

vi 사용법

UNIX/LINUX의 문서편집기

- ed 오래된 유닉스 표준 라인 텍스트 편집기
- ex 유닉스 표준 라인 텍스트 편집기
- emacs Emacs 편집기
- gmacs Emacs 편집기
- vi 유닉스 표준 화면 편집기
- view vi와 같음, 읽기 전용모드

vi 편집기 시작하기

- 셸에서 터미널 환경 맞추기(vi 시작 전)
 - `set term=vt100` // 터미널을 vt100으로 설정
 - `tset` // 터미널 초기화
- 명령어 시작
 - `vi` // vi 시작
 - `vi 파일이름`
 - `vi -R 파일이름` // 읽기전용 모드로 시작
 - `view 파일이름` // 읽기전용 모드로 시작

vi의 3가지 모드

- 명령 모드(ESC 모드)
 - ESC 키를 누르면 들어가는 상태로 명령들을 입력할 수 있는 모드
 - 이 상태에서 파일을 저장하거나, 입력, 찾기 등 다른 명령을 수행
- 입력 모드
 - i, a, o, I, A, O 를 누른 후 텍스트를 입력할 수 있는 모드
- 콜론 모드(ex 모드)
 - ESC를 누른 후 :을 입력한 상태
 - 예를 들면 종료하고자 하는 경우 다음과 같이 입력
 - :wq!

입력모드로 들어가기

- 다음의 명령을 입력하면 입력 모드로 전환
 - i : insert, 현재 커서의 위치에 글자를 삽입
 - I : insert, 커서가 있는 줄(line)의 맨 앞에 글자를 삽입
 - a : append, 현재 커서 위치의 다음 칸에 글자를 추가
 - A : append, 커서가 있는 줄(line)의 맨 뒤에 글자를 추가
 - o : open line, 현재의 줄 다음에 새로운 줄을 삽입
 - O : open line, 현재의 줄 앞에 새로운 줄을 삽입
 - 입력 모드로 전환한 후 입력한 것들이 파일 내용

종료하기

- 콜론 모드에서 수행
- 편집한 데이터를 저장하고 종료
 - :wq // write and quit
- 저장하지 않고, 그냥 종료
 - :q! // quit
- [Tip] 현재 작업 중인 내용의 저장
 - :w // write

명령 모드에서 글자 단위 이동

- h : 커서를 한칸 왼쪽으로 이동하는 명령
- j : 커서를 한줄 아래로 이동하는 명령
- k : 커서를 한줄 위로 이동하는 명령
- l : 커서를 한칸 오른쪽으로 이동하는 명령
- <Backspace> : 커서를 한칸 왼쪽으로 이동하는 명령
- <Space> : 커서를 한칸 오른쪽으로 이동하는 명령

명령 모드에서 줄 단위 이동(1)

- ^ : 빈칸을 무시하고, 커서를 현재 줄의 첫 글자로 이동
- 0 : 커서를 현재 줄의 처음으로 이동
- \$: 커서를 현재 줄의 끝으로 이동
- % : 짝을 이루는 기호 확인하기
- [Tip] 프로그램에서 열린 괄호({)에 대응하는 닫힌 괄호(})를 찾고자 하는 경우 % 명령을 이용
- + : 커서를 다음 줄의 처음으로 이동
- - : 커서를 이전 줄의 처음으로 이동
- <Return> : 커서를 다음 줄의 처음으로 이동
- [Tip] 실제로 20 ← 하는 경우 21번째 줄로 이동 (20 + <Return> 효과)

명령 모드에서 줄 단위 이동(2)

- n : 현재 줄의 n 번째 열로 (n 은 숫자)
- H : 커서를 화면상에 처음(Home) 줄로 이동
- M : 커서를 화면상에 중간(Middle) 줄로 이동
- L : 커서를 화면상의 마지막(Last) 줄로 이동
- nH : 화면상의 처음 줄로부터 n 줄 밑으로 이동
- nL : 화면상의 마지막 줄로부터 n 줄 위로 이동
- gg : 파일의 첫 줄로 이동(Go)
- G : 맨 마지막 줄로 이동
- ngg : n 번째 줄로 이동
- nG : n 번째 줄로 이동
- n : n 번째 줄로 이동

명령 모드에서 단어 단위 이동

- w : 커서를 다음 단어(word)의 첫 글자로 이동
- b : 커서를 이전(back) 단어의 첫 글자로 이동
- e : 커서를 다음 단어의 끝(end) 글자로 이동
- E : ?, ! 등 구두점을 무시하고, 현재 단어의 끝으로 이동
-) : 다음 문장의 시작으로 이동
- (: 이전 문장의 시작으로 이동
- } : 다음 문단의 시작으로 이동
- { : 이전 문단의 시작으로 이동
-]] : 다음 섹션의 시작으로 이동
- [[: 이전 섹션의 시작으로 이동

명령 모드에서 화면 단위 이동

- <Ctrl>F : 다음(Forward) 화면으로 이동
- <Ctrl>B : 전(Backward) 화면으로 이동
- <Ctrl>D : 반쪽 화면 밑(Down)으로 이동
- <Ctrl>U : 반쪽 화면 위(Up)로 이동
- <Ctrl>E : 커서는 현재 위치 그대로, 화면만 한 줄씩 위로 이동
- <Ctrl>Y : 커서는 현재 위치 그대로, 화면만 한 줄씩 아래로 이동
- [Tip] <Ctrl>G : 현재 편집하고 있는 문서의 상태 표시
- [Tip] <Ctrl>L : 화면 재표시

정규 표현식의 특수기호

- . : 한 글자를 대표
- * : 여러 개의 문자를 대표
- ^ : 줄의 처음 시작
- \$: 줄의 끝
- % : 처음 줄부터 끝 줄까지
- \w< : 단어의 시작과 대응
- \w> : 단어의 끝과 대응
- [] : 일정한 제한을 두어 글자를 대표
 - [a-z] : 알파벳 소문자 a부터 z까지를 대표
 - [0-9a-c] : 0,1,2,3,4,5,6,7,8,9, a, b, c 를 대표

찾기

- $/abc$: 현재 위치에서 뒤로 abc 찾기
- $?abc$: 현재 위치에서 앞으로 abc 찾기
- n : 뒤로 단어 찾기 반복
- N : 앞으로 단어 찾기 반복
- $/$: 뒤로 단어 찾기 반복
- $?$: 앞으로 단어 찾기 반복

복사하기

- `yy` : 한 줄 전체 복사(yank)
- `nyy` : 현재 줄 이하로 n 개의 줄 복사
- `Y` : 한 줄 전체 복사 (`yy`와 동일)
- `yw` : 한 단어 복사
- `ymw` : n 개의 단어 복사
- `y$` : 현재 위치에서 그 줄의 끝까지 복사
- `y0` : 현재 위치에서 그 줄의 처음까지 복사
- `yG` : 현재 위치에서 파일의 끝까지 복사
- `y/abc` : 현재 위치에서 `abc`가 나올 때까지 복사

지우기(1)

- dmw : n 개의 단어 삭제
- $d\$$: 현재 위치에서 줄의 끝까지 삭제
- D : 현재 위치에서 줄의 끝까지 삭제
- $d0$: 현재 위치에서 줄의 처음까지 삭제
- $d^$: 현재 위치에서 줄의 처음까지 삭제
- dG : 현재 위치에서 문서 끝까지 삭제
- $:lined$: $line$ 줄을 삭제
- $:line1, line2d$: $line1$ 에서 $line2$ 까지 삭제
 - $:30, 35d$: 30번째 줄부터 35번째 줄까지 삭제

지우기(2)

- `dd` : 현재 줄 전체 삭제
- `n dd` : n 개의 줄 삭제
 - `10dd` : 10 줄 삭제
- `dw` : 현재 단어 삭제
- `n dw` : n 개 단어 삭제
- `s` : 한 글자 삭제 후 insert 모드
- `S` : 줄 삭제 후 insert
- `d/abc` : `abc` 단어가 나올 때까지 지우기
- `d/^abc` : 줄의 맨 앞에 `abc` 단어가 나올 때까지 지우기

붙이기

- p : 버퍼에 저장된 내용(복사 혹은 삭제)을 커서의 오른쪽으로 붙이기(paste)
- np : n 번 p 명령을 반복
- P : 저장된 내용을 커서 앞으로 붙이기
- nP : n 번 P 명령 반복

되살리기

- u : 한번 복구하기(undo)
- nu : n 단계까지의 복구

바꾸기

- r : 글자 바꾸기(replace)
- nr : n 개 글자 바꾸기
- cc : 한 줄 바꾸기(change)
- ncc : n 개의 줄 바꾸기
- cw : 한 단어 바꾸기
- cnw : n 개 단어 바꾸기
- ce : 공백을 제외하고 한 단어 바꾸기
- $c\$$: 현재 위치에서 줄 끝까지 바꾸기
- C : 현재 위치에서 줄 끝까지 바꾸기
- $c0(c^{\wedge})$: 현재 위치에서 그 줄 앞까지 바꾸기

줄 합치기

- J : 현재 줄을 앞 줄에 붙이기(join)
- nj : n 개의 줄을 합쳐 한 줄로 만들기

유용한 ex 명령(1)

- 기본 형식 : (범위) (명령어) (수행 위치)
- `:/1,/2 cmd /3` : /1에서 /2까지 `cmd` 수행하여 /3
 - `:1,10 co 50` : 1줄부터 10줄까지 복사(copy)하여 50줄에 붙이기
 - `:30, 50 d` : 30줄부터 50줄까지 삭제(delete)
 - `:100,200 m 50` : 100줄부터 200줄까지 50줄로 이동(move)

유용한 ex 명령(2)

■ 찾기 응용 예

- `:/abc/ d` : `abc`가 들어있는 줄 삭제
- `:/abc/ -nd` : `abc`가 들어있는 줄로부터 앞으로 `n`개 줄 삭제
- `:/abc/ +nd` : `abc`가 들어있는 줄로부터 뒤로 `n`개 줄 삭제
- `:/abc/,/xyz/d` : `abc`가 들어있는 줄부터, `xyz`가 들어있는 줄까지 삭제
- `../abc/ m 10` : 현재 줄부터 `abc`가 들어있는 줄까지를 10번째 줄로 이동

유용한 ex 명령(3)

■ 전체 범위(g) 응용

- `:g/abc` : `abc`가 포함된 마지막 줄로 이동
- `:g/abc/ p` : 파일 전체에서 `abc`가 있는 줄 출력
- `:g/abc/ nu` : 파일 전체에서 `abc`가 있는 줄을 줄번호와 함께 출력
- `:60,100 g/abc/ p` : 60~100줄 사이에서 `abc`가 있는 줄 출력
- `:g/abc/d` : 문서 전체에서 `abc`가 있는 줄 제거

유용한 ex 명령(4)

- 전체적으로 특정 단어 바꾸기
- `:/1,12 s/old/new/g` : 1에서 12까지 *old*를 *new*로 바꾸기
- `:g/abc/ s/old/new/g` : *abc*가 있는 모든 줄에서 *old*를 *new*로 바꾸기
 - `:1,5 s/리눅스/linux/gc` : 1줄부터 5 줄까지 '리눅스'를 'linux' 로 확인(check)하면서 바꾸기
 - `:g/문자 /s/파일/file/g` : '문자'가 있는 줄에서 '파일'을 'file'로 바꾸기
 - `:% s/버퍼/buffer/g` :파일 전체에서, '버퍼' 를 'buffer' 로 바꾸기

유용한 ex 명령(5)

- 현재 위치로 외부 파일 읽어 오기
 - `:r filename`
- vi 잠시 중지하고 셸 명령 수행하기 (수행 후 enter를 입력하면 vi로 복귀)
 - `:! ls`
 - `:! gcc test.c`
 - `:! ./a.out`

환경 변수

- :set으로 현재 설정 확인
- :set number : 화면에 줄 번호 보이기
- :set nonumber : 줄 번호 보이지 않기
- number 외에도 tabstop(탭 간격), filetype(파일 종류), cindent (들여쓰기 형식)등