Robotics Corner

Robotics Corner

Linux

Linux is an open Source operating system that is widely used in most of the companies.

01 Users (Types, Add, Delete, Properties)

02 User Groups

03 File and Directory Permissions

04 chown and chmod keywords

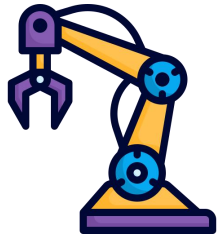
05 sudo

ROBOTICS CORNER

Linux

Users and Permissions

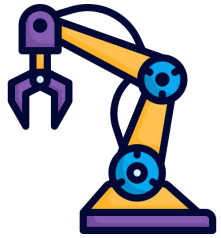
Linux



What are Users in Linux?

- In a Linux system, users refer to individuals or entities that interact with the operating system by logging in and performing various tasks. User management plays a crucial role in ensuring secure access control, resource allocation, and system administration.
- A user in Linux is associated with a user account, which consists of several properties defining their identity and privileges within the system. These properties are a username, UID (User ID), GID (Group ID), home directory, default shell, and password.
- Each user account possesses these unique properties listed above.

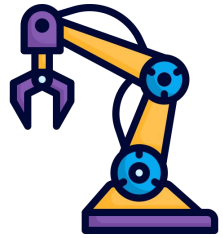
ROBOTICS CORNER



Types of Users in Linux

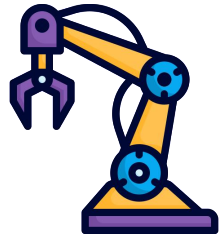
- System users are created by the system during installation and are used to run system services and applications, /var/lib

- Regular users are created by the administrator and can access the system and its resources based on their permissions.



How to Create Users

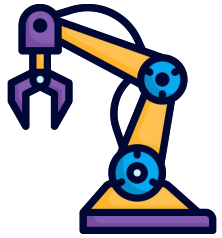
- useradd -u 1002 -d /home/robot -s /bin/bash robot or simply: useradd robot
- This command creates robot's account with uid (-u) as 1002, the home directory (-d) as /home/robot, and sets (-s) /bin/bash as his default shell.
- Verify the new user account by running the id command: id robot



User Account Properties

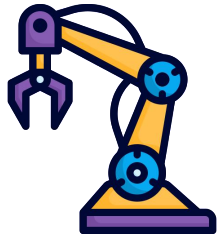
- **Username**: Each user is assigned a unique username that serves as their identifier within the Linux system. For example, the username is “robot".
- **UID (User ID) and GID (Group ID):** Every user account is associated with a UID and GID. The UID is a numerical value assigned to the user, while the GID represents the primary group to which the user belongs. For instance, robot's UID may be 1002, and his primary group's GID is 1002 as well.
- **Home Directory**: Each user has a designated home directory where their personal files and settings reside. robot's home directory is /home/robot.

ROBOTICS CORNER



User Account Properties

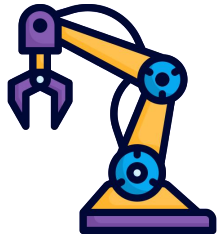
- **Default Shell**: The default shell determines the command interpreter used when a user logs in. It defines the user's interactive environment. In our case, robot's default shell is set to /bin/bash, which is a popular shell in Linux.
- **Password**: User accounts require passwords to authenticate and access the system.
- **Group**: The group membership determines which system resources the user can access, as well as which users can access the user's files. (-g)



cat /etc/passwd

- look at the users on their Linux by running the cat /etc/passwd command. robot:x:1002:1002:,,,:/home/robot:/bin/bash
- robot: username, x: password
- 1002: This is the UID (User ID) of the user account, which is a unique numerical identifier assigned to the user by the system.
- 1002: This is the GID (Group ID) of the user account, which represents the primary group membership of the user.

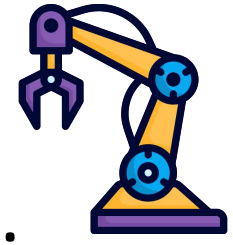
ROBOTICS CORNER



cat /etc/passwd

- This is the GECOS field, which stands for "General Electric Comprehensive Operating System". This field is used to store additional information about the user, such as their full name or contact information. In this case, the field is empty, as no additional information was provided while creating the user account.
- /home/robot: This is the home directory of the user account, which is the location where the user's files and personal data are stored.
- /bin/bash: This is the default shell for the user account, which is the command interpreter used to process commands entered by the user in the terminal. In this case, the default shell is Bash, which is the most commonly used shell in Linux.

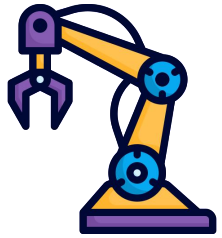
ROBOTICS CORNER



How to Delete Users & Modify User Groups in Linux

```
sudo userdel robot
sudo usermod -aG development john
```

ROBOTICS CORNER



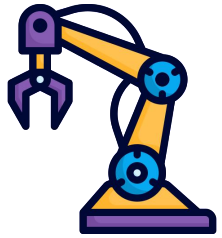
How to Change Default Shell in Linux

- In a case where John prefers to use a different shell other than the default /bin/bash shell. The IT team can modify his account accordingly. For example, to change John's default shell to /bin/zsh, the following command can be used:

```
sudo usermod -s /bin/zsh john
```

- This command updates John's account to use the new default shell — /bin/zsh.
- You can run the cat /etc/passwd again to see that the shell for john has changed from /bin/bash to /bin/zsh.

ROBOTICS CORNER



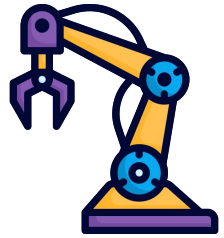
How to Create a New Group in Linux

- To create a new group, such as the marketing group, the following command can be used:

  ```
  sudo groupadd marketing
  ```

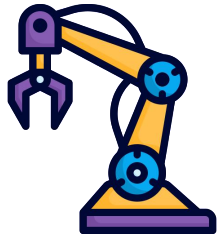
- The command above creates the marketing group, which can be used to grant specific permissions and access to marketing-related resources.
- To view the group you just added, run the command: `cat /etc/group`

ROBOTICS CORNER



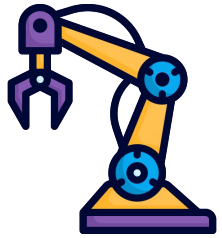
Change group

- To change the group owner of a file: chgrp [USERGROUP] [FILE
- To delete a group: groupdel [GROUP]



File and directory permissions

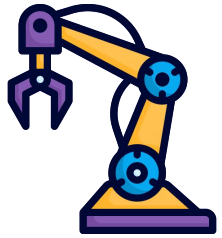
- When working with files and directories in Linux, it’s important to understand how to set permissions. Permissions define who can access and modify files and directories on a system.
- In Linux, each file and directory has three types of permissions: read, write, and execute. These permissions can be set for three different categories of users − owner of file or directory, group to which file or directory belongs, and all other users.



Understanding Linux File Permissions

- The read permission allows users to view contents of a file or directory. write permission allows users to modify contents of a file or directory. execute permission allows users to run a file or access a directory.

- Each file and directory also has an owner and a group. owner is user who created file or directory, and group is a collection of users who share a common set of permissions.



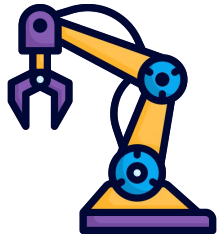
Using chown Command

- The **chown** command is used to change the owner of a file or directory. To change a file or directory owner, you must have root privileges or be the current owner of the file or directory.

  ```
  chown [OPTIONS] [NEW_OWNER] [FILE_OR_DIRECTORY]
  ```

- You can also use the chown command to change the owner of a directory and all of its contents: `chown -R john example`
- The "-R" option tells Chown to change the owner of the directory and all of its contents recursively.

ROBOTICS CORNER

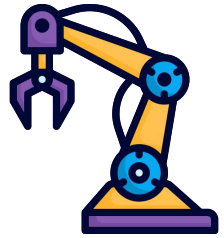


Using chmod Command

- The chmod command is used to change permissions of a file or directory. To change permissions of a file or directory, you must have appropriate permissions to do so.
- chmod [OPTIONS] [PERMISSIONS] [FILE_OR_DIRECTORY]
- Chmod 735 file.txt

Value	Meaning
0	No permission
1	Execute permission
2	Write permission
3	Write and execute permission
4	Read permission
5	Read and execute permission
6	Read and write permission
7	Read, write, and execute permission

ROBOTICS CORNER

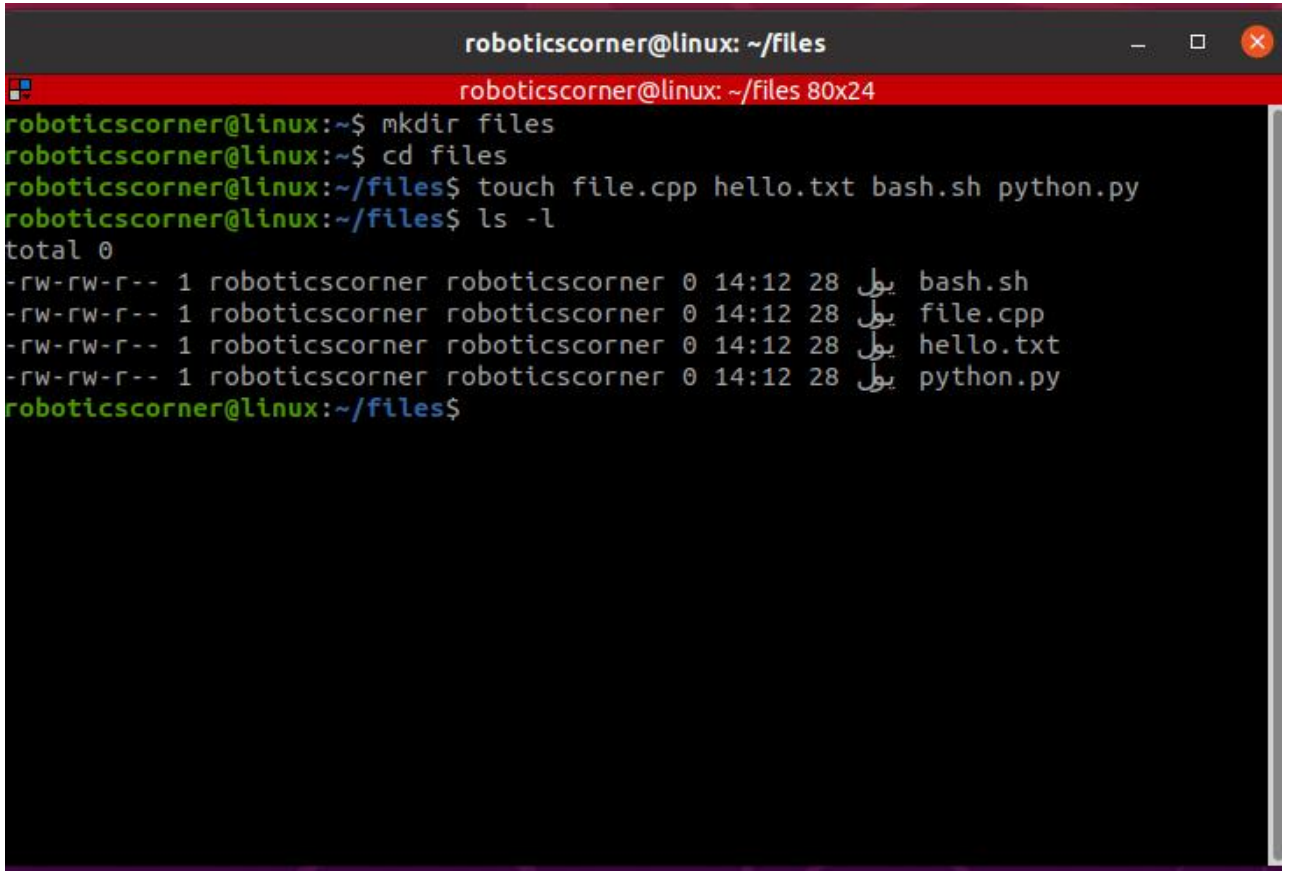


CHMOD example

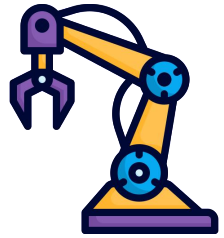
- chmod abc example.txt
- a→ owner
- b→ group
- c→ other users
- In this example, owner of "example.txt" file will have read, write, and execute permissions, while group and all other users will have read and execute permissions.

Understanding permission modes (rwx)

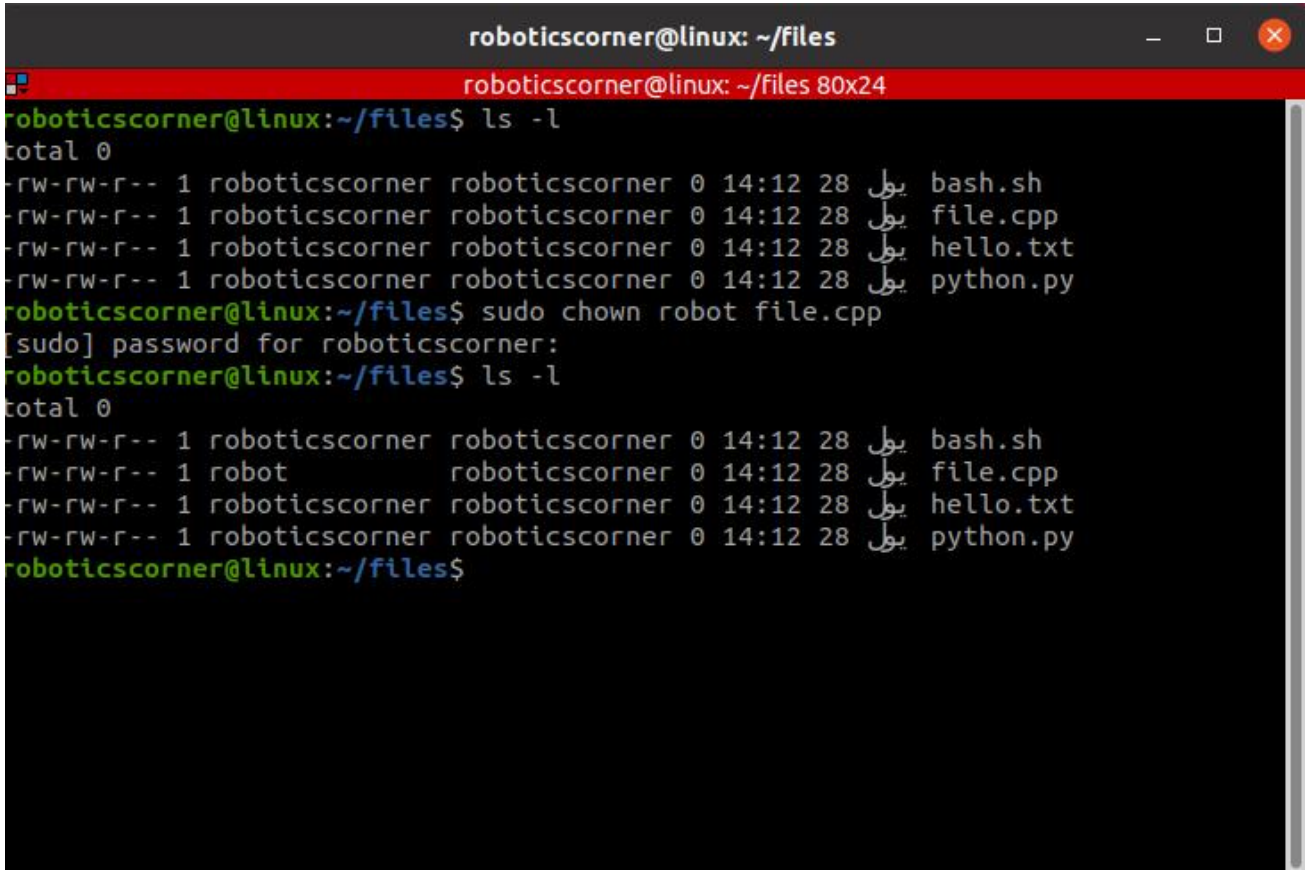
- File type: - , d→ directory
- Permission settings: rw-r--r--
- Extended attributes: dot (.)
- User owner: root
- Group owner: root

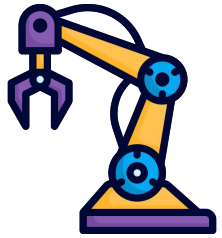


Understanding permission modes (rwx)



- File type: - , d→ directory
- Permission settings: rw-r--r--
- Extended attributes: dot (.)
- User owner: root
- Group owner: root

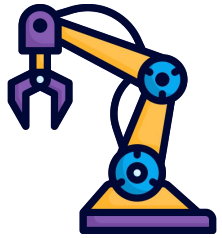




How do you read file permissions?

- rw-r--r–
- The first set of permissions applies to the owner of the file. The second set of permissions applies to the user group that owns the file. The third set of permissions is generally referred to as "others." All Linux files belong to an owner and a group.

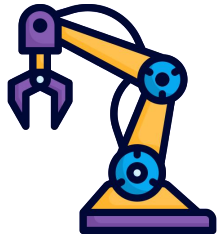
- When permissions and users are represented by letters, that is called symbolic mode. For users, u stands for user owner, g for group owner, and o for others. For permissions, r stands for read, w for write, and x for executable.



Task

- Create 5 text files in a directory called Countries, each file with a country name: Egypt, Lebanon, Oman, Germany, France
- Create another directory inside Countries named Cities and create 3 files, random names as u wish
- Change the permissions of the Arab countries to be read and write for the owner, read and execute for the group, and no permissions for the other users
- Change 2 new users called arab and europe with different ids

ROBOTICS CORNER

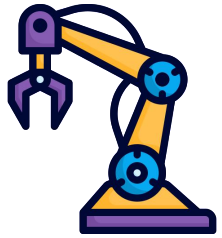


task

- Change the user owner of the arab countries to be arab and for the European countries to Europe
- Create a new group called schengen
- Change the group owner of the European countries to be schengen

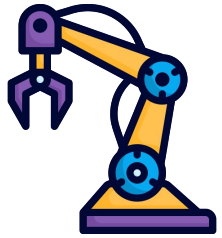
ROBOTICS CORNER

Sudo command



- sudo is a Linux command that is used to temporarily execute programs as another user. It is the abbreviation for substitute user and do and borrows the privileges of another user, for example, the root user. This way, sudo helps you accomplish system administration tasks without logging in as root, super user
- As a regular user on Linux, you have reduced permissions that are sufficient for most of the tasks. The root user is the Linux superuser and the equivalent to the administrator.

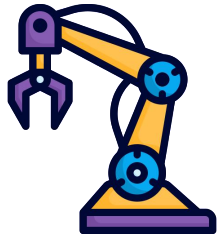
ROBOTICS CORNER



- When running a command prefaced with sudo, the system asks you for the password of the root account. After successful authentication, the command is executed with superuser privileges.

- Depending on the sudo configuration, the elevated privileges persist for a certain period of time and as long as you are working in the same terminal session. So you do not need to provide the root password again when running another sudo command.
- sudo [command]

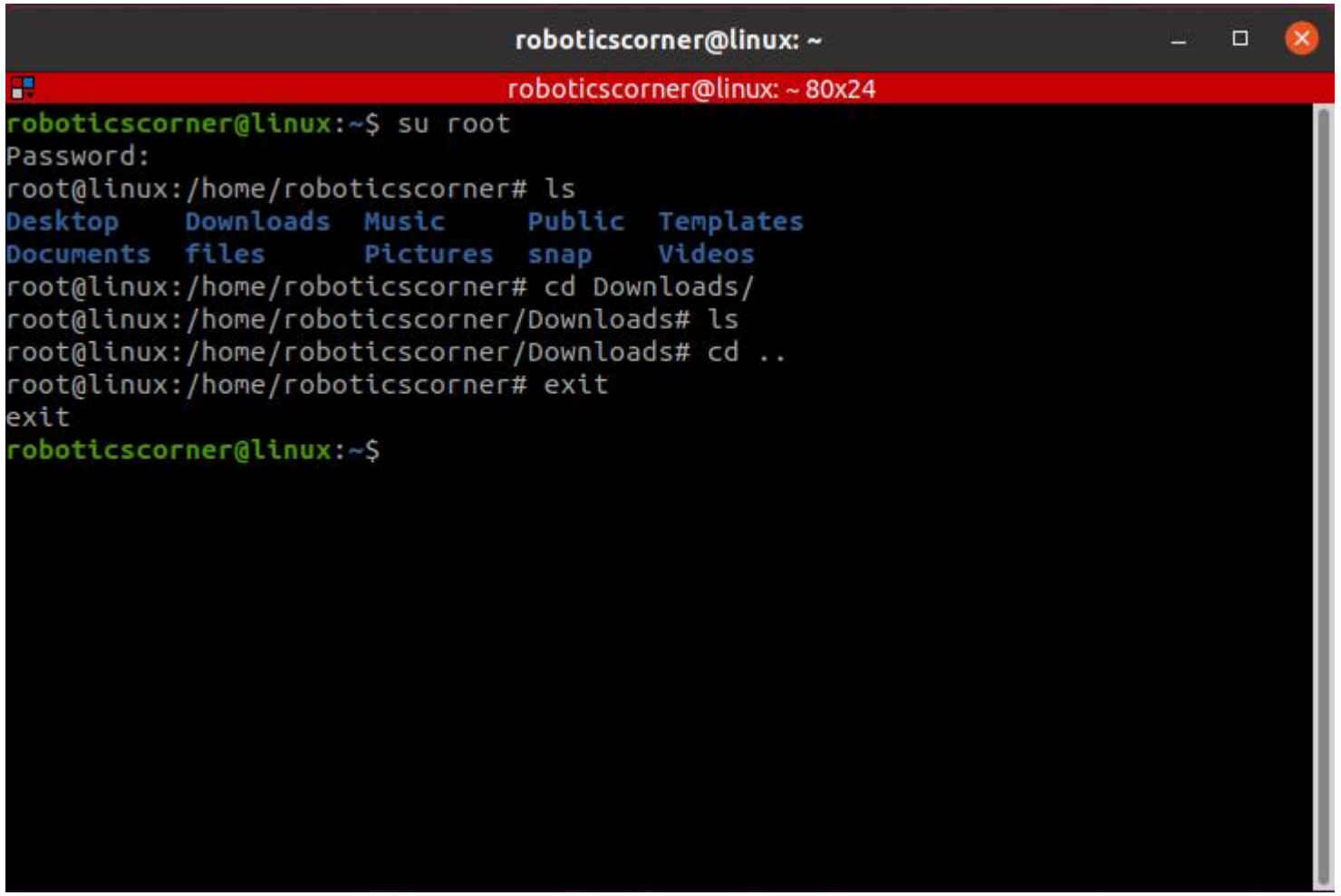
Sudo command



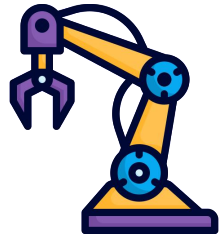
- This is the equivalent of the "run as administrator" option in Windows. The option of sudo lets us have multiple administrators.
- sudo su: Switch to the superuser (root) account.
- sudo mkdir /path/to/new_directory: for directories requiring superuser permissions.
- sudo touch /path/to/new_file.txt: for file creation requiring superuser permissions.

ROBOTICS CORNER

root


roboticscorner@linux: ~
roboticscorner@linux: ~ 80x24
roboticscorner@linux:~$ su root
Password:
root@linux:/home/roboticscorner# ls
Desktop Downloads Music Public Templates
Documents files Pictures snap Videos
root@linux:/home/roboticscorner# cd Downloads/
root@linux:/home/roboticscorner/Downloads# ls
root@linux:/home/roboticscorner/Downloads# cd ..
root@linux:/home/roboticscorner# exit
exit
roboticscorner@linux:~$

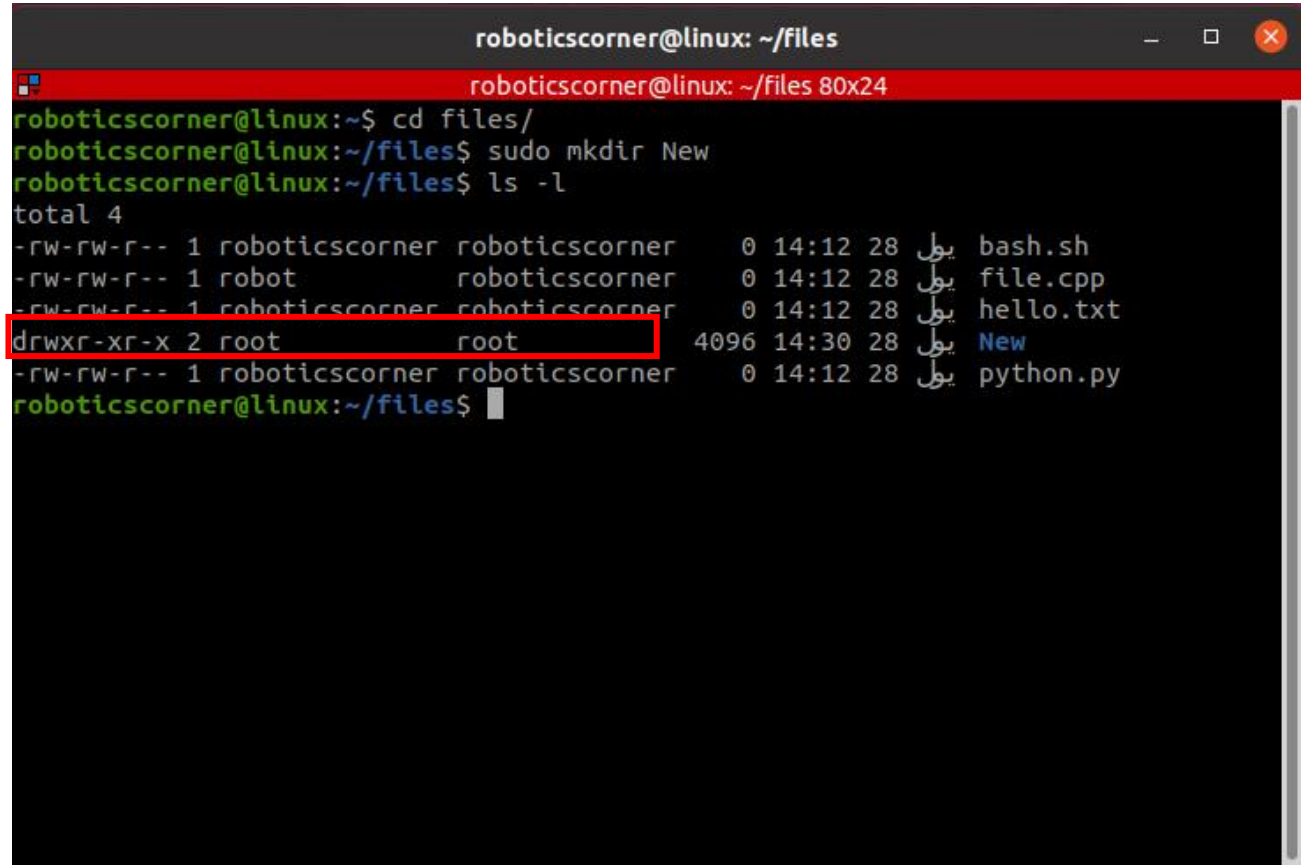
05 Sudo command



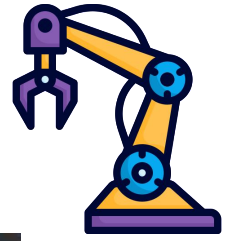
The mode is not the usual one when you create a directory.

Also the owner and group in root

You can not create a file or a directory in this directory without a permission



05 Sudo command



The mode is not the usual one when you create a directory.

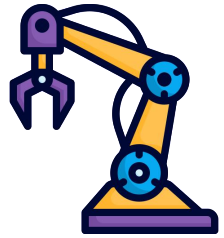
Also the owner and group in root

You can not create a file or a directory in this directory without a permission

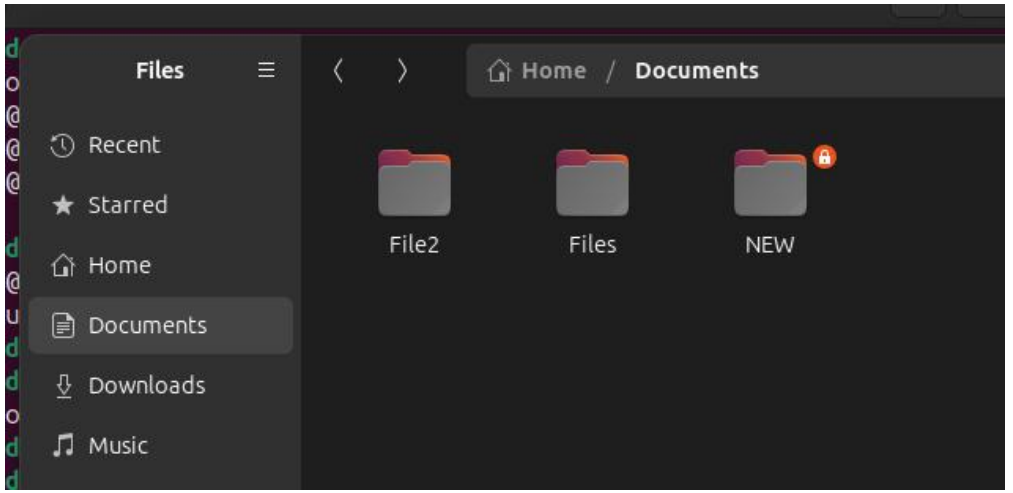
```
roboticscorner@linux: ~/files/New
roboticscorner@linux: ~/files/New 80x24
roboticscorner@linux:~$ cd files/
roboticscorner@linux:~/files$ sudo mkdir New
roboticscorner@linux:~/files$ ls -l
total 4
-rw-rw-r-- 1 roboticscorner roboticscorner    0 14:12 28 يول bash.sh
-rw-rw-r-- 1 robot          roboticscorner    0 14:12 28 يول file.cpp
-rw-rw-r-- 1 roboticscorner roboticscorner    0 14:12 28 يول hello.txt
drwxr-xr-x 2 root           root           4096 14:30 28 يول New
-rw-rw-r-- 1 roboticscorner roboticscorner    0 14:12 28 يول python.py
roboticscorner@linux:~/files$ cd New
roboticscorner@linux:~/files/New$ touch file new.txt
touch: cannot touch 'file': Permission denied
touch: cannot touch 'new.txt': Permission denied
roboticscorner@linux:~/files/New$
```

ROBOTICS CORNER

05 Sudo command

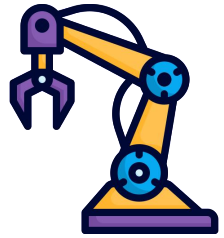


```
roboticscorner@linux: ~/files/New
roboticscorner@linux: ~/files/New 80x24
roboticscorner@linux:~$ cd files/
roboticscorner@linux:~/files$ sudo mkdir New
roboticscorner@linux:~/files$ ls -l
total 4
-rw-rw-r-- 1 roboticscorner roboticscorner    0 14:12 28 يول bash.sh
-rw-rw-r-- 1 robot          roboticscorner    0 14:12 28 يول file.cpp
-rw-rw-r-- 1 roboticscorner roboticscorner    0 14:12 28 يول hello.txt
drwxr-xr-x 2 root           root           4096 14:30 28 يول New
-rw-rw-r-- 1 roboticscorner roboticscorner    0 14:12 28 يول python.py
roboticscorner@linux:~/files$ cd New
roboticscorner@linux:~/files/New$ touch file new.txt
touch: cannot touch 'file': Permission denied
touch: cannot touch 'new.txt': Permission denied
roboticscorner@linux:~/files/New$
```



ROBOTICS CORNER

05 Sudo command



Creating a file with the sudo command

roboticscorner@linux: ~/files

roboticscorner@linux: ~/files 80x24

```
roboticscorner@linux:~$ cd files/
roboticscorner@linux:~/files$ sudo mkdir New
roboticscorner@linux:~/files$ ls -l
total 4
-rw-rw-r-- 1 roboticscorner roboticscorner    0 14:12 28 يول bash.sh
-rw-rw-r-- 1 robot          roboticscorner    0 14:12 28 يول file.cpp
-rw-rw-r-- 1 roboticscorner roboticscorner    0 14:12 28 يول hello.txt
drwxr-xr-x 2 root           root           4096 14:30 28 يول New
-rw-rw-r-- 1 roboticscorner roboticscorner    0 14:12 28 يول python.py
roboticscorner@linux:~/files$ cd New
roboticscorner@linux:~/files/New$ touch file new.txt
touch: cannot touch 'file': Permission denied
touch: cannot touch 'new.txt': Permission denied
roboticscorner@linux:~/files/New$ cd ..
roboticscorner@linux:~/files$ sudo touch hello.cpp
roboticscorner@linux:~/files$ nano hello.cpp
```

roboticscorner@linux: ~/files

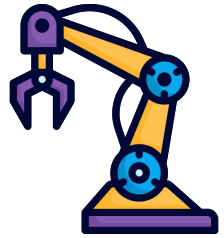
roboticscorner@linux: ~/files 80x24

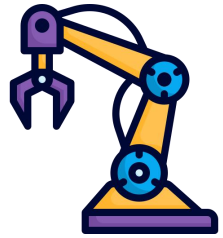
```
 GNU nano 4.8                    hello.cpp

[ File 'hello.cpp' is unwritable ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text^T To Spell  ^_ Go To Line
```

ROBOTICS CORNER

05 Sudo command





Thank You

Do you have any questions?

ROBOTICS CORNER

ROBOTICS CORNER

01211626904

www.roboticscorner.tech

Robotics Corner

Robotics Corner

Robotics Corner

Robotics Corner