

Text Editors

❖ vi

- ◆ pronounce **vee eye**
- ◆ **vi** → Visual Interface
- ◆ **vim** → vim: vi improved (most current systems use this version)
- ◆ Is a screen-oriented text editor originally created for UNIX by **Bill Joy** in 1976
- ◆ Almost always available on all UNIX or Linux systems

❖ emacs

- ◆ Edit Macros
- ◆ It is not just an editor
- ◆ Available on most of Linux distributions
- ◆ Part of GNU Project

Text Editor: vi

- ❖ The **vi** is a modal program with two modes
 - ◆ **Command mode**
 - Write to save or quit files
 - Move the cursor around the files, cut & paste, substitution, etc.
 - ◆ **Insert mode (edit mode)**
 - Text is entered in this mode
 - The **ESC** key ends insert mode and returns to command mode.
- ❖ The **ESC** key always brings you back to command mode
- ❖ Open file: **vi file.txt**
 - ◆ **vi** is in command mode when a file is opened
 - ◆ The file will be created if it does not exist
- ❖ Exit **vi** (in command mode)
 - ◆ Type: ZZ (shift zz), file is saved, and exit **vi**. **↔** :wq or (:x)
 - ◆ Type **:q** or **:q!** in command mode

Browse File Content with vi

vi does not recognize mouse!!!

- ❖ Move cursor around with **h, j, k, l** (lower case) or the 4 arrow keys (in Command Mode)
 - ◆ **h** moving to left
 - ◆ **j** moving down
 - ◆ **k** moving up
 - ◆ **l** moving to right
- ❖ **ctrl-f** Move forward one screen
- ❖ **ctrl-b** Move backward one screen
- ❖ **G**: Move to the last line of the file (H:move to the 1st line)
- ❖ **:n** Goes to the nth line
 - ◆ **:1** goes to the first line
 - ◆ **:\$** goes to the last line, the end of file (this often used for pattern search and substitution)

The Command Mode (vi)

- ❖ To search for a pattern
 - ◆ Just type “/” and followed by the pattern, pattern can be a form of regular expression (regex)
 - ◆ Press **n** for forward search (for the **n**ext occurrence of the pattern)
 - The next occurrence of the search pattern will be highlighted
 - When the end of file is reached, it will go back to the beginning of the file, and highlight the first occurrence of the pattern
 - ◆ Press **N** for backward search (for the previous occurrence of the pattern)
 - The previous occurrence of the search pattern will be highlighted.

Command Line in Command Mode

- ❖ Entering command line by typing “:”
 - ◆ “:” shows **at the bottom of the screen** (vi must be in command mode)
- ❖ What can you do in command line mode (**Table 5.2 p131**)
 - ◆ `:q`

Quitting vi, you will be prompted to do something if the file has been modified
 - ◆ `:q!` → Quit without saving
 - ◆ `:w`

Save file, (do this often and regularly) and continue editing
vi uses “working buffer”, no changes made to file until you save it
 - ◆ `:wq` → Write (save) and quit (exit) vi
 - ◆ `:w newfile`

Write the file to a new file (`:w! filename – overwrite filename`)
 - ◆ `:r filename`

Read in a new and insert to where the cursor is

vi Insert Mode

- ❖ Switch to **insert mode** by one of the followings in the command mode
 - ◆ "i" (insert),
 - ◆ "a" (insert after current character), append
 - ◆ "A" (insert at end of line),
 - ◆ "o" insert after the current line(open a line after current line)
 - ◆ "O" insert before the current line(Open a line above current line) commands.
 - ◆ "r" = replace the character where the cursor is. You will be in command mode after the change
 - ◆ "R" - replace the rest of the line after the cursor with what you are going to type.
 - ◆ "c": clear the line the cursor is, and the line after it, and change to insert mode
 - cw: change/delete the next word
 - c10: change the next 10 lines + current line (total 11 lines)

How to “CUT”

In Command Mode

- ❖ x
 - ◆ Delete one character, and shift to right
- ❖ X
 - ◆ Delete one character, and shift to left
- ❖ dd
 - ◆ Delete the line where the cursor is
- ❖ d10
 - ◆ Delete 10 more lines from where the cursor is (11 line deleted)
- ❖ :2,10d
 - ◆ Delete line 2 to line 10

Cut and Paste (vi)

- ❖ Performed in the **command mode**
- ❖ Whatever you just deleted is saved in the buffer and can be pasted with command “p” or “P”
 - ◆ Before issuing “p”, move the cursor to where you want to paste/insert the content.
 - ◆ “p” will paste the content after the cursor
 - ◆ “P” will paste the content before the cursor

Copy and Paste with “Yank”

❖ yy

- ◆ Copy the current line (where the cursor is) into buffer

❖ <n>yy

- ◆ Copy the current line and the **n-1** lines after it to the buffer

❖ Paste with p or P after “yank”

More Editing Operators

❖ “J”

- ◆ “Join” the current line and the line following it

❖ “~”

- ◆ Toggle letter case

“undo” and “redo” and “repeat”

❖ Undo with “u” in command mode

- ◆ Will undo what you just did
- ◆ Capital “U” will undo the whole line
- ◆ Continue press “u”, like the undo on Word, can bring you all the way to what it is when you just open the file, even you have saved the file during the editing

❖ Redo

- ◆ with “`ctrl-r`”
- ◆ With “`:redo`”

❖ Repeat command: “.”

- ◆ Just press “.” to repeat the previous command.

Customizing vi with “set”

- ❖ In command line
 - ◆ :set ts=4
 - ◆ :set number
 - ◆ :set nonumber
 - ◆ :set ai (auto indent)
 - ◆ :set list (show hidden characters)
 - ◆ :set nolist (no show hidden characters)
- ❖ In configuration file under your home directory (.exrc or .vimrc, create one if not exist)

```
set ts=4
set number
set nonumber
set ai
```

Other Text Editors

❖ **pico**

- ◆ Pine COmposer
- ◆ Not available on all the Linux distributions
- ◆ Pine is a Program for Internet News and Email on UNIX

❖ **nano**

- ◆ A clone of PICO
- ◆ Part of GNU Project now.

❖ **gedit** (console, not through ssh)

- ◆ Lightweight editing tool for the GNOME desktop
- ◆ Can not use remotely (without gnome)

Editor: pico/nano

- ❖ Pico is a text file editor used on UNIX systems, not always available on Linux by default
- ❖ To start: `pico filename`
- ❖ Control commands (at bottom of the screen):
 - ◆ Ctrl-a Beginning of current line
 - ◆ Ctrl-e End of current line
 - ◆ Ctrl-v Forward one screen
 - ◆ Ctrl-y Backward one screen
 - ◆ Ctrl-f Forward a character
 - ◆ Ctrl-b Backward a character
 - ◆ Ctrl-p Previous line Ctrl-n Next line
- ❖ Google “pico” to learn more if you are interested

^G Get Help ^O WriteOut ^R Read File ^Y Prev Pg ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where is ^V Next Pg ^U UnCut Tex ^T To Spell