Experiment No. 3	
Explore Linux Commands	
Date of Performance:	
Date of Submission:	
Marks:	
Sign:	

Aim: Explore user management commands of linux.

Objective:

Explore basic commands of linux

Theory:

A user is an entity, in a Linux operating system, that can manipulate files and perform several other operations. Each user is assigned an ID that is unique for each user in the operating system. In this post, we will learn about users and commands which are used to get information about the users. After installation of the operating system, the ID 0 is assigned to the root user and the IDs 1 to 999 (both inclusive) are assigned to the system users and hence the ids for local user begins from 1000 onwards.

In a single directory, we can create 60,000 users. Now we will discuss the important commands to manage users in Linux.

- useradd - create a new user or update default new user information ,useradd is a low level utility for adding users.
- userdel - delete a user account and related files
- groupadd - create a new group , The groupadd command creates a new group account using the values specified on the command line plus the default values from the system. The new group will be entered into the system files as needed.
- groupdel - delete a group , The groupdel command modifies the system account files, deleting all
- entries that refer to GROUP. The named group must exist
- who - show who is logged on , Print information about users who are currently logged in.
- whoami - print effective userid
- passwd - change user password

The passwd command changes passwords for user accounts. A normal user may only change the password for his/her own account, while the superuser may change the password for any account. passwd also changes the account or associated password validity period.

CSL403: Operating System Lab

1. to enter in root sudo su then password

2. to add new user type useradd csds11 (username)

3. to check a newly added user you have to type cat etc/passwod

4 set a password to new user : sudo passwd csds11

5. create a new group: groupadd csds12

6. Check group cat /etc/group

7. add new user in newly created group useradd -G csds12 piya1 (group name and new user name)

8. to check : cat /etc/group

9. to enter in new user : su - csds11 (username)

10. to delete user type : userdel csds (username that you have to delete)

11. Again check whether it is deleted or not cat /etc/passwd

10. to delete user type : groupdel csds12 (group that you have to delete)

12. Again check whether it is deleted or not cat /etc/passwd

13. who - show who is logged on Print information about users who are currently logged in.

14.whoami - print effective userid

Output:

useradd

```
ubuntu@ubuntu-HP-280-Pro-G5-Small-Form-Factor-PC:~$ sudo useradd test1
[sudo] password for ubuntu:
ubuntu@ubuntu-HP-280-Pro-G5-Small-Form-Factor-PC:~$ sudo useradd test1
useradd: user 'test1' already exists
```

2) userdel

```
ubuntu@ubuntu-HP-280-Pro-G5-Small-Form-Factor-PC:~$ sudo userdel test1
ubuntu@ubuntu-HP-280-Pro-G5-Small-Form-Factor-PC:~$ cat /etc/passwd | grep test1
ubuntu@ubuntu-HP-280-Pro-G5-Small-Form-Factor-PC:~$ test1
Command 'test1' not found, did you mean:
  command 'testr' from deb python3-testrepository (0.0.20-6)
  command 'test' from deb coreutils (8.32-4.1ubuntu1)
Try: sudo apt install <deb name>
```

3)groupadd

```
ubuntu@ubuntu-hp1:~$ groupadd traitors
groupadd: Permission denied.
groupadd: cannot lock /etc/group; try again later.
```

4)groupdel

```
ubuntu-hp1:~$ groupdel traitors
l: group 'traitors' does not exist
```

5)who

```
ubuntu@ubuntu-hp1:~$ who -q
ubuntu
```

6) whoami

```
ubuntu@ubuntu-hp1:~$ whoami
ubuntu
```

7)passwd

```
ubuntu@ubuntu-hp1:~$ passwd -e user1
passwd: Permission denied.
```

Conclusion:

Explain Linux API?

The Linux API, or Application Programming Interface, serves as a bridge between user applications and the Linux kernel. It encompasses a vast collection of functions, data structures, and conventions that programmers can utilize to interact with the underlying operating system. At its core, the Linux API provides a set of system calls, which are entry points into the kernel that enable applications to request services such as file I/O, process management, memory allocation, and networking. Additionally, the Linux API includes libraries and utilities that simplify common programming tasks, such as string manipulation, mathematical operations, and input/output handling. These libraries, such as the GNUCLibrary (glibc), provide a higher-level interface while still ultimately relying on system calls to perform their operations. Furthermore, the Linux API extends beyond system calls and libraries to encompass other interfaces like the /proc and /sys filesystems, which expose information about system configuration, processes, and hardware to user space programs. Overall, the Linux API serves as a comprehensive toolkit for developers to harness the power and versatility of the Linux operating system in their applications.