Bash Help Notes

Hayden Tremethick 04/07 2012

Bash is terminal based, therefore bash help is primarily designed to be,'*cheese eater free'*.
Everything in Linux is a file (including special files and processes executing in RAM).
R U Lost? Use **`pwd`** to find yourself; and wherever you may roam **`cd ~`** takes you home

Document Processing Systems. **`man`** is produced by **troff** and uses a builtin **`less`** to navigate
info is produced and navigated using **emacs**

Searching documents – Basic Navigation for man & info

Man Keystrokes (vim keybindings)	**Action**	**Info** Keystrokes (emacs key bindings)
Enter	Down 1 line	↓
←↑↓→ keys.	Left/Up/Down/Right	←↑↓→ keys.
PgDn key PgUp key	Page Down Page Up	PgDn key PgUp key
Home/End keys	Home/End	Home/End keys
/<expression> ?<expression>	Search forward Search backward	/<expression> or s<expression> ?<expression>
/ key or **n** key	Search forward **n**ext	}
N key	Search reverse **N**ext	{
h key	Navigation **h**elp	**h** key
q key	**q**uit -	**q** key
Spacebar or **d** key **b** key	**d**own 1 screen **b**ack 1 screen	Spacebar Delete key
	menu item by name. **Enter** item under cursor	**m** key (with Tab-Tab completion) **Enter** key
	Go back to **l**ast node Go to the **d**irectory (top node)	**l** key **d** key
	Cancel the current operation	Ctrl+g key

The Bash Environment is a collection of NAME=var-val pairs maintained by the Bash process.
It can be viewed using **`pidof bash`** then **`cat /proc/<pid>/environ| tr '\0' '\n'`** or
`printenv` – will print all the NAME=var-val, pairs in the environment.
`printenv <`*`VAR-NAME`*`>` or **`echo $<`*`VAR-NAME`*`>`** – prints individual *VAR-NAME* values.
for example **`printenv PATH`** or **`echo $PATH`** – will print the PATH var-val.

Note: upper-case is used in Bash for environment (universal) VARIABLE-NAMES, this prevents conflict with lower-case command-names. **Hint:** if you are writing a bash script, consider starting your local variable names with an uppercase letter followed by all lower case.

To change an environment value use **`export <`*`VAR-NAME`*`>=<value>`** **Note:** no spaces around = sign. For a permanent change edit the file **`~/.bashrc`** . (&/or **`~/.bash_profile`** for login shell).
When scripting use bash functions clearenv(), getenv(), setenv(), putenv(),unsetenv(), environ().

Bash Built-ins and External Commands
Bash commands may be,

1) built-ins – these are component parts of the bash interpreter,
2) external programs that are contained in directories pointed to by PATH.

Caution – some commands may have two versions one a built-in and one external e.g. **echo**; in this case if you want to use the external command, use its full path name e.g. **/bin/echo**

/usr/share/doc/ is the information source of last resort,it contains non man/info documents which will sometimes provide the information you need, Have a look through it to familiarise yourself.

Bash command search. When a command is input to a terminal, **/bin/bash** checks, 'Is it is a built-in?', if true the command is executed, if false, Bash searches the PATH directories to find and and execute the file. To list the enabled built-ins use **enable | mawk '{print $2;}'**.

Help for Built-ins,

help help – provides a description of the bash builtin help
help – provides a synopsis of all the built-in commands usage.
help <command-name> – will provide detailed built-in <command-name>.
man bash then using **less** commands, navigate to the required bash built-in.
Built in commands do not usually have man or info pages.

Man Help (for external commands).

Man pages are usually installed when you install a package, so no package installed, no man.
whatis <command-name> brief man page description
which <command-name> – provides a command's path name.
whereis <command-name> – this locates the binary, source, and man page files for a command

man sections
man is divided into the following sections

1 Executable programs or shell commands
2 System calls (functions provided by the kernel)
3 Library calls (functions within program libraries)
4 Special files (usually found in /dev)
5 File formats and conventions eg /etc/passwd
6 Games
7 Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)
8 System administration commands (usually only for root)
9 Kernel routines [Non standard]

man <command-name> man will display the man page from the section man considers appropriate
man -a <command-name> By default, **man** displays what it considers to be the man page from the most appropriate section. The **-a, or --all** switch will show you the man page starting at the lowest numbered section. After you **q**uit you will be offered the option to read from the next section or skip.
man <section-number> <command-name> will show the man info page from <section-number>

To find a specific term, pipe your man command through the **grep** command
For example, use the following command to search for the term "options" in the grep man page.
man grep | grep -in "options" (-i = ignore case -n = show line numbers)

apropos – Each manual page has a short description available within it under the heading '**NAME**'.
apropos (man -k) searches these descriptions for instances of the given keyword or **"**key phrase**"**.
whatis (man -f) searches the descriptions and displays descriptions of any matching name.

Some other useful 'man' command options (switches)
-h, --help or **-?** following a command name will provide a brief command synopsis.
-aw will provide a list of all the available man pages for a topic
-S list, **-s list**, **--sections=list** will view the man pages from a 'list' of sections where 'list' is a colon or a comma-separated list of the required sections.
-f, **--whatis** call the command **whatis**.
-k, --apropos calls the command **apropos** which searches for keywords or "key phrases"

Info Help. (info may not be installed by default on some systems)
info has more detail than man on some commands, but if it has no information at all on a command it will display the appropriate man page. Info has tree file structure. Pages are called nodes. To start input the command **info**. This will open in the top or **d**irectory node.

At the top of the 'node' are some brief navigation instructions (read and try them) At the bottom of the window, in reverse colour highlight is the *mode line*. immediately below the mode line is the *read line* that accepts input from the keyboard. Use **info rluserman** (**r**ead **l**ine **user man**ual) for detail.

info <command-name> will open **info** for the given command name, from there if you want to go to the top level node called the **d**irectory the '**d**' key will get you there or '**q**' for **q**uit. To go to a new command use **m** then enter the <command-name> you can use **Tab-Tab completion.**

For Starters try the following commands
man man, the man page for the man command
man info, the man page for the info command
info info, the info page for the info command
info man, the info page for the man command
man intro and **info intro**, provide an introduction to user commands
man bash and **info bash** for bash operation and **inbuilt bash commands**
help lists a synopsis of the Bash built-in commands

Some Useful Tricks & Tips

- View/Browse man pages in your web browser at **http://localhost/cgi-bin/man/man2html** (this may not work with all systems). You can bookmark this page and use it to access man.
- **If you have KDE,** Doldhin and Konqueror will convert man and info pages into html, use **man:/** for the Manual Index & **man:<command-name>** or **#<command-name>** for a command **info:/** for the Directory & **info:<command-name>** or **##<command-name>**for a command
- Unsure about the exact name of a command, option, or a file? use **Tab-Tab completion**
- Save man-page as pdf – **man -t <command-name> | ps2pdf – <command-name>.pdf**
- Dump man page to a text file – **man <command-name>| col -b > <command-name>.txt**
- Send manpage to default printer – **man <command-name>|col -b|lpr**
- Clear a terminal screen – **Ctrl-l**
- Run the previous command **!!** run previous command as Root – **sudo !!**
- **Ctrl+Alt+t** Brings up a Terminal Emulator Ctrl+Shift+t – Opens a new TE tab.
- The info pages for GNU programs are available on at: http://www.gnu.org/manual/manual.html Many commands, like GRUB, gawk, & bash, have an "entirely on one web page" version.
- An A-Z Index of the Bash command line for Linux. **http://ss64.com/bash/**
- Get your external IP address. – **curl ifconfig.me**
- mtr is traceroute and ping combined – **mtr linuxlsga.net**

Useful Bash Information

Bash Initialization Files.
`~/.bashrc` is read when an interactive bash is called. **`~/.bash_profile`** is read only on login.
When Bash starts, **`~/.inputrc`** , is read, variables and key bindings contained in this file are set.
`Ctrl-x` **or** **`Ctrl-r`** will re-read **`~/.inputrc`**
To avoid maintaining both files, you can add the following lines to `.bash_profile`:

```
if [ -f ~/.bashrc ]; then
   source ~/.bashrc
fi
```

Standard I/O files (sometimes called Standard Streams), are three channels, one input and two outputs, set-up at boot-time. These standard I/O files are called.
Standard Input (STDIN) accepts input from keyboard.
Standard Output (STDOUT) outputs text to the terminal screen.
Standard Error (STDERR) outputs error messages to the terminal screen.

Redirection Operators + cat

`>`	redirects STDOUT to create a new file (**caution:** it will overwrite an existing file)
`2>`	redirects STDERR to create a new file (**caution:** it will overwrite an existing file)
`&>`	redirects both STDOUT and STDERR
`>>`	append to the end of a file instead of overwriting.
`<`	accepts input from a source other than STDIN
`|`	pipes the output from the command on the LHS as input to the command on the RHS
`cat`	catenates (links) one or more files to STDOUT (can also be piped to other files)