

VI editor

Dept. of Computer Multimedia Engineering,
Dongguk University

<http://ce.dongguk.ac.kr>

Introduction

- VI (visual editor)는 Unix에서 사용되는 기본적인 화면 편집기
 - UNIX 시스템의 편집기들 : ed, ex, vi, EMACS 등
- 특징
 - 마우스 지원, 풀다운 메뉴, 페이지 포맷 설정 등을 할 수 없으나 UNIX 환경 컴퓨터에서 자원의 소모가 적으며 편집 작업을 위한 다양한 도구 제공
- 구성
 - vi는 mode형 편집기로서 insert mode와 command mode로 구
 - command mode : 커서이동, 텍스트 삭제, 복사, 붙여 넣기
 - insert mode : 텍스트 추가, 변경
- Linux 에서의 vi
 - 개선된 vi인 vim(vi improvement)가 실행됨
 - 참조 man vim

2

VI 실행과 종료

- vi 실행
 - [spec_user@ce spec_user]\$ vi^{Enter}
 - vi 프로그램이 실행되며 vi의 명령모드에 있게 된다
- vi 종료
 - vi에서 : q^{Enter}
 - vi 프로그램의 실행이 끝나고 다시 shell로 돌아온다
 - 파일의 저장은 : w^{Enter}
 - q 명령은 편집하려고 열었던 파일을 저장하지 않고 vi를 종료하게 함(편집이 이루어진 경우 : q!^{Enter} 명령으로 종료)
 - ex) : q, : q!, : wq, : wq!

3

VI에서의 커서 이동

- \$ vi hello.c^{Enter}
 - 테스트를 위해 hello.c 파일을 vi로 연다
- 방향키를 이용한 커서이동
 - ← : 커서가 왼쪽으로 한칸 이동
 - → : 커서가 오른쪽으로 한칸 이동
 - ↑ : 커서가 위쪽으로 한칸 이동
 - ↓ : 커서가 아래쪽으로 한칸 이동
- 문자키를 이용한 커서이동
 - h : 커서가 왼쪽으로 한칸 이동
 - j : 커서가 오른쪽으로 한칸 이동
 - k : 커서가 위쪽으로 한칸 이동
 - l : 커서가 아래쪽으로 한칸 이동

4

VI에서의 커서 이동(계속)

- 단어 단위와 현재 라인에서의 커서 이동
 - **W**: 커서가 다음단어 첫 문자로 이동
 - **E**: 커서가 단어의 마지막 문자로 이동
 - **B**: 커서가 이전단어 첫 문자로 이동
 - **O** or **^**: 커서가 라인의 첫 문자로 이동
 - **\$**: 커서가 라인의 마지막 문자로 이동
- 문자키를 이용한 커서이동
 - **H**: 커서가 화면의 첫 번째 줄 첫 문자로 이동
 - **L**: 커서가 화면의 마지막 줄 첫 문자로 이동
 - **M**: 커서가 화면의 중간 첫 문자로 이동
 - **G**: 커서가 vi로 연 파일의 마지막 줄의 첫 문자로 이동
 - 1G: 파일의 첫 라인으로 이동
 - 15G: 파일의 15번째 라인으로 이동

5

VI에서의 커서 이동(계속)

- 검색
 - 검색하고자 하는 단어 혹은 문자열(공백포함가능)의 시작 문자로 커서를 이동시킨다
 - 사용법:
 - 순차검색 : /검색어
ex) / printf **Enter**
 - 역방향검색 : ?검색어
ex) ? printf **Enter**
 - 순차적으로 계속 검색을 하고 싶을 경우 : **n**
 - 역방향으로 계속 검색을 하고 싶을 경우 : **N**

6

VI에서의 커서 이동(계속)

- 화면표시 위치 이동
 - **Ctrl f**: 파일의 내용을 한 화면 아래로 이동 표시
 - **Ctrl b**: 파일의 내용을 한 화면 위로 이동 표시
 - **Ctrl d**: 파일의 내용을 반 화면 아래로 이동 표시
 - **Ctrl u**: 파일의 내용을 반 화면 위로 이동 표시

7

명령모드에서 텍스트 삭제

- 한 문자 삭제 : **x**
- 변경한 내용 되돌리기 : **u**
- 한 라인 삭제 : **d d**
- 한 단어 삭제 : **d W**
- 커서위치부터 라인 끝까지 삭제 : **D**
- 명령어 앞에 숫자 옵션을 주면 여러 라인 혹은 여러 문자가 삭제된다
 - 3dd : 커서가 위치한 라인과 아래 두 라인 삭제
 - 5x : 커서가 위치한 문자 포함 5문자 삭제
 - 4dw : 커서가 위치한 단어 포함 4단어 삭제(공백 제외)

8

파일에 라인번호 붙이기

- 라인마다 번호 붙이기 명령어:
 - : set nu or :set number
- 번호 붙이기 취소 명령어:
 - : set nonu
- vi의 : 명령은 ex 편집기의 영향이며 명령을 받기 위해 커서는 화면 하단으로 이동한다
 - : set 명령어의 취소는 : set no 명령어로 한다
 - setting 되어있는 값들을 보기 원할 경우는 : set 혹은 : set all 입력

9

파일에 텍스트 추가

- 커서가 위치한 문자 왼쪽에 추가 :
- 커서가 위치한 문자 오른쪽에 추가 :
- 커서 위치 밑으로 한 라인 추가 :
- 커서 위치 위쪽으로 한 라인 추가 :
- i, a, o, O 명령 실행 후에 vi는 insert mode에 있게 되며 텍스트의 추가를 가능하게 한다
- Insert mode에서 command mode 로의 전환은 사용
- 현재 진행중인 모든 종류의 작업은 key를 사용해 command mode 로 전환 할 수 있다

10

스크립트 실행

- 현재 디렉토리에 존재하지 않는 파일인 scrp을 vi의 인자로 주어 파일을 연다
- Insert mode를 사용해 shell에서 실행 할 수 있는 명령어인 echo, cal, date를 써 넣는다
- : wq 명령은 vi를 열 때 받았던 scrp 인자와 같은 파일에 내용을 저장하고 vi를 종료 시킨다
- scrp 파일을 실행 파일로 만들어 실행 하면 shell은 순차적으로 한 라인씩 읽어 명령을 수행한다

11

문자의 대체

- 한 문자 대체 :
- 문자열 덮어쓰기 :
- 한 문자 삭제 후 insert mode로 전환 :
- R, s 명령은 덮어쓰기와 insert mode에서 빠져 나오기 위해 사용하여 command mode로 전환해야 한다
- 한 단어 삭제 후 insert mode로 전환 :
- 한 라인 삭제 후 insert mode로 전환 :
- 현재 커서의 위치부터 라인 끝까지 삭제 후 insert mode로 전환 :

12

파일전체 또는 특정영역에서의 텍스트 변경

- 지정된 모든 라인에서 "목표어"을 찾아 해당 라인에서 처음 나타나는 "목표어"을 "대체 문자열"로 변경 :
 - : 범위 s/목표어/대체문자열/
 - ex) : 1, \$ s/pirntf/printf/
- 특정 문자열이 존재하는 라인에서, 해당라인의 처음 나타나는 "목표어"를 "대체 문자열"로 변경 :
 - : g /특정문자열/s/목표어/대체문자열/
- 텍스트 변경 명령 끝에 / g플래그(global)을 주면 "목표어"가 나타나는 라인에서 일치하는 모든 "목표어"를 "대체 문자열"로 변경한다

13

텍스트의 이동 및 복사

- 한 라인 복사 : `y|y`
- 커서가 위치한 아래쪽으로 붙여 넣기 : `p`
- 여러 라인 복사 : 숫자 `y|y`
- 한 단어 복사 : `y|w`
- 한 라인 삭제 : `d|d`
- 여러 라인 삭제 : 숫자 `d|d`
- 한 단어 삭제 : `d|w`
 - dd 명령은 라인을 삭제 한 후에 vi가 버퍼에 삭제한 내용을 보관하도록 한다

14

파일의 저장 및 종료

- 원본과 변경된 파일 모두 저장 :
 - : w 새로운파일이름
- 특정 영역을 파일로 저장 :
 - : 4,10 w 새로운파일이름
 - : 4,10 w !파일이름 으로 명령 시에는 파일이 이미 존재 할지라도 덮어쓰기가 된다
- 파일 끝에 추가 :
 - : 4,10 w >> 파일이름
 - 4~10 라인이 파일 끝에 추가된다

15

작업중인 파일에 다른 파일 붙여 넣기

- 현재 작업중인 파일의 커서가 위치한 아래쪽으로 다른 파일의 전체 내용 붙여 넣기 :
 - : r 다른파일이름
- 현재 작업중인 파일의 특정위치 아래쪽으로 다른 파일의 전체 내용 붙여 넣기 :
 - :14 r 다른파일이름

16

VI에서 shell 명령어 실행

- vi 실행 중에 shell 명령어 입력
 - : ! 명령어
명령어는 shell에서 사용될 수 있는 명령어를 입력 한다
ex) ls, cal, who ...
- : r ! 명령어를 입력하면 현재 편집중인 파일 안에 shell 명령어로 얻은 내용을 읽어 들일 수 있다

17

VI로 여러 개의 파일 열기

- 여러 개의 파일 열기
 - %vi 파일1 파일2 ...
다음 파일로 이동
 - : n
처음 파일로 이동
 - : rew
파일 저장
 - : w
vi 종료
 - : q!

18

VI 설정 파일

- vi의 초기화 파일은 .exrc 이며 vim의 초기화 파일은 .vimrc 이다.
- vi 는 실행이 되면서 초기화 파일을 읽어 설정되어 있는 기능을 사용 할 수 있도록 한다.

19

Q & A