Emacs

Emacs is another powerful editor. Some people really find themselves drawn to vi while others thoroughly enjoy using emacs. It's a bit of a rivalry in the Linux world, actually. Experiment with emacs and vi to see which one works for you. You can't make a bad choice as they are both great editors.

emacs [file] - Edit file.

When reading emacs documentation know that C-<char> means to hold down the Ctrl key while pressing <char>. For example, C-h means to hold down the Ctrl key while pressing the h key. If you see C-h t, that means to hold down Ctrl key while pressing the h key, release the Ctrl key and then type the letter t.

When you see M-<char>, that means hold down the "meta" key, which is the Alt key, while pressing <char>. You can also substitute the Esc key for the Alt key. So M-f translates to holding down the Alt key and pressing f or pressing and releasing Esc followed by typing the f key. You may need to use Esc for the meta key since Alt may be intercepted by your terminal program, for instance. If you want to simply things, always use Esc for the meta key as it will work in all situations.

Here are some helpful emacs commands.

C-h - Help.

C-x C-c - Exit. While holding down Ctrl press x, continue to hold down Ctrl and press c.

C-x C-s - Save the file.

C-h t - Emacs has a nice built-in tutorial.

C-h k <key> - Describe key. Use this to get help on a specific key command or key combination.

Navigating

C-p - Previous line.

C-n - Next line.

C-b - Backward one character.

C-f - Forward one character.

M-f - Forward one word.

M-b - Backward one word.

C-a - Go to the beginning of the line.

C-e - Go to the end of the line.

M-< - Go to the beginning of the file.

M-> - Go to the end of the file.

Deleting Text

C-d - Delete a character.

`M-d` - Delete a word.

Copying and Pasting

`C-k` - Kill (cut) the rest of the current line of text. To kill the entire line, position the cursor at the beginning of the line.

`C-y` - Yank (or paste) from the previously killed text.

`C-x u` - Undo. Keep repeating for multi-level undo.

Searching

`C-s` - Start a forward search. Type the text you are looking for. Press `C-s` again to move to the next occurrence. Press `Enter` when you are done searching.

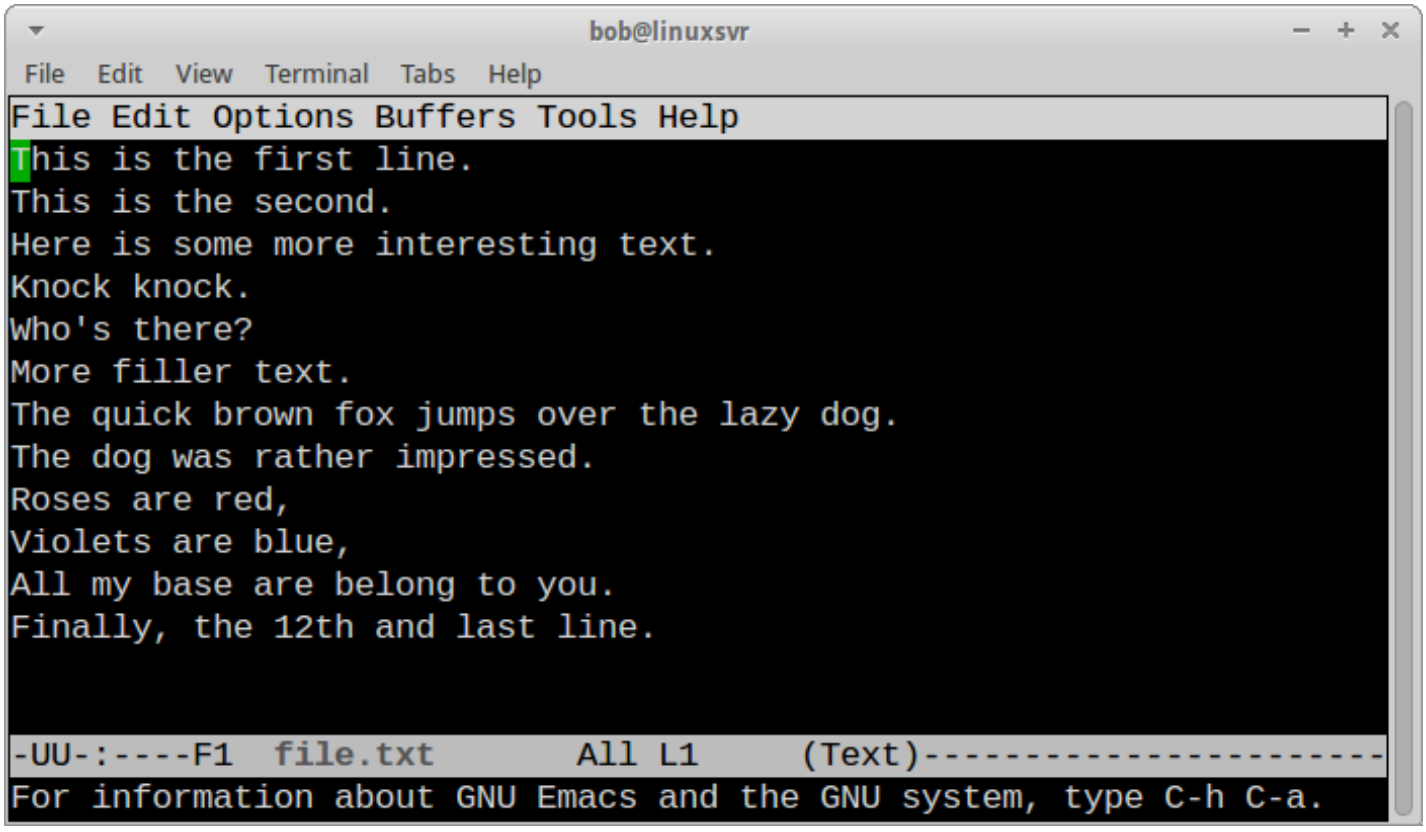
`C-r` - Start a reverse search.

Repeating

Like `vi`, `emacs` provides a way to repeat a command.

`C-u N <command>` - Repeat <command> `N` times.

For instance, to kill three lines of text type `Ctrl-U 3 Ctrl-k`.



You have only scratched the surface with the `vi` and `emacs` editors. There is so much more to learn if you are interested. Both editors have features that include macros, global replace, and more. Entire books have been written on each of the these editors.

Graphical Editors

So far you have learned about command line editors that are appropriate to use when you connect to a server via ssh. However, if you are running Linux as a desktop operating system you might be interesting in some graphical text editors and word processors. Here are some for your

consideration.

- `emacs` - Emacs has a graphical mode, too.
- `gedit` - The default text editor for the Gnome desktop environment.
- `gvim` - The graphical version of `vim`.
- `kedit` - The default text editor for the KDE desktop environment.

If you are looking for a MicroSoft Word replacement, consider AbiWord or LibreOffice. LibreOffice not only includes a word processor, but it is a complete office suite with a spreadsheet program, a database, and presentation software.

If you are looking for a source code editor to aid in computer programming, look at Geany, jEdit, or Kate. Sublime Text is another option. It is a commercial product that runs on Windows, Mac, and Linux.

Specifying a Default Editor

Some commands rely on the `$EDITOR` environment variable to tell them which program to use for editing. Since cron's primary purpose is to schedule jobs, it delegates the task of editing files to another program. The `crontab -e` command invokes the editor specified by the `$EDITOR` environment variable. You can set `$EDITOR` in your personal initialization files to ensure your favorite editor is used, be it `nano`, `emacs`, `vi`, or something else.

```
$ echo $EDITOR
vi
```