

리눅스시스템 및 응용

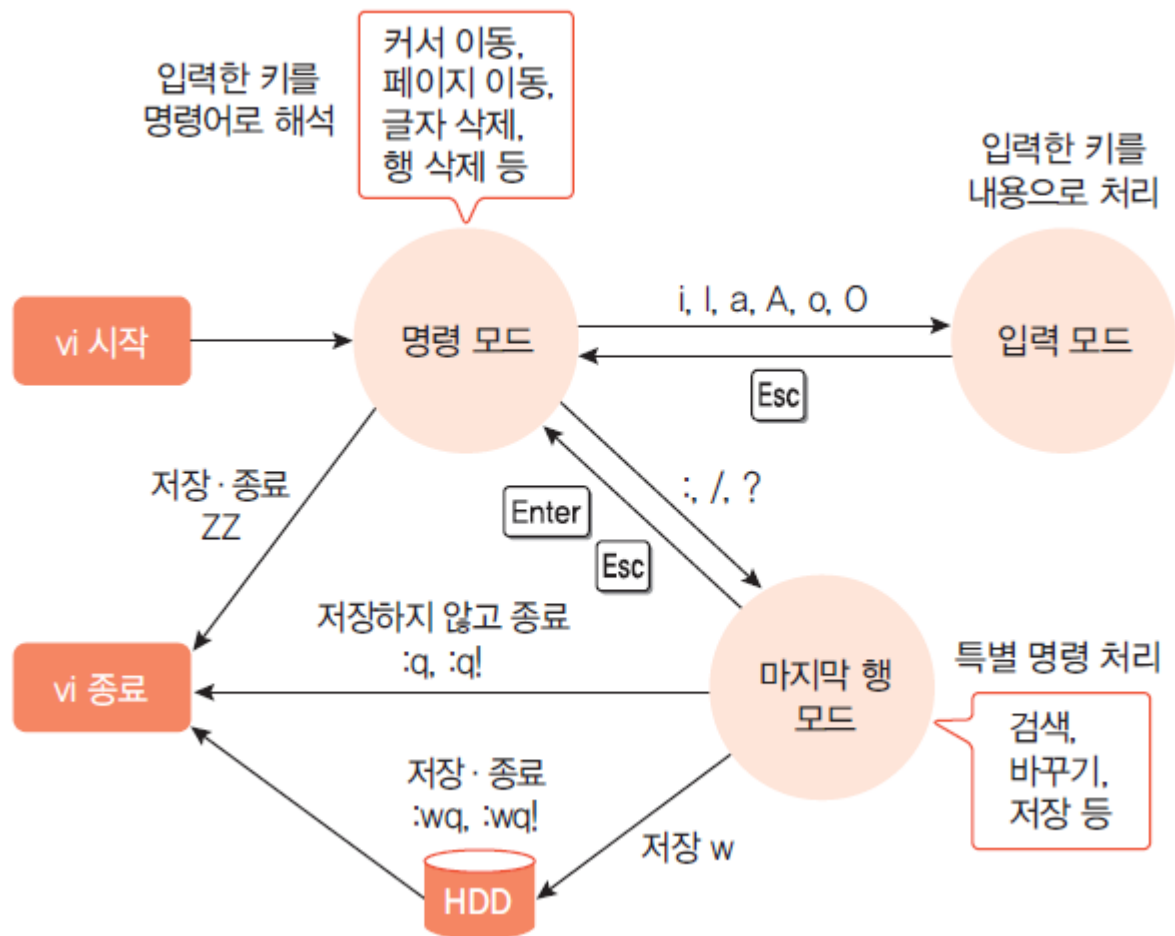
Week 5

Vi 에디터 사용법

학습목표

- vi editor 의 여러가지 기능 사용하기

학습목표



vi 시작하기

- vi *file* Invoke vi on file
- vi *file1 file2* Invoke vi on files sequentially
- view *file* Invoke vi on file in read-only mode
- vi -R *file* Invoke vi on file in read-only mode
- vi -r *file* Recover file and recent edits after a crash
- vi -t *tag* Look up *tag* and start editing at its definition
- vi -w *n* Set the window size to *n*; useful over a slow connection
- vi + *file* Open *file* at last line
- vi +*n* *file* Open *file* directly at line number *n*

커서 Cursor 위치 변경

- *Character*

- h, j, k, l Left, down, up, right (←, ↓, ↑, →)

- *Text*

- w, W, b, B Forward, backward by word
- e, E End of word
-), (Beginning of next, previous sentence
- }, { Beginning of next, previous paragraph
-]], [[Beginning of next, previous section

- *Lines*

- ENTER First nonblank character of next line
- 0, \$ First, last position of current line
- ^ First nonblank character of current line
- +, - First nonblank character of next, previous
- line
- n | Column n of current line
- H, M, L Top, middle, last line of screen
- n H n (number) of lines after top line
- n L n (number) of lines before last line

- *Scrolling*

- CTRL-F , CTRL-B Scroll forward, backward one screen
- CTRL-D CTRL-U Scroll down, up one half-screen
- CTRL-E CTRL-Y Show one more line at bottom, top of window
- z ENTER Reposition line with cursor: to top of screen
- z . Reposition line with cursor: to middle of screen
- z - Reposition line with cursor: to bottom of screen
- CTRL-L Redraw screen (without scrolling)

- *Searches*

- */pattern* Search forward for *pattern*
- *?pattern* Search backward for *pattern*
- *n, N* Repeat last search in same, opposite direction
- */, ?* Repeat previous search forward, backward
- *f x* Search forward for character *x* in current line
- *F x* Search backward for character *x* in current line
- *t x* Search forward to character before *x* in current line
- *T x* Search backward to character after *x* in current line
- *;* Repeat previous current-line search
- *,* Repeat previous current-line search in opposite direction

- *Line number*

- CTRL-G Display current line number
- *n* G Move to line number *n*
- G Move to last line in file
- : *n* Move to line *n* in file

- *Marking position*

- `m x` Mark current position as *x*
- `' x` Move cursor to mark *x* (grave accent)
- `''` Return to previous mark or context (two grave accents)
- `' x` Move to beginning of line containing mark *x* (single quote)
- `''` Return to beginning of line containing previousmark (two single quotes)

- *Insert*

- i, a Insert text before, after cursor
- I, A Insert text before beginning, after end of line
- o, O Open new line for text below, above cursor

- *Change*

- cw Change word
- cc Change current line
- *c motion* Change text between the cursor and the target of *motion*
- C Change to end of line
- r Replace single character
- R Type over (overwrite) characters
- s Substitute: delete character and insert new text
- S Substitute: delete current line and insert new text

- *Delete, move*

- x Delete character under cursor
- X Delete character before cursor
- dw Delete word
- dd Delete current line
- d *motion* Delete text between the cursor and the target of *motion*
- D Delete to end of line
- p, P Put deleted text after, before cursor
- " *n* p Put text from delete buffer number *n* after cursor (for last nine deletions)

- *Yank*

- yw Yank (copy) word
- yy Yank current line
- " a yy Yank current line into named buffer a (a–z); uppercase names append text
- y motion Yank text between the cursor and the target of motion
- p, P Put yanked text after, before cursor

- *Other commands*

- . Repeat last edit command
- u, U Undo last edit; restore current line
- J Join two lines

- *ex edit commands*

- :d Delete lines
- :m Move lines
- :co or :t Copy lines
- :.,\$d Delete from current line to end of file
- :30,60m0 Move lines 30 through 60 to top of file
- :.,/pattern/co\$ Copy from current line through line containing *pattern* to end of file

- Exit Commands

- *Command Meaning*

- :w Write (save) file
 - :w! Write (save) file, overriding protection
 - :wq Write (save) and quit file
 - :x Write (save) and quit file
 - ZZ Write (save) and quit file
 - :30,60w *newfile* Write from line 30 through line 60 as *newfile*
 - :30,60w>> *file* Write from line 30 through line 60 and append to *file*
 - :w %.*new* Write current buffer named *file* as *file.new*
 - :q Quit file
 - :q! Quit file, overriding protection
 - Q Quit vi and invoke ex
 - :e *file2* Edit *file2* without leaving vi

-
- :n Edit next file
 - :e! Return to version of current file as of time of last write (save)
 - :e # Edit alternate file
 - :vi Invoke vi editor from ex
 - : Invoke one ex command from vi editor
 - % Current filename (substitutes into ex command line)
 - # Alternate filename (substitutes into ex command line)

- Buffer Names

- 1–9 The last nine deletions, from most to least recent
- a–z Named buffers to use as needed; uppercase letters append to the respective buffers

- Buffer and Marking Commands

- " b command Do command with buffer b
- m x Mark current position with x
- ' x Move cursor to character marked by x (grave accent)
- ' ' Return to exact position of previous mark or context (two grave accents)
- ' x Move cursor to first character of line marked by x (single quote)
- ' ' Return to beginning of the line of previous mark or context (two single quotes)

- The Substitute Command

- The general form of the substitute command is:

- **:*[addr1[,addr2]]s/old/new/[flags]***

- flags

- c Confirm each substitution
 - g Change all occurrences of *old* to *new* on each line (globally)
 - p Print the line after the change is made

- Address Symbols

- 1,\$ All lines in the file
- x,y Lines x through y
- x;y Lines x through y, with current line reset to x
- 0 Top of file
- . Current line
- n Absolute line number n
- \$ Last line
- % All lines; same as 1,\$
- x-n n lines before x
- x+n n lines after x
- -[n] One or n lines previous
- +[n] One or n lines ahead
- ' x Line marked with x (single quote)
- ' ' Previous mark (two single quotes)
- /pat/ or ?pat? Ahead or back to the line where pat matches

-
- :sh
 - :! shell command
 - set number
 - :set nu
 - :set nonu
 - 특수문자 보기
 - :set list
 - 다음 파일 불러오기
 - :n
 - :n!
 - 사용전에 반드시 이전 파일을 저장 하기를 결정해야 한다

파일 묶기.압축하기

- tar
 - 묶기 `tar -cvf [묶음파일명] [묶을 파일리스트]`
 - 풀기 `tar -xvf [묶음파일명]`

- 압축하기
 - gzip
 - gunzip
 - Compress
 - Uncompress

Question?