

# GET HELP FROM THE COMMAND LINE

The basic commands you need to get advice from the Linux command line

**I**n our beginner's guide to the command line (*The MagPi* #54), we looked briefly at 'man', the manual you can access from the command line.

The man tool is so important that we think it deserves a more thorough explanation. And man isn't alone in offering help on the command line. Other commands like **whatis**, **info**, and **apropos** all offer support and assistance. And let's face it, support and assistance are what you will often need at the command line.

Even seasoned coders don't always know the correct command to type into the Linux terminal. This guide is all about the various ways to get help at the command line, so no matter what command you come

across, you'll be able to find out more information on how to use it.

**man**

Your first point of call for getting help on the command line is man (short for 'manual'). Enter **man** followed by the name of a command to get detailed information about it. For instance, enter:

**man passwd**

...and you will see detailed information about the tool used to change your password. Man screens are displayed one page at a time. Press the **SPACE** bar to

The synopsis gives a brief outline of how to use the command. In this case, you need to enter the command, an option, and an account name. The parts in brackets are optional

The description gives a detailed outline of the tool. It also offers information on how it works

Many commands have options, typically a hyphen followed by a letter (or double-hyphen followed by a word). You'll find each option outlined in detail in the man page

```

pi@raspberrypi:~$ man passwd
passwd(1)                                User Commands
passwd - change user password
SYNOPSIS
passwd [-options] [LOGIN]
DESCRIPTION
The passwd command changes passwords for user accounts. A normal user may only change the password for his/her own account, while the superuser may change the password for any account. passwd also changes the account or associated password validity period.
Password Changes
The user is first prompted for his/her old password, if one is present. This password is then encrypted and compared against the stored password. The user has only one chance to enter the correct password. The superuser is permitted to bypass this step so that forgotten passwords may be changed.
After the password has been entered, password aging information is checked to see if the user is permitted to change the password at this time. If not, passwd refuses to change the password and exits.
The user is then prompted twice for a replacement password. The second entry is compared against the first and both are required to match in order for the password to be changed.
Then, the password is tested for complexity. As a general guideline, passwords should consist of 6 to 8 characters including one or more characters from each of the following sets:
- lower case alphabetic
- digits 0 thru 9
- punctuation marks
Care must be taken not to include the system default erase or kill characters. passwd will reject any password which is not suitably complex.
HINTS FOR USER PASSWORDS
The security of a password depends upon the strength of the encryption algorithm and the size of the key space. The legacy UNIX System encryption method is based on the IBM DES algorithm. More recent methods are now recommended (see ENCRYPT METHOD). The size of the key space depends upon the randomness of the password which is selected.
Compromises in password security normally result from careless password selection or handling. For this reason, you should not select a password which appears in a dictionary or which must be written down. The password should also not be a proper name, your license number, birth date, or street address. Any of these may be used as guesses to violate system security.
You can find advices on how to choose a strong password on http://en.wikipedia.org/wiki/Password_strength
OPTIONS
The options which apply to the passwd command are:
-a, --all
This option can be used only with -S and causes show status for all users.
-d, --delete
Delete a user's password (make it empty). This is a quick way to disable a password for an account. It will set the named account passwordless.
-e, --expire
Immediately expire an account's password. This in effect can force a user to change his/her password at the user's next login.
-h, --help
Display help message and exit.
-i, --inactive INACTIVE
This option is used to disable an account after the password has been expired for a number of days. After a user account has had an expired password for INACTIVE days, the user may no longer sign on to the account.
Manual page passwd(1) line 1 (press h for help or q to quit)

```

```

File Edit Tabs Help
pi@raspberrypi: ~
File: "manpages". Mode: bash, Up: {dir}
BASH(1)          General Commands Manual          BASH(1)

NAME
  bash - GNU Bourne-Again Shell

SYNOPSIS
  bash [options] [command_string | file]

COPYRIGHT
  bash is Copyright (C) 1989-2013 by the Free Software Foundation, Inc.

DESCRIPTION
  Bash is an sh-compatible command language interpreter that executes
  commands read from the standard input or from a file. Bash also incor-
  porates useful features from the Korn and C shells (ksh and csh).

  Bash is intended to be a conformant implementation of the Shell and
  Utilities portion of the IEEE POSIX specification (IEEE Standard
  1003.1). Bash can be configured to be POSIX-conformant by default.

OPTIONS
  All of the single-character shell options documented in the descrip-
  tion of the set builtin command can be used as options when the shell
  is invoked. In addition, bash interprets the following options when it
  is invoked:

  -c      If the -c option is present, then commands are read from the
         first non-option argument command_string. If there are argu-
         ments after the command_string, they are assigned to the
         positional parameters, starting with $0.
  -i      If the -i option is present, the shell is interactive.
  -l      Make bash act as if it had been invoked as a login shell (see
         INVOCATION below).
  -r      If the -r option is present, the shell becomes restricted
         (see RESTRICTED SHELL below).
  -s      If the -s option is present, or if no arguments remain after
         option processing, then commands are read from the standard
         input. This option allows the positional parameters to be
         set when invoking an interactive shell.
  -o      A list of all double-quoted strings preceded by $ is printed
         on the standard output. These are the strings that are sub-
         ject to language translation when the current locale is not C
         or POSIX. This implies the -n option: no commands will be
         executed.
  [-]ID [shopt_option]
         shopt_option is one of the shell options accepted by the
         shopt builtin (see SHELL BUILTINS COMMANDS below). If
         shopt_option is present, -o sets the value of that option: +o
         unsets it. If shopt_option is not supplied, the names and
         values of the shell options accepted by shopt are printed on
         the standard output. If the invocation option is +o, the
         output is displayed in a format that may be reused as input.

```

The info screens are similar to man, and in many cases, they offer the same content. But some info screens (like bash) provide much more detailed content

move to the next page, and press Q to exit the page and return to the command prompt.

Man pages can be a bit tricky to read at first, but you'll soon get the hang of it.

At the top are the Name, Synopsis, and Description sections. Read these to get an overview of the command. Below them you'll find options and parameters; read these carefully to discover ways to expand your usage of each command. It's a good idea to use man on any commands you know, and read the manual for any new Linux commands you come across.

You can even read a man page for man:

```
man man
```

Press H in the man screen to view a summary of navigational key presses. These are worth learning so you can do more than press space to move to the next page.

Man's lesser-known partner is 'info', which is used to display information pages associated with commands. Sometimes these are the same as the man pages. In other cases they provide a different description. Try these:

```
man bash
info bash
```

While **man bash** gives you a brief description of the GNU Bourne-Again Shell and the options used with the bash command; **info bash** gives the whole history and hundreds of pages of detailed information.

Press H on an info screen to view the controls for navigating such long documents. As well as SPACE to move down, you use DELETE to go back a screen, TAB to highlight links, and RETURN to use them. Press Q to exit the help screen.

```

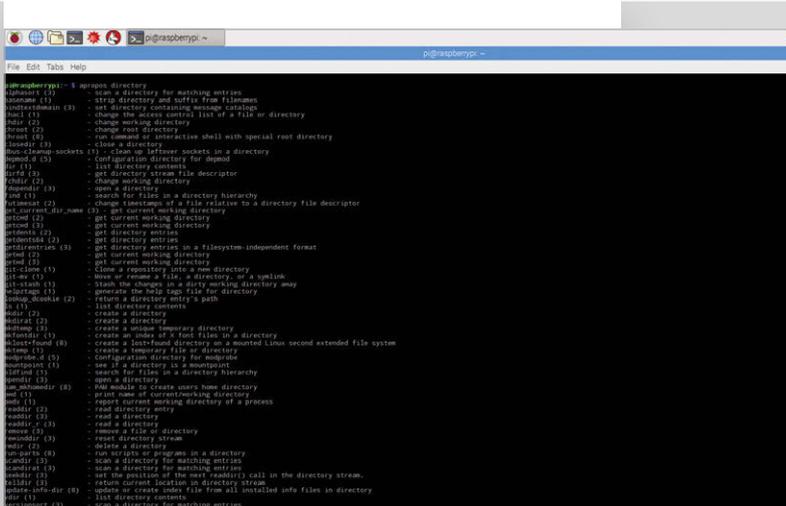
File Edit Tabs Help
pi@raspberrypi:~$ mktemp --help
Usage: mktemp [OPTION]... [TEMPLATE]
Create a temporary file or directory, safely, and print its name.
TEMPLATE must contain at least 3 consecutive 'X's in last component.
If TEMPLATE is not specified, use tmp.XXXXXXXXXX, and --tmpdir is implied.
Files are created u+rw, and directories u+rw, minus umask restrictions.

-d, --directory      create a directory, not a file
-u, --dry-run        do not create anything; merely print a name (unsafe)
-q, --quiet          suppress diagnostics about file/dir-creation failure
--suffix=SUFF       append SUFF to TEMPLATE; SUFF must not contain a slash.
                   This option is implied if TEMPLATE does not end in X
-p DIR, --tmpdir=DIR interpret TEMPLATE relative to DIR; if DIR is not
                   specified, use $TMPDIR if set, else /tmp. With
                   this option, TEMPLATE must not be an absolute name;
                   unlike with -t, TEMPLATE may contain slashes, but
                   mktemp creates only the final component
-t                  interpret TEMPLATE as a single file name component,
                   relative to a directory: $TMPDIR, if set; else the
                   directory specified via -p; else /tmp [deprecated]
--help              display this help and exit
--version           output version information and exit

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/mktemp>
or available locally via: info '(coreutils) mktemp invocation'
pi@raspberrypi:~$

```

Many commands feature a built-in help option, accessed with -h or --help. Using it offers a brief outline



## FINDING COMMANDS

As you become more familiar with `man` and `info`, you'll start searching for commands to look up. Here, the `man -k` command comes in useful. In particular, try this:

```
man -k directory | more
```

This command lists all available `man` entries. Press `SPACE` to run through them one at a time. The `man -k` option is worth remembering. If you use `man man`, it tells you the `-k` option is 'equivalent to `apropos`'. `apropos` is used to search manual page names and descriptions. It's a handy way to find commands when you don't know their names.

For instance, enter:

```
apropos directory
```

...and you'll get a list of all the commands that have the word 'directory' in their description or page name. Here you'll find common commands such as `ls`, `cd`, and `pwd`, but you'll also find less obvious commands, such as `mktemp`.

Next to each command is a number, like (1) or (2). These correspond to the section numbers of the manual (view using `man man`).

The section numbers are useful for guiding you to the commands that can be used on the command line. As a general rule, **1**: Executable programs or shell commands, and **2**: System calls, both tend to be worth investigating. Higher numbers are for library calls, special files, and kernel routines for advanced users.

You can find out more information about any command using `man`:

```
man mktemp
```

This command gives you detailed information on how to create temporary directories.

## TAB AUTOCOMPLETE

Another way to find files is to use 'tab autocomplete'. By pressing the `TAB` key, you can automatically complete commands, files, and directories on the command line. If you're not doing so already, learn to press `TAB` a lot on the command line: it's a good way to discover new commands.

Take the `apt` tool, for example. There are `apt-get` and `apt-cache`, but did you know about `apt-config` and `apt-key`?

Enter:

```
apt
```

And press the `TAB` key twice. It will display all the different types of `apt` available.

You can even run through the letters of the alphabet. Enter the letter 'a':

```
a
```

And press `TAB` twice to view all the commands beginning with 'a'. You can then use `man` to look up commands. It's a great way to broaden your knowledge of the command line.

## EXPRESS HELP

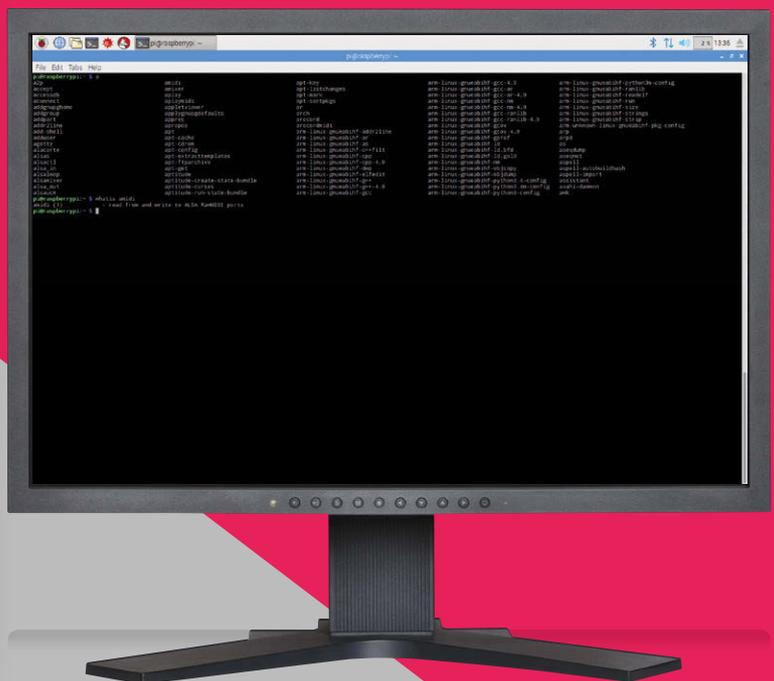
Many commands also offer a help feature as an option. Help is typically accessed using `-h` or `--help`:

```
mktemp --help
```

This command displays the options offered by the `mktemp` command. It's the same as the first page of `mktemp`'s `man` file, but saves you digging in and out of the full document.

`apropos` is used to search the manual for matching words. With it, you can find commands based on subjects, such as directory, password, or links

Enter a single letter and press `TAB` twice to view all the commands available in that letter. Then use `whatis` to get a short one-line description of each command



Not all commands make use of `--help`. Some, like `ls --help`, display the full man document (you can pipe this through `less`):

```
ls --help | less
```

...but it's typically easier to use `man ls`. Some commands don't implement the help option at all.

```
pwd --help
```

...returns 'invalid option'. But it's worth trying when you are experimenting with new commands.

One final command worth using when searching for commands is `whatis`:

```
whatis pwd
```

This example returns 'print name of current/working directory'. Often, this brief description is enough to let you know what it does, or at least tell you if it's something you'd like to investigate further with `man` or `info`.

These are just some of the tools you can use to get help at the command line in Linux. While the command line may seem intimidating at first, you're far from alone in this text-only environment.

## WEB SEARCH

One of the advantages when using a desktop interface, like Raspbian, is that a web browser – and a search engine – is just a click away.

Getting online from the command line is a lot easier than you'd imagine. There are many different text-based web browsers that enable you to access Google, Bing, DuckDuckGo, and other websites without having to boot into the PIXEL desktop interface.

We're going to use:

```
sudo apt-get update && sudo apt-get upgrade
sudo apt-get install elinks
```

Now you can open the web browser from the command line using:

```
elinks
```

The `elinks` interface is full-screen, so it replaces the command line. Press `g` to open a URL field. You can enter full URLs, such `http://www.google.com` or just shortened versions, such as `raspberrypi.com`.

Better yet, there are a few key bindings for helpful sites. Press `g` then enter these shortcuts:

```
d - dict.org search
```

```
sd - Slashdot
```

```
g - Google search
```

You can also enter Google search terms in the URL field. Press `g`, then enter 'g the magpi' to search for our website in Google.



Other keyboard shortcuts can be used to navigate the program:

```
g - Goto URL
```

```
Down Arrow - Next link
```

```
Up Arrow - Previous link
```

```
Return - Select link
```

```
Left Arrow - Back
```

```
u - Forward
```

```
q - Quit
```

```
. - Toggle link numbering
```

```
% - Toggle colours
```

```
t - New tab
```

```
T - Open link in new tab
```

```
> - Next tab
```

```
< - Previous tab
```

```
c - Close tab
```