biswaraj sahoo's blog

Dashboard

File Management in Linux

Biswaraj Sahoo
·Jun 10, 2023·
6 min read

TABLE OF CONTENTS

In Linux, most of the operations are performed on files. And to handle these files Linux has directories also known as folders which are maintained in a tree-like structure. Though, these directories are also a type of file themselves. Linux has 3 types of files:

1. **Regular Files:** It is the common file type in Linux. it includes files like – text files, images, binary files, etc. Such files can be created using the touch command. They consist of the majority of files in the Linux/UNIX system. The regular file contains ASCII or Human Readable text, executable program binaries, program data and much more.

2. **Directories:** Windows call these directories as folders. These are the files that store the list of file names and the related information. The root directory(/) is the base of the system, /home/ is the default location for user's home directories, /bin for Essential User Binaries, /boot – Static Boot Files, etc. We could create new directories with **mkdir command**.

3. **Special Files:** Represents a real physical device such as a printer which is used for IO operations. Device or special files are used for device Input/Output(I/O) on UNIX and Linux systems. You can see them in a file system like an ordinary directory or file.

There are two types of special files for each device, i.e. character special files and block special files.

1. Files Listing

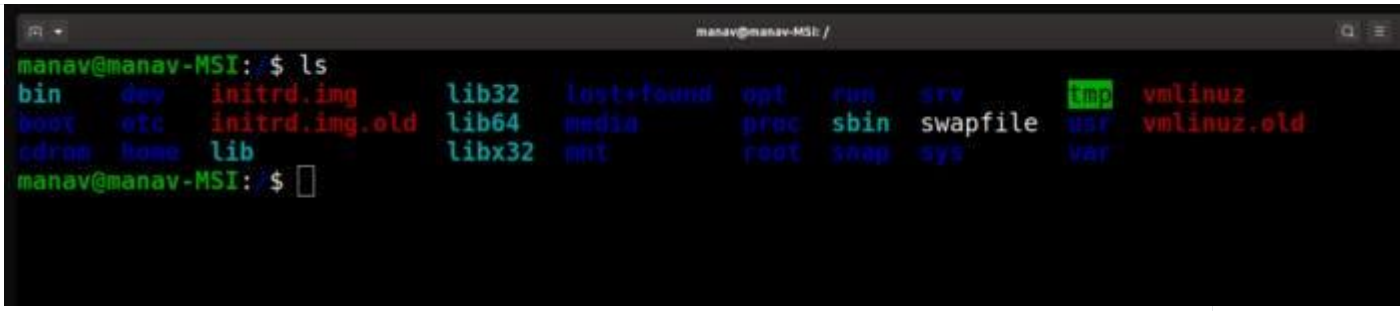
To perform Files listings or to list files and directories **ls command** is used

COPY

COPY

COPY
COPY

```
$ls
```



2. Creating Files

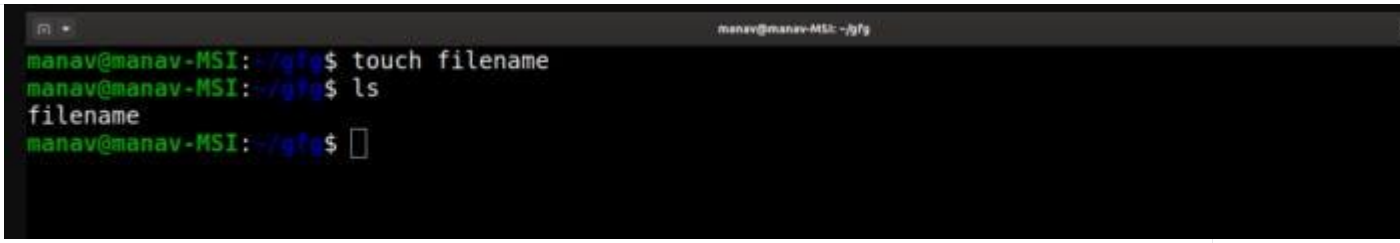
touch command can be used to create a new file. It will create and open a new blank file if the file with a filename does not exist. And in case the file already exists then the file will not be affected.

COPY

COPY

COPY
COPY

```
$touch filename
```



3. Displaying File Contents

cat command can be used to display the contents of a file. This command will display the contents of the 'filename' file. And if the output is very large then we could use more or less to fit the output on the terminal screen otherwise the content of the whole file is displayed at once.

COPY

COPY

COPY
COPY

```
$cat filename
```

4. Copying a File

cp command could be used to create the copy of a file. It will create the new file in destination with the same name and content as that of the file 'filename'.

COPY

COPY

COPY

COPY

```
$cp source/filename destination
```

5. Moving a File

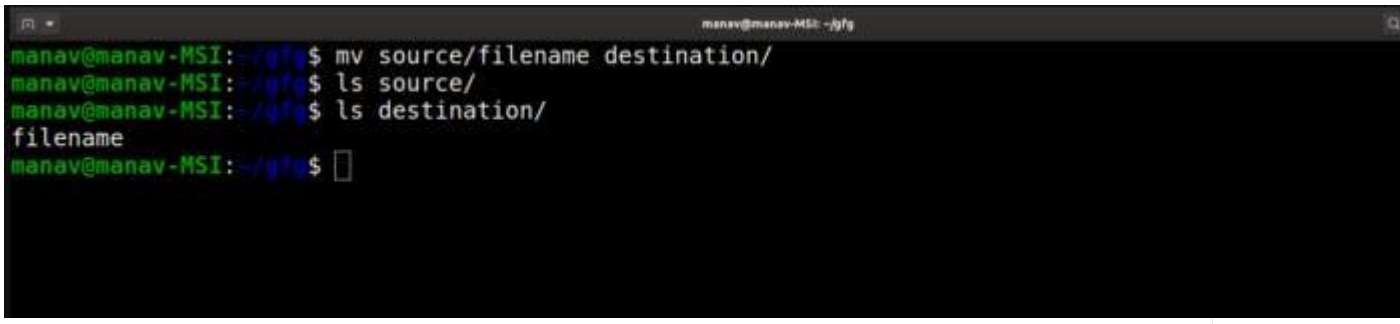
mv command could be used to move a file from source to destination. It will remove the file filename from the source folder and would be creating a file with the same name and content in the destination folder.

COPY

COPY

COPY
COPY

```
$mv source/filename destination/
```



6. Renaming a File

mv command could be used to rename a file. It will rename the filename to new_filename or in other words, it will remove the filename file and

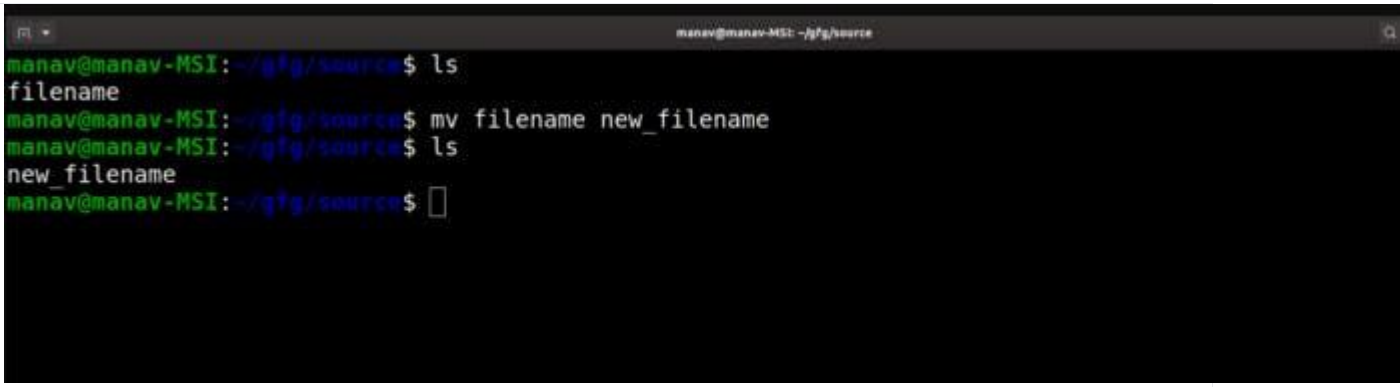
would be creating a new file with the new_filename with the same content and name as that of the filename file.

COPY

COPY

COPY
COPY

```
$mv filename new_filename
```



7. Deleting a File

rm command could be used to delete a file. It will remove the filename file from the directory.

COPY

COPY

COPY
COPY

```
$rm filename
```

Directory Manipulation Commands in Linux

- `pwd` Get the full path of the current working directory.
- `cd -` Navigate to the last directory you were working in.
- `cd ~` or just `cd` Navigate to the current user's home directory.
- `cd ..` Go to the parent directory of current directory (mind the **space** between `cd` and `..`)

Listing Files Inside a Directory

- `ls -l` List the files and directories in the current directory in long (table) format (It is recommended to use `-l` with `ls` for better readability).
- `ls -ld dir-name` List information about the directory `dir-name` instead of its contents.
- `ls -a` List all the files including the hidden ones (File names starting with a `.` are hidden files in Linux).
- `ls -F` Appends a symbol at the end of a file name to indicate its type (`*` means executable, `/` means directory, `@` means symbolic link, `=` means socket, `|` means named pipe, `>` means door).
- `ls -lt` List the files sorted by last modified time with most recently modified files showing at the top (remember `-l` option provides the long format which has better readability).

- `ls -lh` List the file sizes in human readable format.
- `ls -lR` Shows all subdirectories recursively.
- `tree` Will generate a tree representation of the file system starting from the current directory.

File or Directory - Create, Copy and Remove Commands in Linux

- `cp -p source destination` Will copy the file from `source` to `destination` directory. `-p` stands for preservation. It preserves the original attributes of file while copying like file owner, timestamp, group, permissions etc.
- `cp -R source_dir destination_dir` Will copy source directory to specified destination recursively.
- `mv file1 file2` In Linux there is **no** rename command as such. Hence `mv` moves/renames the `file1` to `file2`.
- `rm -i filename` Asks you before every file removal for confirmation. IF YOU ARE A NEW USER TO LINUX COMMAND LINE, YOU SHOULD ALWAYS USE rm -i. You can specify multiple files.
- `rm -R dir-name` Will remove the directory `dir-name` recursively.
- `rm -rf dir-name` Will remove the directory dir recursively, ignoring non-existent files and will never prompt for anything. BE CAREFUL USING THIS COMMAND! You can specify multiple directories.
- `rmdir dir-name` Will remove the directory `dir-name`, if it's empty. This command can only remove empty directories.
- `mkdir dir-name` Create a directory `dir-name`.
- `mkdir -p dir-name/dir-name` Create a directory hierarchy. Create parent directories as needed, if they don't exist. You can specify multiple directories.

- `touch filename` Create a file filename, if it doesn't exist, otherwise change the timestamp of the file to current time.

File or Directory - Permissions and Groups Commands

- `chmod < specification > filename` Change the file permissions. Specifications = `u` user, `g` group, `o` other, `+` add permission, `-` remove, `r` read, `w` write, `x` execute.
- `chmod -R < specification > dir-name` Change the permissions of a directory recursively. To change the permission of a directory and everything within that directory, use this command.
- `chmod go=+r myfile` Add read permission for the owner and the group.
- `chmod a +rwx myfile` Allow all users to read, write or execute `myfile`.
- `chmod go -r myfile` Remove read permission from the group and others.
- `chown owner1 filename` Change ownership of a file to user `owner1`.
- `chgrp grp_owner filename` Change primary group ownership of file `filename` to group `grp_owner`.
- `chgrp -R grp_owner dir-name` Change the primary group ownership of directory `dir-name` to group `grp_owner` recursively. Use this command to change group ownership of a directory and everything within that directory.

WRITTEN BY

Biswaraj Sahoo

--AWS Community Builder | DevOps Engineer | Docker | Linux | Jenkins | AWS | Git | Terraform | Docker | kubernetes

Empowering communities via open source and education.

Connect with me over linktrr: https://linktr.ee/biswaraj333

Subscribe to my newsletter

Read articles from **biswaraj sahoo's blog** directly inside your inbox. Subscribe to the newsletter, and don't miss out.

SUBSCRIBE

File Management in Linux

WRITTEN BY

Biswaraj Sahoo

--AWS Community Builder | DevOps Engineer | Docker | Linux | Jenkins | AWS | Git | Terraform | Docker | kubernetes

Empowering communities via open source and education.