

# 문서 편집기 – VIM (vi improved)

## 리눅스, 유닉스 의 대표적인 문서 편집기

- vi, emacs
- emacs는 vi 에 비해서 상대적으로 복잡하여 비 선호

## vi (Visual display editor)

- vi 의 여러가지 클론 중 요즘 대부분의 배포판에는 vim 이 포함 되어 있음
- vim (vi improved)
  - 완벽하게 한글을 지원, 원래의 vi의 기능을 충실하게 갖고 있을 뿐만 아니라 여러 가지 좀더 편리한 툴들을 제공

# VIM

## 실행과 종료, 파일 저장

- 실행 : 셸 프롬프트에서 vi<CR> 입력
- 종료
  - :q<CR>
  - :w [file]<CR>
  - :wq<CR>

## 편집 모드의 변경

- vi 는 '명령 모드'와 '입력 모드' 로 구분되며 기본적으로는 명령 모드
- 입력모드로 들어가는 문자를 입력하면 입력 모드로 변경 (a, A, i, I, o, O)
- 입력 모드에서 명령 모드로 복귀하려면 '<ESC>'키

# 기본 명령어

## ■ 커서의 이동

h	한칸 왼쪽으로 이동		한칸 오른쪽으로 이동
j	한줄 아래로 이동	k	한줄 위로 이동
w	다음 단어의 첫 글자로 이동	W	다음 단어의 첫 글자로 이동
b	이전 단어의 첫 글자로 이동	B	이전 단어의 첫 글자로 이동
e	단어의 마지막 글자로 이동	E	단어의 마지막 글자로 이동
^	그 줄의 첫 글자로 이동	\$	그 줄의 마지막 글자로 이동
0	그 줄의 처음으로 이동	<CR>	다음 줄의 첫 글자로 이동
+	다음 줄의 첫 글자로 이동	-	윗줄의 첫 글자로 이동
(	이전 문장의 첫 글자로 이동	)	다음 문장의 첫 글자로 이동
{	이전 문단으로 이동	}	다음 문단으로 이동

❖ 대문자 명령 : 무조건 띄어쓰기 전까지를 한 단어로 취급

❖ vi는 빈 줄이 나오기 전까지를 한 문단으로 취급

# 기본 명령어

## ■ 입력

a	커서 위치의 다음 칸부터 끼워넣기(append)
A	커서가 있는 줄의 끝에서부터 끼워넣기
i	커서 위치부터 끼워넣기(insert)
I	커서가 있는 줄의 맨 앞에서부터 끼워넣기
o	커서 바로 아래에 줄을 만들고 끼워넣기(open line)
O	커서 바로 위에 줄을 만들고 끼워넣기

# 기본 명령어

## ■ 수정

r	커서 위치의 한 글자 교체
R	커서 위치부터 <ESC>를 누를때까지 다른 글자로 교체
s, cl	커서 위치의 글자를 한개를 여러 글자로 교체
ch	커서 바로 앞의 글자를 한개를 여러 글자로 교체
cw	커서 위치의 한 단어를 교체
c0	커서 위치부터 줄의 처음까지 교체
C, c\$	커서 위치부터 줄의 끝까지 교체
cc	커서가 있는 줄을 교체
cj	커서가 있는 줄과 그 다음 줄을 교체
ck	커서가 있는 줄과 그 앞줄을 교체

# 기본 명령어

## ■ 삭제

x, dl	커서 위치의 글자 삭제
X, dh	커서 바로 앞의 글자 삭제
dw	한 단어를 삭제
d0	커서 위치부터 줄의 처음까지 삭제
D, d\$	커서 위치부터 줄의 끝까지 삭제
dd	커서가 있는 줄을 삭제
dj	커서가 있는 줄과 그 다음 줄을 삭제
dk	커서가 있는 줄과 그 앞줄을 삭제

# 기본 명령어

## ■ 복사

yw	커서 위치부터 단어의 끝까지 복사
yO	커서 위치부터 줄의 처음까지 복사
y\$	커서 위치부터 줄의 끝까지 복사
yy	커서가 있는 줄을 복사
yj	커서가 있는 줄과 그 다음 줄을 복사
yk	커서가 있는 줄과 그 앞줄을 복사
p	커서의 다음 위치에 붙여넣기
P	커서가 있는 위치에 붙여넣기

# 기본 명령어

## ■ 기타 명령어

u	작업 취소(undo)
U	그 줄에 행해진 작업 모두 취소
Ctrl+r	작업 재실행(redo)
.	조금 전에 했던 명령을 반복
J	현재 줄과 아래 줄을 연결
~	대문자를 소문자로, 소문자를 대문자로 바꿈
%	괄호의 반대쪽 짝으로 이동
Ctrl+l	현재 화면을 지우고 다시 그림
Ctrl+g	파일에 관한 정보를 표시



# 고급 명령어

## ■ 커서 이동

H	커서를 화면의 맨 위로	z<CR>	현재 줄을 화면의 맨 위로
M	커서를 화면의 중앙으로	z.	현재 줄을 화면의 중앙으로
L	커서를 화면의 맨 아래로	z-	현재 줄을 화면의 맨 아래로
[n]H	커서를 위에서 [n]번째 줄로	[n]L	커서를 아래에서 [n]번째 줄로
Ctrl+u	반 화면 위로 스크롤	Ctrl+b	한 화면 위로 스크롤
Ctrl+d	반 화면 아래로 스크롤	Ctrl+f	한 화면 아래로 스크롤
gg, 1G	문서의 맨 처음 줄로	G	문서의 맨 마지막 줄로
[n]G	[n]번째 줄로 이동		

# 고급 명령어

## ■ 책갈피

ma	현재 위치를 `a'로 마크
`a	마크된 `a'로 이동
``	직전의 커서 위치로 이동
'a	마크된 `a'가 있는 줄의 처음으로 이동
''	직전에 커서가 위치하던 줄의 처음으로

# 고급 명령어

## ■ 검색 및 교체

/[pattern]<CR>	현재 위치에서부터 아래 방향으로 패턴 검색
?[pattern]<CR>	현재 위치에서부터 위쪽 방향으로 패턴 검색
n	검색하던 방향으로 계속 패턴 검색
N	검색하던 반대 방향으로 계속 패턴 검색
:s/old/new<CR>	현재 줄의 처음 old를 new로 교체
:s/old/new/g<CR>	현재 줄의 모든 old를 new로 교체
:1,20s/old/new/g<CR>	1부터 20번째 줄까지 모든 old를 new로 교체
:-2,+4s/old/new/g<CR>	커서 2줄 위부터 4줄 아래까지 old를 new로 교체
:%s/old/new/g<CR>	문서 전체에서 old를 new로 교체
:%s/old/new/gc<CR>	문서 전체에서 old를 new로 확인하며 교체
:g/pattern/s/old/new/g<CR>	pattern이 있는 모든 줄의 old를 new로 교체
:g/pattern/s//new/g<CR>	:%s/old/new/g<CR>과 동일

# 고급 명령어

- 끝줄 모드

:q<CR>	vi를 종료함
:w<CR>	편집중인 문서를 저장
:w [file]<CR>	편집중인 문서를 [file]로 저장
:w >> [file]<CR>	편집중인 문서를 [file]에 덧붙여서 저장
:e [file]<CR>	[file]을 불러옴
:e#<CR>	이전에 편집하던 파일을 불러옴
:e%<CR>	현재 파일을 다시 불러옴, 즉 저장하지 않은 작업 취소
:r [file]<CR>	[file]을 커서위치에 끼워넣기
:set [option]<CR>	[option]을 켜기
:set [nooption]<CR>	[option]을 끄기
:[command]<CR>	[command] 실행
:r ![command]<CR>	[command] 실행 결과를 끼워넣기

# VIM 설정

set autoindent : 자동 들여쓰기

set cindent : C 프로그래밍을 할 때 자동 들여쓰기

set smartindent : 좀더 똑똑한 들여쓰기

set textwidth=79 : 만약 79번째 글자를 넘어가면

set wrap : 자동으로 <CR>를 삽입하여 다음 줄로 넘어감

set nowrapscan : 검색할 때 문서의 끝에서 다시 처음으로 돌아가지 않음

set nobackup : 백업 파일을 만들지 않음

set visualbell : 키를 잘못 눌렀을 때 뽕 소리를 내는 대신 번쩍이게 함

set ruler : 화면 우측 하단에 현재 커서의 위치(줄,칸)를 보여줌

set tabstop=4 : Tab을 눌렀을 때 8칸 대신 4칸 이동

set shiftwidth=4 : 자동 들여쓰기를 할 때 4칸 들여 쓰도록 함