

man VS. info

Say you want some online documentation on your Linux™ system. What do you use, `man` or `info`? I have met many people (in real life, email lists, and newsgroups) who seem to think that `man` and `info` present identical information. While there is certainly some overlap, this is not true. By far the most important thing is that **texinfo is easier to write than `man`**, but here is a quick rundown of the differences:

	man	info
Formatting:	Normally formatted by <code>n/groff</code>	Preformatted from <code>texinfo</code> source
Goal:	Concise coverage of usage/options	In-depth guide to using the software
Linking:	Text references	Support for hyperlinks
Organization:	Section and program/function name	Program name or package (e.g., <code>fileutils</code>)
Size:	Normally one small file	Often several large files for complex software
Writing:	Must know a <code>roff</code> formatting language	Describe contents with simple tags (much like HTML)

History

The history of the `man` command goes all the way back to early 1970's with the [first AT&T UNIX releases](#). The manual pages were written in one of the `roff` family of text formatting languages. (Incidentally, text processing is why UNIX ever made it out of Bell Labs according to [some rumors](#).) Having documentation written in a text-formatting language meant that specific pages could be read online with `man` or printed out without losing any formatting. Unfortunately, to quote from Dale Dougherty and Tim O'Reilly in [UNIX Text Processing](#):

The formatting markup language required by `nroff` and `troff` is quite complex, because it allows detailed control over the placement of every character on the page,...

Contrast that statement with this quote from the Free Software Foundation's [texinfo manual](#):

A bit of history: in the 1970's at CMU, Brian Reid developed a

program and format named Scribe to mark up documents for printing. It used the @ character to introduce commands, as Texinfo does. Much more consequentially, it strived to describe document contents rather than formatting, an idea wholeheartedly adopted by Texinfo.

The [GNU Project](#) found two problems with `man`:

- **practical**: the `man` system was cumbersome to use on a large project (such as a compiler or an operating system)
- **philosophical**: there was no Free `nroff` to format `man` pages

Characteristically, the FSF attacked on both fronts. There were already several high-quality Free projects that dealt with text processing. They developed a generalized system for documentation that would allow you to write one file, then

create a printed document with the normal features of a book, including chapters, sections, cross references, and indices. From the same Texinfo source file, you can create a menu-driven, online Info file with nodes, menus, cross references, and indices. You can also create from that same source file an HTML output file suitable for use with a web browser, or an XML file. ¹

This became the standard documentation for all GNU software, including `bash` and `gcc`. The typically more concise `man` output was replaced with the `--help` option to all GNU software, and now `man` pages for most GNU software are generated automatically with [help2man](#). Meanwhile, the GNU `roff` formatter, `groff`, reached a stable release in 1991 and is used to format `man` pages from software packages that do not include `info` documentation.

Using `man` and `info`

A recurring complaint from non-`emacs` users is that the `info` program is complicated and non-intuitive, while `man` just gets the job done (`emacs` has a built-in `info` reader). To these users, I always recommend [pinfo](#), an `info` *and* `man` reader with `lynx`-like key bindings. It is now installed by default in several Linux™ distributions.

References

¹ "Using Texinfo" *GNU Texinfo 4.2* http://www.gnu.org/manual/texinfo-4.2/html_node/Using-Texinfo.html

Questions or comments? Email [joshuadfranklin\(at\)yahoo.com](mailto:joshuadfranklin@yahoo.com)