# File Ownership:

Every file and directory in Linux has three kinds of owners.

1. **User**: User is the owner of the file. When you create a file, you become the owner of the file.
2. **Group**: Every user is a part of a certain group. A group consists of several users. This is one way to manage users in a multiuser environment.
3. **Others**: Others consists of all the users on the system.

# File permissions:

Every file and directory in Linux has three permissions for all three kind of users

Permissions for files:

1. **Read**: Can view and copy.
2. **Write**: Can modify the contents.
3. **Execute**: Can run the file.

Permissions for directories:

1. **Read**: Can list and copy files from directories.
2. **Write**: Can add or delete files.
3. **Execute**: Can enter the directory

## How to check the file permissions:

We can use ls -l to check the file permissions.

Each letter denotes a particular permission:

1. r: Read permission
2. w: Write permission
3. x: Execute permission
4. -: No permission set

# Changing File permissions:

We can use chmod to change permissions. Each file permission is represented by a number.

* Read: 4
* Write: 2
* Execute : 1
* No permissions: 0
* 3(1+2) : -wx
* 5(4+1): r-x
* 6(4+2): rw-
* 7(4+2+1): rwx

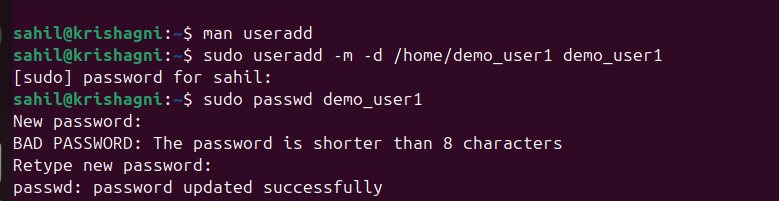
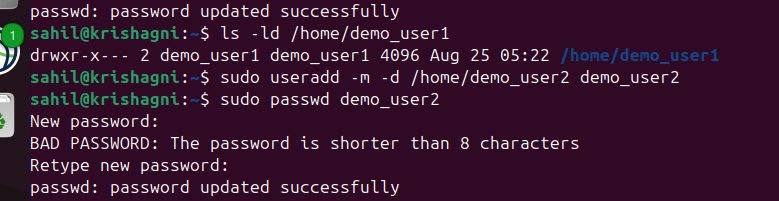
# Change file Ownership

To change the ownership of a file or directory, we can use the chown command.

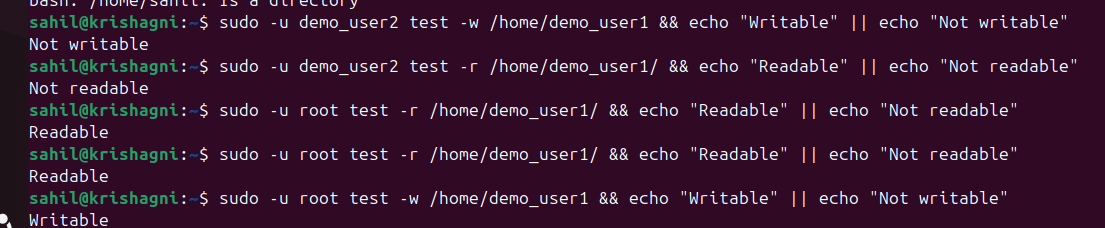
1. To change the use owner of a file: chown <new\_user> <file\_name>
2. To change user as well as group: chown <new\_user>:<new\_user\_group> <file\_name>
3. To change only group: chown :<new\_user\_group> <filename>

Create two non-sudo users. Each user has a private directory inside their home.

Created users and private directory in home.



**Verification:**

Only the root and the user itself is able to read and write   
  


Bandit

### How to read files with - prefix

cat ./-filename

### How to find a file based on properties like

* human-readable
* 1033 bytes in size
* not executable

**find ./inhere -type f -size 1033c ! -executable -exec cat {} \;**

./inhere: its the location.

-type : f (file), d(directory)

-size: if size is a parameter.

! -executable: not executable

-exec: executes the command on each file that matches the criteria.

Cat: reads the file

{}: placeholder for the file found.

**find / -type f -user bandit7 -group bandit6 -size 33c 2>/dev/null**

2>/dev/null : directs all error msgs in /dev/null silencing them.

### Find word in file

grep 'millionth' data.txt

### Find Unique line in a file

sort data.txt | uniq -u

Sort the data using sort and pass the output as input to uniq -u which gives only uniq lines in the file.

1. Extract Strings from binary file

strings data.txt

1. Show active processes   
   ps -ef
   1. ps: shows list of processes running on the system
   2. -e: show all processes
   3. -f: show full format
2. Count the number of files in a directory
   1. ls -1|wc -l
      * 1. ls -l: lists only file names in a column
        2. wc -l: count the number of lines
3. Tee
   1. Writes output of a command to a file as well as prints in terminal.
   2. ls | tee data.txt
   3. Writes the output of ls to data.txt and prints it in terminal screen.
4. Awk: text processing tool in linux
   1. It fast.
   2. Can do filtering, counting and formatting
   3. awk '{ print $1 }' server.log : print first column of each line.
   4. awk '$3=="ERROR" { count++ } END { print "ERROR count:", count }' server.log: count error messages.
   5. Begin: Runs once before running any line.
   6. End: Runs once after all lines are read.
      1. Good for reporting and summaries
      2. Printing max /min
5. Sed: (short for **Stream Editor**) in Linux is a powerful command-line tool used to **search, find, replace, insert, and delete text** in a stream (like a file or input from a pipeline).
   1. -E extended expression
   2. -s substitute
   3. -d delete