쓴다.

-i : 옮겨질 디렉토리에 존재하는 파일이 있으면 확인한다.

-u : 옮겨질 디렉토리에 구 버전의 파일이 있을 경우만 옮긴다.

-v : 옮기기 전에 파일명을 출력한다.

✧ 예

1. cp_test2 디렉토리를 cp_test1 로 rename

```
# mv   cp_test2   cp_test1
```

※ 파일내용보기

파일의 내용을 볼 수 있는 명령어는 cat 과 pg 그리고 head 와 tail, strings 란 명령어가 있으며 도스의 type 명령어와 같으나 이보다 훨씬 다양하게 활용할 수가 있다.

● cat

지정한 파일의 내용을 보고자 할 때 사용하는 명령어이다.

cat 명령어는 보고자 하는 파일이 텍스트파일일 경우는 내용을 알아볼 수 있게 출력하지만, 바이너리 파일일 경우에는 내용을 볼 수는 있으나 알 수 없는 문자들로 그 내용을 알기 어렵다.

cat 파일은 2 개 이상의 파일이름이 지정되면 모든 파일이 연결되어 보여진다.

옵션

-b : 행번호를 앞에 붙여서 출력한다.(빈행은 번호를 붙이지 않는다.)

-n : 행번호를 앞에 붙여서 출력한다.(빈행도 번호를 붙인다.)

✧ 예

#cat /var/log/anaconda.log

#cat -b /var/log/anaconda.log

내용을 추가하며 파일 생성

1. cat > 파일명 → 새로운 파일을 생성하며 내용 입력하기
   내용 입력
   ....
   Ctrl + C → 내용을 모두 입력 후 마지막 라인에서 Ctrl+C 로 종료하기
2. cat >> 파일명 → 생성되어 있던 파일에 맨 마지막 라인에 내용 추가하기
   내용추가
   ....
   Ctrl + C → 내용을 모두 추가한 후 마지막 라인에서 Ctrl+C 로 종료하기

✧ 예

cat > hello
bye → 내용 입력
ctrl + c → Ctrl + C 로 종료
cat >> hello
blank → 내용 추가
ctrl + c → Ctrl + C 로 종료
cat hello
bye
blank

● more

텍스트로 작성된 파일을 화면에 페이지 단위로 출력한다.

<space>는 다음 페이지, b 는 앞 페이지, Q 는 종료이다.

✧ 예

#more /var/log/anaconda/anaconda.log

cat /var/log/anaconda/anaconda.log | more

➢ 파이프라인 [|] : 명령어와 명령어 사이에 쓰이며 각 명령어를 연결하여 사용할 수 있다.

100 행부터 출력

#more +100 /var/log/anaconda/anaconda.log

● head

head 명령어는 파일내용의 첫 부분을 기준으로 출력한다.

✧ 예

/var/log/anaconda/anaconda.log file 의 첫 5 라인만 출력

#head -5 /var/log/anaconda/anaconda.log

● tail

tail 명령어는 파일내용의 뒷부분을 출력할 때 사용하는 명령어이다.

✧ 예

숫자를 지정하지 않은 경우에는 마지막 10 행만을 보여준다.

tail /var/log/anaconda/anaconda.log

파일의 마지막 10 행만을 보여준다.

#tail -10 /var/log/anaconda/anaconda.log

● less

more 와 비슷하지만 더 확장된 명령이다. More 의 키 및 화살표, <page up>,<page down>도 작동한다.

✧ 예

#less /var/log/anaconda/anaconda.log

100 행부터 출력

#less +100 /var/log/anaconda/anaconda.log

● grep

파일 내 or 입력 값에서 특정 패턴을 검색한다.

옵션

-n : 특정 값의 행 번호를 출력하며 검색

-i : 대소문자를 가리지 않고 검색

✧ 예

1. /var/log/anaconda/anaconda.log 파일에서 log 패턴 추출(log 가 포함된 라인 추출)

   # grep log /var/log/anaconda/anaconda.log

2. /tmp 디렉토리 하위 파일명에 log 가 포함된 파일명 출력

   #ls /tmp | grep log

원하는 파일찾기

● find

형식 : find <찾을 디렉토리 경로> <찾기 옵션> <조건> <찾은 후 행할 작업>

특정 파일을 찾는 명령어이다.

<찾을 디렉토리 경로>에는 다음과 같은 것들이 있다.

. : 현재 디렉토리 이하

/ : 루트 디렉토리 이하(파일시스템전체)

~ID : 특정 ID 의 홈 디렉토리 이하

<찾기 옵션>에는 다음과 같은 것들이 있다.

-empty : 비어있는 파일

-gid n : 특정 gid 를갖는파일(n:특정 gid)

-group gname : 특정 group 에 속한 파일(gname : group 명)

-name : 지정한 형식을 갖는 파일이름

-newer : 특정 파일 이후에 생성된 파일

-perm : 특정 허가모드를 가지고 있는 파일

-uid n : 특정 uid 를갖는파일(n:특정 uid)

-used n : 최근에 n 일 이후에 변경된 파일(n:일수)

-user : 특정파일을 소유하고 있는 소유자의 파일

<찾은 후 행할 작업>에는 다음과 같은 것들이 있다.

-print : 가장많이 쓰는 옵션으로 찾은 파일을 보여준다.

-exec : 찾은 파일들에 대해 특정명령을 수행한다.

✧ 예

1. anaconda.log file 찾기

   #find /var -name anaconda.log

2. anaconda.log file 을 찾아서 more 로 출력

   # find /var -name anaconda.log -exec more {} ₩;

   ** -exec 명령어 {} ₩;

   → {} : find 로 찾은 파일들

   → ₩; : -exec 옵션 내용의 끝을 나타냄

3. anaconda.log file 을 찾아서 /tmp/findtest.txt 로 출력(복사)

   # find /var -name anaconda.log -exec more {} > /tmp/findtest.txt ₩;

4. /etc 디렉토리 하위에 확장명이 ".conf"인 파일 검색

   # find /etc/ -name *.conf

5. 현재 사용자의 홈디렉토리 하위에 허가권이 644 인 파일 검색

   # find ~ -perm 644 -exec ls -al {} ₩;

6. /var 디렉토리 하위에 파일 크기가 10KB ~ 100KB 인 파일 검색

   # find /var -size +10k -size -100k -exec ls -al {} ₩;

● which

형식 : which 실행파일명

find 가 특정 파일을 찾아주는 명령어인데 비해 which 라는 명령어는 특정 명령어의 위치가 어디인지를 찾아주는 명령어이다.

리눅스나 유닉스 등에서는 명령어의 위치를 모두 기억하기 어려우므로 이 명령어를 활용하여 찾고자하는 명령어의 위치를 확인할 수 있다.

✧ 예

#which more

● whereis

형식 : whereis 실행파일명

whereis 란 명령어도 이와 유사한 기능을 하는데 다른 점은 패스에 해당하는 모든 디렉토리를 뒤져서 해당 명령어를 찾아준다.

실행 파일 및 소스, man 페이지 파일까지 검색한다.

✧ 예

#which more

●파일의 종류 확인하기→file

리눅스에서 파일의 확장자가 특별한 의미를 갖는 것은 아니다.

그냥 편의상 다른 파일들과 구별하기 위해서 임의로 붙여놓은 것이다.

예컨데 abc.txt 라고 해서 이 파일이 꼭 텍스트파일이란 보장은 없다.

따라서 특정파일이 실제로 어떤 종류의 파일인지 확인하는 것이 필요하다.

즉, 텍스트파일인지 바이너리 파일인지 등에 대한 조사를 하여 보여주는 명령어가 file 이란명령어이다.

✧ 예

file /var/log/audit

/var/log/audit: **directory**

file /var/log/anaconda/anaconda.log

/var/log/anaconda/anaconda.log: UTF-8 Unicode text

file /var/log/wtmp

/var/log/wtmp: **data**

디스크 사용량 확인하기

● du

du 명령어는 Disk Usage 의 약어로 해당 디렉토리의 사용량을 출력해 준다.

그냥 "du"라고 하면 현재 디렉토리 이하의 개별 디렉토리별로 사용량을 체크해 주지만 "-s"(summary)을 주게 되면 전체사용량을 간략히 표현한다.

이때 표현되는 단위는 block 으로 Kbyte 단위이다.

이를 좀더 알기쉽게 표현하려면 "-h"(human-readable)라는 옵션을 사용하며 단위로 표시하여 좀 더 알기쉽게 표현해 준다.

✧ 예

```
# du /var/log
  16        /var/log/gdm
  8         /var/log/vbox
  ...
# du -s /var/log
  1312      /var/log
# du -sh /var/log
  1.3M      /var/log
```

● df

현재 시스템에서 사용중인 파일시스템 즉 마운트된 디스크 정보와 사용량을 보여준다.

```
# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        15G  4.2G   11G  28% /
devtmpfs        898M     0  898M   0% /dev
tmpfs           912M  144K  912M   1% /dev/shm
tmpfs           912M  9.5M  903M   2% /run
```

파이프(pipe), 필터(filter), 리디렉션(redirection)

파이프(pipe)

파이프란 2 개의 프로그램을 연결해주는 연결 통로를 의미한다.
'|'를 사용함(shift + ₩)

[사용 예]

# ls -l /etc | less	ls -l /etc 명령을 입력하면 파일이 너무 많아서 1 페이지에 모두 담을 수 없으므로 1 페이지씩 나눠서 보겠다는 의미이다.

필터(filter)

필터란 필요한 것만 걸러주는 명령어이다. grip, tail, wc, sort, awk, sed 명령어 등이 있다.
주로 파이프와 같이 사용된다.

[사용 예]

# ps -ef | grep hash	ps -ef 명령을 입력하면 모든 프로세스 번호를 출력하므로, bash 라는 글자가 들어간 프로세스만 출력하게 한다.

리디렉션(redirection)

리디렉션은 표준 입출력의 방향을 바꿔준다. 표준 입력은 키보드, 표준 출력은 모니터이지만 이를 파일로 처리하고 싶을 때 주로 사용한다.

[사용 예]

# ls -l > list.txt	ls -l 명령의 결과를 화면에 출력하지 말고, list.txt 파일에 저장하도록 한다. 만약 list.txt 파일이 기존에 있으면 덮어(overwrite)쓴다.
# ls -l >> liste.txt	위와 같다. 단, list.txt 파일이 기존에 있으면 기존의 내용에 이어서(append) 쓴다.
# sort < list.txt	list.txt 파일을 정렬해서 화면에 출력한다.
# sort < list.txt > out.txt	list.txt 파일을 정렬해서 out.txt 파일에 쓴다.