

Linux Commands

Learning the Basic Linux Commands

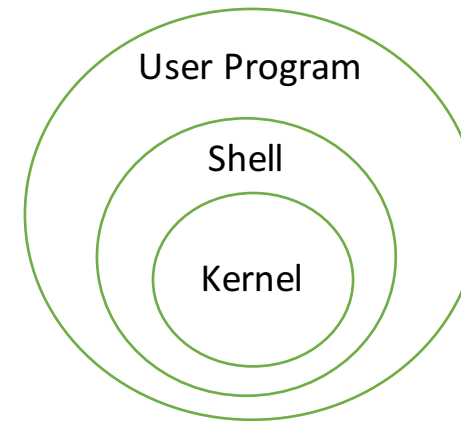
Shell & Terminal

- 셸(Shell)

- 사용자(User)와 운영체제 내부 커널(Kernel) 사이에 존재하는 층
- 사용자가 명령어를 치면 셸이 이를 번역하여 커널에 전달한다.
 - 리눅스 bash 셸 (Bourne Again Shell)
 - 윈도우 cmd

- 터미널(Terminal)

- 시스템에 접속하여 입출력을 제어하는 단말장치
- 터미널 에뮬레이터 실행
 - [Ctrl] + [Alt] + [t]



Linux Basic Commands

- 리눅스에서 기본적으로 제공해주는 명령어들
- 터미널 상에서 명령을 수행하면 셸이 해석하여 적절한 명령을 수행.

ls	show list
cd	change directory
mkdir	make directory
rmdir	remove directory
cp	copy file
rm	remove file
mv	rename(move) file

ls

- ls
 - 현재 경로의 파일 & 디렉토리 목록을 출력한다.
 - 옵션

ls -a	숨긴 파일 출력
ls -l	파일 정보 출력
ls -al	숨긴 파일 + 파일 정보 출력

- -a : 숨긴 파일은 .으로 시작한다.
- -l : drwxr-xr-x에서 맨 처음 d는 디렉토리를 의미한다. (-는 파일)

```
hyeongwon@hyeongwon:~/test$ ls -al
합계 12
drwxrwxr-x  2 hyeongwon hyeongwon 4096  9월  4 23:28 .
drwxr-xr-x 15 hyeongwon hyeongwon 4096  9월  4 23:28 ..
-rw-rw-r--  1 hyeongwon hyeongwon   0  9월  4 23:27 .gitignore
-rw-rw-r--  1 hyeongwon hyeongwon  82  9월  4 23:28 test.c
```

mkdir, rmdir, cd

- mkdir [directory name]
 - directory 생성
 - Ex) \$ mkdir Project
- rmdir [directory name]
 - directory 삭제 (비어있어야 한다)
 - Ex) \$ rmdir Project
- cd [directory name]
 - 해당 directory(경로)로 이동
 - Ex) \$ cd Project : Project 로 이동
 - \$ cd .. : 상위 directory 로 이동

cp, rm, mv

- cp [target file] [copy file]
 - file 복사
 - Ex) \$ cp test.txt test_copy.txt
- rm [target file]
 - file 삭제
 - Ex) \$ rm test.txt
- mv [target file] [renamed file/path]
 - file 이름변경 또는 이동
 - Ex) \$ mv text.txt test.txt : 파일 이름 변경
 - \$ mv test.txt .. : 상위 directory에 test.txt 파일 이동

Etc

- 그밖에 리눅스에서는 많은 명령어를 제공한다.
- 참고 :
 - <http://ss64.com/bash/>
 - http://www.mireene.com/webimg/linux_tip1.htm

pwd	현재 경로 위치를 출력한다.
whereis	찾고자 하는 파일의 위치를 출력한다.
who	현재 접속한 사용자 리스트를 출력한다.
touch	빈 파일을 생성한다.
cat	파일의 내용을 출력한다.
tar	파일 압축 관련
passwd	비밀번호 변경
chmod	파일의 권한 변경

Vim Editor

Learning the usage of vim editor

Vim Editor

- vi
 - 1976년 Bill Joy가 만든 UNIX 계열의 Visual Text Editor.
 - 세 가지 모드를 지원.
 - 터미널에서 간편히 사용 가능.
 - 사용법을 숙달 시 강력한 문서 편집기능.
 - (cf) vi와 양대산맥인 Text Editor로 Emacs가 있다.
- Vim
 - Vi Improved의 축약어로, vi의 확장판 개념.
 - undo, split windows, syntax coloring, #Vim scripts 등 지원.
 - 현재 오리지널 vi는 거의 사용되지 않고,
Linux 배포판에는 Vim이 포함되어 있다.

Vim Install

- 터미널 실행 : [Ctrl] + [Alt] + [t]
- \$ sudo apt-get update
- \$ sudo apt-get upgrade
- \$ sudo apt-get install vim

3 Modes of Vim

1. 명령 모드

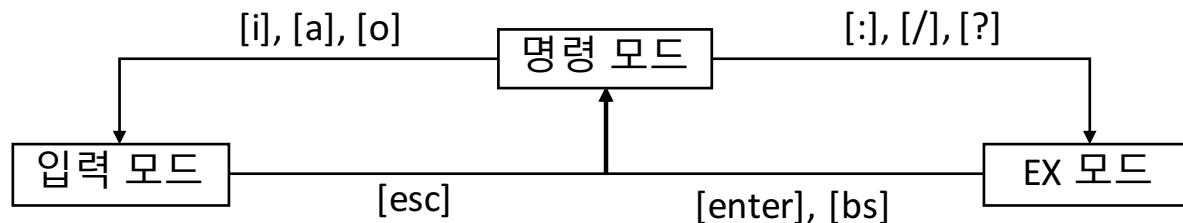
- Vim 을 실행시켰을 때 기본 모드
- 여러가지 명령(커서이동, 행 복사, 삭제 등)들을 수행할 수 있다.

2. 입력모드

- 문서를 편집하는 모드
- 명령 모드에서 [i], [o], [a] 입력

3. EX 모드

- 찾기/치환 혹은 파일 저장, 종료 등을 수행할 수 있다.
- 명령 모드에서 [:], [/], [?] 입력



Practice 1

- Vim 을 이용해서 다음 test.txt 를 작성해본다.
 1. 터미널 에서 vim test.txt 로 파일 생성
 2. 입력 모드로 전환 : [i]
 3. 다음 <예제>의 소스를 작성.
 4. 작성 후 저장 및 종료 : [:wq]

<예제>

This is the first vim example file.
The default mode is command mode.
We can write files by typing [i] from command mode.
[i] means insert.
We can return mode by typing [esc].
We can save and close vim by typing [:wq]
[w] means write and [q] means quit.

Command Mode

- Vim 명령모드에서 지원하는 명령어

h, j, k, l	커서 ←, ↓, ↑, → 이동
0	줄의 맨 처음으로 이동
^	줄의 맨 처음으로(공백 제외)
\$	줄의 맨 마지막으로 이동
gg	문서의 처음으로 이동
Shift + g	문서의 마지막으로 이동
e, E	다음 단어의 끝으로
w, W	다음 단어의 처음으로

i, a, o	입력 모드로 전환
yy	1줄 복사
y5y	5줄 복사
dd	1줄 복사 후 삭제(잘라내기)
d5d	5줄 복사 후 삭제(잘라내기)
p	복사된 줄 다음 줄부터 붙이기
u	작업 취소
v	비주얼 모드 (다중 라인 처리)

EX Mode

- Vim EX 모드에서 지원하는 명령어

:w	저장
:q	종료
:q!	강제 종료
:wq	저장 후 종료
:5	5번째 줄로 이동
:%s/[old]/[new]/g	문서 전체 old 단어를 new로 치환
:%s/[old]/[new]/gc	문서 전체 old 단어를 new로 하나씩 치환

:split [file]	file을 가로 분할로 연다.
:vs [file]	file을 세로 분할로 연다.
:20vs [file]	file을 세로 분할로 20 칸 만큼 할당 후 연다.
Ctrl + w + 방향키	분할 된 창 이동
비주얼 모드 + :norm i//	비주얼 모드에서 선택된 줄 만큼 맨 앞에 // 삽입 (주석처리)
비주얼 모드 + :norm 2x	비주얼 모드에서 선택된 줄 만큼 맨 앞에서 2 단어 삭제 (주석 삭제)

Practice 2

- 이전에 만든 test.txt 를 수정해본다.
 1. 터미널 에서 vim test.txt 로 파일 열기.
 2. [:4] [dd] [Shift + g] [k] [p] 를 차례로 수행.
 3. [u]를 이용해 작업 취소를 수행.
 4. [:wq!]를 통해 저장 후 강제종료.

<실행 결과>

This is the first vim example file.
The default mode is command mode.
We can write files by typing [i] from command mode.
We can return mode by typing [esc].
We can save and close vim by typing [:wq]
[i] means insert.
[w] means write and [q] means quit.

Practice 3

- Vim을 이용하여 다음과 같은 소스 파일을 작성한다.

1. 터미널에서 vim main.c 실행
2. [i] 입력후 <예제> 작성
3. 작성 후 [:wq]로 저장 후 종료

```
<예제>

#include <stdio.h>

int main(void) {
    printf("Hello World\n");
    return 0;
}
```


vimrc 설정

- vimrc 파일을 수정하여 Vim Editor 설정을 변경할 수 있다.
 - 터미널에서 vim ~/.vimrc 을 다음과 같이 작성한다.

```
set number           "Show line number"
set autoindent        "Auto indent"
set si               "Smart indent"
set cindent          "C style indent"
set shiftwidth=4      "Set auto indent width 4 spaces"
set tabstop=4         "Set tab width 4 spaces"
set expandtab         "Set tab to spaces"
set ignorecase       "No classify capital/small characters"
set hlsearch         "Highlight searched characters"
set nocompatible     "Activate arrow keys"
set background=dark  "Highlight dark/light"
set fileencodings=utf-8,euc-kr
set bs=indent,eol,start "Activate backspace key"
set history=1000      "Record upto 1000 cmds"
set ruler            "Show cursor position"
set nobackup         "No backup file"
set title           "Show title"
set showmatch       "Show matched brackets"
set nowrap
set wmnu

syntax on
```

vimrc 적용

- vimrc 변경 전

```
#include <stdio.h>

int main(void) {
    printf("Hello World\n");
    return 0;
}
```



- vimrc 변경 후

```
1 #include <stdio.h>
2
3 int main(void) {
4     printf("Hello World\n");
5     return 0;
6 }
```

- C언어 문법 Coloring 적용.
- tab이 space로 인식.
- tab 간격은 4칸.

수고하셨습니다.
