USERS AND PERMISSIONS

SOFTWARE ENGINEERING

ROBOTICS CORNER

CONTENTS

- Users (Types, Add, Delete, Properties)
- User Groups
- File and Directory Permissions
- chown and chmod keywords
- sudo

ROBOTICS CORNER

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.

ROBOTICS CORNER

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

ROBOTICS CORNER

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)

ROBOTICS CORNER

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

- sudo userdel robot

ROBOTICS CORNER

HOW TO MODIFY USER GROUPS IN LINUX

- Consider a company CTechCo. As CTechCo's workforce evolves, the IT team may need to make adjustments to user accounts. For example, John (the developer), is assigned additional responsibilities within the company. To reflect this change, the IT team can modify John's account using the usermod command.
- CTechCo creates a new group called development to manage access to development-related resources. To add John to the development group, the following command can be used: 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

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.

ROBOTICS CORNER

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.

ROBOTICS CORNER

CHANGE GROUP

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

ROBOTICS CORNER

USING CHOWN COMMAND

- The chown command is used to change owner of a file or directory. To change owner of a file or directory, you must have root privileges or be current owner of file or directory.
- chown [OPTIONS] [NEW_OWNER] [FILE_OR_DIRECTORY]
- You can also use chown command to change owner of a directory and all of its contents: chown -R john example
- The "-R" option tells chown to change owner of 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]

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

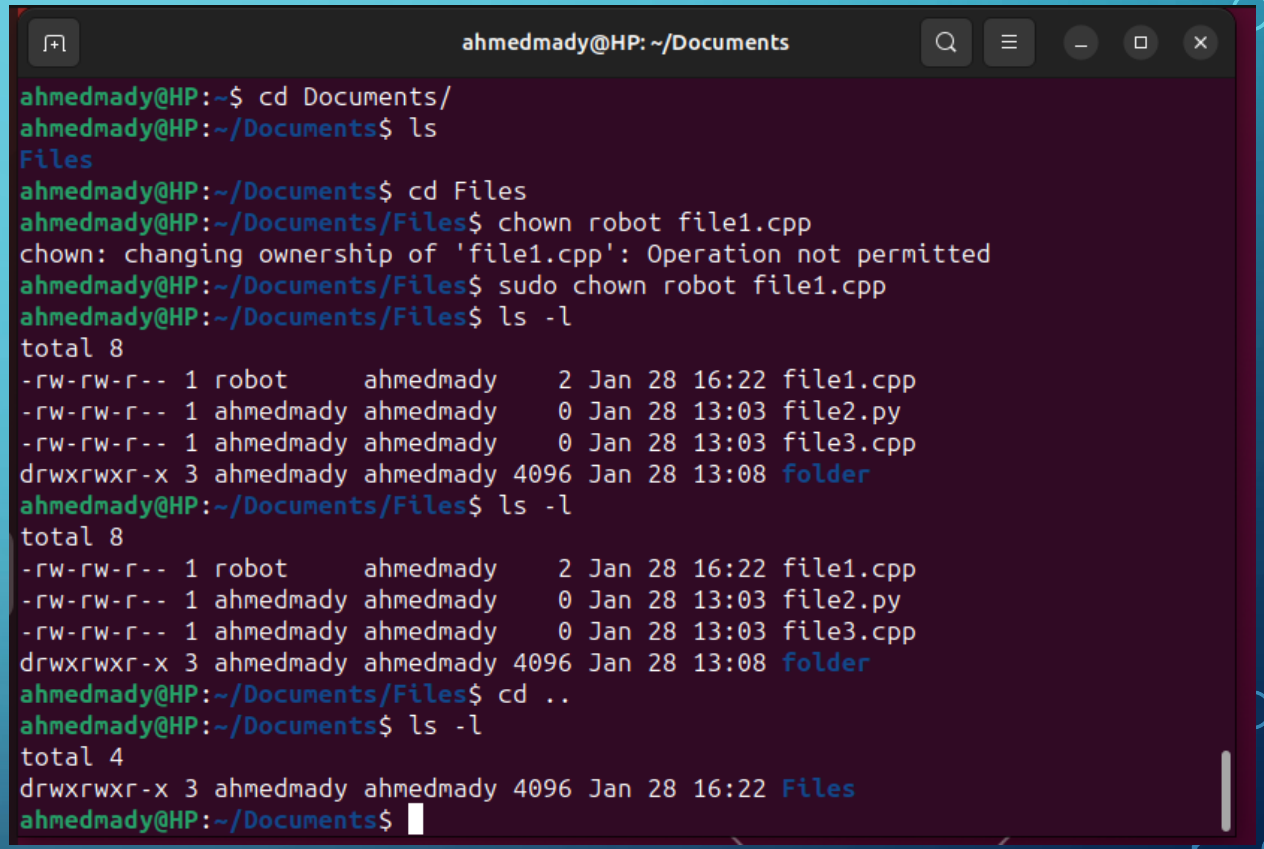
CHMOD EXAMPLE

- 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.

ROBOTICS CORNER

UNDERSTANDING PERMISSION MODES (RWX)

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



ROBOTICS CORNER

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.

ROBOTICS CORNER

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

- 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

SUDO

- 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.

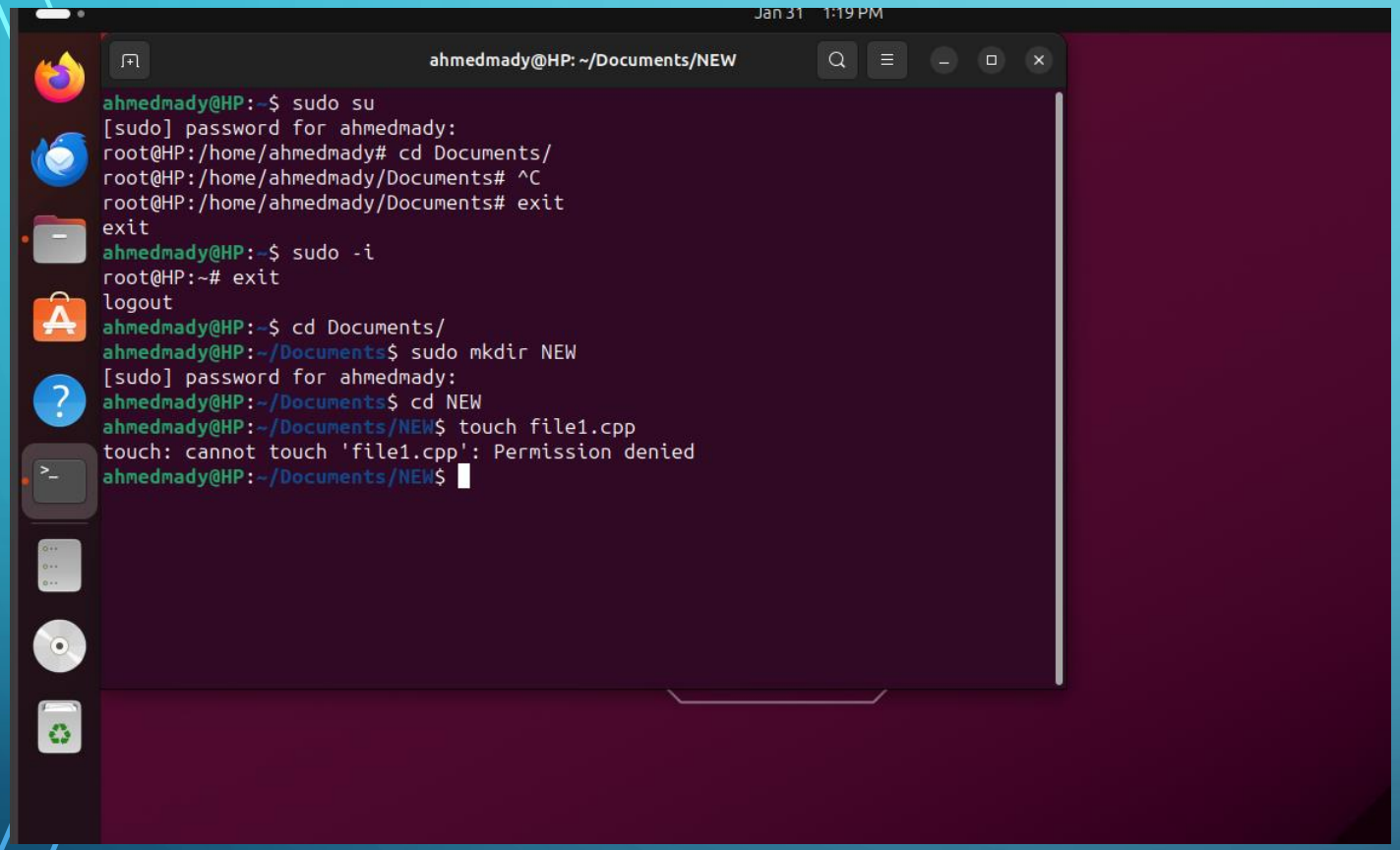
- Depending on the sudo configuration, the elevated privileges persist for a certain period of time and as long as you are working in the same terminal session. So you do not need to provide the root password again when running another sudo command.
- sudo [command]

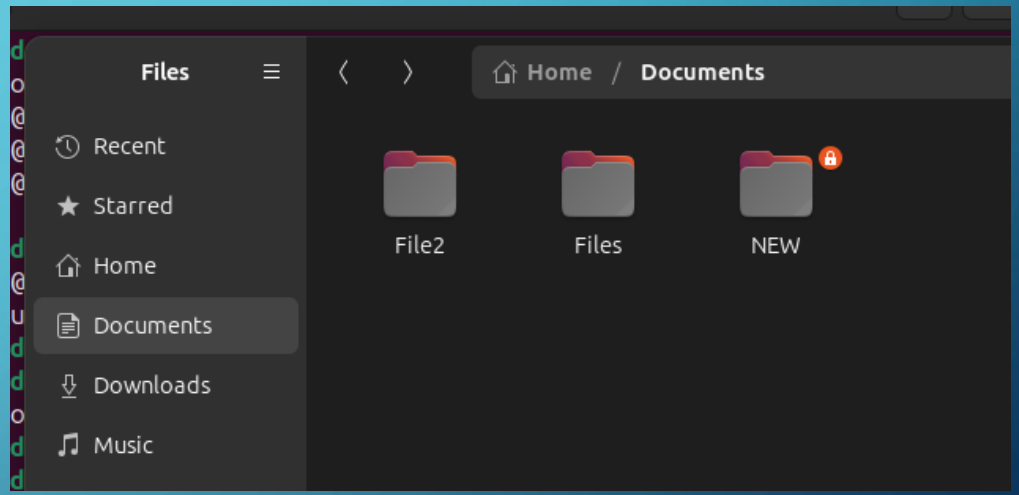
ROBOTICS CORNER

SUDO

- 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.

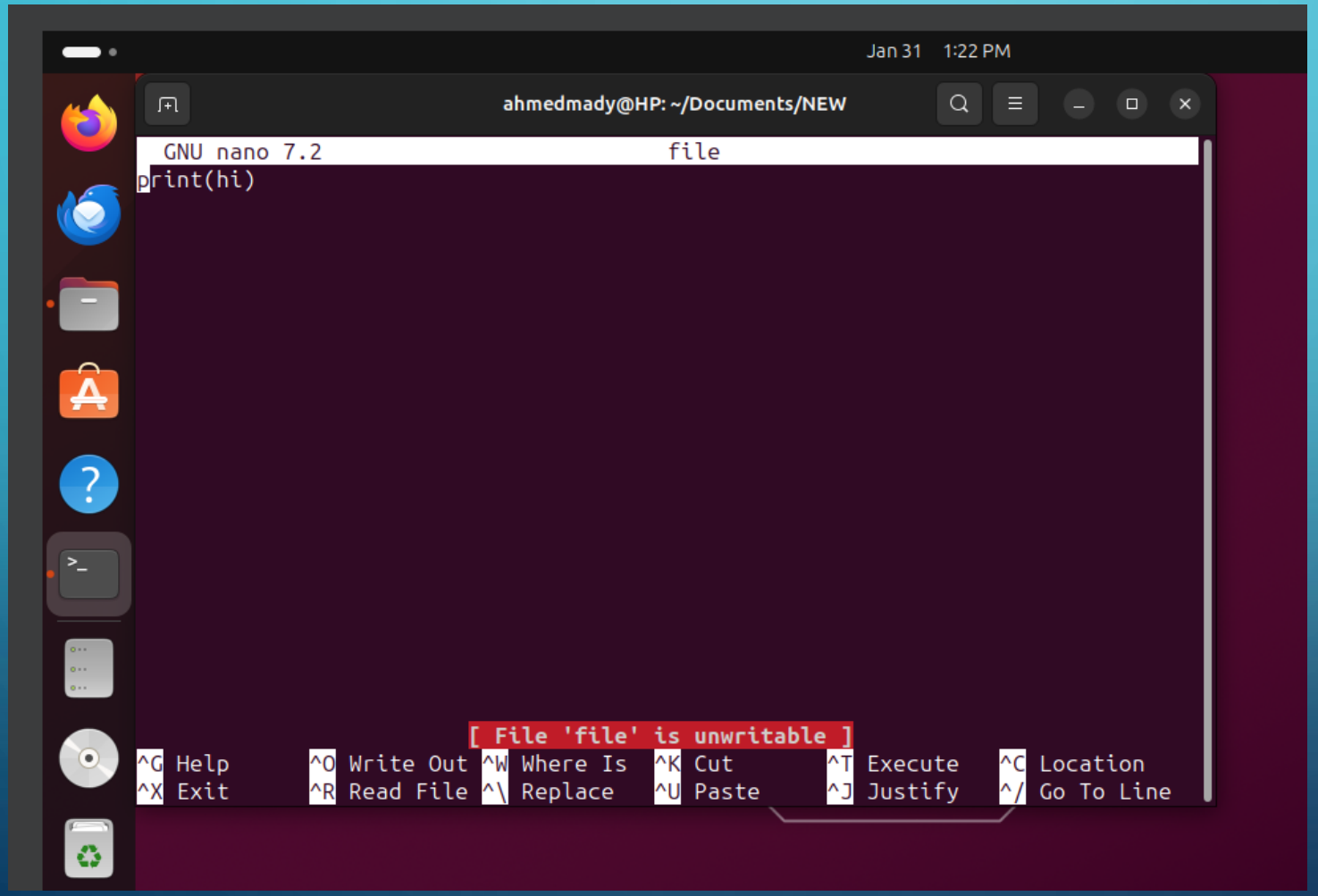
ROBOTICS CORNER





ROBOTICS CORNER

SUDO



ROBOTICS CORNER

REFERENCES

- https://www.freecodecamp.org/news/how-to-manage-users-in-linux/
- https://www.tutorialspoint.com/setting-permissions-with-chown-and-chmod#:~:text=To%20effectively%20manage%20file%20and,and%20chmod%20commands%20in%20combination.&text=700%20example.txt-,In%20this%20example%2C%20%22john%22%20user%20will%20become%20new%20owner,users%20will%20have%20no%20permissions.
- https://www.redhat.com/sysadmin/linux-file-permissions-explained

ROBOTICS CORNER