Work in Progress

Please, feel free to edit/add to this article, or contact me if you need to discuss something about it.

First steps

Now that you have been introduced to the CLI, let's get started. The first thing you need to do is either open up a terminal emulator, or start your system in runlevel 3. Once you are in a terminal, the next section lists some basic commands that you will frequently be using:

I. Basic commands

- TAB: This is the TAB key on your keyboard, pressing this will **auto-complete commands**, and file or directory paths. This is a great time saver.
- pwd: This command **prints the current working directory** which assists the user to know which directory they are working in.
- ls: This command **lists** all the files and directories in a folder.
- cd: This command **changes the directory**.
- su: This command **switches the user** of the currently running virtual console. Running this without giving any other option switches you to the root user.
- rm: This **removes** files or folders

Use the rm command with extreme caution and never run this as root unless you are absolutely sure you need to, and know what will be deleted

- man: This command opens up the **manual** for a program, for example running "***man cd***" will give you the manual for the cd command. A HOWTO is here How to search and read Manpages efficiently

Now that you know what the commands are, I recommend reading the manual for each of them by using the correct command above. Once you have completed that, you can move onto text editors.

II. Choosing a text editor

In this section we will talk about the different text editors, so let's get started.

The following text editor applications are included with the "Full Installation" of Slackware.

App Name	Description	Difficulty
ed		Advanced
elvis		
sed		Advanced
jed		
joe		
jove		
nano	A simple CLI text editor with no mode and easy to locate on-screen commands	Simple
vim		Advanced
emacs		Advanced
kate		Simple
gvim		Advanced

In general the editors in Linux/Unix can be divided into three categories: vi derivatives, emacs derivatives and those which are similar to the typical Windows editors. Ed and sed are ancestors of vi; vim/gvim and elvis are its successors (please note that in Slackware vi is actually a symlink to elvis.) Jed, joe and jove are emacs derivatives whereas nano and kate have their roots in the Windows-world. Both Emacs and vi (and their clones) are worth the effort of learning because it helps you to get your work done. Note that sed (the stream editor) works in a completely different way.

a. Nano

Nano is a very easy to use CLI text editor, that I recommend for beginners. To start Nano, you simply type in the command nano. Once nano is up, you can just begin typing into the document. To save your work, you simply press Ctrl+X on the keyboard, give the file a name, and press enter. While nano is open, you will notice along the bottom of your screen several letters preceded by ^. These are the different keyboard shortcuts within Nano. ^ represents the Ctrl key, while the letter represents the corresponding key on your keyboard. One more thing to cover is how to open up an existing file in Nano, to do this you run Nano as shown below.

```
nano /path/to/text/file
```

Doing this opens the text file you entered the path to in Nano.

b. Emacs

Emacs is a highly customisable text editor. Thanks to a wide range of extensions, it offers other functionalities, such as IDE, a project planner, news and mail reader, file manager with an ftp client built in. This section, however, will guide you through a basic usage of Emacs.

Emacs Initialisation

Emacs can be started from a menu of your desktop environment or window manager. If you want to run without the X Server, you need to start it with the `-nw` flag:

```
user@darkstar:~$ emacs -nw
```

or to create a new file:

```
user@darkstar:~$ emacs -nw my_new_file
```

As Emacs relies heavily on keyboard shortcuts, it is essential to cover them first to get you started.

Emacs Keybindings

In the Emacs world, a widely used convention is to refer to Ctrl and Alt as C and M respectively. Consequently, the C-h combination denotes clicking Ctrl+H simultaneously, whereas M-x means pressing the Alt+X keys at the same time. You will also encounter more complex keybindings, such as C-x s which means pressing Ctrl+X and then s. Similarly, C-x C-c refers to pressing Ctrl+X and then Ctrl+C. Other commonly used keys are S (= Shift) and Esc. Here's a list of most basic keybindings in Emacs:

Keys	Description
C-h	Access help
C-h t	Start Emacs tutorial
C-h r	Read Emacs manual
C-x C-f	Create a new file
C-x C-s	Save
C-x C-w	Save as
C-x u	Undo
M-w	Copy
C-y	Paste
C-x C-c	Exit Emacs

The above information should get you started with Emacs. For more advanced usage, please visit this link.

c. Vim

Vim and the other vi-clones are very different from other types of editors. vi has two modes, **insert-mode** and **command-mode**.
This means, the editor distinguishes between *inserting text* and *moving and making changes*.
One needs only the alphanumeric keys to work with vi. Therefore vi and it's clones are primarily useful for people who can touchtype.

When you start vi you're in command-mode, with i you switch to insert-mode. From insert-mode you get with ESC back to command-mode. Leave vi with :q in command-mode.

Vim comes with a very useful tutorial which I strongly recommend for newbies. Open a terminal and simply type

```
vimtutor
```

Useful external links

- After you've made the first steps you may want to use the *vi cheat-sheet*
- A nice interactive vim tutorial
- More information is in the *wikibook*
- *this page* will give you an even deeper insight.
- Here's a very helpful text about *efficient text editing*

The help system

Vim has it's own very large help-system. When you're using Vim and need help, you can (in command-mode) type

```
:help
```

If you need help regarding special subject (here as an example macros), simply type

```
:help macro
```

For an overview of the Vim-help type

```
:help help
```

and if you're totally confused the command

```
:help!
```

will surely help you.

The dokuwiki Plugin

If you use Vim when writing pages for this dokuwiki, you can find a helpful plugin which you'll only have to copy into the /usr/share/vim/vim73/syntax/ directory.

You can load this plugin when editing a dokuwiki file with the command:

```
set ft=dokuwiki
```

If you need syntax highlighting for the <key> plugin of dokuwiki (which is not by default supported by the dokuwiki plugin), you can insert the lines:

```
syn region dokuwikiKeyboard start="<key>" end="</key>"
hi link dokuwikiKeyboard Comment
```

to the plugin.

Chapter Navigation

Previous Chapter: Introduction

Next Chapter: Shells

Sources

- Originally written by Andrew Daniel
- Contributions by Markus Hutmacher and Marcin Herda

work in progress, author b1ueb1aze