COMPUTER POWER USER

Tips & Tricks

Click To Print

Tips & Tricks
November 2008 • Vol.8 Issue 11
Page(s) 98-99 in print issue

Warm Up To Penguins

The Many Editors Of Linux

Whether it's writing a shopping list or tweaking a Linux configuration file, editing text files is one of the more mundane but important tasks that users need to perform on a regular basis. As such, one of the most heated debates among hardcore Linux users is what Linux text editor reigns supreme. The "Vi vs. Emacs" war has been on a continual boil for at least 20 years, while others swear by nano. There are even some pragmatists (we count ourselves among them) who use a mixture of editors depending on the task. So, presented for your inspection are some of the many choices you have when you need to edit a file under Linux.

Groceries, For Example

As a standard for comparison, we'll assume that there's a file called "shopping-list" that we wish to edit, which has the following contents:

```
1 loaf bread
1 jar strawberry jam
1 jar peanut butter
1 pint strawberry ice cream
```

We want to add a quart of milk to the list, remove the peanut butter, and change strawberry to blueberry in the two locations it occurs.

Ed

The ed editor is probably the most "primitive" editor available on a modern Linux system. How primitive? Well, ed dates back to the days before there were computer terminals that could move the cursor around on the screen. Think of the old teletype terminals you see in movies from the '60s, and you're not too far off. Everything you do in ed is done by commands typed to a prompt. Like all the editors we'll look at, you edit a file in ed by typing **$ ed shopping-list**. In other words, you type the name of the editor followed by the file you wish to edit.

When we run ed on our file, we receive the helpful response "83." What in the world is 83? Why, the number of characters in the file, of course! We'll begin by adding the quart of milk to the end of the list. We do this by typing **a** and pressing ENTER, which enters append mode. Then we'd type **1 quart milk** and press ENTER, then a lone period (.) all on the next line to exit append mode. When ed first starts, the append command will append to the end of the file.

Assuming we added the milk at the end, we'll next delete the peanut butter. To do that, we move to the line we want to delete, then use the 'd' command. We type **3**, press ENTER, then type **D** and press ENTER.

Finally, we want to replace strawberry in two locations with blueberry. We can actually do that in a single command, typing **1,$strawberry/blueberry/**. This says, "between line 1 and the end of file ($), substitute blueberry for strawberry." Finally, we type

w to write the file and **q** to quit.

Vi

Although ed may be great if all you have is a dumb terminal, most old-school Linux power users would rather use the vi editor. You can still use all the same ed commands, but you get the advantage of having a constantly updated view of the file. You can also maneuver around inside the file using single keystroke commands. So, to perform the same editing tasks, you'd type **vi shopping-list**, and the window would clear, leaving only the file (or as much of the file as will fit on the screen) displayed. To delete the peanut butter, we press the "J" key twice, which moves the cursor down to the line in question. Then, we type **dd** to delete the line. The line will instantly disappear from the display, leaving the cursor blinking on the ice cream line. Typing **$** moves the cursor to the end of the line, and **a** enters append mode. Then we can press ENTER, type in our line for milk, and press ESCAPE to exit append mode.

The substitution command in vi is identical to ed, with one exception: place a colon (:) before the command to run it. The display will instantly update to show the change. The "w" and "q" commands write the file and exit the editor, respectively, but again, you must put a colon in front of each command.

Nano

Many people these days prefer nano to vi, as it's a little simpler to use. When you invoke nano on the shopping list, you can open a main window that displays the file and a small help window below that shows commonly used commands. You can navigate around the file using the arrow keys, so to add the milk, we press the Down arrow four times and start typing. Then we press the Up arrow to move to the peanut butter line and press CTRL-K to delete it. Finally we move our cursor back up to the top and press CTRL-\, which is the replace string command. This causes nano to prompt for the string we're searching for, at which point we'd type **strawberry** and press ENTER. Nano will ask for the string to replace it with; in this case, we type **blueberry**. When we press ENTER again, nano will move to the first occurrence and ask if we want to replace it. You can press Y for yes, N for no, or A to replace all occurrences in the file. Press CTRL-X to exit, and nano will ask if you want to save the file.

Emacs

The joke about Emacs is that it's as much an operating system as it is an editor. It's huge compared to ed and vi, requiring much more memory. It also starts up much slower than ed or vi. Because of this, vi is often preferred for quick file edits, reserving Emacs for more complex tasks, such as software development. Like vi, when we invoke emacs on our file, we get a window that displays the contents of the file with the cursor at the upper left.

Emacs is an incredibly powerful editor that you can run from a terminal window or as a GNOME or KDE application.

Most commands in Emacs involve using CTRL, ALT, or both. For example, to move down to the peanut butter line, we press CTRL-N twice. This drops the cursor to the beginning of the third line. We then press CTRL-K (kill line) twice. The first time deletes the contents of the line, leaving it empty. The second time deletes the blank line.

You could then press ALT-> (go to end of file) and enter the quart of milk. Pressing ALT-< returns you to the beginning of the file, and you could then use the query-replace function to swap strawberry for blueberry, just like we did in nano. To do this, you'd press ALT-X, which prompts you for the command name; type **query-replace** and press ENTER. Emacs then prompts for the string to search for, at which point you'd type **strawberry** and press ENTER. Then, type **blueberry** and press ENTER. Like nano, Emacs moves the cursor to the first occurrence of strawberry and asks if you want to change it. You can press Y for yes, N for no, or ! to replace the rest of the occurrences in the file. Then press CTRL-X and CTRL-S to save the file.

Although we explained how to navigate around files using keystrokes in Emacs, you can also use the mouse with modern, window-system aware versions of Emacs. Emacs is an incredibly powerful editor, with built-in functions such as a mail reader and the ability to view two or more files at once on the same screen, moving back and forth between them.

As you can see, Linux has a wide variety of editors available, including some we didn't even mention, such as GNOME's Text Editor. Selecting which one to use can be a matter of taste, or using the right tool for the right job. There's really no wrong choice you can make.

by James Turner

Infinite Loop:
Dead Sea Scrolls Get Digital Upgrade

If you've ever wanted to get up close and personal with the Dead Sea Scrolls, you'll soon be able to do so online. A team of Israeli scientists has undertaken the task of snapping digital photographs of the two-millennia-old text, which consists of approximately 9,000 individual fragments. The Israel Antiquities Authority, the scrolls' caretaker, estimates the project will take over two years to complete.

Source: africa.reuters.com/odd/news/usnLR699315.html