EDITORS

Lab 11

EECS 448

Meenakshi Mishra and Dr Swapan Chakrabarti

Text Editors

- Writing program is similar to writing an essay in a specialized language
- Editors used to make writing as efficient as possible
- Editors for English should have preferably
  - Spelling and Grammar check
  - Copy, Cut and Paste Options
  - Easy formatting options
  - Manageable citations
  - Reference tool
  - Sectioning and Listing
  - Text search and replacement

Editors in Programming

- Why do we need editors?
  - Syntax highlighting and Code Formatting
  - Syntax Reminders
  - Debugging
  - Line numbering
  - Do not have your preferred document editor
  - Preferred editor does not load
  - Different working environment
  - No Mouse
  - Maintain Mobility without carrying your computer everywhere
- What do you expect to learn from this lab?
  - Focus on two major text editors, vi and emacs
  - Commands that will facilitate navigation and editing the text fast
  - Focus on Command Line execution

Editors

- Which Editor shall I use?
  - It depends on your preference and availability
- How do I remember so many commands?
  - No one remembers all the commands
  - Each one of us remembers the set of commands we use regularly and keep adding to the set
  - Important objective is to get efficient and fast in programming
- Vi and Emacs are most popular as are entirely keyboard driven
  - Mouse usually slows people down
- Other popular editors are gedit, notepad, eclipse (it is an IDE)

Existing Editors

- Nano, Pico
  - Default on Debian and Gentoo Distributions
- Text Editors (Notepad etc.)
- Vi, Vim, Gvim
  - Available on all Unix/Linux Systems
  - Can be installed on DOS, Windows and MacOS
  - Vim (Vi Improved) has both text and GUI interface
  - It is also extensively customizable
- Emacs
  - Specifically designed with programmers in mind
  - Has many useful features and plug-ins helpful for programmers
- Eclipse
  - Powerful open source GUI IDE
  - Similar to Microsoft Visual Studio

VI editor

- Standard editor
  - Usually present in all Linux/UNIX systems
  - Often the only editor you will have access to in machines newly installed or you cannot modify
  - Uses very little resources
  - Easy to use
  - Commands are similar across boards

Open a file

- Create a directory Lab11 (mkdir Lab11)
- Go inside your newly created directory (cd Lab11)
- Open a new or existing file in edit mode(vi tryusingvi.txt)
  - Open the file in read only mode (vi –R *filename*)

Modes

- Command Mode
  - Used for administrative process e.g. saving, quitting
  - Parse through text
  - Cut, copy, paste
  - Search and replace
  - Anything typed is interpreted as command
- Insert Mode
  - Can insert text into the file
  - Anything typed is interpreted as input to the file and inserted
- Normal Mode
  - Can provide some basic instructions like quitting, navigation

Modes

- Command Mode
  - Enter command mode from normal mode
    - Press ‘:’ or ‘/’
  - Exit command mode to normal mode
    - Press esc or return
- Insert Mode
  - Enter Insert mode from normal mode
    - Press ‘i’ or ‘a’
  - Exit Insert mode to normal mode
    - Press Esc

Try this

- In the file open, type “The quick brown fox jumped over the lazy dogs.”
  - To type, you must first press ‘i’ to enter the insert mode
- Save it by typing ‘:w’
  - To give command, you must first enter normal mode by pressing esc
- Quit by typing ‘:q’
- Open the file again (vi filename)
- Type something else in the file
- Quit without saving (:q!)
- Quit with saving (:wq)

Navigation and insertion

- Navigate through text (h/j/k/l → left/down/up/right)
- Scroll down one half page (ctrl-d)
- Scroll up one half page (ctrl-u)
- Scroll down one full page (ctrl-b)
- Scroll up one full page (ctrl-f)
- Move cursor to middle of page (M)
- Move cursor to top of the page (H)
- Move cursor to bottom of the page (L)
- Move cursor forward word at a time (w or 5w)
- Move cursor back word at a time (b or 5b)
- Move cursor to beginning of line/ end of line (0/$)
- Move cursor to beginning of sentence/ end of sentence '(' / ')'
- Move cursor to matching bracket '%'

Editing

- Insert at the cursor (i)
- Append after the cursor (a)
- Undo last change (u)
- Undo all changes to entire line (U)
- Delete line (dd or 3dd)
- Delete contents of line after cursor (D)
- Delete word (dw)
- Delete character at cursor (x)
- Copy (yy or move cursor to the start position and press v, move cursor to end position and press y)
- Cut (d)
- Paste (P for before cursor and p for after the cursor)

Try This

- Open the file tryusingvi.txt
- Copy the sentence
- Paste the sentence multiple times
- Play around with the commands for navigation
  - Move cursor up, down, left right
  - Move page up, page down, half page up, half page down
  - Move the cursor to middle of the page
- Play around with the commands for editing text
  - Type another sentence of your choice
  - Delete a word
  - Delete the line
  - Undo deletion

Other commands

- Search for *search_seq* (/*search_seq* or ?*search_seq*)
- Search and replace in current line (:s/search/replace/g)
- Search and replace in entire file (:%s/search/replace/g)
- Run linux commands (:!command)

Try This

- Play with the commands for searching and replacing
  - Search for all the “The” in the text
  - Change all the “The” to “A”
  - Save this file
  - See what other files are in your current directory

Emacs editor

- Started by Richard Stallman
- Free, portable, extensible
- Particularly good for programmers
- Modeless
- Only a small difference in GUI and SSH

Commands in Emacs

- All keys are commands
  - Letters a-z and A-Z are commands to insert the letters on the file
  - Every Command has a long name which you can look up in the documentation
    - Examples are: kill-line, delete-backward-char, self-insert-command
  - We will learn key-strokes which make editing fast
- Key combinations produce different commands
  - 95 printable ASCII characters execute self-insert-command
  - Ctrl-key (C-x) in combination with other keyboard keys constitute additional commands
  - Meta-key (M-x) gives rise to fresh set of commands
    - Usually ‘Alt’ key is the meta key
    - Esc can also be used as meta key
    - Esc is an ASCII character, so don’t hold it

Prefixes used

- C-c → Used for commands specific to particular modes
- C-h → Used for Help Commands
- C-x → Used for manipulating files, buffers and windows
- M-x → Type the full name of the command
  - E.g. M-x compile
- C-u -N → Repeat the following command N times

Data Structures in Emacs

- File
  - Actual file on disk
  - Never edit this file directly
  - Emacs reads a copy of the file in a buffer, and makes changes to buffer instead
  - Saving file means writing copy of the buffer onto a file
- Buffer
  - Internal data structure that holds the text you actually edit
  - Can have any number of buffer active at any time
  - Buffers have names
    - The buffer has almost always the same name as the file it is visiting
  - At any given time, only one buffer is selected
  - Buffers can be deleted without effecting the files
- Window
  - The view of the buffer
  - Can split screens horizontally or vertically to view different buffers at once
  - Windows can be created or deleted without effecting the buffer

Commands to Manipulate Files

- Open emacs (emacs filename or emacs)
- C-x C-f (find-file): Read a file
  - When executed, emacs prompts you to enter filename
  - Then it checks to see if file is still being edited
  - If you are then it switches buffer
  - Else creates a new buffer named same as the file, initialized with copy of the file
- C-x C-s (save-buffer): Save the file current buffer is in
  - Write the current buffer to the file with same name
- C-x s (save-some-buffers): save all the buffers visiting file one by one
  - Allows you to save all buffer visiting files, querying for each one
- C-x C-c: exit emacs
- C-g: Your best friend
  - Exit the command

Movement Commands

- C-f , C-b: Move forward/backwards one character
- M-f , M-b: Move forward/backwards one word
- C-p , C-n: Move to previous/next line
- C-a , C-e: Move to beginning/end of line
- M-a , M-e: Move to beginning/end of sentence
- C-v (scroll-up)
- M-v (scroll-down)

Try This

- Open tryusingvi.txt
- Play around the movement commands
  - Move up/down/left/right
  - Move page up/ page down
  - Move to previous line/ previous sentence/ next line/ next sentence

Commands to Manipulate Windows

- C-x o (other-window)
- C-x 0 (delete-window)
- C-x 1 (delete-other-windows)
- C-x 2 (split-window-vertically)
- C-x 3 (split-window-horizontally)
- C-M-v (scroll-other-window)

Commands to manipulate buffers

- C-x b (switch-to-buffer): Switches buffer of current window
  - Prompts for a buffer name
  - Switches buffer of current window to that buffer
  - Creates a new empty buffer if you type a new name
    - This buffer does not visit any file
- C-x C-b (list-buffers): Gives list of buffers
  - Pops a new window that has the list
- C-x k (kill-buffer):
  - Prompts for a buffer name
  - Removes entire data-structure for that buffer
  - Asks you if you want to save it first
  - Does not delete file
- C-x C-q (vc-toggle-read-only):
  - Make buffer read only or read-write only

Try This

- Open any two programs from your file so that you can see both the programs and the text file you created for vi
- On both programs, add a line to print your name at the beginning of the main function
- Split the screen and open the shell (M-x shell)
- You can run all the shell commands from here
- You can even compile and run the programs here
- See the list of buffers

Editing Command

- C-spc : Mark the starting of the text to be selected
- C-w: Kill the text from the mark to the cursor
- C-y: Paste the text recently killed
- C-s: Search forward
  - Search for next match by typing C-s
- C-r: Search backward
- M-%: Replace
- C-g: Return to Start of Search
- M-u : make uppercase from cursor to end of word
- M-l : Make lowercase from cursor to end of word
- C-d: Delete character

Try This

- Play around with these commands like you did for vi editor

Help Commands

- C-h m (describe-mode): Help page for current mode (C++, Python)
- C-h a (command-apropos): Prompts for a keyword, then lists all commands with that keyword
- C-h k (describe key): Prompts for a key-stroke and displays its description
- C-h i (info): Enters the Info hypertext document

Try This

- Obtain a list of all documents with the word “window”
- See if you find any commands you know
- Obtain the description related to key-stroke “C-x C-f”

Editor War

- Ref: en.wikipedia.org/wiki/Editor_war

	Vi	Emacs
Keystroke Execution	Retains each permutation of typed key creating path in the decision tree	Emacs commands are key combination, so decision tree of commands formed but not of individual keystrokes
Memory Usage and Customizability	Vi: smaller, faster but not much customizable Vim: Evolved to provide more functionality and customization	Longer to startup, requires more memory, highly customizable, has large number of features, an execution environment for a Lisp program designed for text-editing
User Environment	Originally used inside text-mode consoles	Initially designed for text-mode console, but grew GUI very early
Function/Navigation Interface	Has distinct editing mode	Has metakey chords
Keyboard	Does not use Alt key, and rarely use Ctrl key	Uses Alt, Ctrl and Shift keys for commands