Getting help

Max Carter-Brown
Anglia Ruskin University, Wellcome Sanger Institute
max.carter-brown@aru.ac.uk

Contents

1. Introduction

1.1. Why get help?

All base commands which you call on the command line in your terminal will have in-built help. Getting help, or looking at the manual pages (the man pages) is a really important part of developing your skills. There are multiple situations in which you might want to get help:

- You don't know what the command does.
- To get more information about the command, and potentially examples.
- To look up syntax and how the command is called.
- To look up the flags (e.g. `-e` - a dash followed by a character), options (e.g. `-e <value>` where `<value>` is a user input like a number, or a file path), or positional arguments.

1.2. A generalised tool

Before we look at a specific example, I am going to walk through a fake example, and define some terms. I have a made up program called `fetch`.

Shell 1: After running `fetch`, we get this output.

```
fetch v1.0 - A program to fetch files from the internet

fetch [-lh] [--verbose=low] [url, ...]
```

The first set of options `[-lh]` are what I call flags. They are put after the name of the program and alter its behaviour. For example, in this case `-h` might cause the program to print the help, and `-l` might create a log file. Secondly we have `[--verbose=low]`. This is what I call an option - it takes an argument or parameter after you have called `--verbose`. In the example, it is `low`. If you call `--verbose` by itself, there will be an error. Lastly we have what are called positional arguments. Usually positional arguments are at the end of the program call, and generally point to a file of some sort. All together:

Shell 2: A made up example where an optional argument `--verbose` is used, and a positional argument which is a URL. `-l` creates a log file of the output.

```
fetch -l --verbose=high "www.example.com/file.txt"
```

1.3. How to get help

1.3.1. `man`

`man`, which is short for 'manual' shows a page on your terminal that you can scroll through. It lists all the flags, options, and behaviour of the program. The output of this program can be quite overwhelming, so let's take an example.

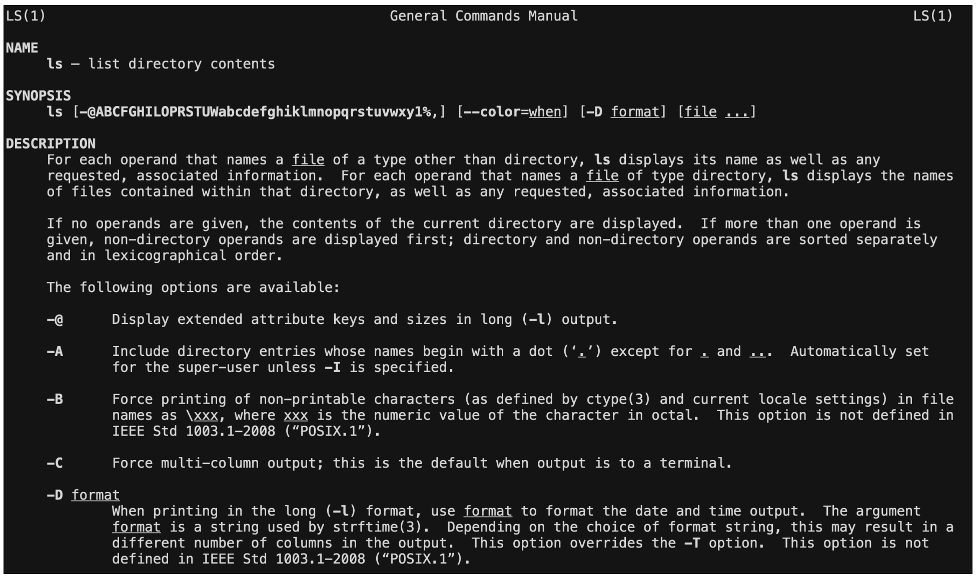
Shell 3: Getting help for the `ls` command using the `man` command.

```
man ls
```

The `man` page for `ls` is really long, perhaps unexpectedly for such a seemingly simple command. We will break down each relevant section.

1.3.1.1. Synopsis

An overview of the tool. It gives a very brief overview of how the tool is used. For `ls` we can see there are a *lot* of flags, but only a few (two) options. There's also one or more positional arguments - so we can actually pass as many paths as we like to `ls`, that's kind of interesting.



```
LS(1)                              General Commands Manual                             LS(1)

NAME
     ls – list directory contents

SYNOPSIS
     ls [-@ABCFGHILOPRSTUWabcdefghiklmnopqrstuvwxy1%,] [--color=when] [-D format] [file ...]

DESCRIPTION
     For each operand that names a file of a type other than directory, ls displays its name as well as any
     requested, associated information.  For each operand that names a file of type directory, ls displays the names
     of files contained within that directory, as well as any requested, associated information.

     If no operands are given, the contents of the current directory are displayed.  If more than one operand is
     given, non-directory operands are displayed first; directory and non-directory operands are sorted separately
     and in lexicographical order.

     The following options are available:

     -@      Display extended attribute keys and sizes in long (-l) output.

     -A      Include directory entries whose names begin with a dot (‘.’) except for . and ...  Automatically set
             for the super-user unless -I is specified.

     -B      Force printing of non-printable characters (as defined by ctype(3) and current locale settings) in file
             names as \xxx, where xxx is the numeric value of the character in octal.  This option is not defined in
             IEEE Std 1003.1-2008 (“POSIX.1”).

     -C      Force multi-column output; this is the default when output is to a terminal.

     -D format
             When printing in the long (-l) format, use format to format the date and time output.  The argument
             format is a string used by strftime(3).  Depending on the choice of format string, this may result in a
             different number of columns in the output.  This option overrides the -T option.  This option is not
             defined in IEEE Std 1003.1-2008 (“POSIX.1”).
```

Figure 1: A screenshot of `man ls` on my Mac.

1.3.1.2. Description

The description gives us an overview of the most important information, then proceeds to list every single flag and option, and tells us what they do. You might want to spend a little time in the `man` page looking at the options - you will always find something new and interesting. You will notice that there is a lot of jargon; computer science and file system terms which you will have to look up and get used to. In particular in `ls` there is a whole section on the so called 'long format' which is important as it's very widely used.

1.3.1.3. Examples

You will have to scroll all the way to the end of the `man` page to see the examples, but they are worth looking at, as they will give you an idea of how the developers use the tool.

1.3.2. `-h/--help`

It is worth noting that while `man` is somewhat standardised in its help, this next part is very variable and depends on the type of program you are looking at. In *general* (but not always) you can get help in another way by passing the `-h` flag (short version usually has a single dash) **or** the `--help` flag (long version usually has two dashes). If the program recognises these flags, they will print help to the terminal. Sometimes if you run a program without any arguments at all, it will print the help too.

Many of the base tools you have on your shell will *not* recognise `-h/--help`. Many tools you will download later on *will* recognise `-h/--help`. It's tool specific, so be wary.