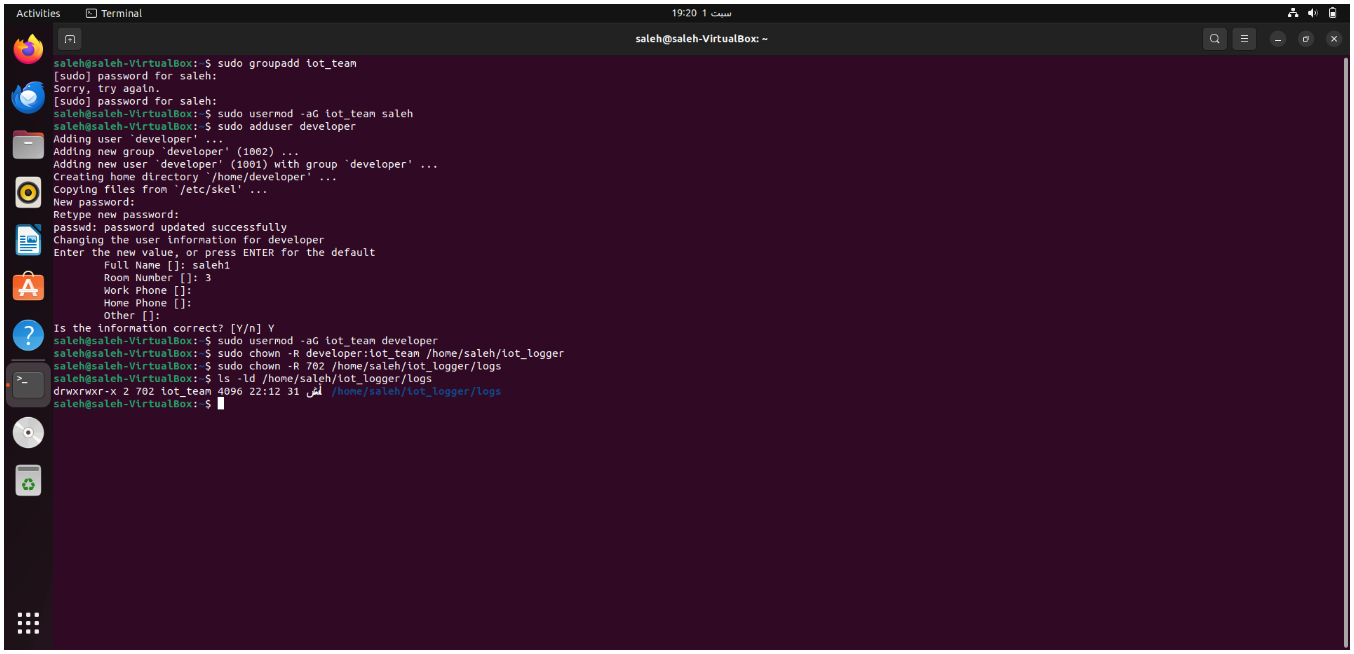
Day 3 – Phase 3 User, Group, and Permissions Management

Tasks:



```
saleh@saleh-VirtualBox:~$ sudo groupadd iot_team
[sudo] password for saleh:
Sorry, try again.
[sudo] password for saleh:
saleh@saleh-VirtualBox:~$ sudo usermod -aG iot_team saleh
saleh@saleh-VirtualBox:~$ sudo adduser developer
Adding user `developer' ...
Adding new group `developer' (1002) ...
Adding new user `developer' (1001) with group `developer' ...
Creating home directory `/home/developer' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for developer
Enter the new value, or press ENTER for the default
        Full Name []: saleh1
        Room Number []: 3
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
saleh@saleh-VirtualBox:~$ sudo usermod -aG iot_team developer
saleh@saleh-VirtualBox:~$ sudo chown -R developer:iot_team /home/saleh/iot_logger
saleh@saleh-VirtualBox:~$ sudo chown -R 702 /home/saleh/iot_logger/logs
saleh@saleh-VirtualBox:~$ ls -ld /home/saleh/iot_logger/logs
drwxrwxr-x 2 702 iot_team 4096 22:12 31 أغس /home/saleh/iot_logger/logs
saleh@saleh-VirtualBox:~$ 
```

Open-Ended Questions:

How do Linux file permissions (r, w, x) work for files vs directories? Give an example using ls -l?

In Linux, files have three basic permissions: read (r), write (w), and execute (x). For regular files, read allows viewing the file contents, write permits editing or deleting the file, and execute lets the user run the file as a program or script.

For directories, the permissions have different connotations: read (r) permits listing the contents of the directory, write (w) allows the addition or removal of files within the directory, and execute (x) permits entering or navigating into the directory.

For example, a directory with r-- permits viewing a listing of the files, but not opening them. On the other hand, with --x you can't list the files in the directory, but you can enter it if you know the name of a file inside. Permissions are viewed using the ls -l command, which shows the file type and permissions in the first 10 characters.

Explain the types of processes in Linux: daemon, zombie, orphan. How can you detect them?

Permissions may be expressed in a numerical form based on octal notation. In this form, reading (r) is coded as 4, writing (w) as 2, and executing (x) as 1. The three numbers are then summed and used to indicate the permission of each category. For example, rwx is 7 as a result of 4+2+1, rw- is 6 as a result of 4+2, and r- is 4. An example is `chmod 750 file.txt`, which assumes full rights (7) for the owner, read and execute (5) for the group, and no rights (0) for others. The umask command sets default permissions to be removed for new files and directories, which is based on system defaults. With a directory default mode of 777 and an umask of 027, the new directories will have a mode of 750, since 777 − 027 = 750. In this manner, a umask prevents overly permissive default access to potentially sensitive files.

Why do we need Inter-Process Communication (IPC)? List some IPC mechanisms and real-life examples?

In Linux, the root user is the superuser with unrestricted access to the entire operating system. In Linux, the root user is a special user with unrestricted access to all files in the system. Such a user can read, edit, and delete any file, and change any system setting. Such a user can also install or remove any software. Unlike the root user, who has an operating system dedicated to him, an ordinary user gets to use the system with very limited rights, and such rights are usually limited to the user's home directory and the user's processes. Such a separation ensures that the system is safe from any accidental or intentional harmful changes. The root user is dangerous because of the potential to destroy the system with a single misstep. For example, the command rm -rf / would do such damage. Therefore, a normal user should be used for everyday activities, switching to root (using sudo) only if essential for a task.