ITE315 Module 1 Part B - Text Editors: `Emacs`

Athens State University

Contents

1 Editor Wars

Emacs: The "Other" Editor

```
!START! j 0aua                ! jump to beginning, load 1st char in register A !
!CONT! l 0aub                 ! load first char of next line in register B !
qa-qb"g xa k -l ga 1uz '      ! if A>B, switch lines and set flag in register Z !
qbua                          ! load B into A !
l z-."g -l @o/CONT/ '         ! loop back if another line in buffer !
qz"g 0uz @o/START/ '          ! repeat if a switch was made on last pass !
```

- Like `vi`, `Emacs` was created as a macro package for a line editor; in this case, the 1960's era `TECO` editor on DEC main-frames
- Became popular in the LISP programming community at MIT

The Free Software Foundation & The GNU Project

- The widely-used version of Emacs is *GNU Emacs*
- One of the first open-source projects

In the early 1980s, the LISP programming community at MIT developed a set of specialized workstations for developing LISP applications. This hardware used a version of the Emacs editor as it's shell and text editing environment. Parts of this environment were ported to the UNIX workstations that were popular in that time frame.

One product of that era was GNU Emacs: an open-source version of Emacs that implemented many of the same features as one would find on the LISP machines. It did this by using LISP as the scripting language for the editor (still true today!). Development of this version of Emacs inspired an effort to create an fully open source version of AT&T's UNIX operating system: The GNU Project, where "GNU" is an acronym for "GNU is Not Unix". A large chunk of the GNU Project's software ended up being incorporated into the Linux operating system. For example, Bash is develiverable of the GNU Project.

The Editor Wars

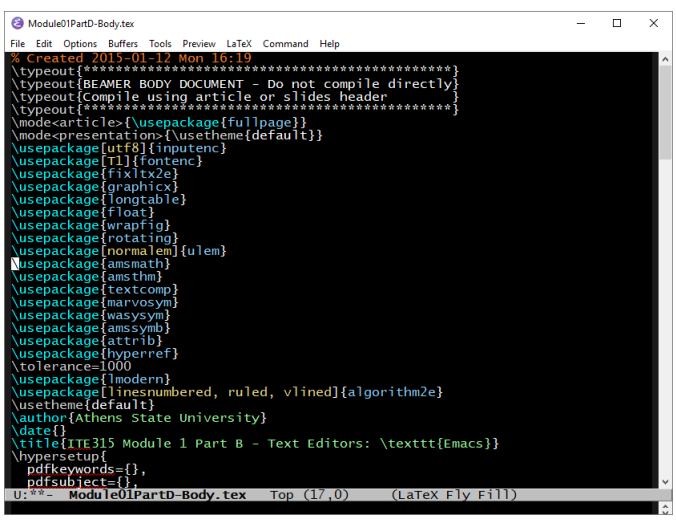
- There has always been a bit of tension within the UNIX community over the choice of programming editor
- One could be either a member of the "Church of Emacs" or the "Cult of VI"
- At the end of the day, the real winner of the Editor Wars was Visual Studio

Why Choose `vi` or `Emacs`

- There is very little difference feature-wise between the two tools
  - Esp. with the introduction of the `vim` version of `vi`
- Dr.Lewis's Opinion: The learning curve for new programmers is less steep for `Emacs` than *vi*

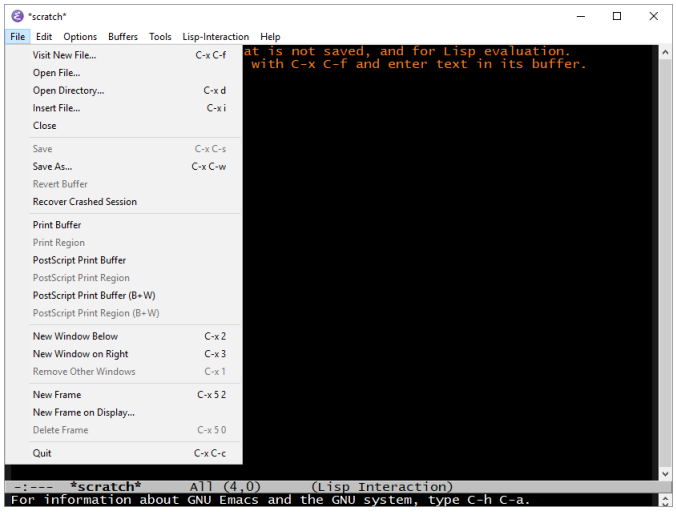
2 A Guided Tour of Emacs

A Guided Tour of Emacs



Note the components of the user interface. Emacs operates on "buffers" stored in "windows" that placed in "frames" in the window manager. The application start in the *Scratch* buffer which is a system buffer you can use as a "scratch pad". Other useful buffers include *Messages* buffer and the *Compilation* buffer.

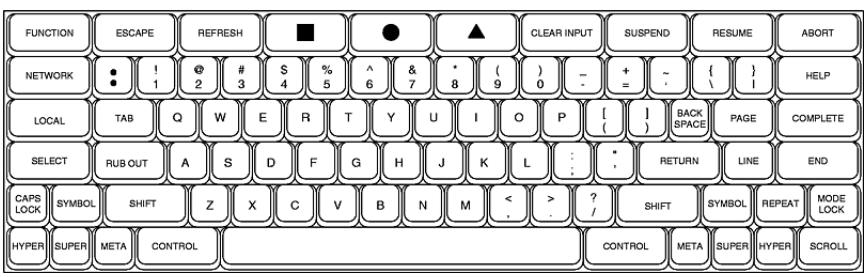
A Guided Tour of Emacs



GNU Emacs is a GUI application. Novice users start using the menu system and pickup on the command set as they gain proficiency. Take note that the key sequences for commands are listed for each menu item.

It's possible to start the editor as a text-only application in the same fashion as `vi`. From the terminal, use the command `emacs -nw` to load the application with "no windows".

A Guided Tour of Emacs: The Keyboard



Understand the Emacs command set requires an understanding of the history of keyboards. The Emacs command set was developed for the keyboard used on the Symbolics LISP Machine. That keyboard had four "modifier" keys: `CONTROL`, `META`, `SUPER`, and `HYPER`. PC keyboards are based upon the standard used with IBM terminals in the 1970s and 1980s that had only two modifier keys: `CTRL` and `ALT`. The Macintosh and later version of Windows adopted a variant of the `SUPER` key for use of the "operating system" key.

Emacs is a "chordal" editor: commands are entered using key sequences called "chords". If you see one mention command such as "CTRL-x", this means that you hold down the `CTRL` and "x" keys at the same time. The `META` key is mapped on modern keyboard to the `ALT` key.

One of the classic jokes about Emacs is that Emacs command key chords were designed to create test subjects for cures for repetitive stress injuries.

A Guided Tour of Emacs: Key Prefixes

- One will find yourself often entering key sequences that begin with one of three *prefix* keys

CTRL-x	Emacs editing commands
CTRL-c	Mode-specific commands
CTRL-u	Repeat prefix

Most of the system-releated commands in Emacs will begin with the *CTRL-x* prefix. For example, the key sequence for exiting the editor is `CTRL-X CTRL-c` while saving a file is done with the `CTRL-x CTRL-s` sequence. The equivalent to the Windows "Save-As" menu option is the write-file command: `CTRL-x CTRL-w`.

Modes in Emacs define specific editing tasks. For example, the editor has a C/C++ programming mode that adds special commands for programming in those languages. I edit the course materials for this course (and others) using Auctex mode for developing materials using the LaTeX documenting formatting system. Mode-specific commands are tied to the `CTRL-c` prefix. For example, I can issue the command to compile a program in C/C++ mode by entering the sequence `CTRL-c c`.

The `CTRL-u` sequence is the repeat prefix sequence. For example, the sequence `CTRL-u 3 CTRL-f` moves the point forward by three characters.

A Guided Tour of Emacs: Basic Navigation

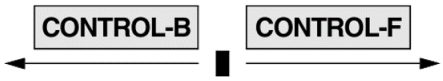


Figure 7-4 Moving the cursor by characters

Basic cursor movement is done with the `CTRL-f` and `CTRB-b` keys: these map to the `forward-char` and `backward-char` Emacs Lisp commands.

A Guided Tour of Emacs: Basic Navigation

Figure 7-5 Moving the cursor by words

The `META-f` and `META-b` key sequences move you forward and backward by words.

A Guided Tour of Emacs: Basic Navigation

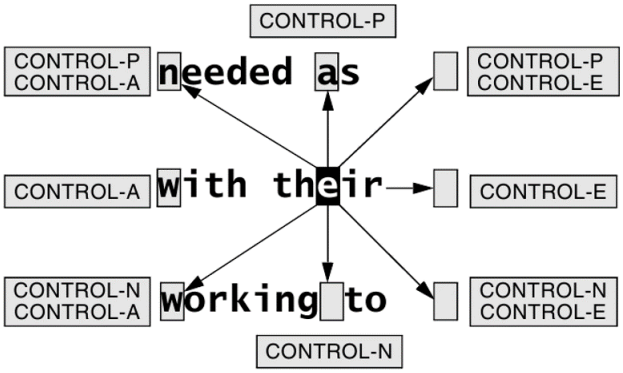
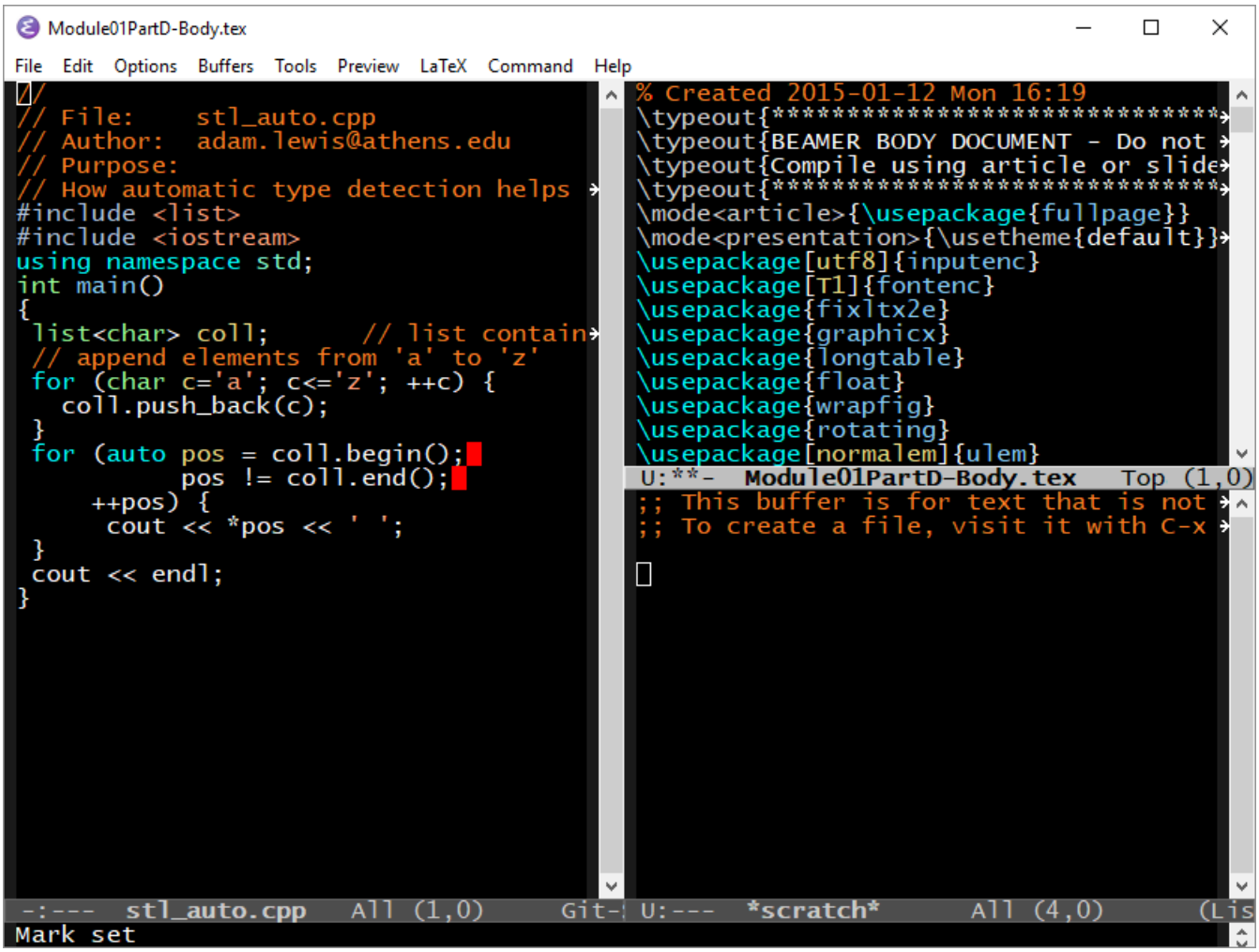


Figure 7-6 Moving the cursor by lines

These key sequences move you around in the buffer by lines. Note that the cursor key are correctly bound for most configurations.

A Guided Tour of Emacs: Buffers, Windows, Frames



- Each file opened in the editor is stored in a *buffer*
- Buffers are displayed in `windows`
- Windows can be displayed in one or more frames

3 Key Points

Key Points

- Text editors: why, what, and how
- The basic use of the `Emacs` editor