. **Tips For Linux** BACK TO HOMEPAGE

>> How to find files in Linux using 'find'

Files can be found under Linux in many different ways. Using the find tool is one of the best ways to find files. The find tool has a huge number of parameters which can be set so that Linux finds exactly those files that you were searching for. Many users use the find tool with just the basic parameters. They get the results that they were looking for. Unfortunately most of the users don't spend time to learn all about find. If they do, they can make excellent use of this tool and I am sure you would be surprised at the possibilities.

In case you just want to know where a particular file exists on your system, and nothing else is required, then use *locate* tool. Article No.20 explains how to use *locate*.

Here are a few ways to use find

-

```
$ find / -name 'program.c' 2>/dev/null
$ find / -name 'program.c' 2>errors.txt
```

/	Start searching from the root directory (i.e / directory)
-name	Given search text is the filename rather than any other attribute of a file
'program.c'	Search text that we have entered. Always enclose the filename in single quotes.. why to do this is complex.. so simply do so.

Note : 2>/dev/null is not related to find tool as such. 2 indicates the error stream in Linux, and /dev/null is the device where anything you send simply disappears. So 2>/dev/null in this case means that while finding for the files, in case any error messages pop up simply send them to /dev/null i.e. simply discard all error messages.

Alternatively you could use 2>error.txt where after the search is completed you would have a file named error.txt in the current directory with all the error messages in it.

-

```
$ find /home/david -name 'index*'
$ find /home/david -iname 'index*'
```

The 1st command would find files having the letters index as the beginning of the file name. The search would be started in the directory /home/david and carry on within that directory and its subdirectories only.
The 2nd command would search for the same, but the case of the filename wouldn't be considered. So all files starting with any combination of letters in upper and lower case such as INDEX or indEX or index would be returned.

-

```
$ find -name met*
```

The above command would start searching for the files that begin with the letters 'met' within the current directory and the directories that are present within the current directory. Since the directory is not specified as the the second parameter, Linux defaults to using the current directory as the one to start the search in.

-

```
$ find /mp3collection -name '*.mp3' -size -5000k
$ find / -size +10000k
```

The 1st command would find within a directory called /mp3collection, only those mp3 files that have a size less than 5000 Kilobytes (< 5MB)
The 2nd command would search from the / directory for any file that is larger than 10000k (> 10MB)

-

```
$ find /home/david -amin -10 -name '*.c'
$ find /home/david -atime -2 -name '*.c'
$ find /home/david -mmin -10 -name '*.c'
$ find /home/david -mtime -2 -name '*.c'
```

The 1st commmand searches for those files that are present in the directory /home/david and its subdirectoires which end in .c and which have been accessed in the last 10 minutes.
The 2nd command does the same but searches for those files that have been accessed in the last 10 hours.
The 3rd and the 4th commands do the same as the 1st and 2nd commands but they search for modified files rather than accessed files. Only if the contents of the files have been modified, would their names be returned in the search results.

-

```
$ find / -mount -name 'win*'
```

This command searches for files starting with the letters 'win' in their filenames. The only difference is that the mounted filesystems would not be searched for this time. This is useful when you have your Windows partitions mounted by default. And a search for 'win' might return many files on those partitions, which you may not be really interested in. This is only one use of -mount parameter.

-

```
$ find /mp3-collection -name 'Metallica*' -and -size +10000k
$ find /mp3-collection -size +10000k ! -name "Metallica*"
$ find /mp3-collection -name 'Metallica*' -or -size +10000k
```

Boolean operators such as AND, OR and NOT make find an extremely useful tool.
The 1st command searches within the directory /mp3-collection for files that have their names beginning with 'Metallica' and whose size is greater than 10000 kilobytes (> 10 MB).
The 2nd command searches in the same directory as above case but only for files that are greater than 10MB, but they should not have 'Metallica' as the starting of their filenames.
The 3rd command searches in the same directory for files that begin with 'Metallica' in their names or all the files that are greater than 10 MB in size.

-

The **exec** option is probably the most important feature of the find tool. The exec command allows you to execute a particular command on the results of the find command. A simple demonstration of this feature is shown below. Its upto your imagination to make maximum use of this feature. Suppose you wanted to see the details of the files (read, write, execute permission, file size, owner etc..) that have been returned as a search result you could do the following

```
$ find / - name 'Metallica*' -exec ls -l {\}\ \;
```

This command would find all the files on your system that begin with the letters 'Metallica' and would then execute the 'ls -l' command on these files. So basically you would be able to see the details of the files that were returned according to your search criteria.

The words following the -exec option is the command that you want to execute i.e. ls -l in this case.
{\}\ is basically an indicator that the filenames returned by the search should be substituted here.
\; is the terminating string, and is required at the end of the command

-

Find has plenty of more options, which I shall discuss in some other article.

> BACK TO HOMEPAGE