

Editors for the shell

# EASY EDITORS

David Dan Calin, Fotolia

More is not always better, especially at the command line. Lean text editors such as nano, JOE, and ee fit Linux novices and experts alike.

BY HEIKE JURZIK

**T**ext editors on Linux are two a penny. The two major desktop environments, KDE and Gnome, have their own graphical counterparts, all of which support mouse control and offer a multitude of functions with menus to match. But how will you manage if your GUI environment is broken or if you need to use SSH to work on a remote machine?

Classical text editors for the command line can be useful because you can con-

trol them with just the keyboard. Many users prefer one of the “major league” editors – Vi(m) [1] or (X)Emacs [2] – but if you prefer a leaner approach, you may find that a program like nano, JOE, or ee could be just as useful.

To establish your Linux environment, you can set a number of environmental variables. To define a default editor, which will automatically be launched by many programs whenever something needs to be edited, you need to define the `$EDITOR` variable.

To define the `$EDITOR` variable for the current shell session, you can just run, for example, the `export EDITOR = nano` command. To make this variable setting permanent, you need to add the com-

**Table 1: Command-Line Parameters for nano**

<code>nano -V (--version)</code>	Displays the version number of the text editor.
<code>nano -h (--help)</code>	Displays the command-line parameters.
<code>nano -B (--backup)</code>	Creates automatic backups; you can identify a backup by the tilde (~) added to a file name.
<code>nano -H (--historylog)</code>	Creates a file called <code>~/.nano_history</code> that stores terms you have searched. Because this feature is persistent between sessions, you can search for older keys; to do so, press Ctrl+W and then Up arrow or Down arrow to scroll through the history.
<code>nano -I (--ignorerfiles)</code>	Ignores the global and personal configuration for the current session.
<code>nano -s program</code> <code>(--speller=program)</code>	Defines the spell-checking tool; for example, <code>nano -s "ispell -x -T latin1" file.txt</code> .
<code>nano -v (--view)</code>	Opens a file in read-only mode.
<code>nano -x (--nohelp)</code>	Hides Help at the bottom of the screen.

## GLOSSARY

**pico:** The pico text editor is part of the Pine [4] mail and news clients, which was the default email client on Unix systems for many years. Pico is easy to use and displays the commands at the bottom of the screen.



Figure 1: The nano default setup looks a lot like pico.

mand to your bash configuration file, `~/bashrc`.

**Lean and Fast - nano**

Nano (“nano’s ANOther editor” [3]) is a free pico clone that was developed in 1999 by Chris Allegretta.

Nano emulates the look and feel of its predecessor, while adding a number of new features. All popular Linux distributions include nano; on Debian GNU/Linux, the editor replaces its predecessor, pico, which is not included with the distribution for licensing reasons.

To create a new file, you just launch the editor by typing `nano` at the command line.

As an alternative, you can pass the filename to the command to open the file when the program launches.

If you want to jump to a specific line of the text when you launch the editor, just specify which line you want at the command line:

```
nano +10 file.txt
```

**Editing Commands**

Nano shows the name of the file you are currently editing at the top of the screen, and common editing commands are displayed at the bottom of the window (see Figure 1).

The circumflex (^) represents the Ctrl key – the shortcut Ctrl + G displays Help, for example.

Some functions are accessible via the meta key (which is shown as M- in the on-screen Help). Depending on your

keyboard settings, this is either the Esc or Alt key.

**Additional Help**

Besides the very readable on-screen Help, nano comes with a man page that supplies more advanced information – for example, how to control nano at launch time by passing in command-line parameters.

Table 1 is an overview of the most important options. As an alternative, you can edit the configuration files to modify the text editor’s behavior.

To edit behavior globally (for all users), you need to edit the `/etc/nanorc` file to set up nano.

To modify your personal version, you just copy this file to your home directory as a template:

```
cp /etc/nanorc ~/.nanorc
```

The entries in the configuration file are self-explanatory for the most part, with useful comments in the file.

You can use the configuration file to specify whether to use automatic indenting (`set autoindent`), whether to let the editor create automatic backups (`set backup`), which spell-checking tool to use (`set speller program`), and many other things. Also, a couple of entries for syntax highlighting are in the configuration file. The examples for C, HTML, and TeX are easy to follow, and you just need to remove the comment signs to reinstate them. On the basis of these examples, it should be easy enough to set up syntax highlighting for other programs.

**Cotton-Eyed JOE**

JOE [5] is another lean, easy-to-use text editor. The name is an acronym for “JOE’s Own Editor,” referring to the program’s developer, Joseph Allen. Again, any popular Linux distribution will include the editor.

To launch the text editor without specifying a file to edit, you just type `joe`. See Figure 2.

Like nano, JOE will jump directly to a line number if you specify `+ <line number>` as a command-line option:

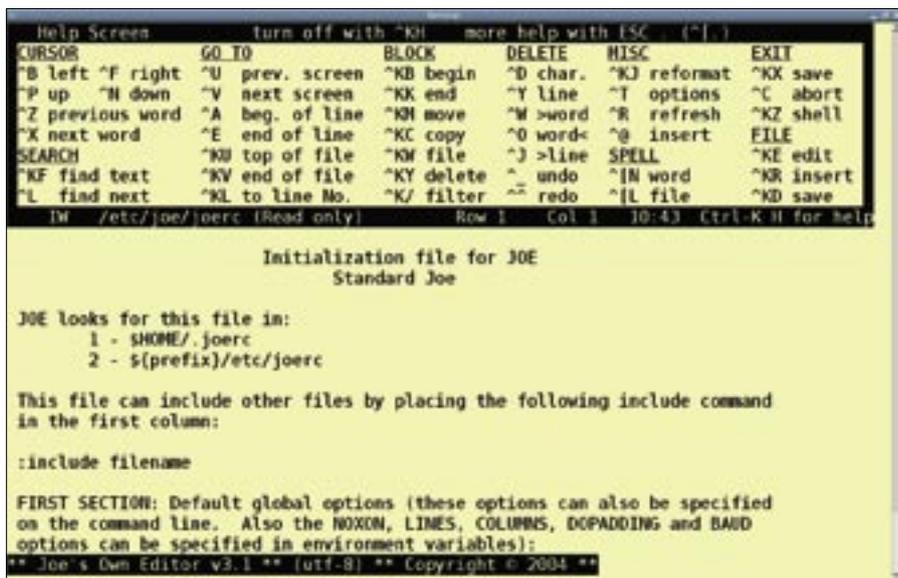


Figure 2: If you type `-help` when launching the program, JOE automatically displays the on-screen Help. Those of you who recall the good old days of computing might be reminded of the DOS word processor, WordStar.

```
joe +30 file.txt
```

At the top of the window, you can see both the file name as well as the cursor position.

After modifying a file, a *(Modified)* tag is added to the file name.

### Customizing

The line also tells you that pressing Ctrl + K and then the H key (Ctrl + K-H; again, the circumflex stands for the Ctrl key) will display the on-screen Help.

Configuration files are used to customize JOE. Again, a global configuration file, */etc/joe/joerc*, is available that you can copy as a template for your personal configuration:

```
cp /etc/joe/joerc ~/.joerc
```

The configuration file syntax is slightly quirky. There is no such thing as a comment sign (such as the pound sign). Instead, to enable a function, you simply remove the blank at the start of the line. For example, if you want JOE to automatically display Help on launching, just delete the space in front of the *-help* entry. If you prefer not to create backup copies, just remove the space in front of *-nobackups*, and so on.

Parameters that modify JOE's startup behavior are listed in Table 2.

### ee - Easy Editor

The lean ee [6] text editor really is easy to use and – like nano and JOE – is included with any modern Linux system. Also, as with the other editors, you can specify a file to load and a line number to jump to when launching the program:

```
ee +13 file.txt
```

ee displays the on-screen Help at the top of the screen by default.

### Menu

Again, the circumflex ^ stands for the Ctrl key, and ^/ means Esc. This key

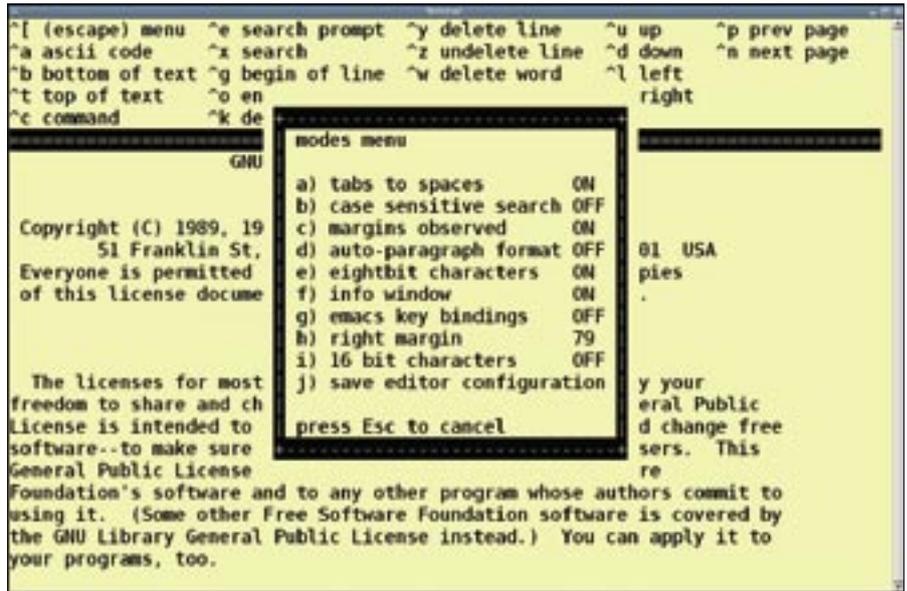


Figure 3: The Esc key takes you to a menu from which you can access common functions.

opens the menu that lets you access common functions (Figure 3).

### Navigate

Inside the menu, you can press the arrow keys to navigate to an entry and then press Enter when you get there.

Another way to navigate is to press the indicated button to access the menu item you need – pressing A, for example, quits ee.

Pressing Esc again will then close out the menu.

### Basic Features

ee does not support more advanced features like syntax highlighting or setup files, and the editor has a fairly spartan collection of command-line options.

But once you know the commands, you can tell the editor to hide the on-screen Help by pressing *-i*, and you can switch off the bold characters in help texts and menus by pressing *-h*.

One keyboard shortcut takes some extra getting used to: Ctrl + C does not cancel the program but displays an input box for editor commands. The program shows the available commands at the top of the screen.

Besides the programs we have looked at today, Linux has a whole bunch of editors for the command line, all of which have their strengths and weaknesses. Which editor you opt for is a matter of personal taste, or possibly a question of how powerful your computer is.

What's most important for efficient use is that your fingers can remember the keyboard shortcuts. ■

### INFO

- [1] "Vim Workshop" by Heike Jurzik, *Linux Magazine*, November 2006, <http://www.linux-magazine.com/issue/72>
- [2] Emacs: <http://www.gnu.org/software/emacs/>
- [3] nano: <http://www.nano-editor.org/>
- [4] Pine Mail and Newsreader: <http://www.washington.edu/pine/>
- [5] JOE: <http://joe-editor.sourceforge.net/>
- [6] ee: <http://mahon.cwx.net/>

Table 2: Joe Command-Line Parameters

<i>joe -help</i>	Automatically displays Help at the top of the screen.
<i>joe -nonotice</i>	Hides the licensing information at the bottom of the screen.
<i>joe -nosta</i>	Hides the status line at the top of the window.
<i>joe -nobackups</i>	Disables automatic backups.
<i>joe -overwrite</i>	Enables type-over mode.

### THE AUTHOR

Heike Jurzik studied German, Computer Science and English at the University of Cologne, Germany. She discovered Linux in 1996 and has been fascinated with the scope of the Linux command line ever since. In her leisure time you might find Heike hanging out at Irish folk sessions or visiting Ireland.

