Mount and Umonut

mount command is used to mount the filesystem found on a device to big tree structure(**Linux** filesystem) rooted at '*/*'. Conversely, another command **umount** can be used to detach these devices from the Tree.

Syntax: mount [-t type] device dir
umount DIRECTORY/Device

Some Important Options:

l : Lists all the file systems mounted yet.
h : Displays options for command.
V : Displays the version information.
a : Mounts all devices described at /etc/fstab.
t : Type of filesystem device uses.
T : Describes an alternative fstab file.
r : Read-only mode mounted.

Note: The /etc/fstab file, short for "file systems table," is a system configuration file in Unix-like operating systems, including Linux. It is used to define how various storage devices, partitions, and network shares are mounted during the system's boot process.

Example:

1) sudo mount -l -t ext4 → Display all ext4 FS.
2) …

File Type	Command to create the File	Located in	The file type using “ls -l” is denoted using	FILE command output
Regular FIle	touch	Any directory/Folder	–	PNG Image data, ASCII Text, RAR archive data, etc
Directory File	mkdir	It is a directory	d	Directory
Block Files	fdisk	/dev	b	Block special
Character Files	mknod	/dev	c	Character special
Pipe Files	mkfifo	/dev	p	FIFO

Symbol Link Files	ln	/dev	l	Symbol link to <linkname>
Socket Files	socket() system call	/dev	s	Socket

Link: https://www.geeksforgeeks.org/how-to-find-out-file-types-in-linux/

Creation of Regular Files:

In Linux, we can create files in the following ways:
1) By using touch command (to create empty file)
2) By using cat command
3) By using editors like gedit, vi, nano etc

Stat

The stat is a command which gives information about the file and filesystem.
Stat command gives information such as the size of the file,
access permissions and the user ID and group ID,
birth time access time of the file.
Syntax: stat filename
Note: It also has some options…. Do it Yourself

Touch

The touch command is a standard command used in the UNIX/Linux operating system which is used to create, change and modify the timestamps of a file.
It is used to create a file without any content. The file created using the touch command is empty. This command can be used when the user doesn't have data to store at the time of file creation.
You can update the modification and access time of each file with the help of touch command.If we are using Touch Command, but the File is already available then what will happen?
The content of the file won't be changed. But last modified date and time (i.e., timestamp) will be updated

Syntax: `touch [options] file_name`

1) touch file1 → create or change timestamp (both access and modify time)

2) touch file1 file2 file3 → create multiple files
3) touch -a file1 → change access file of file1
4) touch -m file1 → change modify time of file1
5) touch -c fileName → don't create any file
6) touch -d DateString fileName → create a file with access and modify date as Date String. Used in conjunction with c to just update the time.
7) touch -t YYMMDDHHMM fileName → This is used to create a file using a specified time.
8) touch -r file1 file2 -----> use file1's timestamp as reference and change timestamp of file2 (now file2 timestamp will change and become same as file1)
9) try :- touch -r file2 -a file1 and observe using stat command what happens

Cat

Cat(concatenate) command is very frequently used in Linux. It reads data from the file and gives its content as output. It helps us to create, view, and concatenate files.
Create Files:

1) cat > newfile_name (In next lines write the content and do crtl+d for saving and exiting the file). → This command is used to create and add content to the file.If a file doesn't exist it creates a new file. If the file exists then it overwrites the previous content.
2) cat >> file_name(In next lines write the content and do crtl+d for saving and exiting the file). → This command appends the file if it is already created, else it creates the file and adds content to it.

Concatenate Files:

3) cat file1 > file2: Creates file2 and add file1 content to file2. If file2 already exist it is overwritten.
4) cat file1 >> file2: Creates file2 and add file1 content to file2. If file2 already exist, file1 is appended at the end of file2.
5) cat file1.txt file2.txt file3.txt > file4.txt → Same as 3 but content consists of multiple Files.

Regular Expressions and Wildcard Characters

If we want to represent a group of strings according to a particular pattern, then we should go for regular expressions.
By using wildcard characters, we can build regular expressions.
A wildcard character can be used as a substitute for required sequence of characters in the
regular expression.
1) * Represents zero or more characters
2) ? Represents only one character
3) [] Range of characters
4) [abc] Either a or b or c
5) [!abc] Any character except a,b and c
6) [a-z] Any lower case alphabet symbol
7) [A-Z] Any upper case alphabet symbol
8) [a-zA-Z] Any alphabet symbol
9) [0-9] Any digit from 0 to 9
10) [a-zA-Z0-9] Any alphanumeric character
11) [!a-zA-Z0-9] Except alpha numeric character (i.e special symbol)
12) [[:lower:]] Any lower case alphabet symbol
13) [[:upper:]] Any upper case alphabet symbol
14) [[:alpha:]] Any alphabet symbol
15) [[:digit:]] Any digit from 0 to 9
16) [[:alnum:]] Any alpha numeric character
17) [![:digit:]] Any character except digit
18){} List of files with comma separator
examples
1) To list out all files present in current working directory $ ls *
2) To list out all files with some extension $ ls *.*
3) To list out all files starts with a $ ls a*
4) To list out all files starts with a and ends with t $ ls a*t
5) To list out all .java files $ ls *.java
6) To list out all files where file name contains only 2 characters and first character should be 'a' $ ls a?
7) To list out all files where file name contains only 3 characters $ ls ???
8) To list out all files where file name contains atleast 3 characters $ ls ???*
9) To list out all files where file name starts with a or b or c $ ls [abc]*
10) To list out all files where file name should not starts with a, b and c $ ls [!abc]*
11) To list out all files starts with lower case aphabet symbol
$ ls [a-z]* OR $ls [[:lower:]]*
12) To list out all files starts with upper case aphabet symbol
$ ls [A-Z]* OR $ls [[:upper:]]*
13) To list out all files starts with digit.
$ ls [0-9]* OR $ls [[:digit:]]*

14) To list out all files where first letter should be upper case alphabet symbol, second
14) To list out all files where first letter should be upper case alphabet symbol, second letter should be digit and third letter should be lower case alphabet symbol.
$ ls [[:upper:]][[:digit:]][[:lower:]]
15) To list out all files starts with special symbol
$ls [![:alnum:]]*
16) To list out all files with .java and .py extension
$ ls {*.java, *.py}
Note: We can use these wildcard characters with the following commands also.
cp, mv, rm
17) To copy all files starts with digit to dir1 directory.
$cp [[:digit:]]* dir1
$cp [0-9]* dir1
18) To move all files starts with alphabet symbol and with .txt extension to dir2 directory?
$mv [[:alpha:]]*.txt dir2
19) Remove all files starts with a or b or c and ends with e or t.
$rm [abc]*[et]