1 . Users

2 . Groups

3 . File Permissions and Ownership

Table of contents

01

Types of users

What is a user

- A user account is used to provide security boundaries between different people and programs that can run commands.
- Users have usernames to identify them to human users and make them easier to work with. Internally, the system distinguishes user accounts by the unique identification number assigned to them, **the user ID or UID**.
- User accounts are fundamental to system security. Every process (running program) on the system runs as a particular user. Every file has a particular user as its owner. File ownership helps the system enforce access control for users of the files.

There are three main types of user account:

1) Superuser (root) -> UID = 0:

The Administrator of the system and has all permissions

- Modify the system configurations
- Manage other users and permissions
- Access or delete any file

2) System User (Service Account) -> UID (1: 999):

created automatically by the system or when installing a software to run background service (**daemons**)

- Security Isolation
- Least Privilege Principle
- Track the actions of different services

3) Regular User -> UID >= 1000 :

Achieve day-to-day tasks and have limited permissions to ensure the system remains secure and stable.

02

Groups

Group

A group can contain multiple users. All users belonging to a group will have the same permissions access to the files.

Groups have group names to make them easier to work with. Internally, the system distinguishes groups by the unique identification number assigned to them, the group ID or GID.

Primary Groups

A primary group is the default group assigned to a user when they are created on the system. By default, this group has the same name as the user. Any files created by the user will belong to their primary group unless specified otherwise. Each user can have only one primary group.

Supplementary Groups

Any other group a user belongs to other than the primary group. The user can be member of multiple secondary groups.

03

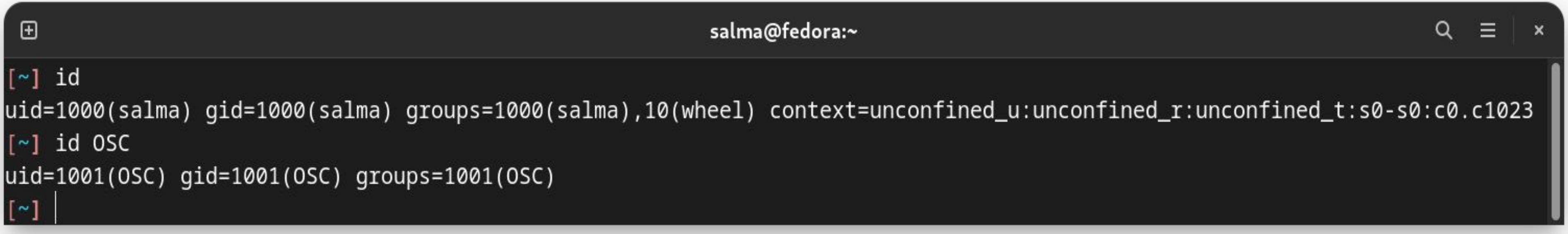
Dealing with users and groups

The id command

To show informaton about the current logged in user.

id username

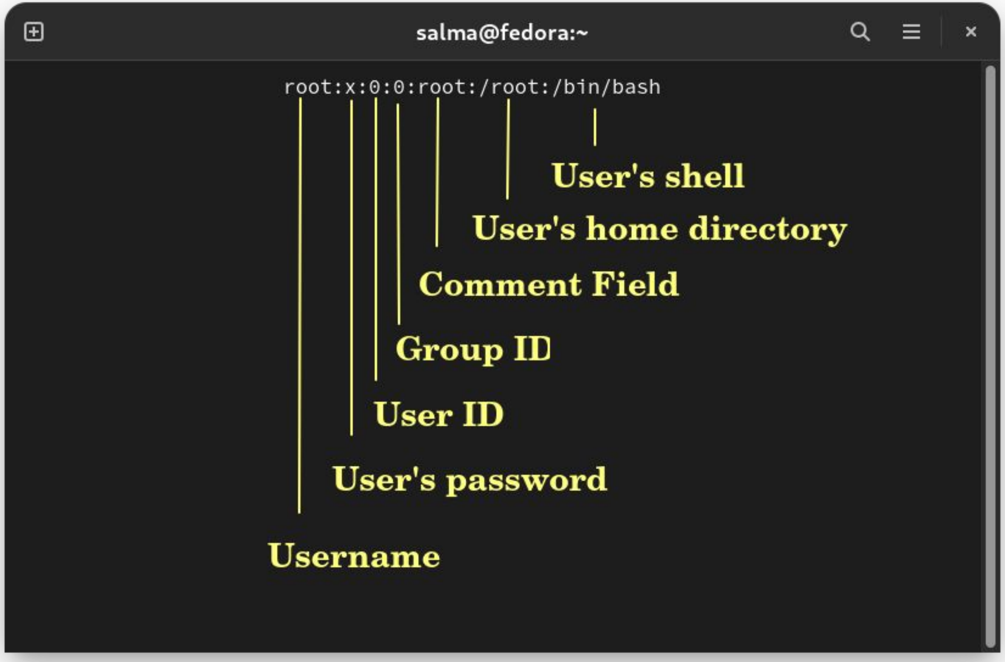
Shows information about the user that you are asking for.



```
[~] id
uid=1000(salma) gid=1000(salma) groups=1000(salma),10(wheel) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[~] id OSC
uid=1001(OSC) gid=1001(OSC) groups=1001(OSC)
[~]
```

/etc/passwd File

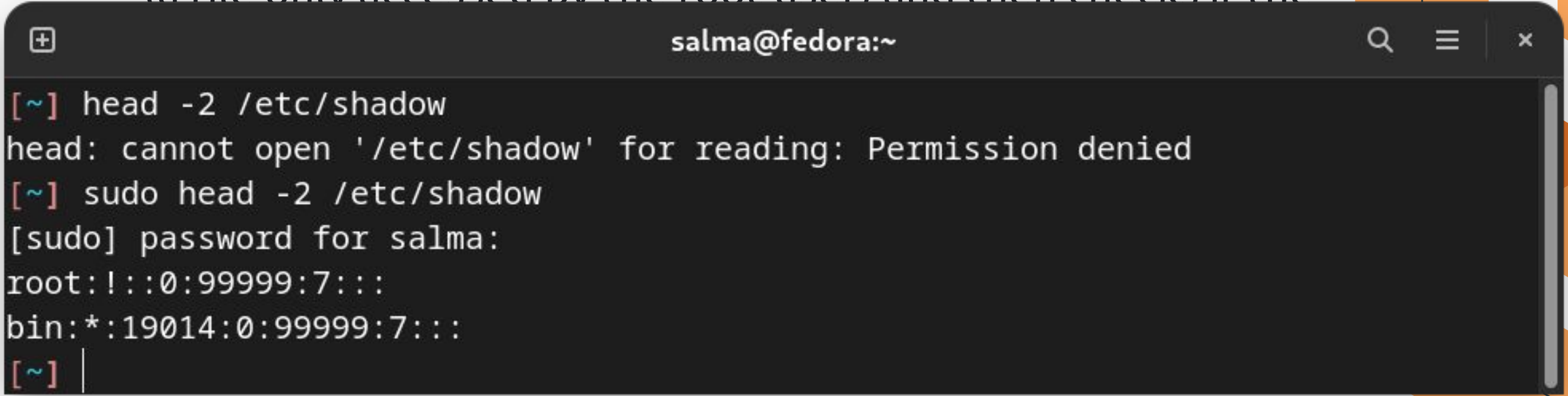
Stores information about the users on the system, each line is giving information about a different user.



```
root:x:0:0:root:/root:/bin/bash
```

Gaining super access

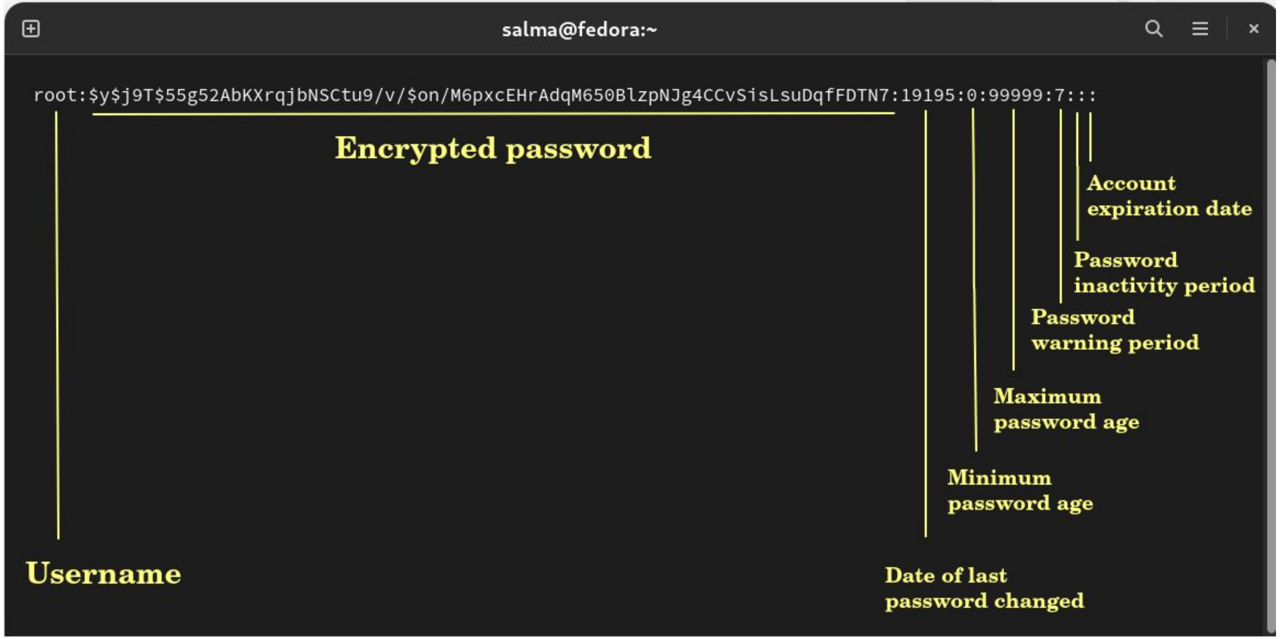
- When we use the sudo command before a command that is reserved for the root user only , sudo searches the “sudoers ” file (a file only accessed by the root user) and then checks if the



```
[~] head -2 /etc/shadow
head: cannot open '/etc/shadow' for reading: Permission denied
[~] sudo head -2 /etc/shadow
[sudo] password for salma:
root:!::0:99999:7:::
bin:*:19014:0:99999:7:::
[~]
```

/etc/shadow File

Store information about user authentication. It requires superuser read permissions.



```
root:$y$j9T$55g52AbKXrqjbNSCtu9/v/$on/M6pxcEHrAdqM650BlzpNJg4CCvSisLsuDqfFDTN7:19195:0:99999:7:::
```

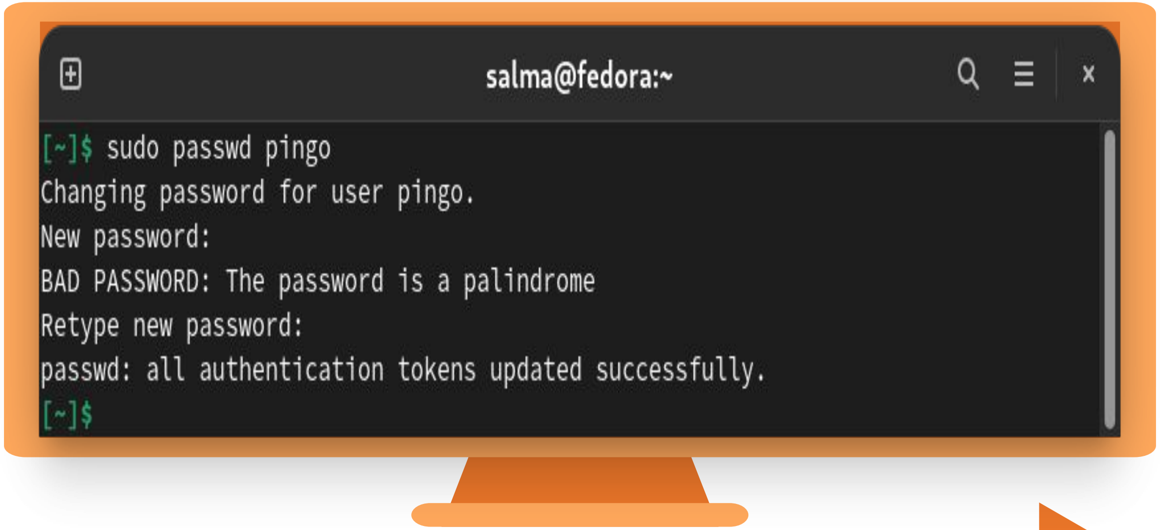
Creating users

- Only the root or a user with sudo privileges can create new user accounts.
- Syntax: useradd [options] <username>
- When executed without any option, useradd creates a new user account using the default settings specified in the /etc/default/useradd file. The variables defined in this file differ from distribution to distribution, which causes the useradd command to produce different results on different systems.
- **-m:** option to create a user with home directory.

Setting/Changing Passwords

- To be able to log in as the newly created user, you need to set the user password. To do that run
  the **passwd** command followed by the username

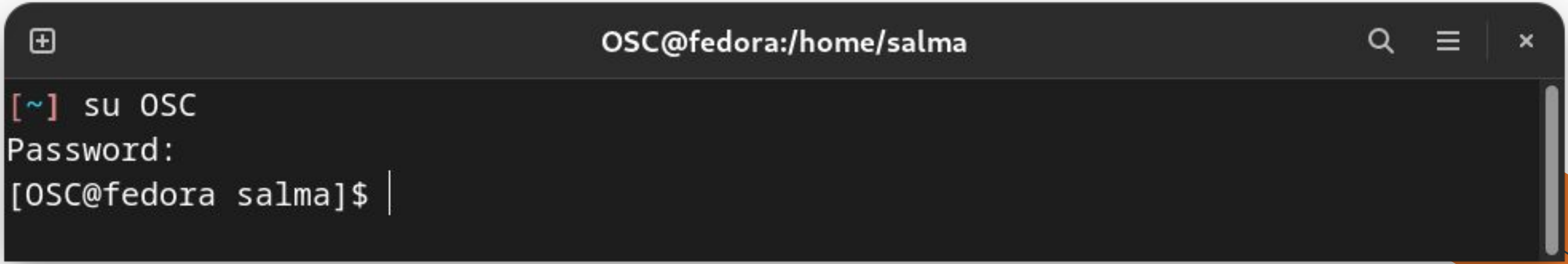
- Of course only the root user or a user with sudo privilege to change or set passwords for other accounts.



```
[~]$ sudo passwd pingo
Changing password for user pingo.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: all authentication tokens updated successfully.
[~]$
```

Switching users

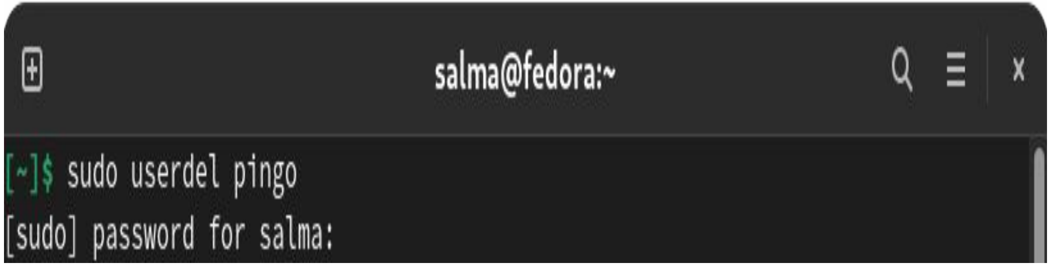
- We can switch between users using **su username**
- The password for the user we are switching to is needed unless you are the root user.



```
[~] su OSC
Password:
[OSC@fedora salma]$
```

Deleting users

- **userdel** is used to remove the details of username from **/etc/passwd** without removing the user's home directory by default. If the -r flag is specified, the **userdel** command also removes the user's home directory.



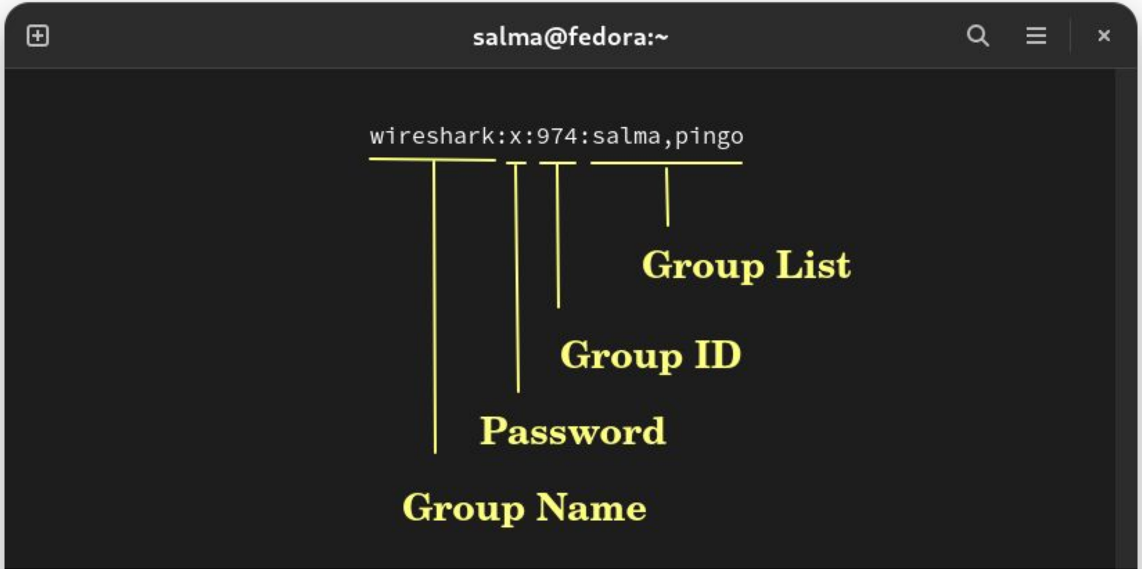
```
[~]$ sudo userdel pingo
[sudo] password for salma:
```

Note :

If a user is deleted without removing its own home directory the system will have files that are owned by an unassigned UID. This situation can lead to information leakage and other security issues.

Dealing with groups

- **/etc/group** file stores information about all groups in the system.



```
wireshark:x:974:salma,pingo
```

groups command

- You can use the command groups to find all the groups you are a member of.

To list all groups you are a member of: **groups**

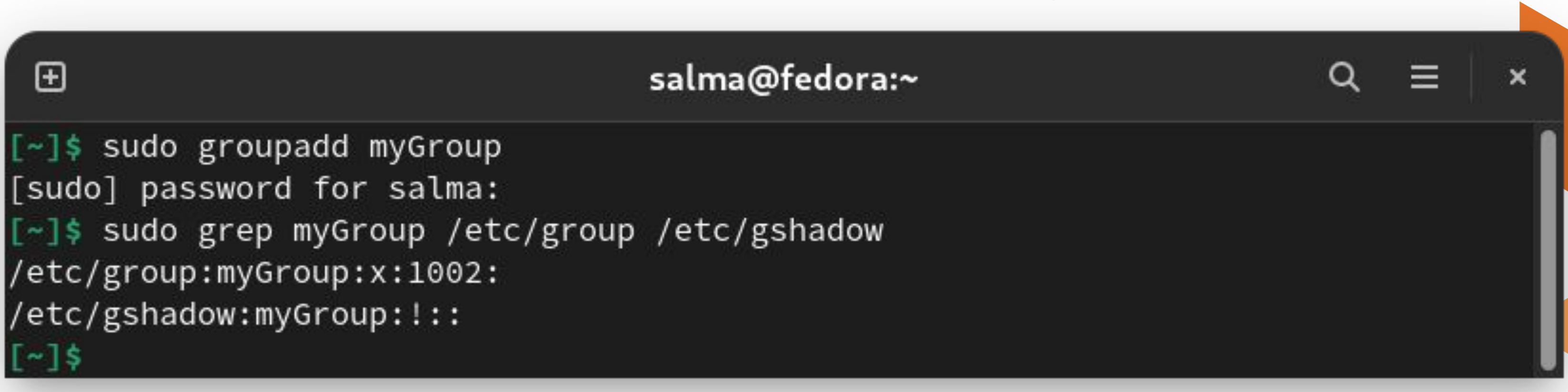
To list all groups of a specific user: **groups username**

Creating groups

- Syntax : groupadd [OPTIONS] Group_name

- Only the root or a user with sudo privileges can create new groups.

- When invoked, groupadd creates a new group using the options specified on the command line plus the default values specified in the /etc/login.defs file.



```
[~]$ sudo groupadd myGroup
[sudo] password for salma:
[~]$ sudo grep myGroup /etc/group /etc/gshadow
/etc/group:myGroup:x:1002:
/etc/gshadow:myGroup:!::
[~]$
```

Adding user to group

- Existing users accounts are added to groups using the usermod command.

- syntax : **usermod [options] <group name> <username>**

So to add the user OSC to the group test Group we will write **usermod -a -G testGroup OSC**

- The -G option tells the command that we will add the user to a supplementary group . The -a option puts the command in append mode ; other wise , the command will remove the user from

all groups unspecified in the command.

Deleting groups

- Groups are deleted using the **groupdel** command
- syntax : **groupdel <group name>**
- You cannot delete the primary group of a user account

04

File Permissions and ownership

Every file or directory on Unix/Linux system has 3 possible permissions:

Read (r)

File: gives you the authority to open and read a file.

Directory: gives you the ability to list its content.

Write (w)

File: gives you the authority to modify the contents of a file.

Directory: gives you the authority to add, remove, and rename files stored in the directory.

Exectute (x)

File: In Unix/Linux, you cannot run a program unless the execute permission is set.
By default, any newly created files are not executable regardless of their file extension suffix.

Directory: The contents of the directory can be accessed.

Linux File Ownership

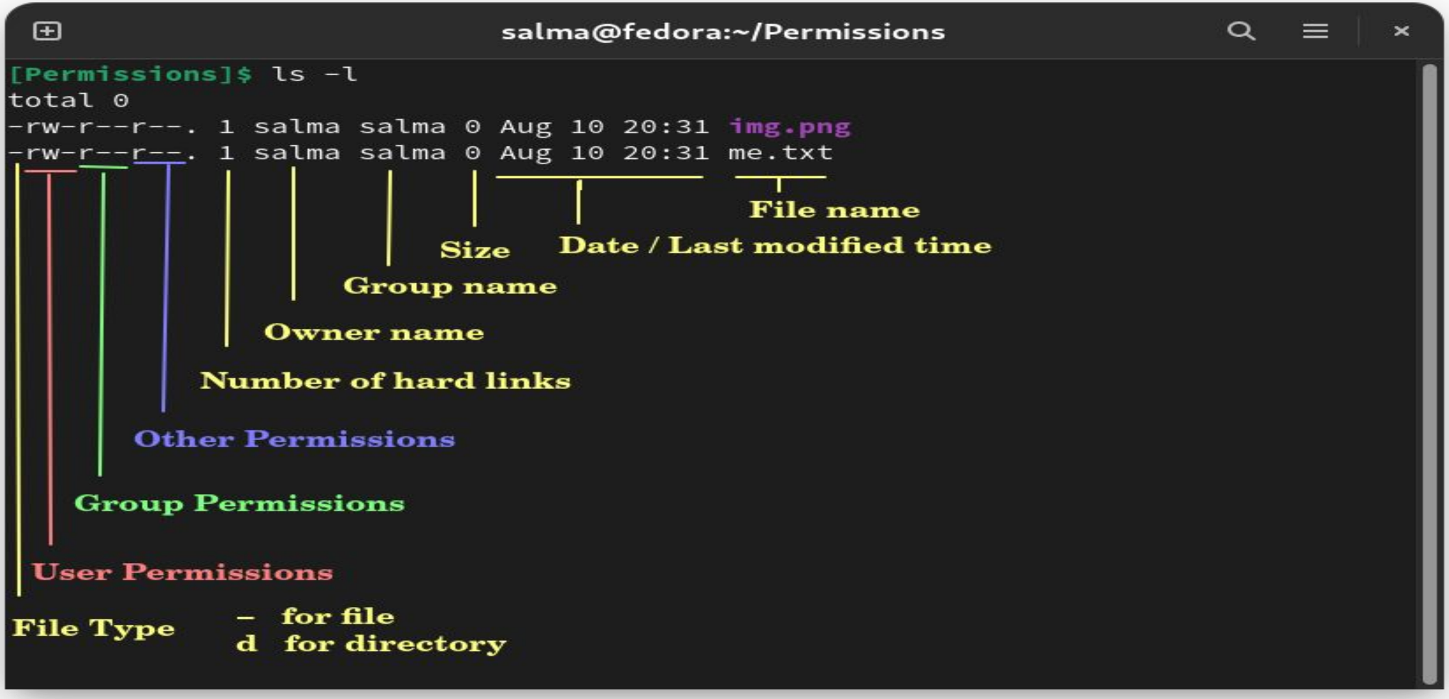
Every file and directory on your Unix/Linux system is assigned **3 types** of owner :

❑ **User (Owner)**
A user is the owner of the file. By default, the person who created a file becomes its owner.

❑ **Group**
A group can contain multiple users. All users belonging to a group will have the same Linux group permissions access to the file.

❑ **Other**
Any other user who has access to a file. This person has neither created the file, nor he belongs to a group who could own the file. Practically, it means everybody else. Hence, when you set the permission for others, it is also referred as set permissions for the world.

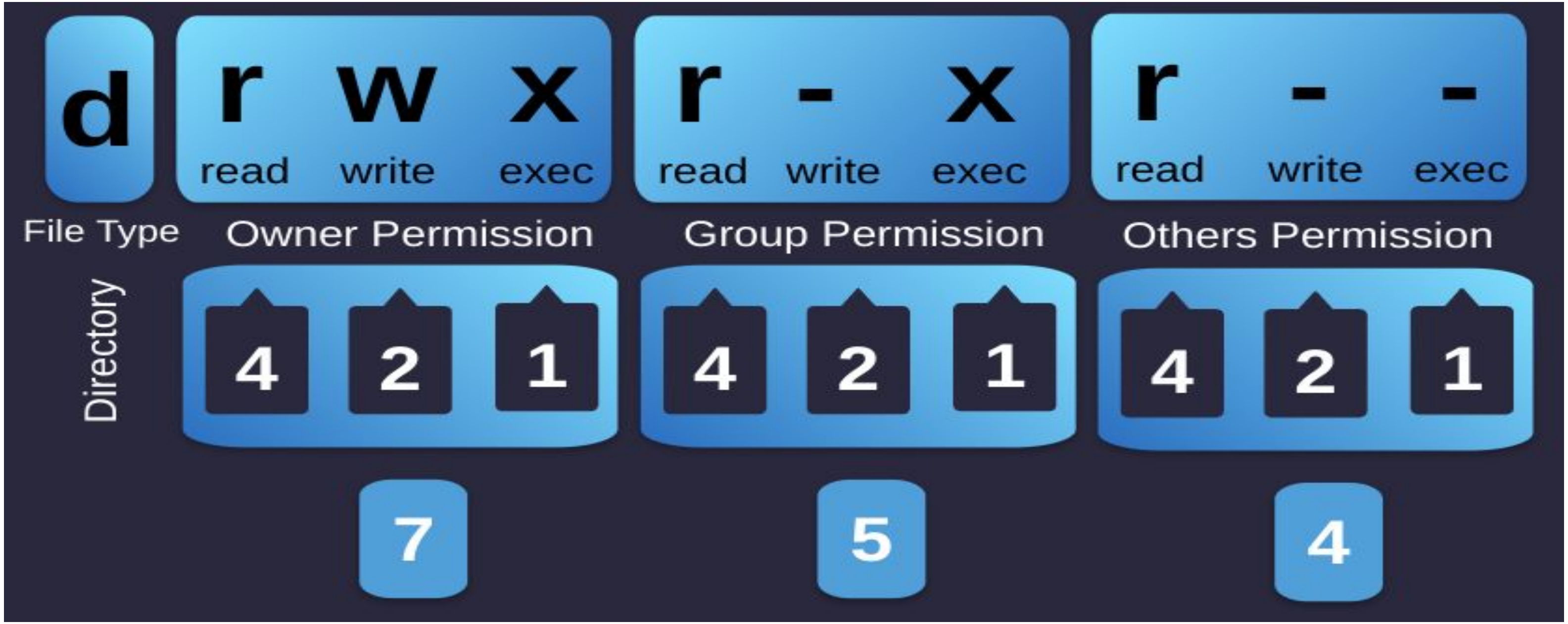
Linux File Ownership



salma@fedora:~/Permissions

```
[Permissions]$ ls -l
total 0
-rw-r--r--. 1 salma salma 0 Aug 10 20:31 img.png
-rw-r--r--. 1 salma salma 0 Aug 10 20:31 me.txt
```

Linux File Ownership



05

Changing permissions

Changing permissions

The **chmod** command is used to change file/ directory's permissions.

There are two ways:

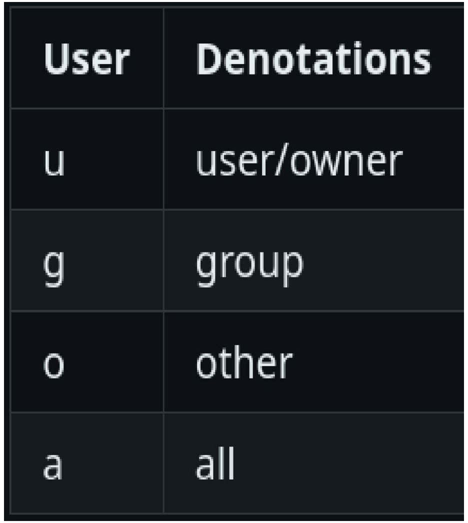
1) Symbolic mode
2) Absolute mode

Symbolic mode

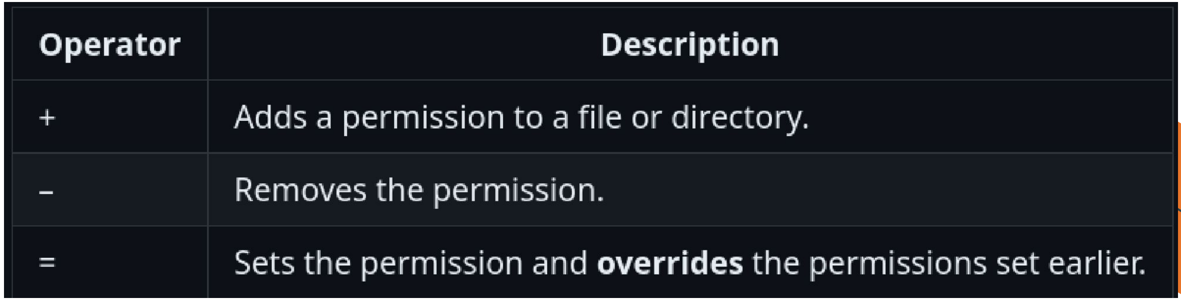
In symbolic mode, you can modify the permissions of a specific owner.

Syntax:

chmod [ownerType] [operator] [new permission] [file name]

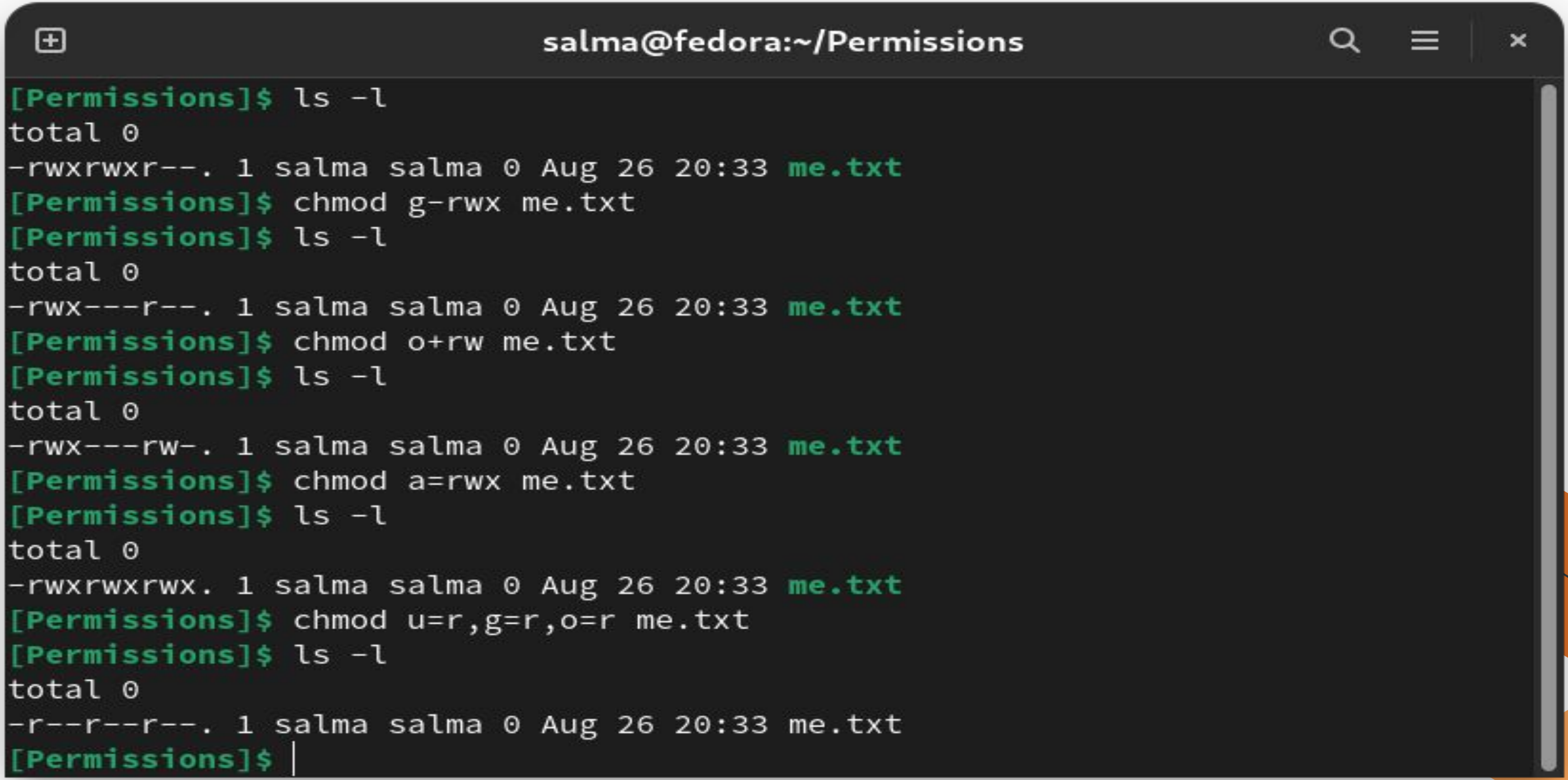


User	Denotations
u	user/owner
g	group
o	other
a	all



Operator	Description
+	Adds a permission to a file or directory.
–	Removes the permission.
=	Sets the permission and **overrides** the permissions set earlier.

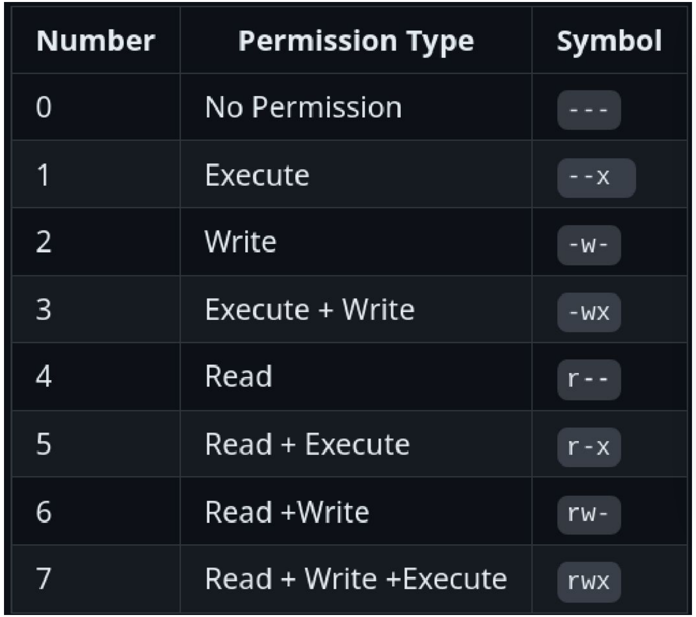
Symbolic mode



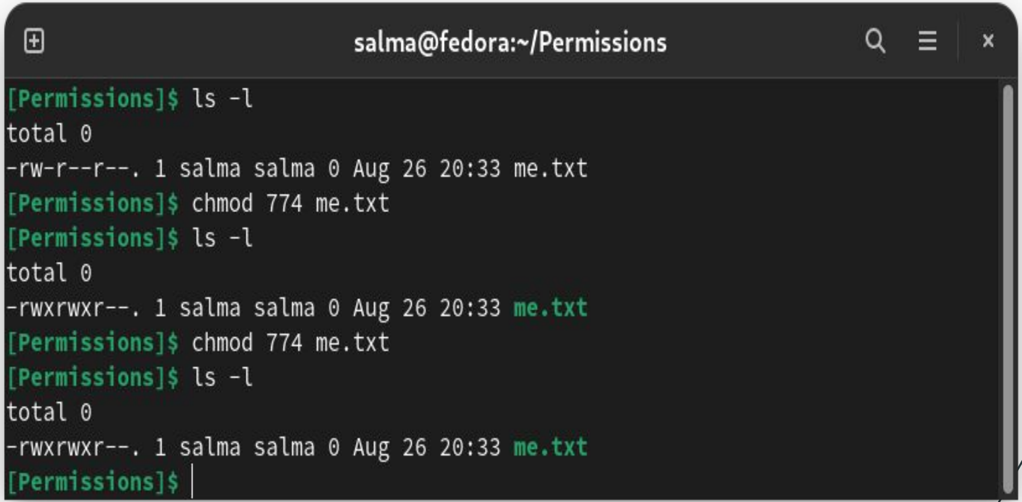
```
[Permissions]$ ls -l
total 0
-rwxrwxr--. 1 salma salma 0 Aug 26 20:33 me.txt
[Permissions]$ chmod g-rwx me.txt
[Permissions]$ ls -l
total 0
-rwx---r--. 1 salma salma 0 Aug 26 20:33 me.txt
[Permissions]$ chmod o+rw me.txt
[Permissions]$ ls -l
total 0
-rwx---rw-. 1 salma salma 0 Aug 26 20:33 me.txt
[Permissions]$ chmod a=rwx me.txt
[Permissions]$ ls -l
total 0
-rwxrwxrwx. 1 salma salma 0 Aug 26 20:33 me.txt
[Permissions]$ chmod u=r,g=r,o=r me.txt
[Permissions]$ ls -l
total 0
-r--r--r--. 1 salma salma 0 Aug 26 20:33 me.txt
[Permissions]$ 
```

Absolute mode

In this mode, you can specify the permissions in the following way:



Number	Permission Type	Symbol
0	No Permission	`---`
1	Execute	`--x`
2	Write	`-w-`
3	Execute + Write	`-wx`
4	Read	`r--`
5	Read + Execute	`r-x`
6	Read +Write	`rw-`
7	Read + Write +Execute	`rwx`



```
[Permissions]$ ls -l
total 0
-rw-r--r--. 1 salma salma 0 Aug 26 20:33 me.txt
[Permissions]$ chmod 774 me.txt
[Permissions]$ ls -l
total 0
-rwxrwxr--. 1 salma salma 0 Aug 26 20:33 me.txt
[Permissions]$ chmod 774 me.txt
[Permissions]$ ls -l
total 0
-rwxrwxr--. 1 salma salma 0 Aug 26 20:33 me.txt
[Permissions]$ 
```

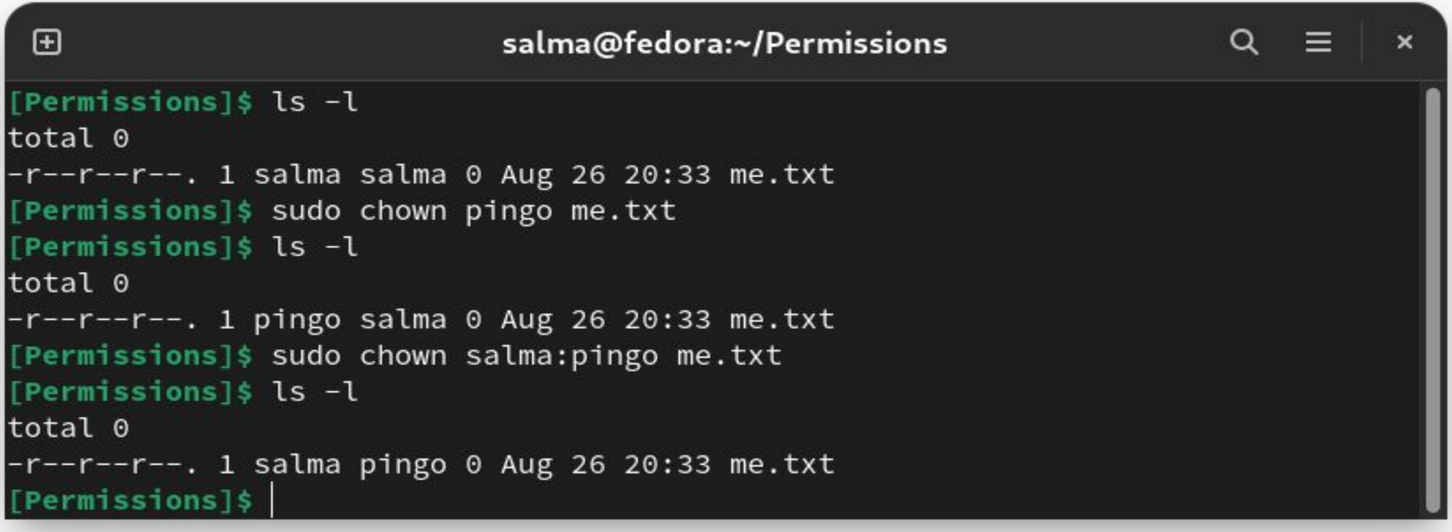
06

Changing ownership and group

chown

- For changing the ownership of a file/directory, you can use : **chown user filename**

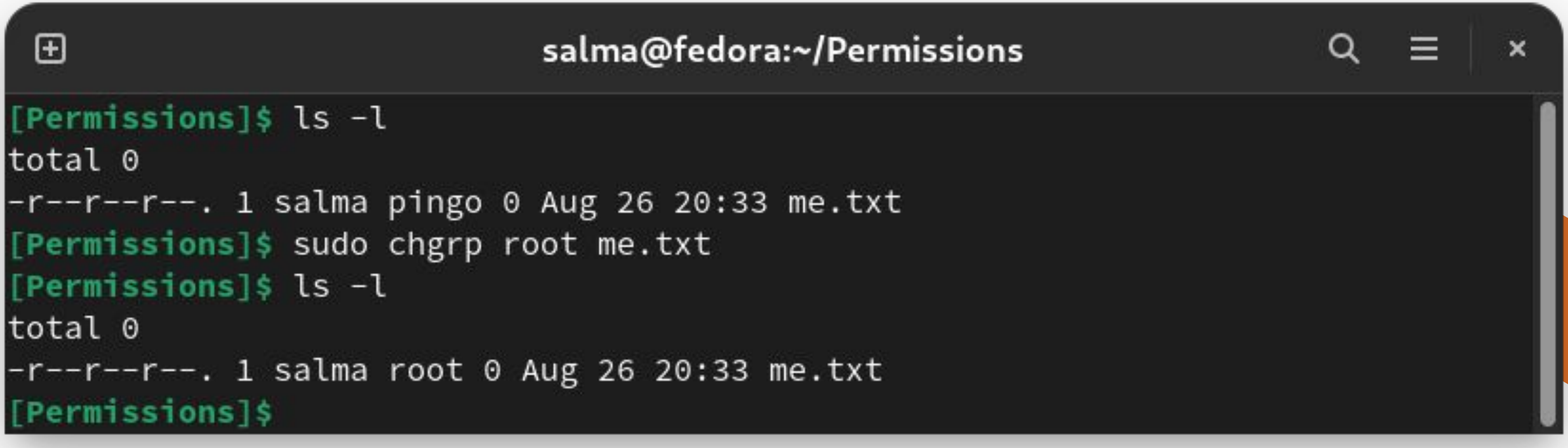
- In case you want to change the user as well as group for a file or directory use the command : **chown user:group filename**



```
[Permissions]$ ls -l
total 0
-r--r--r--. 1 salma salma 0 Aug 26 20:33 me.txt
[Permissions]$ sudo chown pingo me.txt
[Permissions]$ ls -l
total 0
-r--r--r--. 1 pingo salma 0 Aug 26 20:33 me.txt
[Permissions]$ sudo chown salma:pingo me.txt
[Permissions]$ ls -l
total 0
-r--r--r--. 1 salma pingo 0 Aug 26 20:33 me.txt
[Permissions]$
```

chgrp

- In case you want to change group-owner only : **chgrp group_name filename**



```
[Permissions]$ ls -l
total 0
-r--r--r--. 1 salma pingo 0 Aug 26 20:33 me.txt
[Permissions]$ sudo chgrp root me.txt
[Permissions]$ ls -l
total 0
-r--r--r--. 1 salma root 0 Aug 26 20:33 me.txt
[Permissions]$
```

Thanks!