Why use apt and not apt-get or another command? The apt command has been available since 2014. It has a command structure that is similar to apt-get but was created to be a more pleasant experience for typical users. Simple software management tasks like install, search and remove are easier with apt.

**In this article, we will explain how to delete directories in Linux using the rmdir, rm, and find commands.**

Before You Begin

When removing a directory using a desktop file manager, the directory is actually moved to the Trash and can be easily recovered.

Be extra careful when removing files or directories from the command line because once the directory is deleted using the commands explained in this article, it cannot be fully recovered.

On most Linux file systems, deleting a directory requires write permission on the directory and its content. Otherwise, you will get “Operation not permitted” error.

Directory names with a space in them must be escaped with a backslash (/).

**Removing Directories with rmdir**

rmdir is a command-line utility for deleting empty directories. It is useful when you want to delete a directory only if it is empty, without needing to check whether the directory is empty or not.

To delete a directory with rmdir, type the command followed by the name of the directory you want to remove. For example, to delete a directory named dir1 you would type:

rmdir dir1

If the directory is not empty, you will get the following error:

rmdir: failed to remove 'dir1': No such file or directory

In this case, you will need to use the rm command or manually remove the directory contents before you can delete it.

**Removing Directories with rm**

rm is a command-line utility for deleting files and directories. Unlike rmdir the rm command can delete both empty and non-empty directories.

By default, when used without any option rm does not remove directories. To delete an empty directory, use the -d (--dir) option and to delete a non-empty directory, and all of its contents use the -r (--recursive or -R) option.

For example to delete a directory named dir1 along with all of its contents you would type:

rm -r dir1

If a directory or a file within the directory is write-protected, you will be prompted to confirm the deletion. To remove a directory without being prompted, use the -f option:

rm -rf dir1

To remove multiple directories at once, invoke the rm command, followed by the names of the directories separated by space. The command below will remove each listed directory and their contents:

rm -r dir1 dir2 dir3

The -i option tells rm to prompt you to confirm the deletion of each subdirectory and file. If the directory contains a lot of files, this can be a little annoying, so you may consider using the -I option what will prompt you only once before proceeding with the deletion.

rm -rI dir1

To remove the directory type y and hit Enter.

rm: remove 1 argument recursively? y

You can also use regular expansions to match and delete multiple directories. For example, to remove all first-level directories in the current directory that ends with \_bak, you would use the following command:

rm -r \*\_bak

Using regular expansions when removing directories may be risky. It is recommended first to list the directories with the ls command so that you can see what directories will be deleted before running the rm command.

**Removing Directories with find**

find is a command-line utility that allows you to search for files and directories based on a given expression and perform an action on each matched file or directory.

The most common scenario is to use the find command to delete directories based on a pattern. For example, to delete all directories that end with \_cache in the current working directory, you would run:

find . -type d -name '\*\_cache' -exec rm -r {} +

Let’s analyze the command above:

/dir - recursively search in the current working directory (.).

-type d - restricts the search to directories.

-name '\*\_cache' - search only directories that end with \_cache

-exec - executes an external command with optional arguments, in this case, that is rm -r.

{} + - appends the found files to the end of the rm command.

Removing all empty directories

To remove all empty directories in a directory tree you would run:

find /dir -type d -empty -delete

Here is an explanation for the options used:

/dir - recursively search in the /dir directory.

-type d - restricts the search to directories.

-empty - restricts the search only to empty directories.

-delete - deletes all found empty directories in the subtree. -delete can delete only empty directories.

Use the -delete option with extreme caution. The find command line is evaluated as an expression, and if you add the -delete option first, the command will delete everything below the starting points you specified.

Always test the command first without the -delete option and use -delete as the last option.

/bin/rm: Argument list too long

This error message appears when you use the rm command to remove a directory that contains a huge number of files. This happens because the number of files is larger than the system limit on the size of the command line argument.

There are several different solutions to this problem. For example, you can cd to the directory and manually or using a loop to remove sub-directories one by one.

The easiest solution is first to delete all files within the directory with the find command and then delete the directory:

find /dir -type f -delete && rm -r /dir

Conclusion

With rm and find you can delete directories based on different criteria fast and efficient.

Deleting directories is a simple and easy process, but you must be cautious not to delete important data.