

4. 파일 편집기

4.1 텍스트 편집기(text editor)

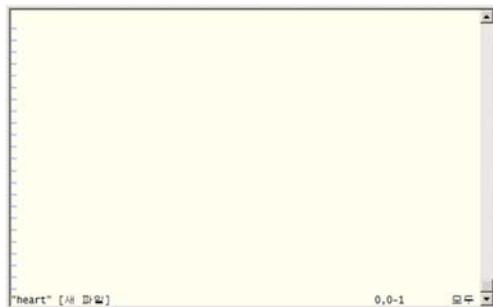
- 텍스트 편집기 종류
 - 라인 편집기(line editor)
 - (1) **ed** : original line editor, 명령어 방식
 - (2) **ex** : ed의 개선된 버전
 - 화면 편집기(screen editor)
 - (1) **vi** : visual screen editor (original UNIX screen editor)
 - (2) **vim** : vi IMproved, vi에 없는 기능 추가 (Linux의 vi는 vim임)
 - (3) **view** : vi가 read only mode에서 시작됨
 - (4) **emacs** : Editor MACroS
Lisp-based AI community에서 시작됨

2

4.2 vi 편집기의 기본 사용

■ vi의 시작

- % vi** ; 저장할 때에 파일이름 지정해야 함
- % vi filename** ; filename파일을 편집. 주로 사용하는 방식



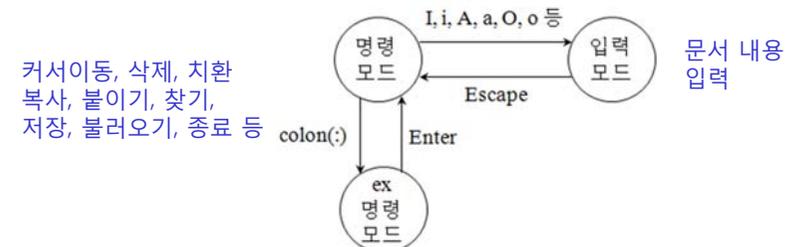
- 마지막 입력 줄 다음에 ~가 표시됨 (~가 표시된 줄은 빈 줄임)

3

vi 편집기의 특징

■ 특징

- 특수기를 사용하지 않고 편집 가능
- 2가지 동작 모드
 - (1) **명령 모드** - 명령모드, ex 명령 모드
 - (2) **입력 모드**
- ex 명령 모드는 라인 편집기의 명령을 사용하는 모드
 - **:**로 시작하며 화면 아래에 나타남

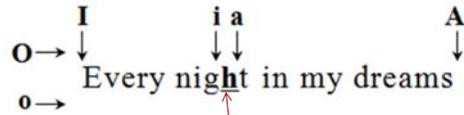


4

입력모드

- vi 시작 시에 명령 모드
- text 입력 모드 전환
 - i, I : insert
 - a, A : append
 - o, O : open line
 - ...

59쪽 표 참조



현재 커서 위치

- ESC키를 사용하여 명령 모드로 돌아옴

5

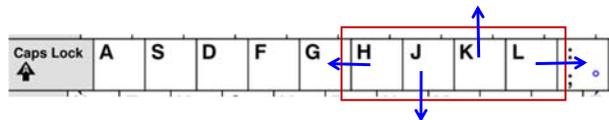
명령 모드

- 명령 모드에서의 동작
 - 커서 이동
 - 삭제, 치환/변경
 - 복사, 붙이기
 - 탐색, 바꾸기
 - 파일 저장, 불러오기
 - 종료

6

커서 이동

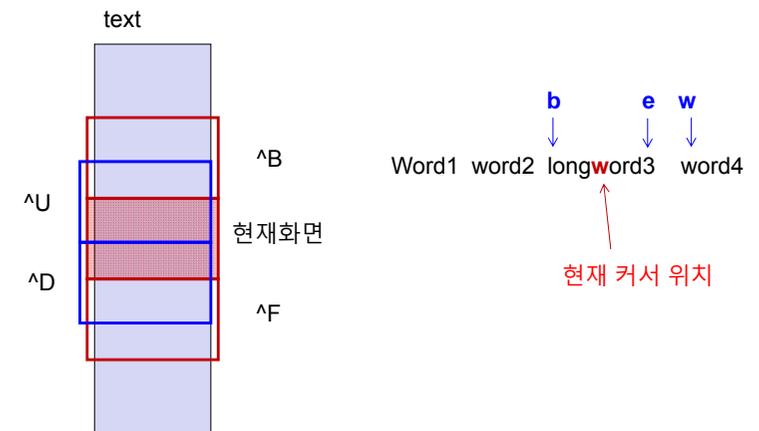
- 커서 이동
 - 입력모드에서 ESC를 눌러서 명령모드로 이동한 후에 커서를 이동함
- h j k l : (한문자 이동) ← ↓ ↑ →
- ^ \$: line 처음, 끝
- b w : word 단위 앞/뒤
- e : word 끝
- ^B ^F : 한 화면 위/아래 (backward / forward)
- ^U ^D : 반 화면 위/아래 (up / down)
- 100G 또는 :100 : 100행
- 명령키 반복
 - 5h 6w : 명령어 앞의 숫자만큼 반복 수행



UNIX시스템

7

커서 이동(2)



현재 커서 위치

UNIX시스템

8

텍스트 삭제

■ 삭제(delete)

- x (문자) X (왼쪽 문자)
- dw (단어)
- dd (줄)
- D (현재부터 줄 끝까지)

(cf) 8x 3dw 2dd

커서 이동과 삭제는 명령 모드로 전환 후 수행함

- dm : 현재부터 커서이동 문자 *m*으로 이동되는 곳까지 삭제

■ 삭제 - ex 모드

- :1,3d ; 1행부터 3행까지 삭제
- :.,.+2d ; 현재 행부터 3행을 삭제 (. 은 현재 행)

텍스트 수정

■ 치환/대치

- r (문자) - 한 문자만 치환
- cw (단어)
- cc (줄)
- R (현재커서부터 계속하여 치환)

} 입력모드로 전환됨

(cf) 4cw

- cm : 현재부터 커서이동 문자 *m*으로 이동되는 곳까지 치환(변경)

텍스트 복사 및 붙이기

■ 복사(copy) 및 붙이기(paste, put)

- yy (yank line) 또는 Y (줄 복사)
(cf) 4yy (4줄 복사)
- :10,14y (10-14행까지 5행을 복사)
- yw (yank word)

- p 현재 줄(커서) 다음에 붙이기 (put)
- P 현재 줄(커서) 이전에 붙이기
- :pu 현재 줄(커서) 다음에 붙이기
- :10pu 10행 다음에 붙이기

- ym : 현재부터 커서이동 문자 *m*으로 이동되는 곳까지 복사
(cf) yw (단어 복사)
y\$ (줄 끝까지 복사)

편집 취소 및 반복

■ 취소

- u (undo) : 가장 최근 편집 명령 취소
- U (undo) : 현재 줄에 대한 편집 명령 취소
- 3u
- :e! : 현재 파일의 편집 내용을 버리고, 최근 저장 내용으로 작업

■ 반복

- . (period) : 가장 최근의 텍스트 편집 명령을 반복

■ 행 연결 - 두 줄 결합

- J (join) : 현재 줄을 다음 줄과 결합하여 한 줄로 만들

■ 화면을 다시 그림

- Ctrl-L : 편집화면에 다른 프로그램의 출력이 표시된 경우 화면을 다시 그리면 이 출력이 없어짐

찾기(search) 및 바꾸기

■ 문자열 검색(search)

- /string[Enter] 앞쪽으로 찾기
- ?string[Enter] 뒤쪽으로 찾기
- ^<ten> 단어 문자열 찾기

■ 검색 반복

- n (마지막 검색 명령 반복)
- N (반대 방향으로 검색 명령 반복)

■ 대치 (substitute) – 찾아 바꾸기

- :1,10s/sss/ttt/ ; 각 행의 발견된 첫 문자열(sss)을 ttt로 바꾸기
- :10,\$s/sss/ttt/g ; 발견된 모든 문자열(sss)을 ttt로 바꾸기
- :.,.+9s/\/10/g ; 현재부터 10줄의 단어 ten을 10으로 치환
often의 ten은 치환되지 않음

파일 저장 및 종료

■ 저장

- :w ; 파일 저장
- :w filename ; filename에 파일 저장 (새이름으로)
- :1,10w filename ; 일부 행 만 저장

■ 종료

- :q ; 종료(변경된 내용을 저장 해야 종료됨)
- :q! ; 무조건 종료

■ 저장 후 종료

- :x ; 저장 후 종료
- :wq
- ZZ

4.3 vi의 고급사용

■ 파일 저장/불러오기

- :r filename ; 현재 위치에 다른 파일 읽기(삽입)
- :e filename ; 다른 파일 불러오기
- :n ; 다음 파일 편집
% vi *.c (편집할 파일이 여러 개임)

■ 다른 명령어 실행

- :!command ; command를 수행한 후 vi로 되돌아감
- :r!command ; command 수행 결과를 현재 위치에 삽입
- !!command ; 위와 같음
- :sh ; 셸을 실행, 프롬프트 출력

버퍼 사용, 옵션 설정, 초기화 파일

■ 버퍼 사용

- “a3yy ; 3줄 복사, 버퍼a에 저장 (버퍼 이름 = 알파벳 문자)
- “ap ; 버퍼a 내용 붙이기

■ 옵션 설정 - :set 명령 (69쪽 참조)

- 옵션 변수 설정 여부 또는 설정 값에 따라서 vi 동작에 영향을 줌
- :set autoindent 또는 se ai
- :set number 또는 se nu
- :set tabstop=4 또는 se ts=4
- :set ai nu ; 한꺼번에 설정
- :set all ; 모든 변수 설정 상태 출력

■ 옵션 설정 취소 - 변수이름 앞에 no를 붙임

- :set noautoindent 또는 se noai

■ 초기화 파일 - 기본값과 다른 옵션 변수 설정을 할 때 사용

- vim은 .vimrc을, 원래의 vi는 .exrc를 사용

4.4 emacs 편집기

- 많은 확장성과 강력한 매크로 기능을 가진 텍스트 편집기
- 초기에 Lisp 언어로 작성되었으며, 현재에는 C로 작성된 GNU emacs가 널리 사용됨
- emacs 시작
 - \$ emacs file
 - 기본적으로 입력 모드로 시작
- 명령키 – Ctrl 또는 Alt와 함께 문자 입력 또는 ESC 다음에 문자 입력
 - C-문자 (Ctrl-문자)
 - M-문자 (Alt-문자 또는 ESC 문자)
 - 대소문자 구분

주요 명령어

- 저장, 종료
 - C-x C-s 저장
 - C-x C-c 종료
- 커서 이동
 - 화살표 키
 - C-p C-n C-b C-f (위, 아래, 왼쪽, 오른쪽)
 - ...
- 문자 삭제
 - Backspace(이전) Del(현재) 또는 Del(이전) C-d(현재)
- 찾기
 - C-s (이후 검색) C-r (이전 검색)
- 기타 기능
 - 교과서 및 기타 자료 참고