# **Linux/Unix Pipes , Grep & Sort Command**

|  |
| --- |
| **What are pipes in Linux?**  Anyone new to Linux might wonder, what role do pipes play in running the operating system? Well, they are not the "real" pipes that you have in mind !  Linux/Unix  Pipes , Grep & Sort Command |

The symbol **'|'** denotes a pipe. If you want to use two or more commands at the same time and run them consecutively, you can use pipes. Pipes enables Unix/Linux users to create powerful commands which can perform complex tasks in a jiffy.

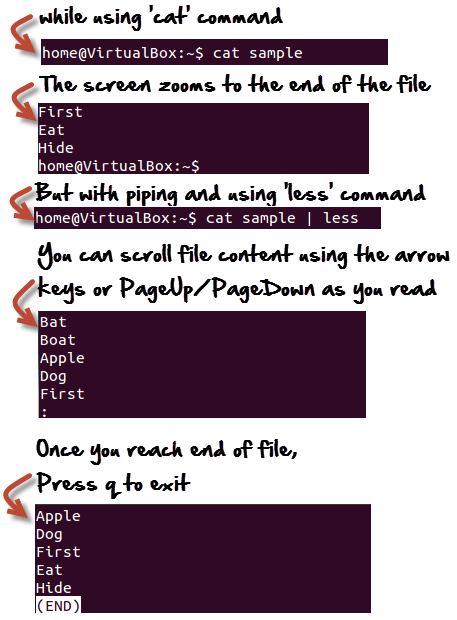
Let us understand this with an example.

When you use 'cat' command to view a file which spans multiple pages, the prompt quickly jumps to the last page of the file and you do not see the content in middle.

To avoid this, you can pipe the output of the 'cat' command to 'less' which will show you only one scroll length of content at a time.

***cat filename | less***

An illustration would make it clear.



## **'pg' and 'more' commands**

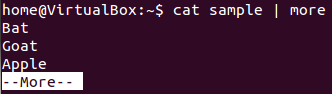
Instead o f 'less' , you can also use

**cat Filename | pg**

or

**cat Filename | more**

And, you can view the file in digestible bits and scroll down by simply hitting the enter key.



## **The 'grep' command**

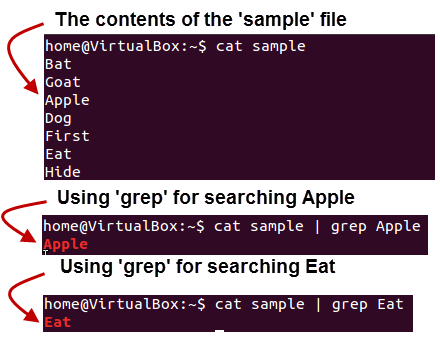
Suppose you want to search a particular information the postal code from a text file .

You may manually skim the content yourself to trace the information. A better option is to use the grep command. It will scan the document for the desired information and present the result in a format you want.

**Syntax:**

***grep***

Lets see it in action -



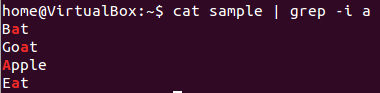
Here, **grep** command has searched the file 'sample', for the string 'Apple' and 'Eat'.

Following options can be used with this command.

|  |  |
| --- | --- |
| **Option** | **Function** |
| -v | Shows all the lines that do not match the searched string |
| -c | Displays only the count of matching lines |
| -n | Shows the matching line and its number |
| -i | Match both (upper and lower) case |
| -l | Shows just the name of the file with the string |

Let us try the first option **'-i'** on the same file use above -

Using the 'i' option grep has filtered the string 'a' (case-insensitive) from the all the lines .



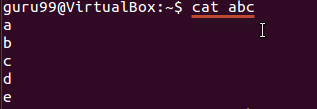
## **The 'sort' command**

This command helps in **sorting out the contents of a file alphabetically.**

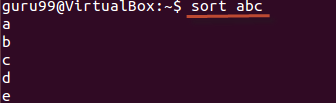
The syntax for this command is:

***sort Filename***

Consider the contents of a file



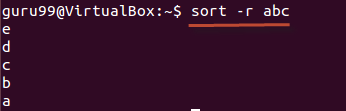
Using the sort command



There are **extensions** to this command as well and they are listed below.

|  |  |
| --- | --- |
| **Option** | **Function** |
| -r | Reverses sorting |
| -n | Sorts numerically |
| -f | Case insensitive sorting |

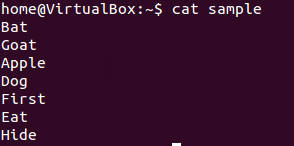
The example below, shows reverse sorting of the contents in file 'abc'.



|  |  |
| --- | --- |
| Linux/Unix  Pipes , Grep & Sort Command | **What is a filter ?** Filter is the **output from first command which becomes the input for the second one**.    When you pipe 2 commands, the "filtered " output of the first command is given to the next |

Let's understand this with help of an example.

We have the following file 'sample'

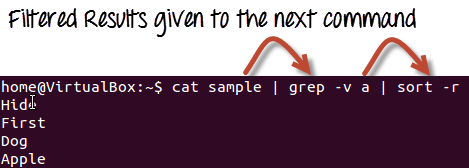


**We want to highlight** only the lines that do not contain the character 'a', but the result should be in reverse order .

For this, the following syntax can be uised.

***cat sample | grep -v a | sort - r***

Let us looks at the result.



Summary:

* Pipes '|' help combine 2 or more commands.
* A filter in a pipe is an output of one command which serves as input to the next.
* The grep command can be used to find strings and values in a text document
* 'sort' command sorts out the content of a file alphabetically
* less ,pg and more commands are used for dividing a long file into readable bits