Linux para Ingeniería:

Editores de Texto

Luis Garreta
luis.garreta@javerianacali.edu.co

Ingeniería de Sistemas y Computación
Pontificia Universidad Javeriana – Cali

9 de febrero de 2018

EDITOR CONCEPTS

- Editing a file is to modify the content of a file
- Text editor:
  - Enter and modify text in a text file
- Word processor:
  - Enter, modify and **format** text in a document
- Line editor:
  - Edit file one line at a time
  - Unix examples: ex, ed and sed
- Full screen editor
  - Shows a whole screen of text at a time

EDITOR FEATURES

- enter text
- search and replace
- copy, cut and paste
- undo and redo
- importing and exporting text
- save and cancel

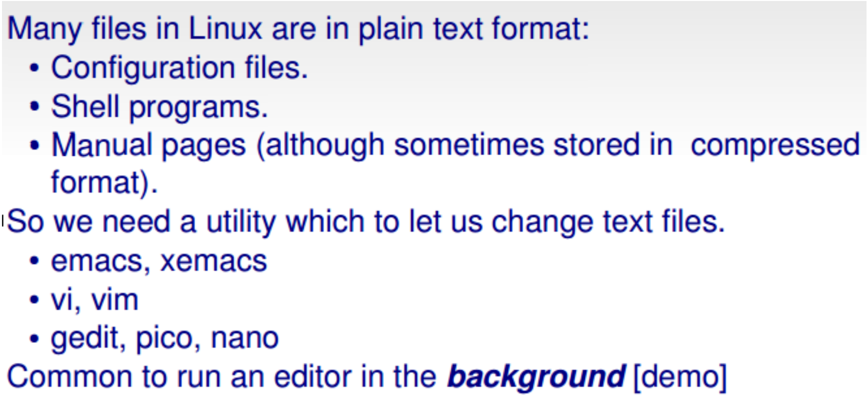
TEXT FILES

- Unix file name does not require file extension
  - Unix file system does not consider the extension when treating files
- However, some extensions are commonly used
  - Program source code: .c .cc .cpp .f .f77 .f95
  - Compiled object code: .o .a .so .sa
  - Compressed files: .z .gz .zip
  - Archive files: .tar .tz
  - Web site source code: .html .shtml .php
  - Text files that will be moved to Windows: .txt
- Executable files typically have no extension

CLASSIC UNIX TEXT EDITORS

- Console
  - vim
  - emacs
  - pico
  - nano
  - mcedit
- GUI editors
  - gvim
  - emacs
  - gedit
- Line editors:
  - x, ed and sed

Console Text Editors



Many files in Linux are in plain text format:

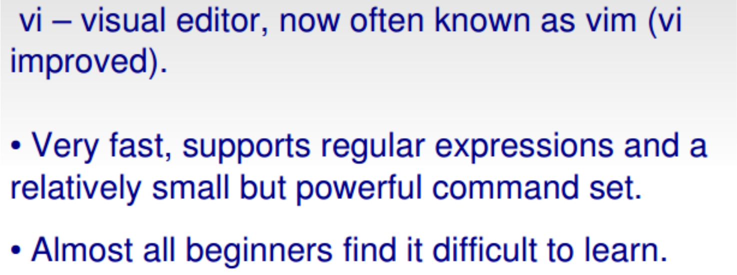
- Configuration files.
- Shell programs.
- Manual pages (although sometimes stored in compressed format).

So we need a utility which to let us change text files.

- emacs, xemacs
- vi, vim
- gedit, pico, nano

Common to run an editor in the ***background*** [demo]

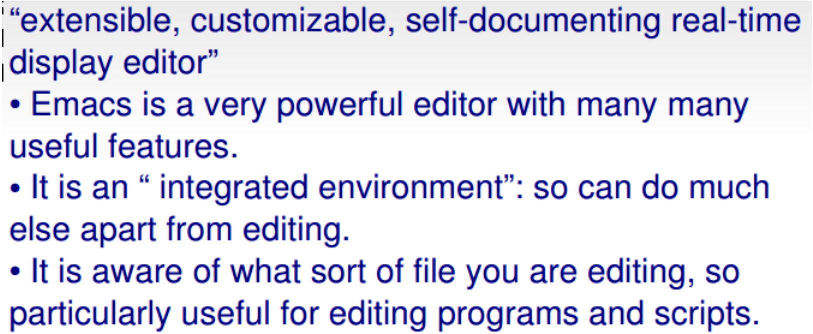
Vi or Vim



vi – visual editor, now often known as vim (vi improved).

- Very fast, supports regular expressions and a relatively small but powerful command set.
- Almost all beginners find it difficult to learn.

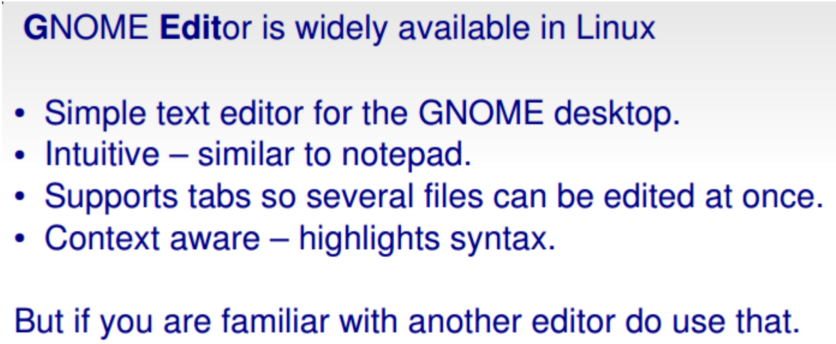
GNU emacs



"extensible, customizable, self-documenting real-time display editor"

- Emacs is a very powerful editor with many many useful features.
- It is an " integrated environment": so can do much else apart from editing.
- It is aware of what sort of file you are editing, so particularly useful for editing programs and scripts.

Gedit

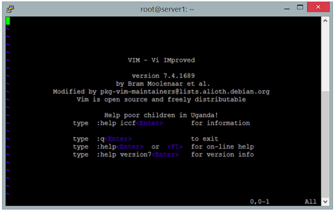


GNOME Editor is widely available in Linux

- Simple text editor for the GNOME desktop.
- Intuitive – similar to notepad.
- Supports tabs so several files can be edited at once.
- Context aware – highlights syntax.

But if you are familiar with another editor do use that.

The Vi Editor



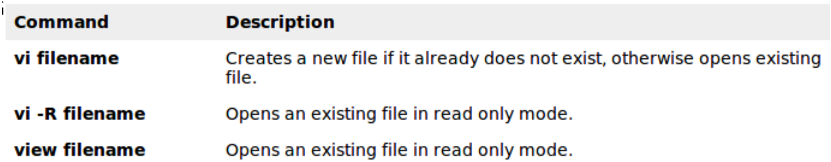
```
~
~
~
~
~                       VIM - Vi IMproved
~
~                        version 7.4.1689
~                    by Bram Moolenaar et al.
~        Modified by pkg-vim-maintainers@lists.alioth.debian.org
~              Vim is open source and freely distributable
~
~                    Help poor children in Uganda!
~           type  :help iccf<Enter>       for information
~
~           type  :q<Enter>               to exit
~           type  :help<Enter>  or  <F1>  for on-line help
~           type  :help version7<Enter>   for version info
~
~
~
~
~
                                                  0,0-1          All
```

- short for: visual editor
- available on all UNIX systems
  - original vi part of BSD Unix
  - written by Bill Joy in 1976
- many derived, improved versions available
- open source vim (vi improved)
  - is part of GNU/Linux
- vi has multiple modes of operation:
  - input mode, command mode, last-line mode

VIM TUTORIAL

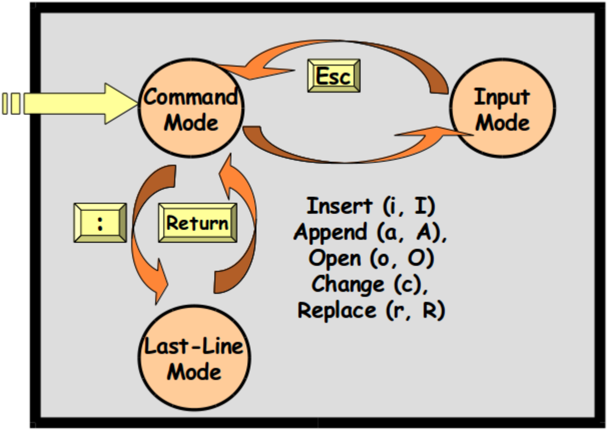
- Online tutorial: vimtutor
- Never start something you can't exit
  - To end vi tutorial in the middle of the session, execute the command :q!
  - :q! = quit without saving
  - :wq = write out (save) and quit
- F1 = help
  - or :help
  - :help <command>
  - :q to exit help window

Starting the vi Editor



Command	Description
vi filename	Creates a new file if it already does not exist, otherwise opens existing file.
vi -R filename	Opens an existing file in read only mode.
view filename	Opens an existing file in read only mode.

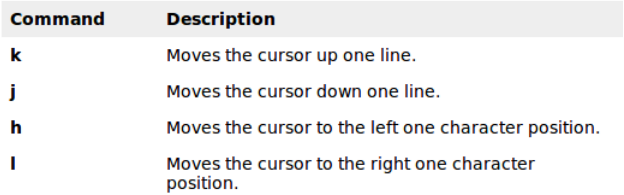
VI EDITING MODES



Getting Out of vi

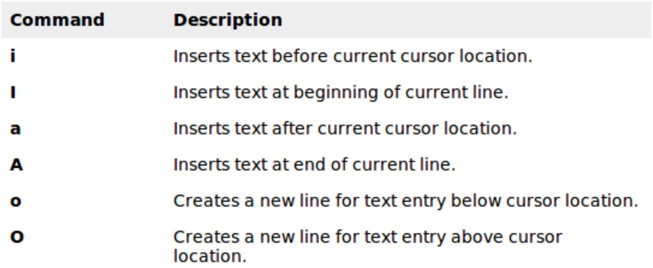
- :q!
- :w
- ZZ command
- :w filename2
- :wq

Moving within a File



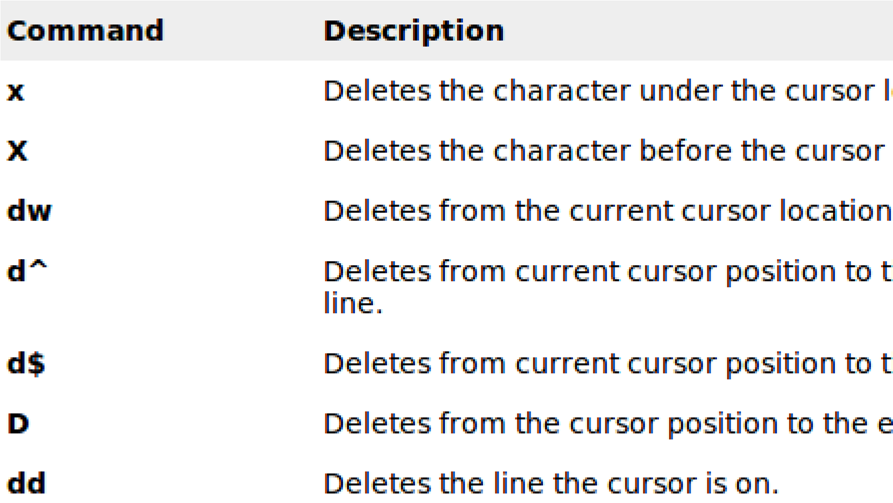
Command	Description
k	Moves the cursor up one line.
j	Moves the cursor down one line.
h	Moves the cursor to the left one character position.
l	Moves the cursor to the right one character position.

Editing Files



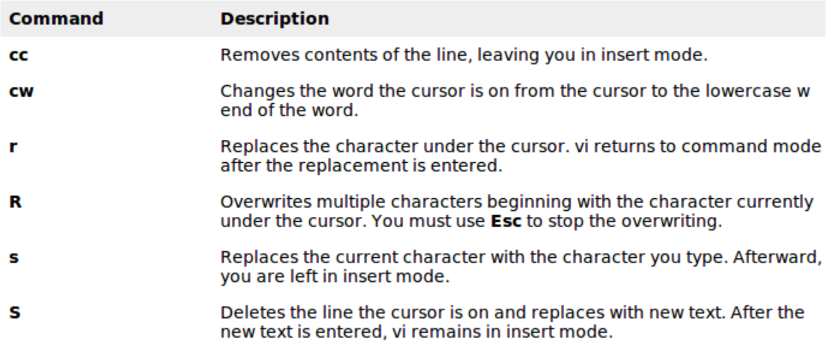
Command	Description
i	Inserts text before current cursor location.
I	Inserts text at beginning of current line.
a	Inserts text after current cursor location.
A	Inserts text at end of current line.
o	Creates a new line for text entry below cursor location.
O	Creates a new line for text entry above cursor location.

Deleting Characters



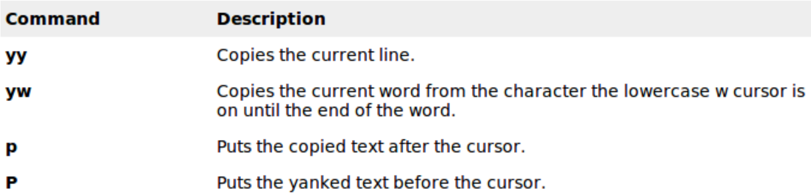
Command	Description
x	Deletes the character under the cursor l
X	Deletes the character before the cursor
dw	Deletes from the current cursor location
d^	Deletes from current cursor position to t line.
d$	Deletes from current cursor position to t
D	Deletes from the cursor position to the e
dd	Deletes the line the cursor is on.

Change Commands



Command	Description
cc	Removes contents of the line, leaving you in insert mode.
cw	Changes the word the cursor is on from the cursor to the lowercase w end of the word.
r	Replaces the character under the cursor. vi returns to command mode after the replacement is entered.
R	Overwrites multiple characters beginning with the character currently under the cursor. You must use **Esc** to stop the overwriting.
s	Replaces the current character with the character you type. Afterward, you are left in insert mode.
S	Deletes the line the cursor is on and replaces with new text. After the new text is entered, vi remains in insert mode.

Copy and Past Commands

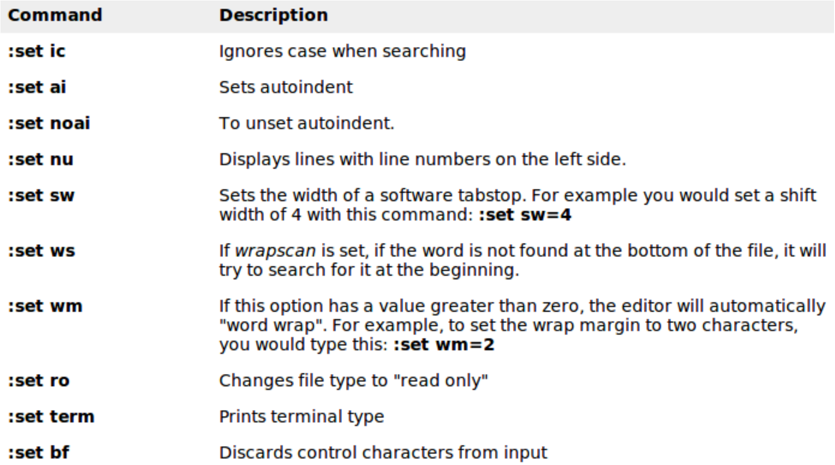


Command	Description
yy	Copies the current line.
yw	Copies the current word from the character the lowercase w cursor is on until the end of the word.
p	Puts the copied text after the cursor.
P	Puts the yanked text before the cursor.

Word and Character Searching

- These two commands differ only in the direction where the search takes place
  - The / command searches forwards **downwards** in the file.
  - The ? command searches backwards **upwards** in the file.
- The n and N commands repeat the previous search command in the same or opposite direction, respectively.

Set Commands



Command	Description
:set ic	Ignores case when searching
:set ai	Sets autoindent
:set noai	To unset autoindent.
:set nu	Displays lines with line numbers on the left side.
:set sw	Sets the width of a software tabstop. For example you would set a shift width of 4 with this command: **:set sw=4**
:set ws	If *wrapscan* is set, if the word is not found at the bottom of the file, it will try to search for it at the beginning.
:set wm	If this option has a value greater than zero, the editor will automatically "word wrap". For example, to set the wrap margin to two characters, you would type this: **:set wm=2**
:set ro	Changes file type to "read only"
:set term	Prints terminal type
:set bf	Discards control characters from input

Replacing Text

- (:s/)
- : s / search / replace / g