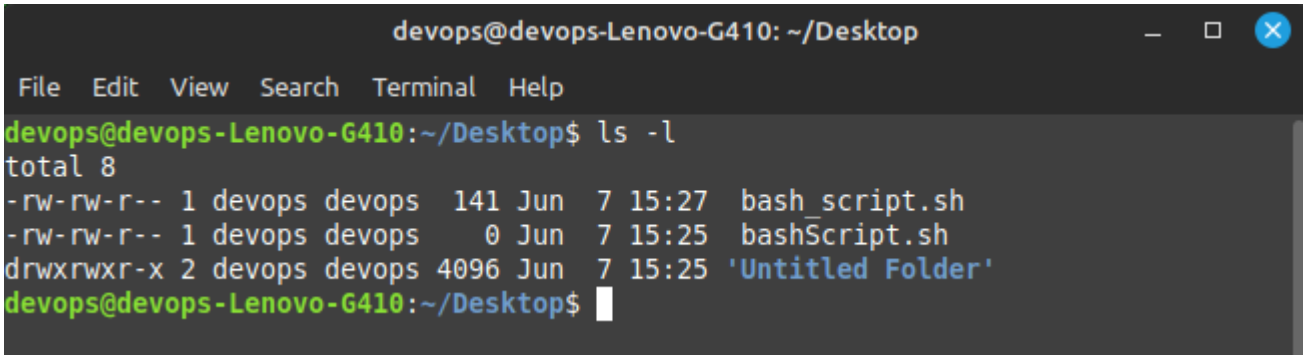
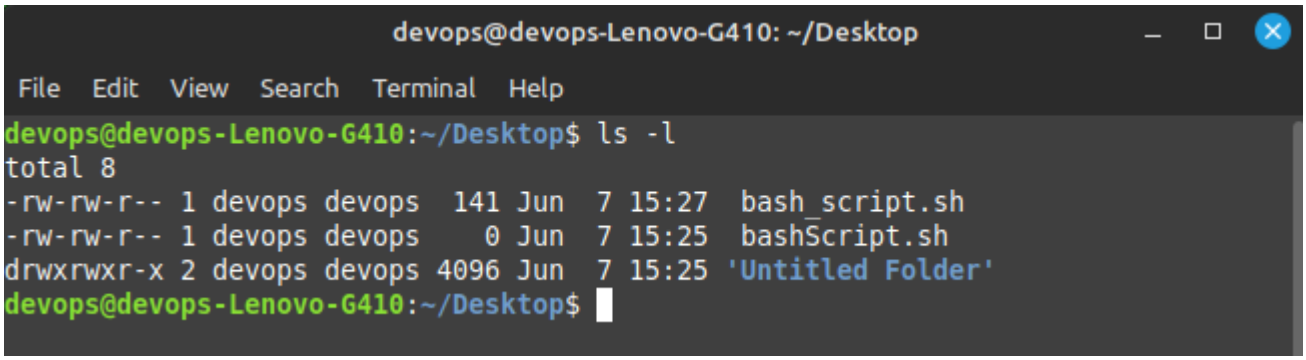
File System and File Permission

Last updated by | Shakibe Hasan | Jun 13, 2023 at 5:07 PM GMT+6

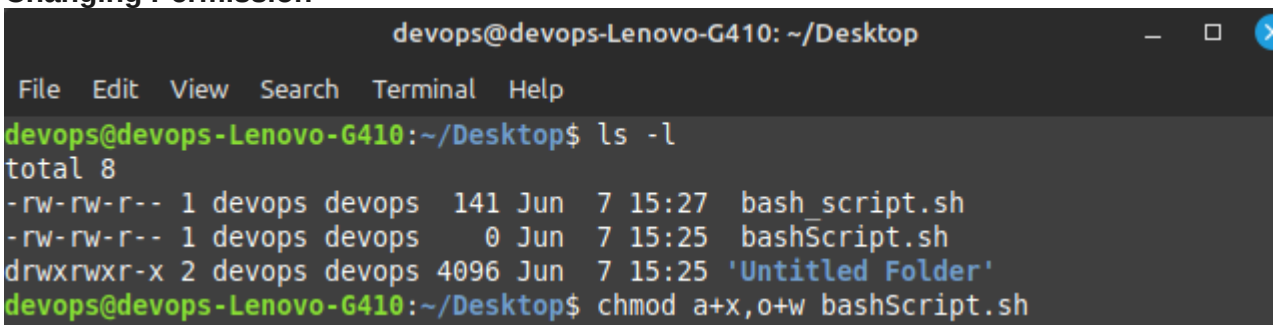


- The **first** column represents the file type and file permissions.
  where,
  (-) regular file
  (d) directory
  (l) link
  (c) special file
  (s) socket
  (p) named pipe
  (b) blocked device
- The **second** column represents the number of memory blocks.
- The **third** column represents the owner of the file or the superuser, who has the administrating power.
- The **fourth** column represents the group of owner/superuser.
- The **fifth** column represents the file size.
- The **sixth** column represents the date and time when the file was created or lastly modified.
- The **last** column represents the name of the file or the directory.

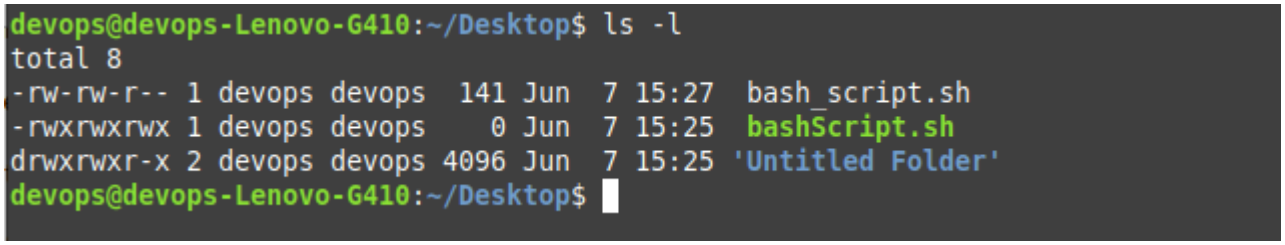


- The first slot(-) represent a file name.
- Next 3 slots(rw-) specify permissions used by assigned owner. Here read and write permission is included and execute permission has been denied.
- Next three slots (rw-) specify the permissions used by the group members who own the directory. These permissions include read and write, but do not include execute permission.
- Next three slots (r--) specify the permissions used by the third=party users. These permissions include read permission only. Here, read and write both the permissions have been denied

Changing Permission



Here changed the permission for bashScript.sh file using **chmod** command.
----------from the top image, we can see, the file "bashScript.sh" has only read and write permission assigned.
For that the file cant executed as execution permission has been denied.
By using chmod command added execution permission{ chmod a+x,o+w bashScript.sh}



In this image, we can see now bashScript.sh file as permission for execution.

Relative Path Vs Absolute Path:

The topmost directory in any filesystem is the root directory denoted by the slash (/). We can describe the location of any file or directory in the filesystem with the absolute path. That means we will take every step starting from the root directory or the absolute beginning of the filesystem.

An absolute path is unambiguous and may be inconvenient to work with, especially if we are working with deeply nested directories. To get a simpler mode of the things, we can use the Relative path instead. Such that, if we are working with files in or near the present working directory, this can save us from a lot of typing.

Ref: javatpoint.com