Softwarica
College of IT & E-commerce

2024

Submitted to:
Rikesh Maharjan

~ Abhishek Adhikari

About

This project is completed as part of Task 2 for Linux Fundamentals, covering a sequence of commands to create users, manage groups, and set file permissions on a Linux system.

Linux User and Group Management Documentation

1. Creating a New User and Adding to the `sudo` Group

1.1 Objective

To create a new user with your name (e.g., abhishek) and add this user to the sudo group, enabling administrative privileges.

1.2 Commands and Execution

Create the User:

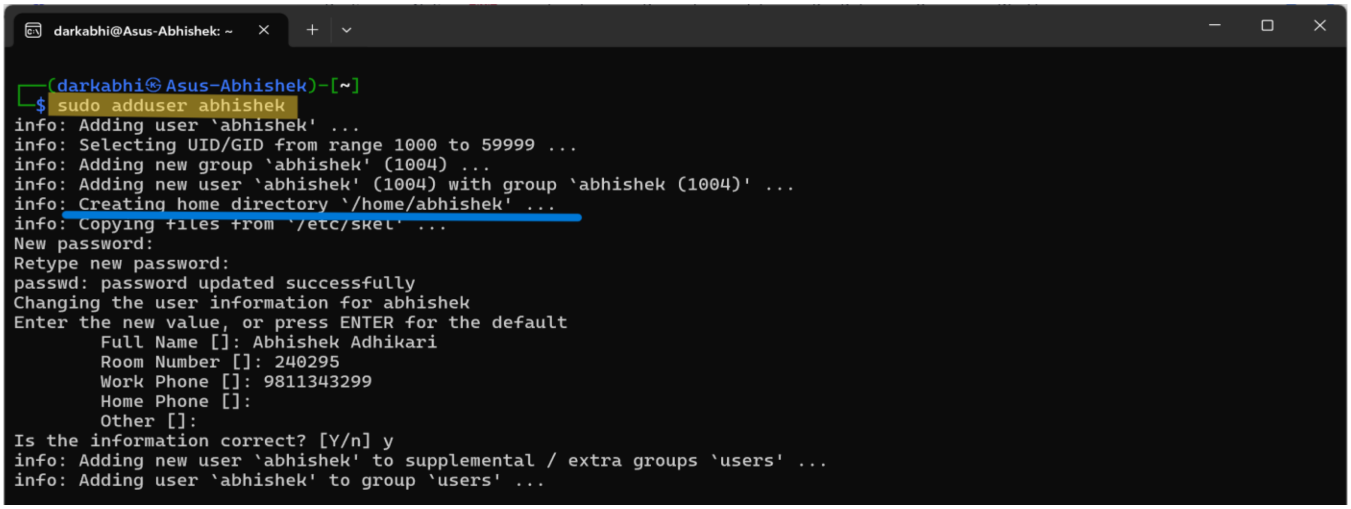
- Use the ` USERADD ` command to create a new user named “abhishek”.
- Command:

  sudo ` USERADD ` -m -s /bin/bash abhishek

  Explanation:

  ➢ `*- m* `: Creates a home directory for the user.

  ➢ `*-s /bin/bash*`: Sets the default shell to bash.
- Visual: The terminal screen shows the command.



```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ sudo adduser abhishek
info: Adding user `abhishek' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `abhishek' (1004) ...
info: Adding new user `abhishek' (1004) with group `abhishek (1004)' ...
info: Creating home directory `/home/abhishek' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for abhishek
Enter the new value, or press ENTER for the default
        Full Name []: Abhishek Adhikari
        Room Number []: 240295
        Work Phone []: 9811343299
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
info: Adding new user `abhishek' to supplemental / extra groups `users' ...
info: Adding user `abhishek' to group `users' ...
```

Add the User to the `sudo` Group:

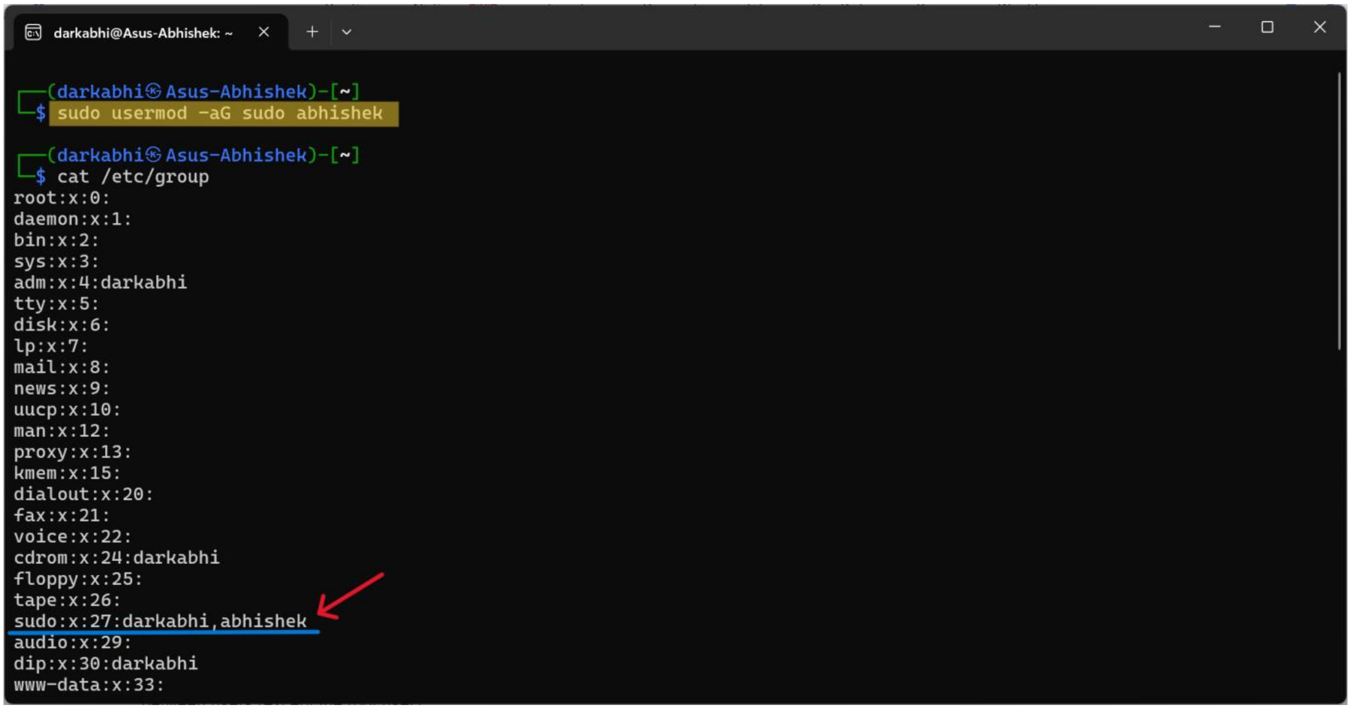
- Add the user “abhishek” to the sudo group for administrative privileges.
- Command:

  sudo ` USERMOD ` -aG sudo abhishek

Explanation:

➢ `-aG`: Adds the user to the specified group without removing them from other groups.

- Visual: The terminal screen shows the command.



```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ sudo usermod -aG sudo abhishek

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:darkabhi
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:darkabhi
floppy:x:25:
tape:x:26:
sudo:x:27:darkabhi,abhishek
audio:x:29:
dip:x:30:darkabhi
www-data:x:33:
```

1.3 Verification

Verify that the user has been added and has sudo privileges by switching to the user and running a sudo command.

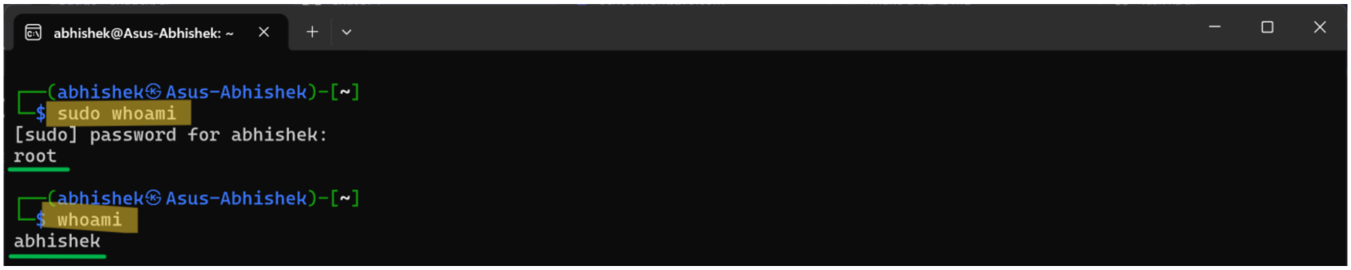
- Commands:

  `SU` abhishek

  `SUDO` whoami ……. (i)

  whoami ……. (ii)

- Expected Output: (i) `root` (ii) `abhishek`
- Visual: The terminal screen shows the command.



```
┌──(abhishek㉿Asus-Abhishek)-[~]
└─$ sudo whoami
[sudo] password for abhishek:
root

┌──(abhishek㉿Asus-Abhishek)-[~]
└─$ whoami
abhishek
```

2. Creating a Group and Adding Multiple Users

2.1 Objective

Create a new group named “batch37” and add three new users (std1, std2, std3) to this group.

2.2 Commands and Execution

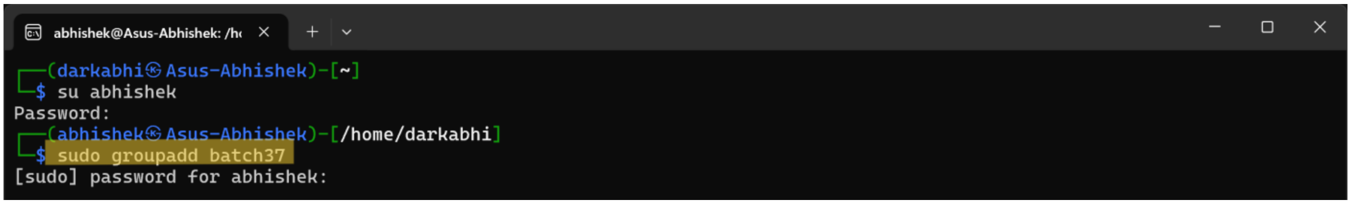
Create the group:

- Use the `GROUPADD` command to create a new group named “batch37”.
- Command:

  sudo `GROUPADD` batch37

  Explanation:

  ➢ This creates a new group in the system.
- Visual: The terminal screen shows the command.



```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ su abhishek
Password:
┌──(abhishek㉿Asus-Abhishek)-[/home/darkabhi]
└─$ sudo groupadd batch37
[sudo] password for abhishek:
```

Create New Users and Assign Passwords:

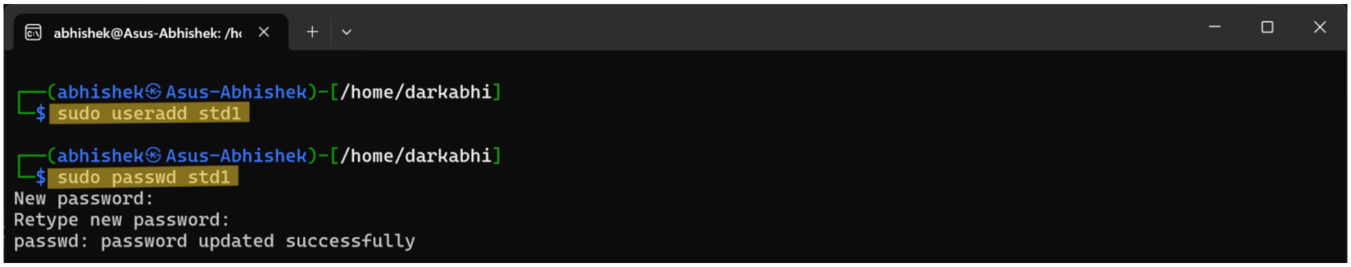
- Create users `STD1`, `STD2` & `STD3` and set their passwords.
- Commands:

  sudo `USERADD` std1
  sudo `PASSWD` std1
  sudo `USERADD` std2
  sudo `PASSWD` std2
  sudo `USERADD` std3
  sudo `PASSWD` std3

  Explanation:

  ➢ Similar to the steps taken to create the user abhishek, these commands create std1, std2 and std3 and set their passwords.
- Visual: The terminal screen shows the commands.

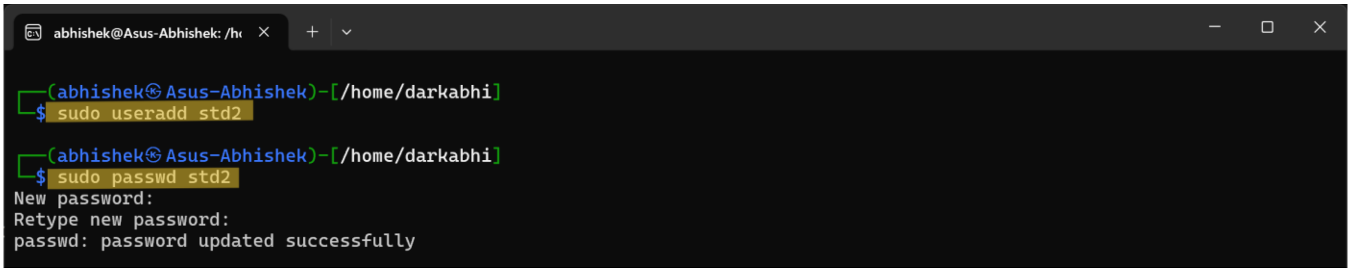
Create and set password for ` STD1 `.



```
┌──(abhishek㉿Asus-Abhishek)-[/home/darkabhi]
└─$ sudo useradd std1

┌──(abhishek㉿Asus-Abhishek)-[/home/darkabhi]
└─$ sudo passwd std1
New password:
Retype new password:
passwd: password updated successfully
```

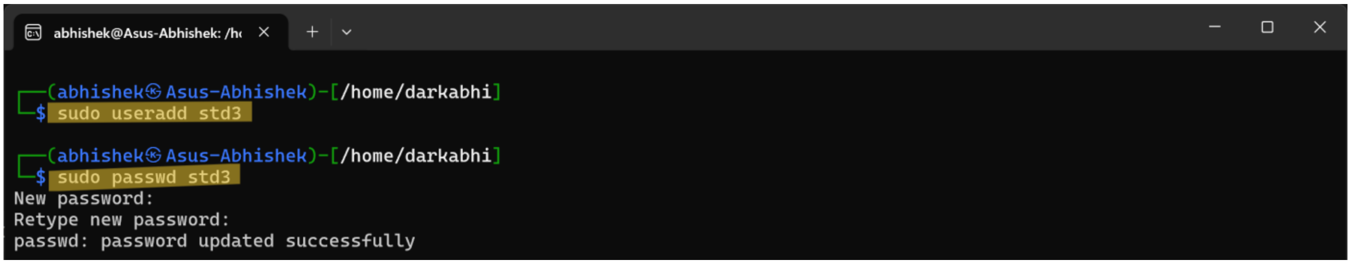
Create and set password for ` STD2 `.



```
┌──(abhishek㉿Asus-Abhishek)-[/home/darkabhi]
└─$ sudo useradd std2

┌──(abhishek㉿Asus-Abhishek)-[/home/darkabhi]
└─$ sudo passwd std2
New password:
Retype new password:
passwd: password updated successfully
```

Create and set password for ` STD3 `.



```
┌──(abhishek㉿Asus-Abhishek)-[/home/darkabhi]
└─$ sudo useradd std3

┌──(abhishek㉿Asus-Abhishek)-[/home/darkabhi]
└─$ sudo passwd std3
New password:
Retype new password:
passwd: password updated successfully
```

Add Users to the `batch37` Group:

- Add the newly created users to the ` BATCH37 ` group.
- Commands:

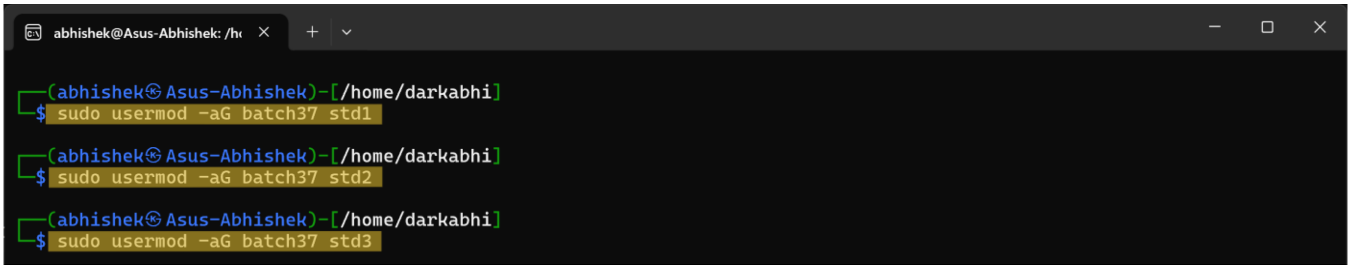
  sudo ` USERMOD ` -aG batch37 std1

  sudo ` USERMOD ` -aG batch37 std2

  sudo ` USERMOD ` -aG batch37 std3

  Explanation:

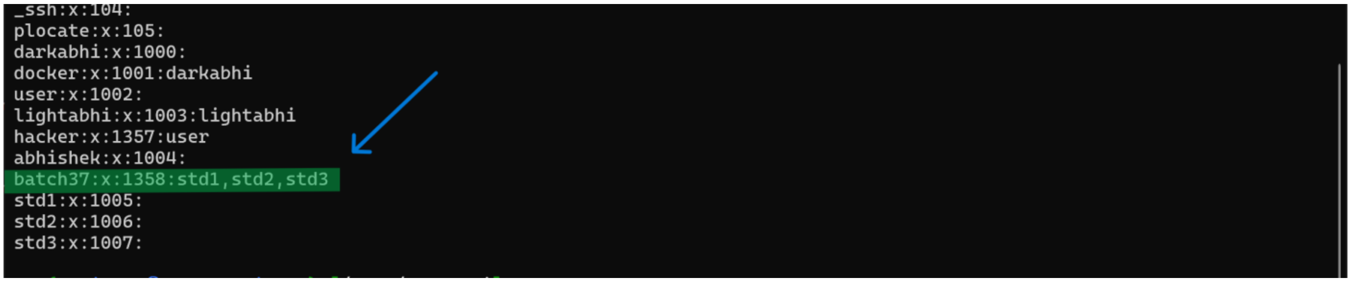
  ➢ This command appends each user to the batch37 group without removing them from any other groups.
- Visual: The terminal screen shows the commands.



```
┌──(abhishek㉿Asus-Abhishek)-[/home/darkabhi]
└─$ sudo usermod -aG batch37 std1

┌──(abhishek㉿Asus-Abhishek)-[/home/darkabhi]
└─$ sudo usermod -aG batch37 std2

┌──(abhishek㉿Asus-Abhishek)-[/home/darkabhi]
└─$ sudo usermod -aG batch37 std3
```

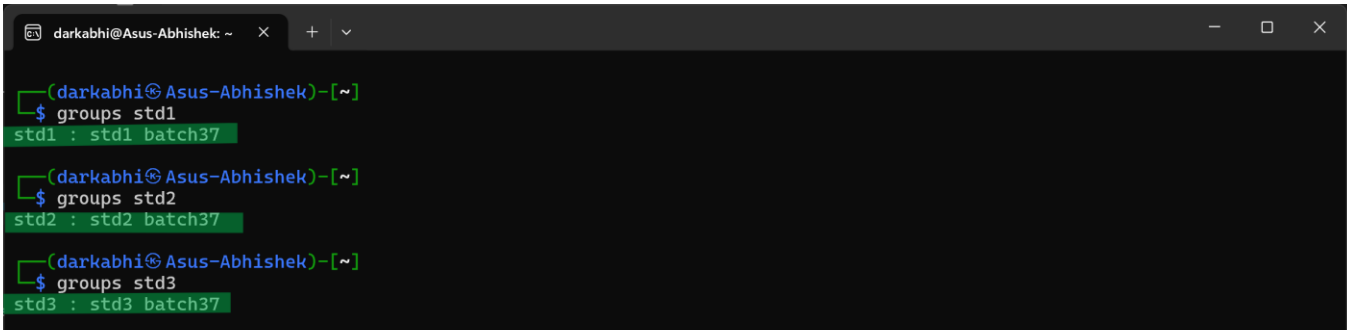


```
_ssh:x:104:
plocate:x:105:
darkabhi:x:1000:
docker:x:1001:darkabhi
user:x:1002:
lightabhi:x:1003:lightabhi
hacker:x:1357:user
abhishek:x:1004:
batch37:x:1358:std1,std2,std3
std1:x:1005:
std2:x:1006:
std3:x:1007:
```

2.3 Verification

Verify the users are part of the batch37 group.

- Commands:
  `GROUPS` std1
  `GROUPS` std2
  `GROUPS` std3
- Expected Output: Each command should list `BATCH37` among the groups.
- Visual: The terminal screen shows the command.



```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ groups std1
std1 : std1 batch37

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ groups std2
std2 : std2 batch37

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ groups std3
std3 : std3 batch37
```

3. File Management and Permission Settings

3.1 Objective

Create a file in `std1's` home directory, manage its ownership and permissions, and verify these settings.

3.2 Commands and Execution

Create Home Directory and Set Permissions:

- Since the home directory for `STD1` was not automatically created during user creation, it needs to be created manually, followed by setting ownership and permissions.
- Commands:

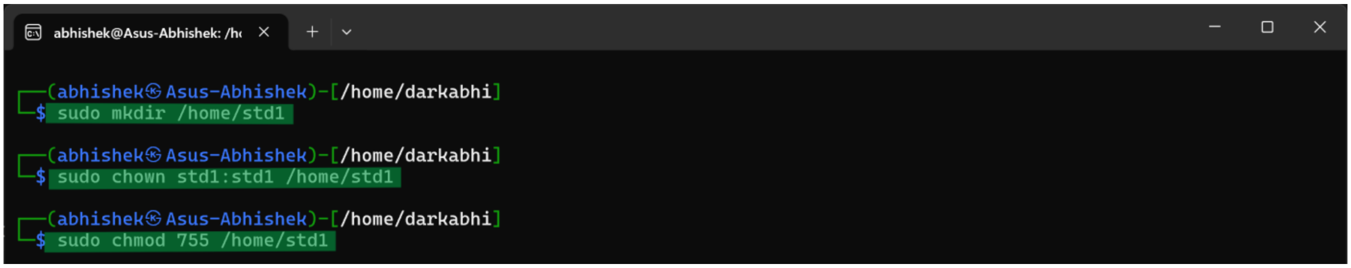
  sudo `MKDIR` /home/std1

  sudo `CHOWN` std1:std1 /home/std1

  sudo `CHMOD` 755 /home/std1

  Explanation:

  - The `mkdir` command is used to manually create the home directory for the user std1.
  - The `chown` command changes the ownership of the /home/std1 directory to user std1 and group std1.
  - The `chmod` 755 command sets the permissions so that the owner has full access (read, write, execute), the group can read and execute, and others can also read and execute.
- Visual: The terminal screen shows the command.



```
┌──(abhishek㉿Asus-Abhishek)-[/home/darkabhi]
└─$ sudo mkdir /home/std1

┌──(abhishek㉿Asus-Abhishek)-[/home/darkabhi]
└─$ sudo chown std1:std1 /home/std1

┌──(abhishek㉿Asus-Abhishek)-[/home/darkabhi]
└─$ sudo chmod 755 /home/std1
```

Create and Write to a File:

- Login as `STD1` and create a new file in the home directory.
- Commands:

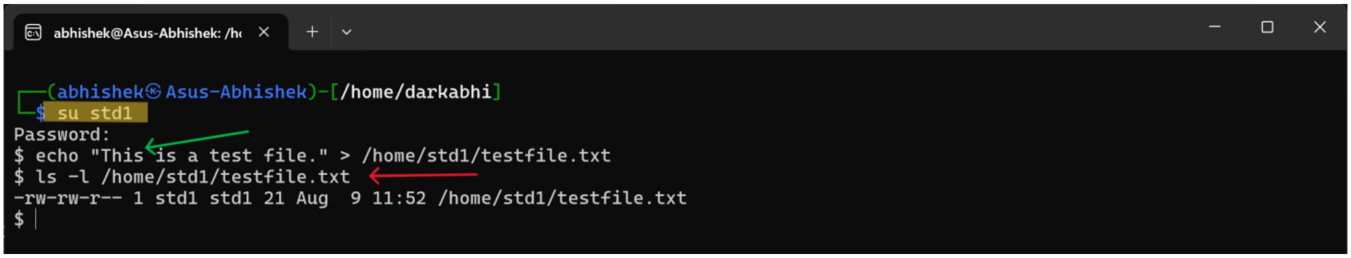
  su `STD1`

  `ECHO` "This is a test file." > /home/std1/testfile.txt

  `LS` -l /home/std1/testfile.txt

  Explanation:

  - The echo command writes the string to the file testfile.txt.
  - The ls command long lists the created file testfile.txt
- Visual: The terminal screen shows the command.



```
┌──(abhishek㉿Asus-Abhishek)-[/home/darkabhi]
└─$ su std1
Password:
$ echo "This is a test file." > /home/std1/testfile.txt
$ ls -l /home/std1/testfile.txt
-rw-rw-r-- 1 std1 std1 21 Aug  9 11:52 /home/std1/testfile.txt
$
```

Change Ownership and Group Ownership:

- Change the ownership of the file to ` STD1 ` and the group ownership to ` BATCH37 `.
- Commands:

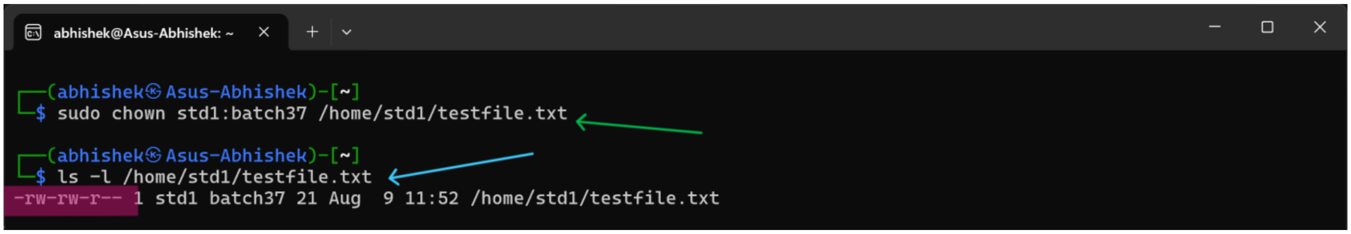
  sudo ` CHOWN ` std1:batch37 /home/std1/testfile.txt

  ` LS ` -l /home/std1/testfile.txt

  Explanation:

  ➢ Chown changes the file's ownership to std1 and group ownership to batch37.

  ➢ The ls command long lists the created file testfile.txt
- Visual: The terminal screen shows the command.



```
┌──(abhishek㉿Asus-Abhishek)-[~]
└─$ sudo chown std1:batch37 /home/std1/testfile.txt

┌──(abhishek㉿Asus-Abhishek)-[~]
└─$ ls -l /home/std1/testfile.txt
-rw-rw-r-- 1 std1 batch37 21 Aug  9 11:52 /home/std1/testfile.txt
```

Modify File Permissions:

- Change the file permissions so that the owner can read and write, the group can only read, and others have no access.
- Commands:

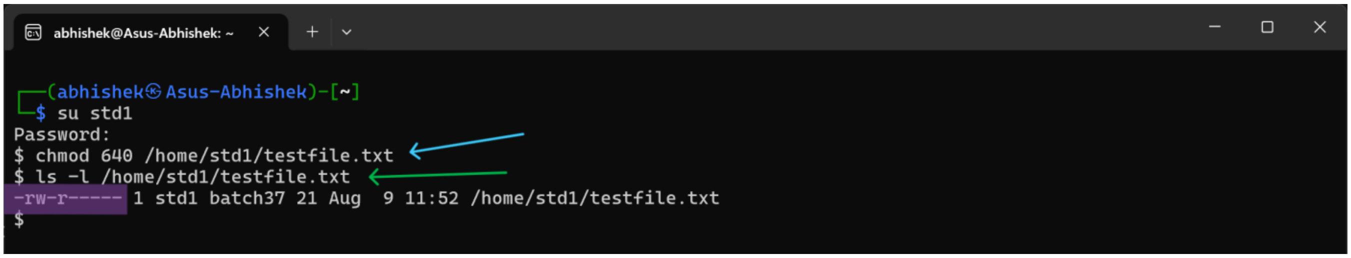
  su ` STD1 `

  ` CHMOD ` 640 /home/std1/testfile.txt

  ` LS ` -l /home/std1/testfile.txt

  Explanation:

  ➢ Chmod 640 sets the permissions such that the owner can read/write, the group can read, and others have no permissions. [**4** = Read (r), **2** = Write (w), **1** = Execute (x), **0** = No permissions (—)]
- Visual: The terminal screen shows the command.



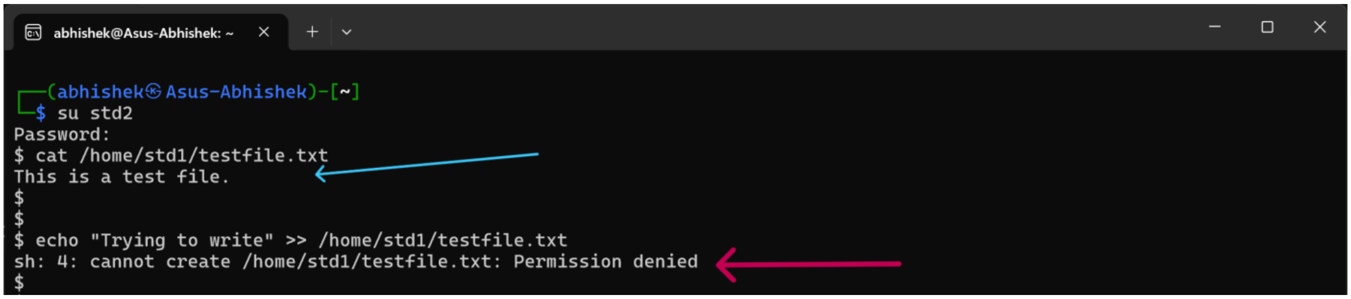
```
┌──(abhishek㉿Asus-Abhishek)-[~]
└─$ su std1
Password:
$ chmod 640 /home/std1/testfile.txt
$ ls -l /home/std1/testfile.txt
-rw-r----- 1 std1 batch37 21 Aug  9 11:52 /home/std1/testfile.txt
$
```

3.3 Verification

Verify Access for Group Members:

Verify Permissions by Switching Users.

- Commands:
  `SU` std2
  `CAT` /home/std1/testfile.txt
  `ECHO` “Trying to write” > /home/std1/testfile.txt
- Expected Output: Both `STD2` & `STD3` should be able to read the file but should not be able to write to it. Members in groups should have only read permission.
- Visual: The terminal screen shows the command.

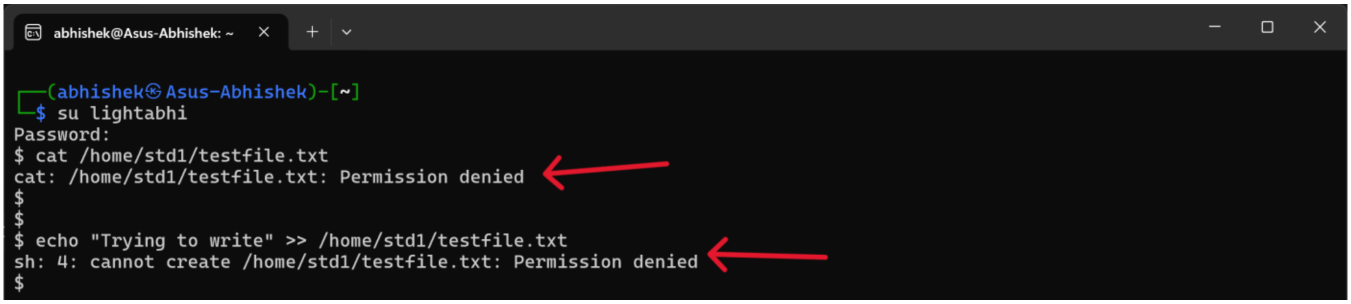


```
┌──(abhishek㉿Asus-Abhishek)-[~]
└─$ su std2
Password:
$ cat /home/std1/testfile.txt
This is a test file.
$
$
$ echo "Trying to write" >> /home/std1/testfile.txt
sh: 4: cannot create /home/std1/testfile.txt: Permission denied
$
```

Verify No Access for Others:

Switch to another user not in the `BATCH37` group and check access.

- Commands:
  `SU` lightabhi
  `CAT` /home/std1/testfile.txt
  `ECHO` “Trying to write” > /home/std1/testfile.txt
- Expected Output: The user should not be able to read or write to the file.
- Visual: The terminal screen shows the command.



```
┌──(abhishek㉿Asus-Abhishek)-[~]
└─$ su lightabhi
Password:
$ cat /home/std1/testfile.txt
cat: /home/std1/testfile.txt: Permission denied
$
$
$ echo "Trying to write" >> /home/std1/testfile.txt
sh: 4: cannot create /home/std1/testfile.txt: Permission denied
$
```

4. Summary and Conclusion

4.1 Summary

In this project, we successfully completed a series of tasks centered around user and group management in a Linux environment. The steps included:

- **Creating a New User and Assigning Sudo Privileges:** We created a user named abhishek, set a password, and added the user to the sudo group, granting administrative privileges.
- **Group Creation and User Assignment:** We created a new group named batch37 and added three users (std1, std2, and std3) to this group.
- **File Management and Permissions:** We manually created a home directory for the std1 user, set the appropriate ownership and permissions, and created a test file within this directory. We adjusted the file permissions to ensure that std1 had full access, while group members had read-only access, and other users had no access.
- **Verification of Permissions:** Finally, we verified the file permissions by switching between users, ensuring that the access control settings were applied correctly.

4.2 Conclusion

User and group management are critical components of system administration in Linux, playing a pivotal role in maintaining both security and organization. By creating and managing users and groups, system administrators can control access to system resources, ensuring that users have the appropriate level of access to files, directories, and commands.

This project highlighted the importance of correctly configuring user directories, ownership, and permissions to prevent unauthorized access and potential security breaches. Additionally, the manual creation of user home directories and subsequent permission setting underscored the flexibility and power of Linux in handling various administrative tasks.

Throughout the process, some challenges were encountered, particularly with ensuring the correct permissions and ownership settings. However, these challenges were addressed by careful command execution and verification, emphasizing the need for precision in system administration.

5. References

Official Ubuntu Documentation: User Management

- Link: https://ubuntu.com/server/docs/administration-users
- https://manpages.ubuntu.com/manpages/focal/man1
- https://manpages.ubuntu.com/manpages/focal/man1/mkdir.1.html

Linux man Pages

- Commands like useradd, usermod, groupadd, chmod, chown, and passwd were referenced through the built-in manual (man) pages in Linux.

The Linux Command Line by William E. Shotts, Jr.

- A comprehensive guide for understanding Linux commands and system administration tasks.