

Unix Editors

*Computer Science Department
CS2211a: Software Tools & Systems Programming
Fall 2013
Instructor: Mahmoud R. El-Sakka
Office: MC-419
Email: elsakka@csd.uwo.ca
Phone: 519-661-2111 x86996*

1

Topic 03: Unix Editors

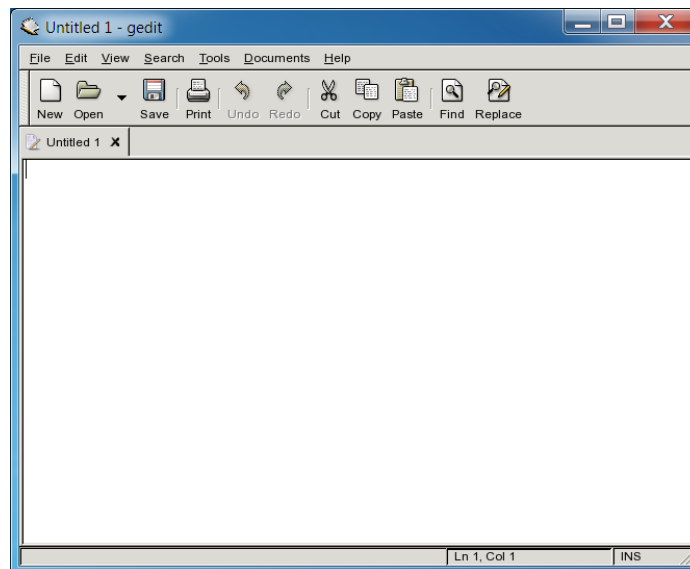
Unix Editors

- Editors in Unix come in two general flavors:
 - ☐ Mode-less editors
 - ☐ Mode editors

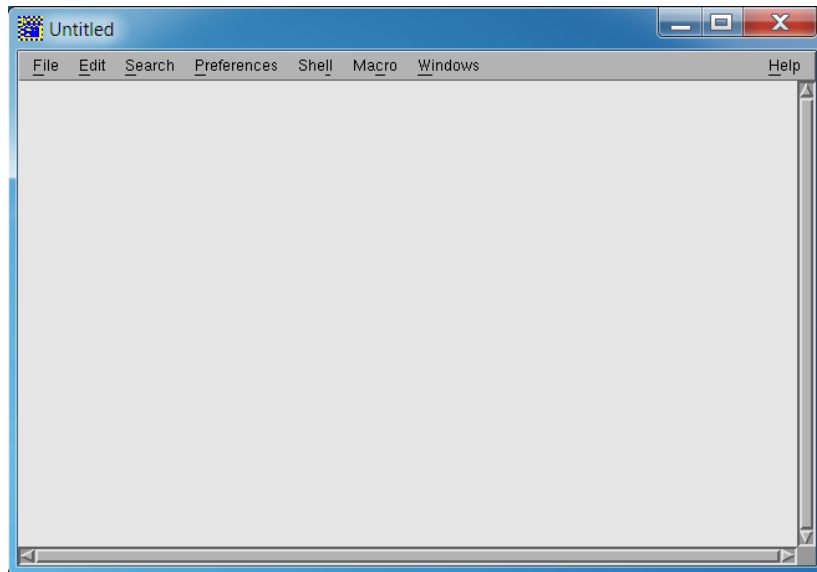
Mode-less Editors

- have **only one mode**
(or **WYSIWYG**: **W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et)
 - positioning and text manipulation are done by special key sequences (like arrow keys and function keys)
 - could also be done by mouse actions or menus
- primary examples: **gedit** , **nedit**, **pico**, **emacs**, **xemacs**

gedit (GNOME)



nedit (GNU)

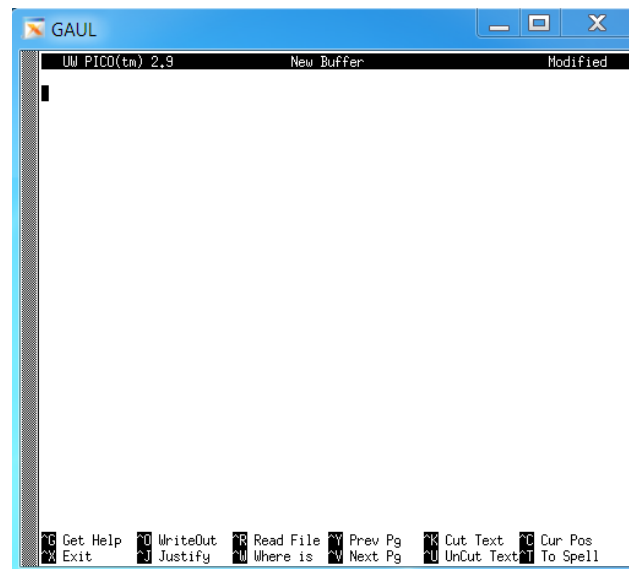


© Mahmoud R. El-Sakka

5

CS 2211: Software Tools & Systems Programming

pico (University of Washington)



© Mahmoud R. El-Sakka

6

CS 2211: Software Tools & Systems Programming

emacs (Richard Stallman-- GNU)

```

emacs@obelix
-----
GNU Emacs 19.34.1 (sparc-sun-solaris2.5, X toolkit) of Wed Jun 17 1998 on obelix
Copyright (C) 1996 Free Software Foundation, Inc.

Type C-x C-c to exit Emacs.
Type C-h for help; C-x u to undo changes.
Type C-h t for a tutorial on using Emacs.
Type C-h i to enter Info, which you can use to read GNU documentation.
('C-' means use the CTRL key.  'M-' means use the Meta (or Alt) key.
If you have no Meta key, you may instead type ESC followed by the character.)

C-mouse-3 (third mouse button, with Control) gets a mode-specific menu.

If an Emacs session crashed recently,
type M-x recover-session RET to recover the files you were editing.

GNU Emacs comes with ABSOLUTELY NO WARRANTY; type C-h C-w for full details.
You may give out copies of Emacs; type C-h C-c to see the conditions.
Type C-h C-d for information on getting the latest version.

-----Emacs: *scratch* (Lisp Interaction)-----All-----
For information about the GNU Project and its goals, type C-h C-p.

```

© Mahmoud R. El-Sakka

7

CS 2211: Software Tools & Systems Programming

xemacs (Richard Stallman-- GNU)

```

emacs: *scratch*
-----
XEmacs 21.4 (patch 12) "Portable Code" (solaris2.8) of Fri Aug 20 2004 on congo
'C-' means the control key, 'M-' means the meta key

Information, on-line help:
XEmacs comes with plenty of documentation...

M-? F: read the XEmacs FAQ (a capital F!)
M-? t: read the XEmacs tutorial (also available through the Help menu)
f1: get help on using XEmacs (also available through the Help menu)
M-? i: read the on-line documentation

M-x describe-project: read about the GNU project
M-x about-xemacs: see who's developing XEmacs

-----XEmacs: *scratch* (Lisp Interaction)-----All-----
Error in init file: Invalid byte code: "variable reference to constant symbol .ty2

```

© Mahmoud R. El-Sakka

8

CS 2211: Software Tools & Systems Programming

Mode Editors

- have multi “modes”, e.g.,
 - input mode
 - allows entry of text
 - command mode
 - allows positioning within the file and
 - allows more sophisticated text modification
- primary examples: `ed`, `ex`, `sed`, `vi`, and `vim`

ed and ex

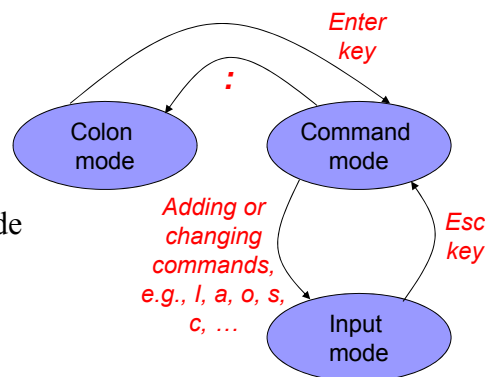
- `ed` is the original line editor
- Later on, it was extended to `ex`
- Does not give you a *local* or *screen picture* of what is in your file (people dislike it)
- Has ability to make massive changes with one command
- Many of its capabilities have been incorporated into `vi` and `emacs`

sed

- `sed` is a stream editor
- It is used to perform basic text transformations on an input stream (input stream can be the output of another program)

vi and vim

- `vi` and `vim` are screen editors
- Have three modes
 - Command mode
 - Colon mode
 - Input mode
- They start in the command mode



vi and vim

■ Saving/quitting commands

□ From colon mode:

- :w write to file
- :wq save and quit
- :q quit (if no change after the last saving)
- :q! exit without save

vi and vim

■ Cursor Movement

□ From command mode:

- arrow keys move the cursor
- h move one char left
- j move one line down
- k move one line up
- l move one char right

- w move to next word
- e move to end of current word
- b move to beginning of previous word
- 0 move to the beginning of the current line
- \$ move to the end of the current line

vi and vim

■ Screen Movement

□ From command mode:

- H move to top of screen
- L move to bottom of screen
- ^F scroll down one page
- ^B scroll up one page
- ^U scroll up one half page
- ^D scroll down one half page

■ Search

□ From command mode:

- / search for the text following /
- ? search backwards for the text following ?

vi and vim

■ Adding Text

□ From command mode:

- o (O) opens new line below (above) the current line
- i (I) inserts text before current char (beginning of line)
- a (A) appends text after current char (end of line)

■ Deletion Commands

□ From command mode:

- x delete character under cursor
- D delete to end of line
- dd delete entire line
- d\$ delete to end of line
- d0 delete to beginning of line
- dw delete the next word
- db delete the previous word

vi and vim

■ Change Commands

□ From command mode:

- s substitute a string for current char (end with ESC)
- r replace current char with another
- R overwrite text (end with ESC)
- C replace to end of current line (end with ESC)
- c0 replace to beginning of current line (end with ESC)
- cw replace the current word (end with ESC)
- cb replace the previous word (end with ESC)

vi and vim

■ Other Commands

□ From command mode:

- . redo last modification command
- u undo the last command

□ From colon mode:

- :w file write the buffer to this file
- :r file read this file into the buffer