COMP 2718: Command Types

By: Dr. Andrew Vardy

Adapted from the notes of Dr. Rod Byrne

Outline

- Command Types — Chapter 5 of TLCL
- 4 Types of Commands
- `type` - Display a Command's Type
- `which` - Display an Executable's Location
- `help` - Help for Shell Builtins
- `man` - Display a Program's Manual Page
- `apropos` - Display Appropriate Commands
- `whatis` - Display a Brief Command Description
- Man Page Organization
- Aside: History of Unix and the GNU Project
- `info` - Display a Program's Info Entry
- Other Sources of Documentation

Command Types — Chapter 5 of TLCL

We'll start to cover material from chapter 5 of the textbook. The following commands will be introduced:

- `type`: Indicate how a command name is interpreted
- `which`: Display which executable program will be executed
- `help`: Get help for shell builtins
- `man`: Display a command's manual page
- `apropos`: Display a list of appropriate commands
- `whatis`: Display a very brief description of a command
- `info`: Display a command's info entry

The first two commands are about the types of commands that exist and where they are located, but the rest are about documentation and getting help.

4 Types of Commands

A command is one of the following:

1. **An executable program**: May be compiled binaries (C, C++, . . .), or scripts running in interpreters (python, perl, . . .).
2. **A command built into the shell itself**: The `bash` shell provides a wide variety of **builtins** such as `cd`.
3. **A shell function**: Shell scripts that are part of the *environment* (covered later).
4. **An alias**: A command or set of customized commands defined by the user (or the system).

type - Display a Command's Type

type is a shell builtin that displays the type of command given:

```
type command
```

e.g.

```
bodhran:~ av$ type ls
ls is /bin/ls

bodhran:~ av$ type cd
cd is a shell builtin

bodhran:~ av$ type z
z is aliased to `exit'
```

which - Display an Executable's Location

which determines the location of an executable:

```
which command
```

e.g.

```
$ which ls
/bin/ls

$ which cd
[No response because cd is a builtin]
```

help - Help for Shell Builtins

You can get help in using a builtin with this command. For example:

```
help cd

cd: cd [-L|[-P [-e]] [-@]] [dir]
    Change the shell working directory.

    Change the current directory to DIR.  The default
    DIR is the value of the HOME shell variable.
```

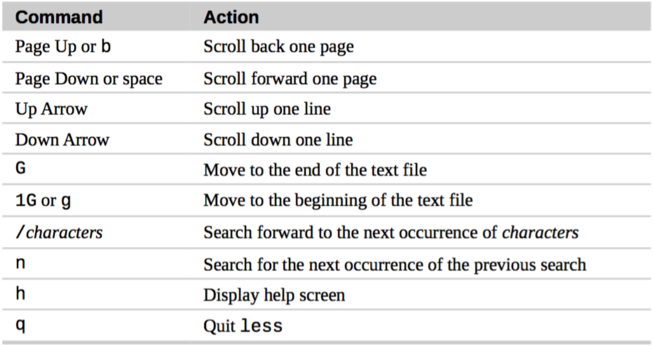
Note the square brackets which indicate optional items. Some options cannot be used together. These are separated by vertical bars. For example, with cd the -L and -P options are mutually exclusive. Nesting indicates when an option can only be used with another. For example, if -e is only possible with -P.

`man` - Display a Program's Manual Page

Most installed executables provide a *manual* or *man page*. `man` is used to view them.

```
man program
```

The man page is usually displayed using `less`. We covered `less` in "The File System: Part 2". Of course, you can also use `man less` to see how to use it. The keys used are given again below:



Command	Action
Page Up or `b`	Scroll back one page
Page Down or space	Scroll forward one page
Up Arrow	Scroll up one line
Down Arrow	Scroll down one line
`G`	Move to the end of the text file
`1G` or `g`	Move to the beginning of the text file
`/`*characters*	Search forward to the next occurrence of *characters*
`n`	Search for the next occurrence of the previous search
`h`	Display help screen
`q`	Quit `less`

apropos - Display Appropriate Commands

If you don't know which command you need, the set of man pages can be searched with apropos.

```
apropos search_string
```

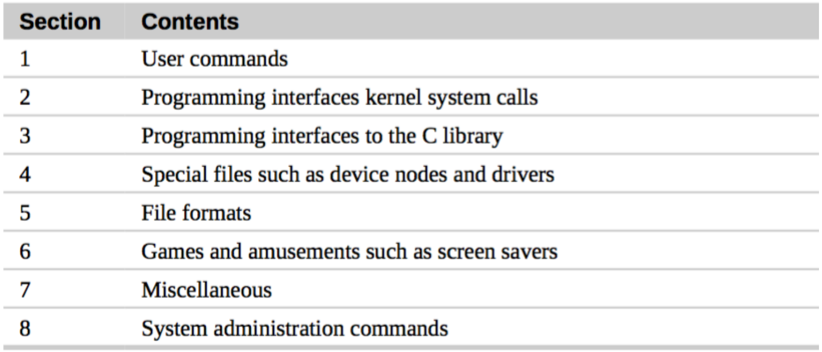
whatis - Display a Brief Command Description

This command simply displays a one-line description of a man page for the given command:

```
whatis command
```

Man Page Organization

Man pages are organized into the following sections:



Section	Contents
1	User commands
2	Programming interfaces kernel system calls
3	Programming interfaces to the C library
4	Special files such as device nodes and drivers
5	File formats
6	Games and amusements such as screen savers
7	Miscellaneous
8	System administration commands

If you want to specify a particular section then enter it as the second argument to `man`. For example, `man 5 passwd`. This is useful here because `passwd` is both a command and a file format.

The sections above correspond to numbers you may see in the output from `apropos` and `whatis`.

Aside: History of Unix and the GNU Project

Unix came out of Bell Labs with development beginning in 1969. In those early days it was a proprietary product, but some versions became open source.

In 1983 the **GNU Project** was launched by Richard Stallman. Its mission is to foster the development of free software, where free means that users are allowed to run, share, and modify the software. Note that “free software” may still be sold and purchased. GNU is a recursive acronym that stands for “GNU’s Not Unix”. Software from the GNU project uses the Unix philosophy but not Unix itself.

Why this historical aside? Because man pages are a Unix convention. The GNU project uses another convention called **info pages**.

info - Display a Program's Info Entry

Info pages, unlike man pages, are hyperlinked like web pages. Most of the program's discussed here are actually part of the GNU Project's coreutils package so we can browse their info pages with:

```
info coreutils
```

Use the following keys within `info`:

Command	Action
?	Display command help
PgUp or Backspace	Display previous page
PgDn or Space	Display next page
n	Next - Display the next node
p	Previous - Display the previous node
u	Up - Display the parent node of the currently displayed node, usually a menu.
Enter	Follow the hyperlink at the cursor location
q	Quit

Other Sources of Documentation

There are many sources of documentation for software, available in a wide variety of formats. If `man` and `info` pages are not available (or are unhelpful), try the following:

- Documents in `/usr/share/doc`
- Documents in the package's source folder (e.g. README, `doc` subdirectory)
- Google!