

# Linux Video Notes

## Intro to Using Linux Part 1 35:04

### Basic Commands Covered in 1<sup>st</sup> Video

- pwd** – print working directory
- ls** – lists contents of a directory  
dir have diff colors  
options –a = all, -l = long file display  
files/dir starting with a period are system/user files. Normally hidden.
- cd** – change directory  
.. goes up one dir
- tab** - cmd completion
- cp** - copy cmd  
leaves orig alone
- Up Arrow** - history cmd
- clear** - clears the screen
- rm** - deletes a file  
options –i = inquiry ( are you sure?)
- mv** - move cmd – or the rename cmd  
deletes the original file
- mkdir** - make dir
- rmdir** - delete dir  
can not delete occupied directories
- cat** - prints out a text file to the screen
- text editors** emacs and vi – vi is always available!  
emacs is much more user friendly  
emacs' short cut syntax C-h t Control plus h and then the t key  
C = control key M = alt key  
Conventions used = filename.txt filename.txt~  
.txt~ = older or original file

## vi command reference

vi has two modes, *Command* and *Input*.

The Esc button is used to enter command mode. Usage: vi [options] *filename*

Input mode: set insertion point	
i	insert text at cursor
a	append text after cursor
I	insert text at beginning of line
A	append text at end of line
o	open line below cursor
O	open line above cursor

Deletion and Change: delete/change words or characters	
dw	delete a single word
cw	change a word
dd	delete entire line
R	replace line
D	delete from cursor position to end of line
C	change from cursor position to end of line
x	delete a single character
r	replace a character

Cursor movement: position the cursor	
l	move right
h	move left
j	move down
k	move up
\$	move to end of line
^	move to beginning of line
w	move to next word
e	move to end of word
1G	move to first line
nG	move to line n (where n is a number)
G	move to last line

File usage: file manipulation	
:w	write buffer
:q	quit
:q!	force quit without saving
:wq	write and quit (save)
:n	next file
:r	read file
:e	edit file
:f	file name
:set	set options on
:set no	set options off
:!	escape to shell
:n	go to line n

Screen movement: move a screen at a time	
ctrl-d	scroll forward half a screen
ctrl-u	scroll back half a screen
ctrl-f	move to next screen
ctrl-b	move to previous screen
ctrl-l	redraw screen

Miscellaneous	
u	undo last action
/	search forward
?	search back
n	repeat search
.	repeat last command
p	put below cursor
P	put above cursor
ZZ	write and quit

## Intro to Using Linux Part 2 28:14

### man pages

- **man cmd** = how to use a cmd with all its options.  
Space bar to page down and **b** key to go back a page.  
**q** key will quit  
To search the man pages hit the / key and type in your search word.  
Hit the n key to find the next occurrence of the word.

### info pages

- **info cmd** = similar to man. More English, less technical.  
Has subsections or categories you can go to.  
Highlight a section and hit the enter key.  
To go back hit the **l** (el) key.  
To search **C-s** and type in the word to search for.  
**C-s** for the next occurrence and so.. **C-g** to stop searching.

### man -k

- ex: **man -k jpg** = lists all cmds that work with jpg files.  
Another way is to use the cmd **apropos jpg** = same result

### Super user

- root account – access via the **su** cmd - switch user  
Need a password

### Dir Structure

- **cd /** = top level dir. Immediately below this are listed some typical dir.  
**bin** = holds binaries or executable files. Actual cmds are listed here.  
Holds the every day cmds that users typical use  
**boot** = holds file responsible for boot up.  
**home** = holds all the users' home dir or where they typically start.  
**opt** = is where you install optional software  
**root** = is typically the root users' home dir.  
**sbin** = hold system executable files. System Admin tools  
**usr** = user dir. Contains the users' executable files.

### User Config Files

- **.bash\_profile** and **.bashrc** file config the users' bash shell.  
The bash shell is signified by the \$ prompt, usually.  
You can vi the **.bash\_profile** file.  
Typically you modify this file's **PATH** line. It lists the dir to search first when you issue cmds. Typically you would add the following: **PATH=\$PATH:/usr/sbin:sbin:**  
Then add a line under that = **export PATH** and save changes.  
These are used for the root or super user acct.  
To activate the changes w/o logging out and back in is to type  
In the cmd: **source .bash\_profile**

The 2<sup>nd</sup> change is how to set an alias

vi **.bashrc**

add the following line:

alias **rm='rm -i'**

remember, this forces you to confirm the deletion of files.

Save your changes.

Now you can activate it with **source .bashrc**

### Dir Names

- **cd /dir1/dir2** = absolute dir names ( slash needed )
- **cd dir1/dir2** = relative dir names ( no slash needed )
- **cd ../scripts** = goes up one dir and then into scripts dir.

