Emacs

What is Emacs?

vi and its clones are very functional and powerful editors. However, they are often considered not particularly extensible. ***vim*** is a successful and powerful ***vi*** variant that shrugs this trend, being both extremely extensible and lightweight. However, many users prefer a more *“heavy”* and extensible editor. This is why many people (including the author of this chapter) prefer ***Emacs***.

Emacs takes extensibility up to eleven. Outside of a core of C, the rest of ***Emacs*** is written in a Lisp variant, nearly all of which is exposed to you, so that you may configure it or even extend it at will (many good Emacs Lisp tutorials can be found on the Internet). People have written all sorts of extensions in Emacs Lisp, from syntax highlighting for an obscure language, to a built-in terminal. In fact, there's even a ***vi*** emulation mode within ***Emacs*** (called viper), so you can still get the modal editing that comes with vi, while having access to the power of the ***Emacs*** core.

Like ***vi***, there are many variants of ***Emacs*** (termed *“emacsen”*). However, the one most commonly used (and the only one in Slackware) is GNU Emacs. When people reference *“Emacs”* directly, they almost always mean GNU Emacs.

Unlike ***vi***, ***Emacs*** operates more like a traditional editor by default. This means that most keyboard shortcuts can be performed without repeatedly changing modes. You can open up a file and start typing away without having to learn what the modes do, or forgetting which one you are currently using.

Starting Out

Emacs can be started simply by running the ***emacs*** command in your terminal. When you first start it in a console without arguments, you will see something that resembles this:

Starting Out

If you are in X windows, ***Emacs*** may start a GUI instead of running in your console. If this is the case and you don't want a GUI, you can invoke it with the flag '*-nw*'.

While here, you can browse around using the keyboard arrow keys. Underlined elements are links, and you can activate them by moving over them and pressing Enter. The documentation mentioned is very good, and can help you get your bearings should you have any problems. Also note how they describe key sequences such as Ctrl+H, meaning press the h key while holding down the CTRL key. Same deal with **M-`**, meaning to hold the the **Meta** key (usually Alt) and press the backtick ` key. When they say (e.g.) Ctrl+X Ctrl+C, this means to press the x key while holding down the CTRL key, then press the x key while also holding down the CTRL key. Conveniently, this is also one of the more important commands in ***Emacs***: to close it.

Alternatively, if you call ***emacs*** with a file name as an argument, it will open that file, just like ***vi***. Upon doing this, you will be presented with the contents of the file in question. Here, you can navigate the document using traditional arrow keys and type in information at will without any issues.

Say you make some edits, and you now want to save your file. The following key sequence will do that: Ctrl+XCtrl+S. If you made a new file, you will be prompted for this in what is called the *“minibuffer”*, the blank line below the gray line at the bottom of the screen. Type in the file name of your choice, then hit Enter. If you don't want to save the file, you can press Ctrl+G, which aborts operations that ask for input. Do note that tab-completion is usually available for operations that use the minibuffer.

Should you want to open a new file within your same ***Emacs*** session, type in Ctrl+XCtrl+F. You will be prompted for a file name in the minibuffer. ***Emacs*** doesn't care whether it exists or not. If it doesn't exist, a new buffer will be created for it (the file will be created upon saving with Ctrl+XCtrl+S), or it will be opened as expected. However, the old file will still be open! You can switch back to it using Ctrl+XCtrl+B, entering in the file's name (or more technically, the buffer's name), then hitting Enter.

How to Move Around

Like ***vi***, ***Emacs*** is also older than the arrow keys on your keyboard. Also, like in ***vi***, using the arrow keys to navigate files is also supported. While the ***vi*** movement keys are more ergonomic, ***emacs***'s are more *“mnemonic”*. However, it is still very possible to operate using the main ***Emacs*** keybindings quickly and efficiently. Here is a table of the basic movement keybindings:

Emacs Cursor Movement

Command	**Result**
Ctrl+F	Move the cursor one character to the right (forward)
Ctrl+B	Move the cursor one character to the left (backward)
Ctrl+N	Move the cursor one line down (next)
Ctrl+P	Move the cursor one line up (previous)

Of course, like with ***vi*** it is also possible to repeat these commands with a numeric argument. If you type in **M-1 M-0** Ctrl+P, or Ctrl+U 10 Ctrl+P, the cursor will move ten lines up. If you type in **M-5** Ctrl+F or Ctrl+U 5 Ctrl+F, the cursor will move five characters to the right.

Getting Help

Emacs contains a great deal of documentation, to the point that it is often called a *“self-documenting”* editor. This is because it provides mechanisms for providing users with documentation while you are using it.

Here are some useful functions that display documentation (they all start with Ctrl+H):

Accessing Emacs Documentation

Command	**Result**
Ctrl+H f FUNCTION-NAME Enter	Show documentation for function FUNCTION-NAME
Ctrl+H k Ctrl+X Ctrl+C	Show documentation for the function bound to the keysCtrl+X Ctrl+C

Command	Result
Ctrl+H t	Show the Emacs tutorial
Ctrl+H ?	Show all help-related functions

Ctrl+H **t** is especially useful if you want or need practice using ***Emacs***.

Calling Functions

As noted earlier, ***Emacs*** exports a large number of functions to for interactive use. Some of these, like those opening and saving files, are mapped to keys. Others (like the ones for moving to the beginning and end of lines) are not. To call them, you have to invoke them. Say we want to call the function *"end-of-line"*. We would do this:

M-x end-of-line Enter

And the cursor would move to the end of the line, as the function name suggests.

Emacs Cheat Sheet

While Emacs can be simple to use, its scope can easily be overwhelming. Below are some useful Emacs commands. Some aspects have been simplified, most notably regarding text selection. These concepts, and more, are described the ***Emacs*** manual, and various on-line tutorials. Decent summaries can also be gleaned from web searches.

Emacs Cheat Sheet

Command	Result
Ctrl+F	Move the cursor one character to the right (forward)
Ctrl+B	Move the cursor one character to the left (backward)
Ctrl+N	Move the cursor one line down (next)
Ctrl+P	Move the cursor one line up (previous)
Ctrl+H f FUNCTION-NAME Enter	Show documentation for function FUNCTION-NAME
Ctrl+H k Ctrl+X Ctrl+C	Show documentation for the function bound to the keys Ctrl+X Ctrl+C
Ctrl+H T	Show the Emacs tutorial
Ctrl+H ?	Show all help-related functions
M-`	Access the Menu Bar
Ctrl+G	Cancel the current operation. This is most useful when in the minibuffer.
M-X FUNCTION-NAME Enter	Call the interactive function FUNCTION-NAME
M-1 M- Ctrl+N	Move the cursor ten lines down
Ctrl+U 1 Ctrl+N	Move the cursor ten lines down (same as above)
M-x beginning-of-line	Move the cursor to the beginning of the current line
M-x end-of-line	Move the cursor to the end of the current line
M->	Move the cursor to the end of the buffer

Command	Result
M-<	Move the cursor to the beginning of the buffer
Ctrl+K	Remove text from the cursor to the end of the line and place it into the kill ring
Ctrl+Space	Enter selection mode (use normal motion keys to move around). Press C-space again to leave it.
Ctrl+W	While in selection mode, delete the selected region, and store the result into the kill ring
M-W	While in selection mode, store the selected region into the kill ring.
C-Y	“Yanks” the contents of the kill ring and places them at the cursor's location
Ctrl+/	Undo the previous action. Unlike most other editors, this includes previous undo actions.
Insert	Enable or disable overwriting characters
Ctrl+S asdf Enter	Forward incremental search for the string “asdf”. Repeat Ctrl+S as needed to search for future items, or Ctrl+R (below) to search backwards.
Ctrl+R asdf Enter	Backward incremental search for the string “asdf”. Repeat Ctrl+R as needed to search for future items, or Ctrl+S (above) to search forwards.
M-% old Enter new Enter	Search for each instance of “old” and prompt you to replace it with “new”. You can force replacement of all items by typing ! at the replacement prompt.
Ctrl+X Ctrl+C	Exit Emacs, prompting you to save each unsaved buffer before doing so
Ctrl+X Ctrl+S	Save the currrent buffer to its file
Ctrl+X Ctrl+W new-file.txt Enter	Save the current buffer to a file *“new-file.txt”*

Chapter Navigation

Previous Chapter: vi

Next Chapter: Networking

Sources

- Original source: http://www.slackbook.org/beta

- Originally written by Alan Hicks, Chris Lumens, David Cantrell, Logan Johnson

slackbook, emacs, text editor