



ABC IT Education

WE'LL TAKE YOU FROM ZERO TO HERO IN A SNAP

Linux Systems Administration

Displaying File Contents
and the Nano Editor

- 
- We'll look at the different ways of displaying the contents of files and also how to use the nano editor in Linux
 - We have previously seen the '**cat**' command and used it to display entire files i.e. display the entire file.
 - Other commands that let you display file contents are '**more**', '**less**', '**head**' and '**tail**' commands. We have also seen the '**head**' and '**tail**' commands in the recent past, so let's look at the '**more**' and '**less**' commands.
 - The '**more**' command will allow you to browse through a file or page through a file.
 - The '**less**' command is like '**more**', except it has more features.
 - **Less is more!**

- 
- When using the commands - more or less,
 - use the space bar to advance to the next page,
 - use the enter key to advance 1 line
 - use 'g' to go to the top of the document
 - use 'G' to go to the bottom of the document
 - and type 'q' to quit viewing the file.
 - The commands are based on the vi editor which we'll see later.
 - As seen previously, head and tail only display 10 lines by default.
 - Assume you want to see the last 15 lines of file for example, you could use '\$ tail -15 filename'.

- 
- The cat command could be a fine way to view files that have static content, but if you're trying to keep up with changes that are being made to a file in real time, cat is not your choice.
 - Instead, you should use 'tail -f' followed by the filename.
 - You would use this to look at files that change often, like log files.
 - As an example, you may want to start a program and look at the programs log file to see exactly what it's doing.
 - Again in this case, use 'tail -f' and the file you want to follow.
 - Looking at this file with a cat command, if we run 'head' against the same file, we see the top 10 lines of that file.
 - To see the top 2 lines, we can run 'head -2'

- 
- To exit, just type 'ctrl+c'.
 - **Editing a file**
 - One way of editing files in Linux is by using the Nano Editor which is a clone of Pico, so if for some reason the 'nano' command isn't available, try 'pico'.
 - Nano is very easy to learn but does not have some of the advanced features of the other like vi or emacs.
 - Let's edit this file with nano. 'nano file.txt'.
 - When nano starts you can see the file's contents, then a list of commands at the bottom of your screen.
 - The '^' symbol represents the control key, so looking in the bottom left hand corner, we see '^x', which means 'control+x' to exit.

- 
- Let's exit - 'control+x'. OK we are exited out of nano.
 - Let's edit the file again with nano. 'nano file.txt'.
 - Editing with nano is pretty easy, as arrow keys work as you'd expect - down, right, up, left.
 - If you want to save a file, hit 'ctrl+o', and hit enter.
 - To make some modifications, just place your cursor wherever you want to go and start typing there.
 - Another important thing is the use of 'ctrl+g' to get help. This will show you more information on how to use nano.

- 
- The locate command, works off of an index, and the index is typically rebuilt everyday.
 - 'locate' may not be enabled on all systems.
 - Instead of evaluating each and every file to see if it matches the pattern 'locate' just does a look up in the index.
 - NOTE: 'locate' results are not in real time because there is a lag between the time that the index gets created and the time that you run the locate command.
 - If you're looking for a file that was created just a few minutes ago and you use 'locate', chances are it's not going to be in the index.
 - To find the file you should use the 'find' command that evaluates files and directories in real time.

- 
- On some systems, 'locate' is not enabled by default.
 - Installing 'locate' on Centos 7
 - i. yum -y update (update the packages)
 - ii. yum -y install mlocate (install locate)
 - iii. Updatedb (update the locate search database)
 - Let's use the locate command to find files and directories that have 'sales' in the name. **\$ locate sales**
 - Let's create a new file. **\$ touch noloate**
 - Now run our locate command again, the file is not found.
 - Do it with the file's name, file still not found because the index is not up to date.
 - Use the find command to evaluate all these in real time and we can find the file with the find command.