Lecture 3: Editors

Youjip Won and Kyungsoo Park

**KAIST EE**

EDITOR#1 - VI

Contents

- What is vi

- Modes of vi

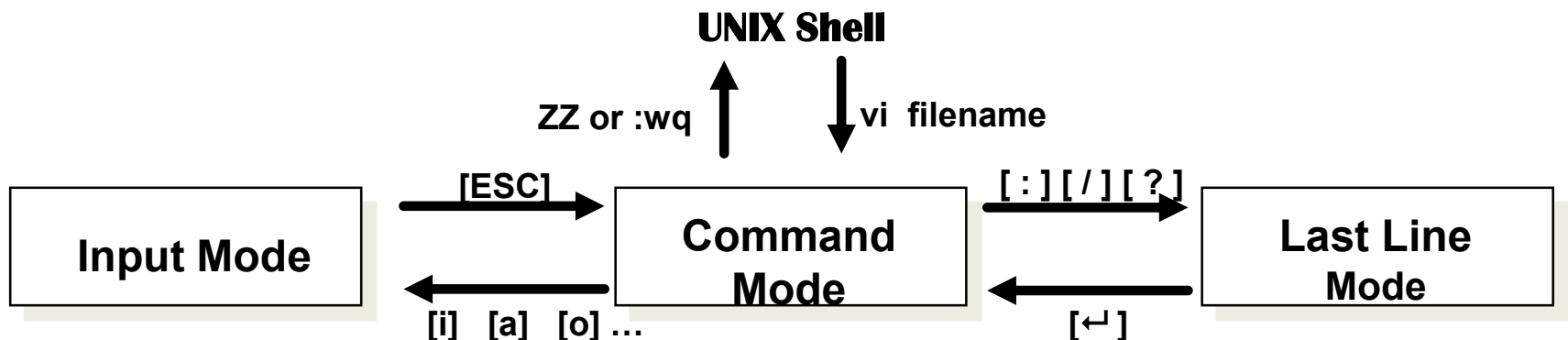- Simple tutorial: from starting to quitting vi

- References

What is vi

- **Vi** is a visual screen text editor developed by Bill Joy, who later becomes co-founder of Sun Micro Systems.

  - It is visual version of **ex**, a Unix line editor.

  - Vi is available on most Unix Systems.

  - Works with a variety of terminals.

  - Allows **ex** command from **vi.**

- **VIM** is acronym for **Vi** Improved, developed by Bram Moolenaar, a extended version of vi and some of enhancements include

  - Completion, comparison, and merging of files

  - Split and tabbed windows

  - Command histories

- All editing session before saving is done in buffer area.

  - Nothing is saved as hard data, until you save it.
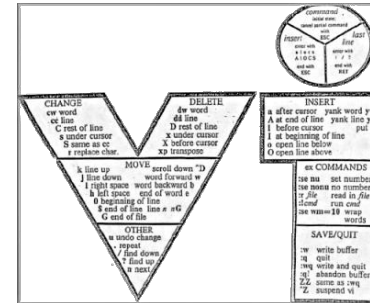
Modes of vi

- There are three modes in **vi.**

  - Command mode – a default mode in vi

    - Everything is command before you enter into other modes.

  - Input mode – what you type is what you see.

    - Anything typed in this mode is considered as data.

    - Pressing [ESC] always leads to Command mode.

  - Last Line Mode – Only can be accessed from Command mode.

    - Three ways to enter Last Line Mode – : (Colon) / (Back Slash) ? (Question Mark)

UNIX Shell

ZZ or :wq ↑   ↓ vi filename

| Input Mode | →[ESC]→ | Command Mode | →[:] [/] [?]→ | Last Line Mode |

[i]   [a]   [o] …

[↵]

Simple Tutorial: From Starting to Quitting vi

- This simple tutorial illustrates how to write, delete, copy, paste, replace, save, and quit

- Start vi by *vi newfile.txt* and type the following

[i] This is how we write [esc] [o] and copy lines [esc] [k] [y][y] [j] [p] [k] [y] [3] [w] [j] [ )] [a] [space bar] [p] [o] ummm [esc] [b] [x] [r] E [l] [r] N [l] [r] D [:] [w] [q] [enter]

- This will produce following and goes back to command prompt

This is how we write
and copy lines
This is how we write and copy lines
End

Simple Tutorial: From Starting to Quitting vi

[i] This is how we write [esc] [o] and copy lines [esc] [k] [y][y] [j]
[p] [k] [y] [3] [w] [j] [ )] [a] [space bar] [p] [o] ummm [esc] [b]
[x] [r] E [l] [r] N [l] [r] D [:] [w] [q] [enter]

- Explaining the commands in the tutorial

| i | Insert | o | Add new line after current line | a | Append |

| k | Moving cursor up | j | Down | l | Right | h | Left |

| y | Yank a line | p | Paste after cursor point |

| w | Moves cursor to next word | b | Previous word |) | End of sentence |

| r | Replace character | x | Erase character | 3 | Number Repeats command |

| : | Enters last line mode | w | Writes to data file | q | Quits vi **(only in last line mode)** |

Save and Exit

- Save

    `:w save`

    `:q exit`

    `:wq save and exit`

    `ZZ save and exit`

    `:q! exit without saving`

- Replace

    - `:s/pt/str/flag` : replace pattern with string.

    - flag

        - g: replace all occurences

        - c: confirm replaces

        - &: repeat last :s command

Searching and replace

- Search

  /string: search forward

  ?string: search backward

  n: go to next match

  N: go to previous match

  :se nu: turn on the line numbers

- Replace

  - :s/pt/str/flag : replace pattern with string.

  - flag

    - g: replace all occurences

    - c: confirm replaces

    - &: repeat last :s command

References

Graphical cheat sheet of Vi and VIM
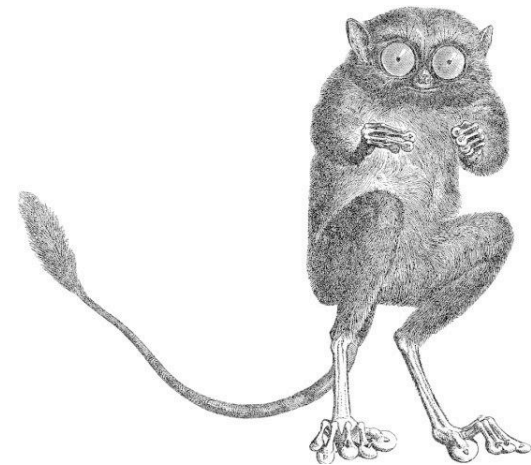
http://www.viemu.com/a_vi_vim_graphical_cheat_sheet_tutorial.html

Cursor movement Commands

http://www.kcomputing.com/vi.html

List of Commands

http://www.smashingmagazine.com/2010/05/03/vi-editor-linux-terminal-cheat-sheet-pdf/

Kyungsoo Park

EDITOR#2 - EMACS

Contents

- What is Emacs

- What is Emacs and why use it ?

- Emacs Preliminaries

- Command Combinations

- Simple Tutorial: From Starting to Quitting Emacs

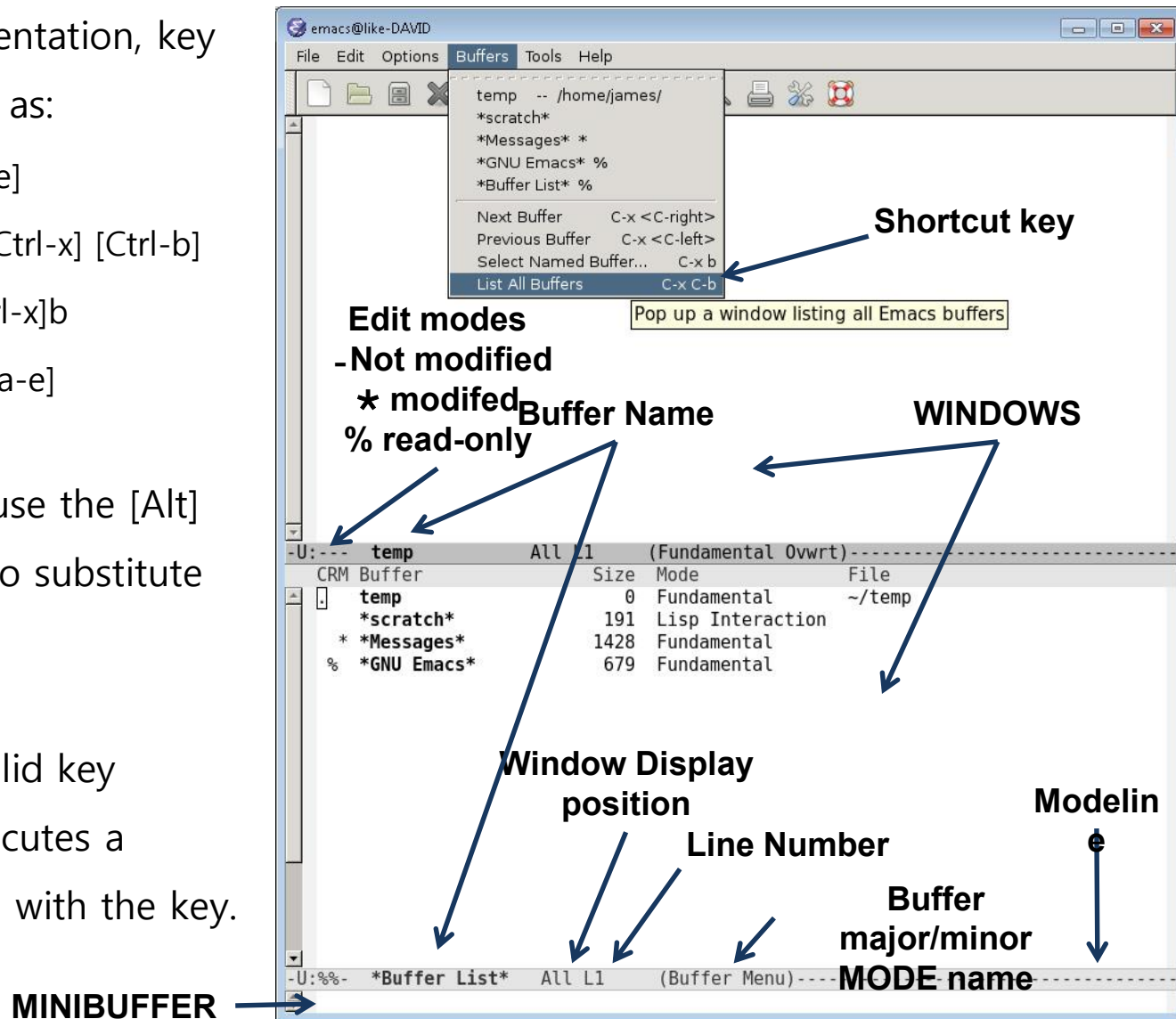- References

What is Emacs and Why use it ?

- Emacs (Editor MACroS) is the extensible, customizable, self-documenting, real-time display editor.

  - Richard Stallman is the author of Emacs; the author of GCC and GDB

  - Runs on LISP engines + lots of LISP libraries.

Richard Stallman
The founder of GNU

- It is not the only good choice, there are options like VI, VIM.

  - Works on many platforms and independent of GUI.

  - Extremely powerful.

  - **vi** often does things with fewer keystrokes, but **emacs** easily surpass vi when it comes to searching and replacing and using macros.

- Some of assumptions of Emacs are.

  - No mouse! – Much more reliable and much faster for experienced user

  - No particular keyboard; No particular GUI environment

  - Runs through telnet (as well as directly).

Emacs Preliminaries

- In the emacs documentation, key sequences described as:

  - C-e – This is [Ctrl-e]

  - C-x C-b – This is [Ctrl-x] [Ctrl-b]

  - C-x b – This is [Ctrl-x]b

  - M-e – This is [Meta-e]

- On the PC, you can use the [Alt] key or [Esc]-release to substitute [Meta] key.

- When you press a valid key sequence, emacs executes a command associated with the key.

Shortcut key

Edit modes
- **Not modified**
- **＊ modifed**
- **％ read-only**

Buffer Name

WINDOWS

Window Display position

Line Number

Modeline

Buffer major/minor MODE name

MINIBUFFER

Command Combinations

- When in Doubt, use ``Get me out of here '' command – [Ctrl- g], c-g

- Emacs uses the control keys to move in the four directions.

  - b (back), f (forward), p (previous line) and n (next line)

  - [Ctrl-b] – move left, [Ctrl-f] – move right, [Ctrl-p] – move up, [Ctrl-n] – move down

- Macros are useful for repeatable key sequences that may be include commands..

- Common macro commands

  - C-x (– begin macro definition (after this, type whatever actions you'd like repeated/stored)

  - C-x) – end macro definition

  - C-e – execute stored macro

  - C-u 5 C-e – execute stored macro 5 times (Note: C-u 5 can prefix any emacs cmd, even a non-cmd)

Simple Tutorial: From Starting to Quitting Emacs

One can type without having to use complex commands but here we go.

Start emacs by **emacs newfile.txt** and type the following.

This is how we write [c−q−j] and copy lines [c−p] [c−a]
[c−spc] [c−e][m−w] [m−}] [c−q−j] [c−y] [c−p] [c−a] [c−spc] [c−u]
[3] [m−f] [m−w] [m−}] [spc] [c−y] [enter] ummm [c−a] [c−d] [c−u]
[3] End [c−x−s] [c−x−c]

This will produce following and goes back to command prompt.

This is how we write
and copy lines
This is how we write and copy lines
End

Simple Tutorial: From Starting to Quitting Emacs

This is how we write [c−q−j] and copy lines [c−p] [c−a]
[c−spc] [c−e] [m−w] [m−}] [c−q−j] [c−y] [c−p] [c−a] [c−spc] [c−u]
[3] [m−f] [m−w] [m−}] [spc] [c−y] [enter] ummm [c−a] [c−d] [c−u]
[3] End [c−x−s] [c−x−c]

⦿ Explaining the commands in the tutorial
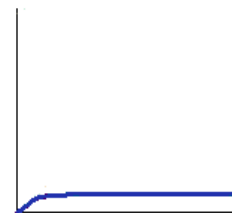
C- is denoted as ^ in this example

^qj Add a new line	**^p** Move to previous line
^a Beginning of the sentence	**^spc** Begin highlighting
^e End of the sentence	**M−w** Copy highlighted	**M−}** End of the paragraph
^y Paste copied text	**^u** Repeat command with following number
M−f Move to next word	**^d** Delete a character	**^xs** Save the document
^xc Quit the document	

References

- Reference card with most commands you'll ever need

  - http://www.clip.dia.fi.upm.es/~logalg/doc/emacsreference.pdf

- Official GNU emacs site

  - http://www.gnu.org/software/emacs/

- An emacs HowTo

  - http://jeremy.zawodny.com/emacs/emacs.html

  - http://www2.lib.uchicago.edu/keit
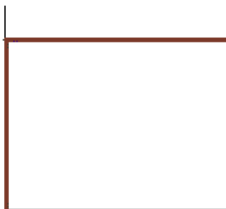
Classical learning curves for some common editors

Notepad

Pico
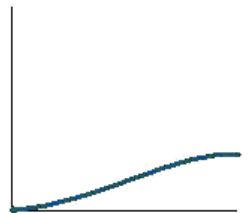
Visual Studio

vi

emacs

Steve Rowe's blog

Homework

Use the text file `blkparse.txt` and do the following tasks. You can choose either vim or emacs at your own disposal.

1. Goto line 51.

2. Find "trace" and replace it with "TRACE".

3. Repeat step 2 ten times. Use repeat command instead of manually repeating the step 2 ten times.

4. Goto line 1.

5. Delete five lines from line 1 to line 5.

6. When a line starts with a space, replace all consecutive spaces with a single space.

7. Save the file under the name "STUDENT_ID.txt" and email to TA.