User specific commands

who -who is logged on with login info

1. Time of last system boot
2. Current run level of the system
3. List of logged in users and more.

```
emertxe@ubuntu:~$ who
emertxe  :0           2022-03-09 08:56 (:0)
emertxe@ubuntu:~$
```

Figure 1 Output of who

w - who is logged on and what they are doing

w command is used to show who is logged on and what they are doing. This shows the information about the users currently on the machine and their processes. The following entries are displayed for each user: login name, the tty name, the remote host, login time, idle time, JCPU, PCPU, and the command line of their current process. The JCPU time is the time used by all processes attached to the tty. The PCPU time is the time used by the current process, named in the “what” field.

```
emertxe@ubuntu:~$ w
 17:14:53 up  7:56,  1 user,  load average: 0.07, 0.03, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
emertxe  :0       :0               09:19   ?xdm?   2:43   0.01s /usr/lib/
emertxe@ubuntu:~$
```

Figure 2 Output of w

whoami - print effective user id

```
emertxe@ubuntu:~$ whoami
emertxe
emertxe@ubuntu:~$
```

Figure 3 output of whoami

uname - to print system information

uname -a -to print all system information
uname -r -to print kernel release information
uname -v -to print kernel version information

```
emertxe@ubuntu:~$ uname
Linux
emertxe@ubuntu:~$ uname  -a
Linux ubuntu 4.15.0-99-generic #100-Ubuntu SMP Wed Apr 22 20:32:56 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
emertxe@ubuntu:~$ uname  -r
4.15.0-99-generic
emertxe@ubuntu:~$ uname  -v
#100-Ubuntu SMP Wed Apr 22 20:32:56 UTC 2020
emertxe@ubuntu:~$
```

Figure 4 Output of uname

adduser-Add user to the system

Adding a new user to the system needs superuser permission. so the command is executed along with the sudo command as shown in the figure below.

The sudo command **allows you to run programs with the security privileges of another user** (by default, as the superuser). It prompts you for your current user password and confirms your request to execute a command

Command

sudo adduser <user_name>

```
emertxe@ubuntu:~$
emertxe@ubuntu:~$ adduser user1
adduser: Only root may add a user or group to the system.
emertxe@ubuntu:~$
emertxe@ubuntu:~$ sudo adduser user1
[sudo] password for emertxe:
Adding user `user1' ...
Adding new group `user1' (1003) ...
Adding new user `user1' (1002) with group `user1' ...
Creating home directory `/home/user1' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for user1
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
emertxe@ubuntu:~$
```

Figure 5 .Add new user

passwd - to change the password of user

passwd command is used to change the password of current user and other user

command

passwd -to change password of current user

sudo passwd <user_name> - to change password of other user

```
emertxe@ubuntu:~$ passwd
Changing password for emertxe.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
emertxe@ubuntu:~$
```

Figure 6 changing password of current user

```
emertxe@ubuntu:~$ passwd user1
passwd: You may not view or modify password information for user1.
emertxe@ubuntu:~$ sudo passwd user1
[sudo] password for emertxe:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Figure 7 changing password of other user

sudo su -switch user

sudo su command is to switch user

```
emertxe@ubuntu:~$ sudo su user1
[sudo] password for emertxe:
user1@ubuntu:/home/emertxe$
user1@ubuntu:/home/emertxe$ cd
user1@ubuntu:~$
user1@ubuntu:~$ pwd
/home/user1
user1@ubuntu:~$ exit
exit
emertxe@ubuntu:~$ pwd
/home/emertxe
emertxe@ubuntu:~$
```

Figure 8 switch user

To switch back to active user use exit command

Note:cd without any argument is the same as cd ~ both will change to the user's home directory.

groups - to print groups user is in

Command

group - to print all the groups which current user is in
group <user_name> - to print the groups which given user is in

```
emertxe@ubuntu:~$ groups
emertxe adm dialout cdrom sudo dip plugdev lpadmin sambashare g1 group1
emertxe@ubuntu:~$
emertxe@ubuntu:~$ groups user3
user3 : user3
emertxe@ubuntu:~$
```

Figure 9 Groups command to check groups of user

addgroup - to create new group

sudo addgroup <group_name>

```
emertxe@ubuntu:~$ sudo addgroup group_1
Adding group `group_1' (GID 1008) ...
Done.
emertxe@ubuntu:~$
```

Figure 10 Creating groups

usermod - modify user account

To add user to group
sudo usermod -a -G <group_name> <user_name>

-a -> to append a group to a user

```
emertxe@ubuntu:~$ sudo usermod -a -G group_1 user3
emertxe@ubuntu:~$
emertxe@ubuntu:~$ groups user3
user3 : user3 group_1
emertxe@ubuntu:~$
```

Figure 11 To add group to user

chown - to change the file and group owner

Command

chown <username> <filename> - to change file owner

chown :<groupname> <filename> - to change group owner

chown <username:groupname> <filename> - to change both file and group owner

```
emertxe@ubuntu:~/NEW$ ls -l
total 0
-rw-rw-r-- 1 emertxe emertxe 0 Mar  8 14:52 file1.txt
-rw-rw-r-- 1 emertxe emertxe 0 Mar  8 14:52 file2.txt
-rw-rw-r-- 1 emertxe emertxe 0 Mar  8 14:52 file3.txt
-rw-rw-r-- 1 emertxe emertxe 0 Mar  8 14:52 file4.txt
emertxe@ubuntu:~/NEW$ chown user1 file1.txt
chown: changing ownership of 'file1.txt': Operation not permitted
emertxe@ubuntu:~/NEW$ sudo chown user1 file1.txt
emertxe@ubuntu:~/NEW$ ls -l
total 0
-rw-rw-r-- 1 user1   emertxe 0 Mar  8 14:52 file1.txt
-rw-rw-r-- 1 emertxe emertxe 0 Mar  8 14:52 file2.txt
-rw-rw-r-- 1 emertxe emertxe 0 Mar  8 14:52 file3.txt
-rw-rw-r-- 1 emertxe emertxe 0 Mar  8 14:52 file4.txt
emertxe@ubuntu:~/NEW$
```

Figure 12 change file owner of a file

```
emertxe@ubuntu:~/NEW$ sudo chown :user1 file2.txt
emertxe@ubuntu:~/NEW$ ls -l
total 0
-rw-rw-r-- 1 user1   emertxe 0 Mar  8 14:52 file1.txt
-rw-rw-r-- 1 emertxe user1   0 Mar  8 14:52 file2.txt
-rw-rw-r-- 1 emertxe emertxe 0 Mar  8 14:52 file3.txt
-rw-rw-r-- 1 emertxe emertxe 0 Mar  8 14:52 file4.txt
emertxe@ubuntu:~/NEW$
```

Figure 13 change group owner of a file

```
emertxe@ubuntu:~/NEW$ sudo chown user1:user1 file3.txt
emertxe@ubuntu:~/NEW$ ls -l
total 0
-rw-rw-r-- 1 user1   emertxe 0 Mar  8 14:52 file1.txt
-rw-rw-r-- 1 emertxe user1   0 Mar  8 14:52 file2.txt
-rw-rw-r-- 1 user1   user1   0 Mar  8 14:52 file3.txt
-rw-rw-r-- 1 emertxe emertxe 0 Mar  8 14:52 file4.txt
```

Figure 14 To change both group and file owner of a file

userdel - to delete user from system

command

sudo userdel <username>

```
emertxe@ubuntu:~/NEW$ userdel user1
userdel: Permission denied.
userdel: cannot lock /etc/passwd; try again later.
emertxe@ubuntu:~/NEW$
emertxe@ubuntu:~/NEW$ sudo userdel user1
emertxe@ubuntu:~/NEW$
```

Figure 15 To delete user from system