**Date Literals:**

**$date +%d :** To display only date.

Ex: 30

**$date +%b :** To display the short name of the month

Ex: Jan

**$date +%m :** To display the month in number.

Ex: 1

**$date +%B :** To display the full form of the month.

Ex: January

**$date +%y :** To display the year.

Ex: 2015

**$date +%H :** To display the current time in hours.

Ex: 20

**$date +%M :** To display the current time minutes.

Ex: 07

**$date +%S :** To display the current time seconds.

Ex: 05

**$date +%r :** To display time in 12 hours format.

Ex: 08:08:58 PM

**$ls :** To display list of directories and files.

**$ls /opt** : To display the list of files and directories in opt directory.

**$ls -f :**  To display the all files in the current working directory. Do not sort, enable -aU, disable -lst

If files starts with .(dot) symbol those are hidden files.

**$ls -a** : (-a, --all) To display the all files even hidden files. do not hide entries starting with’ **.**’

**$ls -r :**  To display the reverse list of files and directories.

**$ls -l :** To display the long list of files and directories.

i.e Permissions,timestamps,owner names etc…

Ex: drwxr-xr-x. 2 root root 6 Mar 10 2016 xinetd.d

drwxr-xr-x. ---- File type and Permissions (ACL) Access Control List.

2 ----- no of links

Root ----- owner

Root ----- Group owner

6 ----- Size of file/directory

Mar 10 2016 -- Time stamp

xinetd.d -- Name of file/Directory

if list starts with “d” it indicates directory.

if list starts with “**-**” it indicates directory.

**$ls -lu :** To display the last access time.

Ls -lu filename/directoryname

**$ls-lr** : To display in reverse order.

**$ls-lt :** To display the recent first time stamp first.

**$ ls -ltu :** It displays the recent accessed file timestamp on top.

**$ ls -ltr :** It displays the recent accessed file timestamp on bottom.

**wildcard character**

These characters are used to write regular expressions.

| **Character** | **Description** | **Example** |
| --- | --- | --- |
| \* | Matches any number of characters. You can use the asterisk (**\***) anywhere in a character string. | **wh\*** finds what, white, and why, but not awhile or watch. |
| ? | Matches a single alphabet in a specific position. | **b?ll** finds ball, bell, and bill. |
| [ ] | Matches characters within the brackets. | **b[ae]ll** finds ball and bell, but not bill. |
| ! | Excludes characters inside the brackets. | **b[!ae]ll** finds bill and bull, but not ball or bell.  **Like “[!a]\*”** finds all items that do not begin with the letter a. |
| - | Matches a range of characters. Remember to specify the characters in ascending order (A to Z, not Z to A). | **b[a-c]d** finds bad, bbd, and bcd. |
| # | Matches any single numeric character. | **1#3** finds 103, 113, and 123. |

**$ls s\* :**  It displays all the files starts with S. and Folders starting with s and all files within that folder.

**$ls -d s\* :** It displays all directories and files starts with ‘s’.

$ **ls -d \*r** : It displays all directories and files ends with ‘r’.

**$ ls -d \*n\* :** it displays files and directories containing the character ‘n’ any where in the name.

**$ ls -d ???? :**  It displays the files and directories having 4 characters only.

**$ ls -d p??? :**  It displays the files and directories having 4 characters only but should start with ‘p’.

**$ ls -d ?e? :**  It displays the files and directories having 3 characters only but middle letter should be ‘e’.

**$ ls -d \*e? :** It displays file and directories having last but second character is ‘e’.

**$ ls -d [np]\* :**  It displays the files and directories starts with either n or p.

**$ ls -d \*[np] :**  It displays the files and directories ends with either n or p.

**$ ls -d [n-s]\* :**  It displays the files and directories starts with “n to s” any letter.

Ex: networks ppp request-key.d smartmontools

**$ls /etc/\*.conf:**  It shows the files from specified location.

$ls –R : It displays the file and directories recursively (or) used to show the aal data in one directory.

**WORKING WITH FILES**

**$ cat** : It is used to create new file, appending to the existing file and print the content of the file on standard output.

Options: -b Displays the line number

-e Append the $ Symbol at the end of the line.

-n Displays the line number even for empty lines also.

Ex: $cat > file1 Creates file with name file1

$cat < file1 Displays the output of file1.

$cat >> file1 Appends data to existing file1.

$ cat file1 >> file2 fiel1 data copied to file2.

**$touch :** To Create one or more empty files at a time and to change the timestamps.

**Touch Command Options**

1. -a, change the access time only
2. -c, if the file does not exist, do not create it
3. -d, update the access and modification times
4. -m, change the modification time only
5. -r, use the access and modification times of file
6. -t, creates a file using a specified time

http://www.tecmint.com/8-pratical-examples-of-linux-touch-command/