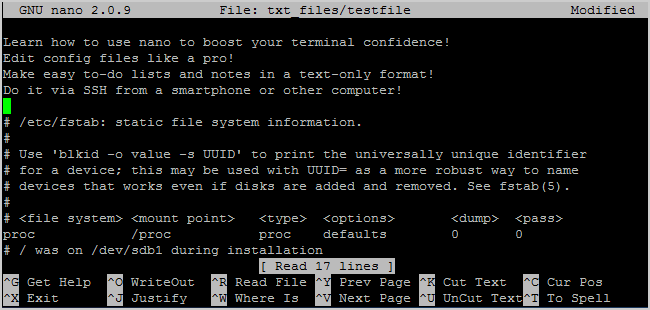
<http://www.howtogeek.com/howto/42980/the-beginners-guide-to-nano-the-linux-command-line-text-editor/>

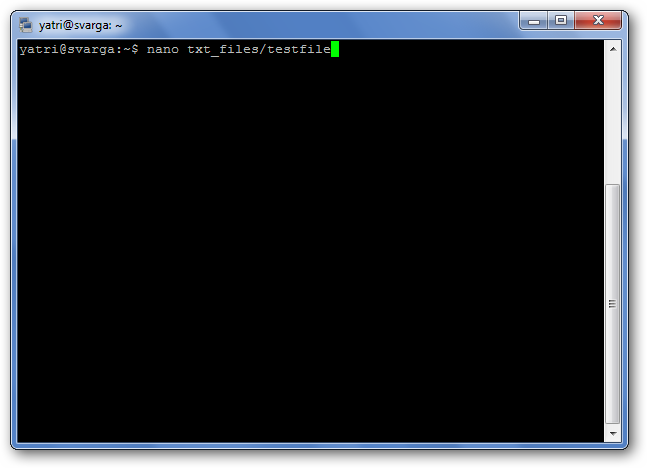
**[The Beginner’s Guide to Nano, the Linux Command-Line Text Editor](http://www.howtogeek.com/howto/42980/the-beginners-guide-to-nano-the-linux-command-line-text-editor/" \o "The Beginner’s Guide to Nano, the Linux Command-Line Text Editor)**



New to the Linux command-line? Confused by all of the other advanced text editors? How-To Geek’s got your back with this tutorial to Nano, a simple text-editor that’s very newbie-friendly.

When getting used to the command-line, Linux novices are often put off by other, more advanced text editors such as vim and emacs. While they are excellent programs, they do have a bit of a learning curve. Enter Nano, an easy-to-use text editor that proves itself versatile and simple. Nano is installed by default in Ubuntu and many other Linux distros and works well in conjunction with sudo, which is why we love it so much.

Running Nano



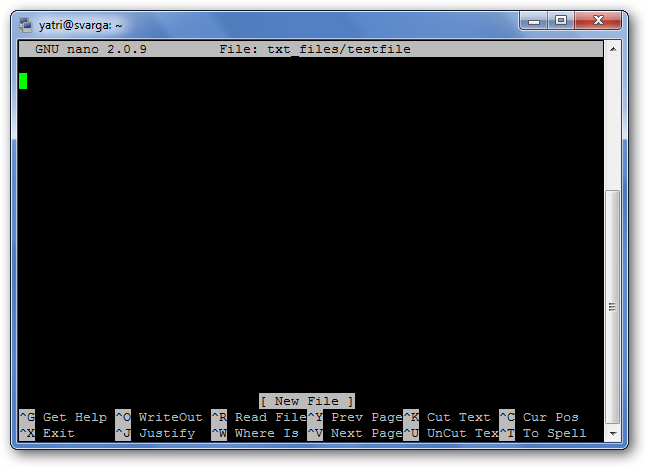
You can run nano in two ways. To open nano with an empty buffer, just type in “nano” at the command prompt.

You can also use the following syntax:

nano /path/to/filename

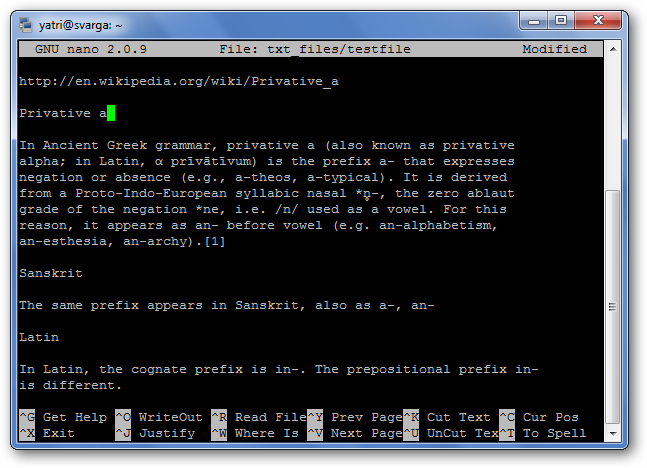
Nano will follow the path and open that file if it exists. If it does not exist, it’ll start a new buffer with that filename in that directory.

Let’s take a look at the default nano screen.



At the top, you’ll see the name of the program and version number, the name of the file you’re editing, and whether the file has been modified since it was last saved. If you have a new file that isn’t saved yet, you’ll see “New Buffer.” Next, you’ll see the contents of your document, a body of text. The third-line from the bottom is a “system message” line that displays information relevant to the program executing a function. Here, you can see that it says “New File.” Lastly, the final two rows at the bottom are what make this program very user-friendly: the shortcut lines.

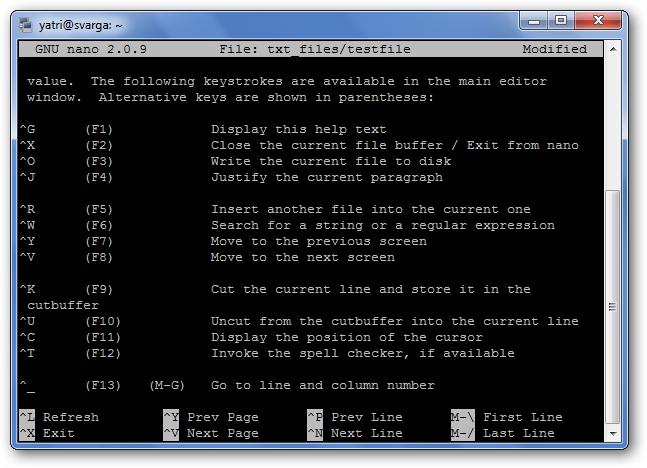
It’s a WYSIWYG editor; “what you see is what you get.” What you type directly goes into the text input, unless you modify it with a key like Control or Meta. It’s pretty simple, so type some text out, or copy something and paste it into your terminal so we have something to play with.



Shortcuts

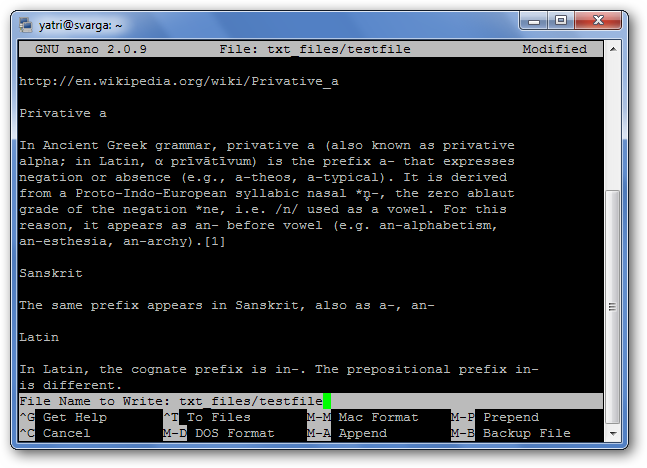
Program functions are referred to as “shortcuts” in nano, such as saving, quitting, justifying, etc. The most common ones are listed at the bottom of the screen, but there are many more that aren’t. Note that nano does not use the Shift key in shortcuts. All shortcuts use lowercase letters and unmodified number keys, so Ctrl+G is NOT Ctrl+Shift+G.

Hit Ctrl+G to bring up the Help documentation and scroll down to see a list of valid shortcuts.

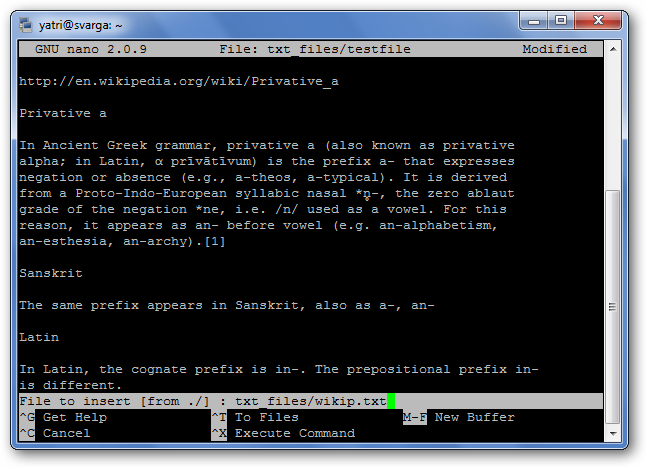


When you’re done looking at the list, hit Ctrl+X to exit help.

Let’s say you’re working on a new text file, or “buffer,” and you want to save it. This is called “writing out” and is executed by hitting Ctrl+O. You’ll be prompted for a filename to use, and the shortcuts at the bottom will change to reflect what you can enter to complete this particular command.



If you want to insert the contents of another file into your current buffer, you’d type Ctrl+R.



You can cancel both of the previous commands by typing Ctrl+C.

You can hit Escape twice instead of holding down the Control key, if you have trouble doing that. There are also some commands that require use of the Meta key. On most keyboard layouts, Meta equates to the Alt button.

When you want to quit nano, you just hit Ctrl+X. Nano will politely ask you if you want to save your buffer, and you can cancel this action as well.

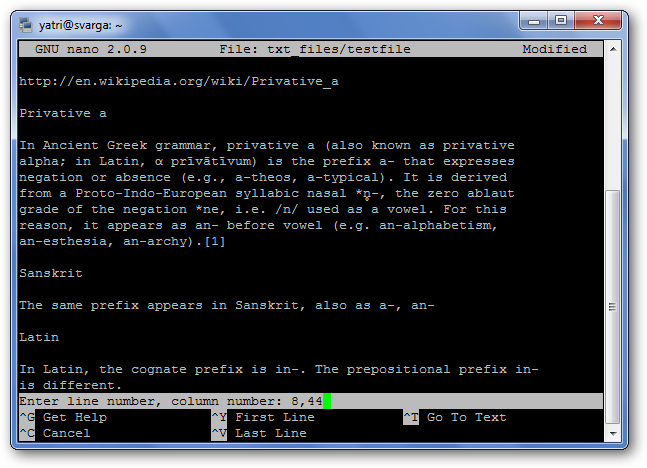


Navigation

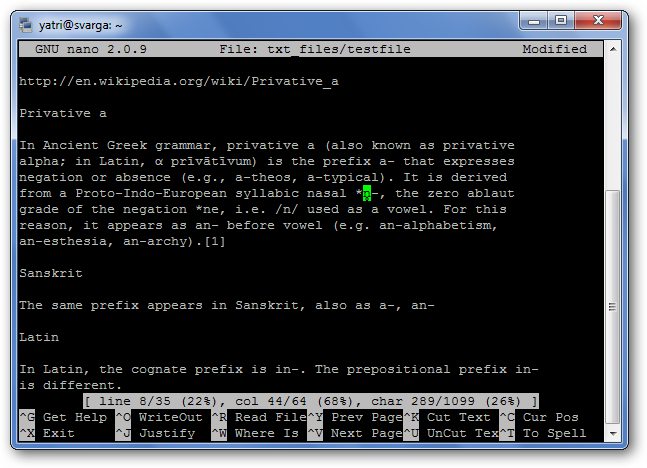
Now that we’ve got a hang of shortcuts, let’s get used to moving around a text file very quickly. Of course, you can always use the Home, End, Page Up, Page Down, and the arrow keys to get around, but that requires moving your fingers from the letters that we all love so much.

To move the cursor forward or backward, you can type Ctrl+F and Ctrl+B. To move up and down one line at a time, you can type Ctrl+P and Ctrl+N. In other words, you can use those keys instead of the Right, Left, Up, and Down arrows, respectively. Missing the Home and End keys? You can use Ctrl+A and Ctrl+E. Want to move pages at a time instead? Ctrl+V moves down a page, and Ctrl+Y moves up a page.

But wait, there’s more! To move forward and backward one word at a time, you can use Ctrl+Space and Meta+Space (remember, that’s Alt+Space). And, if you’re really in a rush, you can hit Ctrl+\_ and then type in the line number, a comma, and the column number to jump straight there.

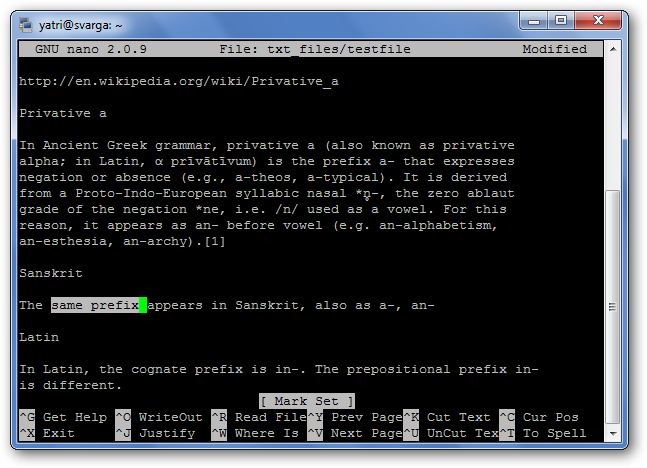


If you want to see where your cursor currently is, sort of like nano-GPS, hit Ctrl+C.



Copying, Cutting, and Pasting

When we want to copy text in graphical environment, we highlight it with the cursor. Similarly, in nano we “mark” it by using the Ctrl+^ command. You simply move the cursor to where you want to start marking, and then you hit Ctrl+^ to “set” it. This will mark everything between the starting point up to and NOT including the cursor.



Note that the cursor is on the empty space, and copying/cutting will not include this space. You can also mark backwards from your “set” point. Be careful, however, as you can edit text while your marking. If you messed up, just hit Ctrl+^ again to unset the marker and you can start over.

To copy the marked text, hit Meta+^. If, instead, you want to cut the text, hit Ctrl+K.

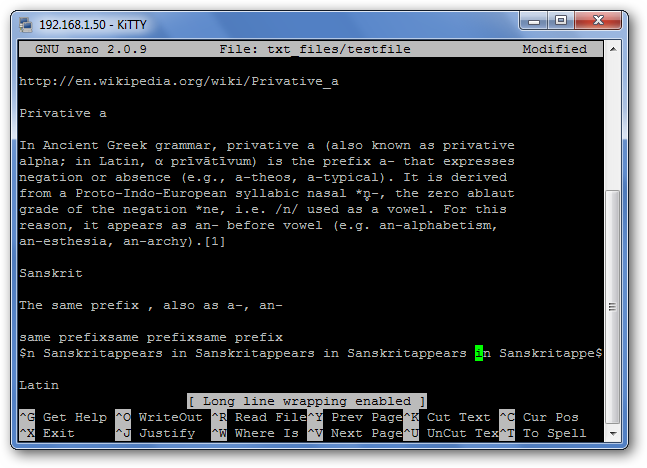
To Paste your text, move the cursor to a suitable position and hit Ctrl+U.



If you want to remove an entire line of text, simply hit Ctrl+K without highlighting anything. This sometimes comes in handy when editing configuration files.

Some Extra Shortcuts

You know how in notepad, you can force long lines of text to wrap-around into what look like paragraphs? You can toggle that feature in nano with the Meta+L shortcut. Since line wrapping is set to “on” by default, this usually comes in handy in the opposite way; for example, you’re writing a config file and want to disable line-wrapping.



You can see that the line the cursor is on has a “$” at both the beginning and end. This signifies that there’s more text both before and after the portion being displayed on screen.

If you want to search for a text string, hit Ctrl+W, and enter your search term. This search can then be cancelled mid-execution by hitting Ctrl+C without destroying your buffer.



The previous search term appears in the square brackets, and leaving the line blank and hitting Enter will repeat that last search.

And after you get really comfortable, you can turn that helpful section at the bottom off by hitting Meta+X to get more screen space for editing!



Some History

Nano was designed to be similar in look and feel to another program called Pico. Pico was the default text editor of Pine, an email program from back in the day that wasn’t distributed with a GPL-friendly license. This meant that redistribution was somewhat of a fuzzy area, and so the TIP project was born. “TIP Isn’t Pico” added some functionality that Pico lacked and was licensed for free distribution, and over time, became the nano we love to use today. For more information, check out the [Nano Project’s History section](http://www.nano-editor.org/dist/v2.2/faq.html#1.4) on their FAQ.

The strength of nano lies in its simplicity of use. The shortcuts work just like in GUI-based word processors like Word and Open Office, so it’s just a matter of learning which ones do what. Everything outside of that is just simple text-editing. Next time you have to edit stuff on the command-line, we hope that you’ll be more comfortable with it now that you’ve gotten familiar with nano.