

BOOTSTRAP – A STARTER FOR THE IMPATIENT

Getting Help from Linux

Once you have Linux installed and running, the most important piece of information you need is how to get help. There are megabytes of documentation right there on your Linux system that can answer all of your questions, if you just know how to access it!

`man` Pages

The standard way to get help, which works on any UNIX system, is a command called **man** (think “manual”, as in user handbook). To get help on the `grep` command, for example you would type: `man grep`

This displays a help screen describing the program and its usage. Man pages, as they are known, are formatted on the fly from special source files, and are displayed using the **pager program** `less`. You can scroll back and forth line by line with the arrow keys or whole screens using the `PgUp` and `PgDn` keys. When you’re done reading, **press Q to quit** (the `escape` key will *not* get you out of `less`) and you’ll get back to the command prompt.

Man pages are stored in a categorized database. These are the standard categories:

1. User Commands
2. System Calls
3. Subroutines
4. Devices
5. File Formats
6. Games
7. Miscellaneous
8. System Administration

Sometimes you will need to know this in order to find the man page you are looking for. For example, there is a user command `passwd` that can be used to change your login password. If you type `man passwd` you will get the man page for this command, since it

is in section 1. However, there is also a file called `passwd` that stores the password database on your system. The format for this file is explained in a man page in section 5, File Formats. To get to it, you need to type `man 5 passwd` so the man command will look in section 5 rather than starting at section 1.

Man pages are often very detailed and might tell you far more than you wanted to know. If you just want a quick idea of what a command does, use the **whatis** command. For example, `whatis grep` returns:

```
grep, egrep, fgrep (1) - print lines matching a pattern
```

This shows you that `grep`, and its related commands `egrep` and `fgrep`, have man pages in section 1, and are used to “print lines matching a pattern.” If you were searching for a command to copy files, you would know that this isn’t it and you could move on. Or you could consult the man page for details on using these commands.

Related to the `whatis` command is **apropos**, which searches the `whatis` database for keywords. If you know what you want to do but don’t know the command, `apropos` will give you list of related commands. For example, if you want to copy files, try `apropos copy` and get:

```
bcopy (3)    - copy byte strings

copy (1)     - copy data to/from a class from/to a Unix file.

copysign (3) - copy sign of a number

cp (1)      - copy files

cpio (1)    - copy files to and from archives

dd (1)      - convert a file while copying it

dvicopy (1) - produce modified copy of DVI file

fcopy (n)   - Copy data from one channel to another.
```

Your list will probably be much longer than this, but by reading through it you will find that the command you want is `cp` for copying files.

If the `apropos` and `whatis` commands don't work on your system, it may mean that the `whatis` database has not been created, or is corrupt. You can rebuild it by running `/usr/sbin/makewhatis` as **SuperUser**. Note that the `apropos` command will accept only one parameter. You cannot search for multiple keywords. If you try, only the first will be used. The `whatis` command will accept multiple search words, but searches for them individually rather than as a group (in other words, combining them with OR rather than AND).

`info`

Many of the utilities included with Linux, especially those created by the **GNU Project**, have documentation in the form of info pages. Info pages look similar to man pages, but there is a lot more to them. Like the man pages, info documents are intended to act as your online "manual", your guide to the GNU/Linux system. But info documents are integrated with each other in a hypertext database that can be read like a book (it even has a table of contents) and accessed from your terminal.

Where both man pages and info pages exist, the info documentation is probably more accurate. To get info on the `grep` command, type:

```
info grep
```

The info viewer looks and behaves much like GNU Emacs. Most of the emacs key combinations will work in the info viewer. If you aren't familiar with emacs, you should probably get familiar with it, as its control keys and syntax are inescapably emulated by hundreds of Linux programs. (See ***Editing Files with Emacs*** for a basic tutorial and list of commands).

Navigating info documentation isn't terribly complicated, nor is it completely intuitive. Thankfully, there is a self teaching tutorial included with the package. At any time you can type a question mark (?) or `Control-h` to access a list of available commands, in

case you need help with the navigation keys. Typing `h` by itself takes you to the info tutorial, the first few pages of which should tell you all you need to know for simple operation. The keys to remember are `SPACE` to page forward, `DELETE` to page backward, and `ENTER` to activate a hyperlink. (Links are normally preceded by an asterisk). `L` takes you to the Last node (page) visited, like the “Back” button in your web browser.

`/usr/doc/*` (or `/usr/share/doc/*`)

Many programs are just too complex to store all their help information in a man page. The `/usr/doc` directory is where these programs will store their information. Sometimes there will just be a text file, sometimes a program will have an entire subdirectory devoted to it. Also in this folder you may find copies of the Linux How-To's to help you with common tasks. Take these with a grain of salt. The information in the How-To's is usually very technical and oriented more toward system administrators than novice users. (Part of the purpose of *Control-Escape* is to translate some of that knowledge into plain English).

Check for a folder called HTML here. If you find it, open up your web browser of choice and use it to view the documentation. You will feel much more comfortable, I'm sure. I did.

The Linux Documentation Project

The LDP is your best and most complete source for Linux information. I don't think a day goes by that I don't visit them for something. Make sure to check their mirrors for one close to you to save bandwidth on their server.

Similarities Between Linux and DOS

Comparing DOS & Windows to Linux

Despite what some folks will tell you, there is a lot of superficial similarity between DOS and Linux, and also between MS Windows and the X Window System. We can leverage the knowledge we already have to get up and running faster in Linux.

What's The Same?

- **Hierarchical Directory Structure:**

DOS, Windows, and Linux all store information in files which are organized into directories (also called folders). Directories may contain files and other directories. Although, the commands differ, files and directories can be created, deleted, renamed, copied, moved and listed from the command prompt.

- Start a program by typing its name at the command prompt.
- Windows can be minimized, maximized, closed using buttons or menus.

What's Different?

Here are some major points you need to know:

- **Linux is Case Sensitive!:** To Linux, a capital letter is NOT the same as a lower case letter. The file Junk.txt is NOT the same file as junk.txt. Watch this carefully! It will cause you monstrous headaches!
- **Linux is quiet:** If a command succeeds, it will just drop you back at the command prompt without a status message. This can be confusing to new users.
- **Linux does not confirm anything:** In DOS or Windows, if you try to delete a file or folder it will ask for confirmation ("Are you sure you want to do that?"). Linux won't ask, it'll just do it. This makes it very easy to accidentally destroy a file, or the entire file system. Be SURE of what you type!
- **Slashes:** Linux uses a forward slash (/) wherever DOS would use a backslash (\). Linux uses a dash (-) to indicate command switches where DOS would use a slash (/).
- **Search Path:** DOS always checks the current directory first, then looks at the PATH environment variable. Linux never looks in the current directory but searches *only* the directories listed in PATH. To run a program in the current directory you have to type `./` before it.

Some other major differences require deeper explanations:

- ***The Linux (Virtual) File System***

Win/DOS to Linux Quick Reference

Below is a quick table roughly mapping Windows or DOS commands to their Linux equivalents. A more comprehensive reference may be added later. Remember that Linux is *case sensitive!*

DOS/Windows Command	Linux Equivalent	What it does
<code>attrib</code>	<code>chmod</code>	Change file or directory attributes or permissions. Note that file attributes on DOS are completely different from permissions on Linux.
<code>cd</code>	<code>cd</code>	Change the current directory. Similar syntax.
<code>cls</code>	<code>clear</code>	Clear the terminal screen
<code>copy</code>	<code>cp -i</code>	Copy files. Similar syntax. Without the <code>-i</code> , <code>cp</code> will not prompt for destructive actions like file replacement. If the target exists, it is overwritten.
<code>del</code>	<code>rm -i</code>	Delete files. Without the <code>-i</code> , <code>rm</code> will not ask for confirmation.
<code>deltree</code>	<code>rm -R</code>	Delete a directory and all its contents, including subdirectories.
<code>dir</code>	<code>ls</code>	List the files in the current directory, or a named directory.
<code>md</code>	<code>mkdir</code>	Create a new subdirectory. Similar syntax.
<code>mem</code>	<code>free -t</code>	Display a summary of current memory usage and availability.
<code>more</code>	<code>more less</code>	Display the contents of a file one page/screen at a time. The <code>less</code> utility allows scrolling back and forth.
<code>move</code>	<code>mv</code>	Move a file.
<code>rd</code>	<code>rmdir</code>	Delete an empty subdirectory. Similar syntax.
<code>ren</code>	<code>mv</code>	Rename a file or folder.
<code>type</code>	<code>cat</code>	Display the contents of a file.
<code>ver</code>	<code>uname -sr</code>	Display the name and version of the OS kernel.

Shutting Down

To log out and return to the `Login:` prompt, just type:

```
logout
```

If you're ready to turn off the computer, you need to instruct Linux to shut it down properly so that your files are not damaged. To turn off the power type:

```
shutdown -h now
```

To reboot the computer type:

```
shutdown -r now
```

Note, however, that the `shutdown` command can only be run by the root user. If you try to run it as a regular user, even using the `su` command, you'll get the message:

```
shutdown: must be root.
```

On Linux systems, an alternative is to use the commands `/sbin/halt` to power down the computer and `/sbin/reboot` to reboot. These commands can be run by regular users with the proper permissions. This is *not* universally correct on UNIX systems, so it is recommended not to try this on non-Linux systems.

Source : <http://www.control-escape.com/linux/bootstrap.html>