

### 8 "Urgent" Things to Know Before You Begin

- ◇ Much you will find on the web is not current.
- ◇ Raspian (like all Linux) is case sensitive. Try avoiding spaces and capitals - enclose strings with spaces in quotes or use underscores.
- ◇ Stuck? try **(ctl-c)**. **q** exits many listings.
- ◇ The prefix command **sudo** gives you elevated or "root" permissions. It is frequently essential if you want something to work.
- ◇ From RPi's **PIXEL desktop** 7 command-line environments can be entered with **<Ctl><Alt><F#>** where **<F#>** is a **function key** from **<F1>** to **<F7>**. Use **<Alt><F7>** to return to **PIXEL**.
- ◇ 2 Sanity savers: **#1 <ctl>l** (lower case L) or **<clear>** clears the screen, and puts the prompt at the top; **#2** up/down arrows (↓ and ↑) scroll through previous commands.
- ◇ To get help: **help**; **help<command>**; **info**; **info<command>**; **"man -k man | less"**; **<command><spc><--help>[| more]**

### Commands You Need Immediately

- ◆ **pwd** displays your **p**resent **w**orking **d**irectory – this is the directory you are "in" at any given moment. (If you have not changed your default user from pi, you start out in /home/pi.)
- ◆ **cd<spc><some path modification string>** changes your working directory. **cd /** moves you to root, **cd /mnt** moves to root/mnt
- ◆ **ls** lists directories in your current directory location. **ls**, like almost all commands, can be modified with "flags" like **-l** or **-a** and these flags can be combined. Try **"ls -lah"**, **"la -l"** for format
- ◆ **sudo** gives root privileges; **su<spc><user>** switches user

### 7 Shortcuts You Need to Know Now

- "." an alias for the current directory
- ".." an alias for the parent directory
- "~" is an alias for the absolute path shortcut to the user's home directory. Type "cd ~" to return to your home directory
- "/" alias for the root directory
- "\*" a wildcard character for one or more possible but unknown character(s), "?" is a wildcard for a single character
- "|" constructs a "pipe" that joins commands output to input – frequently used with the commands **less**, **more**, and **cat**

### The Most Important RPi Command Line Tools?

*Special Note: apt is an updated utility that replaces apt-get. The aptitude command suite combines the best of apt-get and apt-cache.*

According to the Raspberry Pi folks, the two most used command line functions are part of the **apt** utility:

- (1) **apt update** - more likely called as **sudo apt update**, but ... **apt update** only gives you a list of packages that could be updated. You need to then call **apt upgrade** (if nothing has to be removed) or maybe **apt full-upgrade** (if packages need removal) – prefix commands with **sudo** unless you have established root privileges (**sudo -i** locks root on).
- (2) **apt install <a program or utility>** (again you may require **sudo**)

\*Note: Other important apt commands include: **install**, **remove**, **purge**, **autoremove**, **search**, **show\_information**

**#!** - "shebang" or "hashbang" - is an initial character syntax that causes bash to initialize a specific interpreter and run the executed script.

### Working with Files and Directories

**chmod<spc><options><spc><filename>** changes file permissions. In a file and directory listing generated by a command, like **ls -l**, the first 10 characters are the "permissions string". Character 0 defines the entry as a directory (**r**) or a file (**-**). The next 9, in groups of 3, establish **read**, **write**, and **execute** permission for the owner (**u**), the group (**g**), and others (**o**) respectively. One way to set them is to define the desired values in string equations for each set separately. **Ex: "chmod u=rwx,g=rw,o=r myfile"** gives the owner (user) all privileges, the group read or write, and other gets read only - with respect to the file "myfile".

		chmod permission string values								
		owner "u"			group "g"			other "o"		
dir or file	character position	1	2	3	4	5	6	7	8	9
	"-" is a file	-								
Options										
	"r" is a directory	r			r			r		
permission granted (r,w,x)		r	w	x	r	w	x	r	w	x
or permission denied (-)		-	-	-	-	-	-	-	-	-

**cat<spc><file>** lists contents of a file, for long files try **cat | less**  
**cp<spc><file><spc><path or path and file name>** copy a file and put it in the directory as specified

**curl** download or upload a file to or from a server  
**diff<spc><file1><spc><file2>** compares file1 to file2  
**dir** displays a list of directories only, add **-a** to get everything  
**find<spc><options><spc><path-name /for root><spc><file name can use wildcards>** note: <options> are advanced. Also see **locate**.  
**grep<spc><"string"><spc><filename>** looks for a string pattern  
**head<spc><-##>** or **tail<spc><-##>** print first or last ## file lines  
**locate<spc><option><spc><target>** not default installed, must "sudo apt install locate" & maybe "sudo updatedb" - many options.  
**mkdir<spc><new directory name>** create a new directory in pwd  
**mkdir<spc><-p><spc><path/dir name>** make a new directory on the path specified

**mv<spc><file><spc><newfilename>** renames or moves a file  
**mv<spc><file><spc><path or path and file name>** moves a file to the directory specified (*mv works; rename usually does not*)  
**rename<spc><current file name><spc><new file name>** renames  
**rm<spc><file>** removes a file \*no way to recover a deleted file  
**rm<spc><file list>** removes a list of files  
**rm<spc><-r><spc><directory name>** removes a directory. Note: it is gone forever.

**rm<spc><-R><directory name>** removes everything  
**rmdir<spc><directory name>** removes an empty directory  
**touch<spc><newfile name>** create a new empty file in pwd or change its time stamp

**shred<spc><file>** ultra secure file destruction (paranoid a little?)  
**tree** show a tree structure of directories and files  
**vdirc** verbosely list directories – editor's fav

**wget<spc><url of file location>** download a file to Pi from the web

**whereis** finds a command file in standard program location  
**wc<enter>** list the number of lines, words, and characters in a file

**Get Information About****PEOPLE**

**groups** displays a list  
**id** current uid's group  
**logname** user's name  
**users** everybody logged in  
**who** shows users by tty  
**whoami** shows user logged in

**NETWORK ENVIRONMENT**

**ifconfig** network status info  
**hostname** <spc><-l> (*capital eye*)  
 the **host** ip will be first 4, dot separated, number series. It is also the "inet" in ifconfig listing  
**ping** checks communication with another host  
**ssh** the secure shell that makes your RPi into a command-line client - not enabled by default - can be activated in "interfacing options" using the **raspi-config** utility. For a non-permanent solution use: "**sudo systemctl enable ssh**" and then "**sudo systemctl start ssh**"  
**tty** displays active terminal #

**HARDWARE**

**arch** you processor name/id  
**du**<spc><"filename"> shows disk space usage of files and directories; use "**du | less**"  
**pinout** - fun for RPi users - textual graphic diagram of your Pi  
**lscpu** will present summary info on the cpu  
**uname**<spc><-a> extensive critical info about your system  
**vcgencmd** - vast hardware info about RPi, NOT in help or info so Google it Ex: **vcgencmd**<spc><get\_config><int>

**SYSTEM AND SOFTWARE**

**df** mounted partition usage  
**ps**<spc><aux><spc><|><spc><l>ess> view all running processes  
**ps**<spc><-u><spc><your user name> info on your processes, including id needed to **kill** one  
**stat**<spc><filename> get the status information on a file  
**stty** print or change current terminal baud setting  
**top** will list running processes showing real time activity

**Find packages installed:** (see Debian: <https://wiki.debian.org/ListInstalledPackages>)

**dpkg-query**<spc><-l> a very nice table with version and description; "-l" is lower case L.  
**dpkg-query**<spc><-f><spc><'\${binary:Package}\n'><spc><-W> one per line

**dpkg-query**<spc><-l><spc><'search pattern'> add search pattern to list command

*NOTE: The Debian site is a good resource. A place to begin is : <https://wiki.debian.org/WordIndex>. A critical look at the Raspberry Pi : <https://wiki.debian.org/Raspberrypi>*

**MULTIPLE INFORMATION TYPES**

Accessing the <proc> information has more than a hundred status and environment attributes to be displayed.

Try these four, displayed by adding them as options to the <cat> command, i.e., **cat**<spc></proc/version> ↪ RPi version  
**/proc/cpuinfo** processor detail  
**/proc/meminfo** memory use  
**/proc/partitions** how your sd card is divvied up.

**DEVICE SETTINGS** - **Pixel** menu, **sudo raspi-config**, or a few at **config.txt** (*see /boot/config.txt*)

- Raspberry Pi Foundation has an overview at: <https://www.raspberrypi.org/documentation/configuration/config-txt/>  
 option sections include: *Memory, Audio, Camera, License Keys/Codecs, Boot, Video/Display, GPIOs Ports and Device Tree, Overclocking, Conditional Filters, Miscellaneous*

**Configuring bash** (your command line environment) a lot can be changed in the file **.bashrc** (in home directory - back it up first!) but a really good (and fun) place to begin is to open (or create, then open)

**.bash\_aliases** and create your own commands. Try adding alias command **alias up="cd .."**  
*more explanation at:*

<https://www.wikipython.com>

**Additional apt Options** besides update, upgrade, full-upgrade  
**apt**<spc><install><spc><a program or utility> install new package  
**apt**<spc><remove> package removed - leaves configuration file  
**apt**<spc><purge> removes all remnants that it can find  
**apt**<spc><auto-remove> used to remove auto installed packages

**Changing Your Command Line Environment**

**alias**<'command equation'> -create your own command: For example: **alias 'lx=ls -lah'**  
**<ctl-d>** logs a user out, presents log-in que  
**exit** or **logout** terminates a session; sometimes **<ctl-D>** will work  
**kill**<PID, i.e., a process id> stop a process  
**passwd** lets you change your password  
**poweroff** will do just that  
**reboot** will also do just that if you have only one user active  
**shutdown**<spc><-h><spc><now> the safest way to shutdown  
**shutdown**<spc><-r> gives you 1 minute, then restarts computer  
**shutdown**<spc><-c> cancel a shutdown command  
**su**<spc><alternate user> change users - must have account  
**systemctl**<spc><reboot><spc><-i> will restart the Pi

**Helpful Things to Know**

- ° **pip3**<spc><command><[options]> is the command to install **Python** packages. **Commands** include: **help**, **install** (some options are PyPI, VCS, and Local project directories), **uninstall**, **list**, **show**, **search**). Options are -h or -help, -v or -verbose, -V or --version
- ° Kill your PIXEL session: in terminal type: **pkill**<spc><lxsession>
- ° To start a PIXEL session in your active tty type: **startx**  
 ... and note you can open terminals in multiple environments organized by tabs in PIXEL.
- ° Put yourself in root mode: **sudo -i** \*Tip: considered risky
- ° How to give a user temp sudoer privileges: log in as a root user; type **sudo**<spc><usermod><spc><-aG><spc><sudo><user name> ...or add a sudo user permanently with the **adduser** command
- ° **Single Character Shortcuts or Search Commands**
  - \ escapes itself and other special characters
  - [ ] brackets pattern for matching a single character
  - \* matches 0 to many characters ? matches one character
  - ; separates commands on a single line; terminates a pipe
  - " " contents in quotes will be treated as one argument
  - # changes line to a comment
  - & runs a command in the background - the shell is then available in the foreground
  - < if followed by <spc><filename> means 'take input from this file'
  - > if followed by <spc><filename> means 'send output to this file';

**cavaet:** it overwrites the file. Raspberry Pi documentation <https://www.raspberrypi.org/documentation/>  
 ° Helpful file locations:  
 /etc/fstab ~/.bashrc /boot/config.txt

**Text Editors and Other Utilities:** From Pixel use Leaf Pad. From bash (command line) recommended text editor is **nano**. Others include: **ed**, **vi**, **vim**, and **emacs**. **gzip** (compress), **gunzip** (uncompress), **dc** (reverse polish calculator), **elm** (email), **talk** (chat), **ssh** (secure shell to make pi a client)  
**SEND ERRORS/SUGGESTIONS TO** [oakey.john@yahoo.com](mailto:oakey.john@yahoo.com)