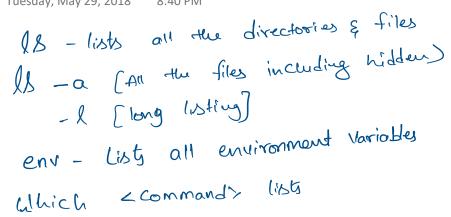
## Linux Commands

Tuesday, May 29, 2018 8:40 PM



**Is** - list the contents of a directory and their attributes

Is -I = long listing, showing file and directory ownership, permissions, and sizes

Is -lh = the 'h' switch shows size in human readable format, must be used with 'l' switch

ls -a = show all files and folders, including hidden ones

Is -R = list directories recursively

Is -S = sort files by size with the largest at the top

Is -t = sort by last time modifed displaying the newest first, most useful with the -l switch

**whoami** - display your currently logged in user

**su** - substitute user, change to another user account on the system

**exit** - leave a shell environment that you are logged in to

init 6 - legacy command for

**env** - list all of the environment variables set for the current shell environment

The **PATH** environment variable contains a llist of all of the directories that Bash will look in for applications or scripts to run.

**echo** = print what follows to the screen

echo \$PATH = print the contents of the PATH environment variable to the screen

**uname** - display the name of the system kernel

**uname -r** - display the kernel release number

**uname -v** - display the kernel build version

**uname -m** -display the machine type

uname -o - display the name of

unacyou and rossou in to

init 6 - legacy command for rebooting a system

init O - legacy command for shutting a system down

halt, poweroff - shuts down (halts) a system

**shutdown** - can be used to log of comma poweroff, reboot or stop a pending Bash prompt shutdown request

**top** - interactively display top running processes on a system

**pwd** - print working directory, the current directory that you are in

cd - change directory to path specified. i.e. cd /path/to/folder if entered by itself with no path, you will change back to the home directory of the user you are currently logged in with

cd .. - go up one directory

cd - -return to last directory

cd ~ - change to the home directory of the currently logged in user

**/etc/bashrc** - system-wide functions and aliases

**/etc/profile** - system-wide environment and startup programs, used during a login shell

/etc/profile.d/ - location of extra

**uname -o** - display the name of the operating system

**uname -a** - display all information that uname can show

.bash\_history - hidden file within the home directory that contains a log of commands entered at the Bash prompt

**HISTFILESIZE** - environment variable that specifies how many lines of history to keep

**HISTCONTROL** - environment variable that modifies Bash's history behavior

**history** - command that prints out commands saved in .bash\_history with each command numbered

!<history number> - re-runs command from .bash\_history

TAB key is your friend when it comes to command completion and having long file and directory names autocompleted at the Bash prompt for you

**env** - lists out environment variables of the currently logged in shell

echo **\$VARIABLE** - prints the value of VARIABLE to the screen

**set** - lists out all environment variables in alphabetical order

**VARIABLENAME=value** - format for declaring a new variable in Bash

**/etc/profile.d/** - location of extra environment setup scripts

The following files are in the home directory of the user (note that not all distributions will use all of these files):

.bash\_profile - used to set user specific shell environment preferences

.bash\_logout - ran when the user logs out of a login shell, not a terminal, does not exist on every system

.bashrc - non-login file that stores user specific functions and aliases

- " " double quotes, contains strings and any variables or commands within them get evaluated or acted on
- •• single quotes, anything within these gets treated literally, disables any special character functionality

\ - backslash, escape character, disables any special character functionality that immediately follows it

Quotes around spaces or an escape character preceding a space will be treated literally.

for declaring a new variable in Bash

**export VARIABLE** - exports variable and its value to other shells

- \* matches zero or more characters
- ? matches any single character

**[abc]** - matches any one of the characters in the list, case sensitive

[^abc] - matches any one character except those in the list, case sensitive

**[0-9]** - matches a range of numbers

**locate** -searches a local database of files and folders looking for items that match the search criteria

## locate cd

**find** - searches the file system for files that match the search criteria find /path/to/folder -name file

When using the find command to search for part of a file name, use globbing within single quotes: find /path/to/folder -name '\*file\*'

**whereis** - locates binary, source and/or manual pages for a command

man - manual pages command, invoked by entering: man < command>

**whatis** - Command that lists summaries and related man pages based on search term, invoked by entering:

whatis <command>
Same results can be obtained by:
man -f <command>

apropos - command that searches man page for appearances of the keyword provided, invoked by entering: apropos <keyword> Same results can be obtained by: man -k <keyword>

Arrow keys and vi key bindings can be used to navigate the man pages. Pressing the 'q' key will exit the man page. mkdir - make a new directoryp = make a parent directory along with a subdirectory

**rmdir** - remove an empty directory

**touch** - create an empty file or update a file's timestamp

cp - copy a file or folder
R = copy a folder recursively
v = verbose, display what the copy command is doing

**mv** - move or rename a file or folder

rm - remove a file or folderr = recursively remoe a folder and its contents