



ASSIGNMENT 3

BASIC LINUX COMMAND

NIMISHA JAMES
REG MCA – B
ROLL NO:11

Usermod

usermod command or modify user is a command in Linux that is used to change the properties of a user in Linux through the command line. After creating a user we have to sometimes change their attributes like password or login directory etc. so in order to do that we use the Usermod command. The information of a user is stored in the following files:

- /etc/passwd
- /etc/group
- /etc/shadow
- /etc/login.defs
- /etc/gshadow
- /etc/login.defs

When we execute usermod command in terminal the command make the changes in these files itself.

```
(root@kali)~# usermod
Usage: usermod [options] LOGIN

Options:
  -b, --badnames          allow bad names
  -c, --comment COMMENT   new value of the GECOS field
  -d, --home HOME_DIR     new home directory for the user account
  -e, --expiredate EXPIRE_DATE set account expiration date to EXPIRE_DATE
  -f, --inactive INACTIVE set password inactive after expiration
                           to INACTIVE
  -g, --gid GROUP          force use GROUP as new primary group
  -G, --groups GROUPS      new list of supplementary GROUPS
  -a, --append             append the user to the supplemental GROUPS
                           mentioned by the -G option without removing
                           the user from other groups
  -h, --help              display this help message and exit
  -l, --login NEW_LOGIN    new value of the login name
  -L, --lock              lock the user account
  -m, --move-home         move contents of the home directory to the
                           new location (use only with -d)
  -o, --non-unique         allow using duplicate (non-unique) UID
  -p, --password PASSWORD use encrypted password for the new password
  -R, --root CHROOT_DIR    directory to chroot into
  -P, --prefix PREFIX_DIR  prefix directory where are located the /etc/* files
  -s, --shell SHELL        new login shell for the user account
  -u, --uid UID            new UID for the user account
  -U, --unlock            unlock the user account
  -v, --add-subuids FIRST-LAST add range of subordinate uids
  -V, --del-subuids FIRST-LAST remove range of subordinate uids
  -w, --add-subgids FIRST-LAST add range of subordinate gids
  -W, --del-subgids FIRST-LAST remove range of subordinate gids
  -Z, --selinux-user SEUSER new SELinux user mapping for the user account
```

1. To add a comment for a user

`usermod -c "This is test user" test_user`

```
(root@kali)~# usermod -c "This is test user" test_user

(root@kali)~# cat /etc/passwd | grep test_user
test_user:x:1002:1002:This is test user:/home/test_user:/bin/sh
```

2. To change the expiry date of a user

`usermod -e 2020-05-29 test_use`

```
(root@kali)~# usermod -e 2021-01-21 test_user

(root@kali)~# chage -l test_user
Last password change           : Aug 13, 2021
Password expires                : never
Password inactive              : never
Account expires                : Jan 21, 2021
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
```

3. To lock a user

`sudo usermod -L test_user`

4. To unlock a user

`sudo usermod -U test_user`

```
(root@kali)~# usermod -L test_user

(root@kali)~# usermod -u test_user
usermod: invalid user ID 'test_user'

(root@kali)~# usermod -U test_user

(root@kali)~#
```

5. To set an unencrypted password for the user

```
(root@kali)~# usermod -p test_password test_user

(root@kali)~# cat /etc/shadow | grep test_user
test_user:test_password:18852:0:99999:7::18648:

(root@kali)~#
```

Groupadd

The groupadd command creates a new group account using the values specified on the command line and the default values from the system. The new group will be entered into the system files as needed.

```
(root@kali)~# groupadd
Usage: groupadd [options] GROUP

Options:
  -f, --force                exit successfully if the group already exists,
                             and cancel -g if the GID is already used
  -g, --gid GID              use GID for the new group
  -h, --help                 display this help message and exit
  -K, --key KEY=VALUE        override /etc/login.defs defaults
  -o, --non-unique            allow to create groups with duplicate
                             (non-unique) GID
  -p, --password PASSWORD    use this encrypted password for the new group
  -r, --system               create a system account
  -R, --root CHROOT_DIR      directory to chroot into
  -P, --prefix PREFIX_DIR    directory prefix
```

1.To add a new group

addgroup groupname

```
(root@kali)~# addgroup groupname
Adding group `groupname' (GID 1003) ...
Done.

(root@kali)~#
```

2.To display version

addgroup --version

```
(root@kali)~# addgroup --version
adduser version 3.118

Adds a user or group to the system.

Copyright (C) 1997, 1998, 1999 Guy Maor <maor@debian.org>
Copyright (C) 1995 Ian Murdock <imurdock@gnu.ai.mit.edu>,
                  Ted Hajek <tedhajek@boombox.micro.umn.edu>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or (at
your option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License, /usr/share/common-licenses/GPL, for more details.

(root@kali)~#
```

Groups

In linux, there can be multiple users(those who use/operate the system), and groups are nothing but the collection of users. Groups make it easy to manage users with the same security and access privileges. A user can be part of different groups.

Important Points:

- Groups command prints the names of the primary and any supplementary groups for each given username, or the current process if no names are given.
- If more than one name is given, the name of each user is printed before the list of that user's groups and the username is separated from the group list by a colon.

Syntax:

groups [username]...

```
(root@kali)-[~]
# groupadd usrgrp

(root@kali)-[~]
# groups
root kaboxer
```

groupmod

groupmod command in Linux is used to modify or change the existing group on Linux system. It can be handled by superuser or root user. Basically, it modifies a group definition on the system by modifying the right entry in the database of the group.

Syntax:

groupmod [option] GROUP

```
(root@kali)-[~]
# groupmod
Usage: groupmod [options] GROUP

Options:
  -g, --gid GID           change the group ID to GID
  -h, --help              display this help message and exit
  -n, --new-name NEW_GROUP change the name to NEW_GROUP
  -o, --non-unique         allow to use a duplicate (non-unique) GID
  -p, --password PASSWORD change the password to this (encrypted)
                           PASSWORD
  -R, --root CHROOT_DIR   directory to chroot into
  -P, --prefix PREFIX_DIR prefix directory where are located the /etc/* files

(root@kali)-[~]
#
```



```
(root@kali)~# groupadd mygroup

(root@kali)~# groupmod -n bettergroup mygroup

(root@kali)~# groupmod
Usage: groupmod [options] GROUP

Options:
  -g, --gid GID           change the group ID to GID
  -h, --help              display this help message and exit
  -n, --new-name NEW_GROUP change the name to NEW_GROUP
  -o, --non-unique         allow to use a duplicate (non-unique) GID
  -p, --password PASSWORD change the password to this (encrypted)
                           PASSWORD
  -R, --root CHROOT_DIR   directory to chroot into
  -P, --prefix PREFIX_DIR prefix directory where are located the /etc/* files

(root@kali)~# groupmod -g 3000 bettergroup

(root@kali)~# groupmod -g 777 3000
groupmod: group '3000' does not exist
```

groupdel

groupdel command is used to delete a existing group. It will delete all entry that refers to the group, modifies the system account files, and it is handled by superuser or root user.

Syntax:

groupdel [options] GROUP

```
(root@kali)~# groupdel
Usage: groupdel [options] GROUP

Options:
  -h, --help              display this help message and exit
  -R, --root CHROOT_DIR   directory to chroot into
  -P, --prefix PREFIX_DIR prefix directory where are located the /etc/* files
  -f, --force              delete group even if it is the primary group of a user

(root@kali)~# groupdel bettergroup

(root@kali)~# groupdel -h
Usage: groupdel [options] GROUP
```

Chmod

In Unix-like operating systems, the chmod command is used to change the access mode of a file.

The name is an abbreviation of change mode.

Syntax :

chmod [reference][operator][mode] file...

Options:

The chmod command supports the following command-line options:

-c, --changes: It is similar to the verbose option, but the difference is that it is reported if a change has been made.

-f, --silent, --quiet: It is used to suppress the error messages.

-v, --verbose: It is used to display a diagnostic for every processed file.

--no-preserve-root: It is used for not treating the backslash symbol ('/'), especially (the default).

--preserve-root: If this option is used, it will fail to operate recursively on backslash ('/').

--reference=RFILE: It is used to specify the RFILE's mode alternatively MODE values.

-R, --recursive: It is used to change files and directories recursively.

--help: It is used to display the help manual having a brief description of usage and support options.

--version: It is used to display the version information.

```
-rw-r--r-- 1 root root 10 Aug 12 13:01 myfile2.txt
```

```
root@kali:~# chmod g+rw myfile2.txt
```

```
-rw-rw-r-- 1 root root 10 Aug 12 13:01 myfile2.txt
```

chown

Linux chown command is used to change a file's ownership, directory, or symbolic link for a user or group. The chown stands for change owner. In Linux, each file is associated with a corresponding owner or group.

The Linux system may have multiple users. Every user has a unique name and user ID. If only a user is available in the system, the user will be the owner of each file.

The Linux system may have multiple users. Every user has a unique name and user ID. If only a user is available in the system, the user will be the owner of each file.

Users can be listed in different groups. The group allows us to set permission on the group level instead of setting permission on an individual level.

Options:

Following are the command-line options of the chown command:

-c, --changes: It is used to display the detailed output like verbose, but it is reported when only a change is made.

-f, --silent, --quiet: It is used to suppress the error messages.

-v, --verbose: It is used to display a diagnostic for every processed file.

--dereference: It is used to affect the referent of each symbolic link.

-h, --no-dereference: It is used to affect the symbolic links instead of any referenced file.

--from=CURRENT_OWNER:CURRENT_GROUP: It is used to change the specific owner and group.

- no-preserve-root:** It is used for not treating the backslash ('/') especially.
- preserve-root:** If the chown is failed to operate recursively on backslash ('/').
- reference=RFILE:** It is used to specify the RFILE's owner and group rather than their values.

```
(rootkali)~[~/college]
# chmod g+rw nm.txt

(rootkali)~[~/college]
#

(rootkali)~[~/college]
# ls -l
total 20
drwxr-xr-x 2 root root 4096 Jun 14 11:55 assignments
drwxr-xr-x 3 root root 4096 Jun 21 12:23 exam
drwxr-xr-x 2 root root 4096 Jun 14 12:22 lab
-rw-rw-r-- 1 root root 18 Jun 20 10:36 nm.txt
drwxr-xr-x 2 root root 4096 Jun 14 09:24 work

(rootkali)~[~/college]
# ls -l nm.txt
-rw-rw-r-- 1 root root 18 Jun 20 10:36 nm.txt

(rootkali)~[~/college]
# chown master nm.txt
chown: invalid user: 'master'

(rootkali)~[~/college]
# chown test_user nm.txt

(rootkali)~[~/college]
# chown -c test_user nm.txt
```

id

id command in Linux is **used to find out user and group names** and numeric ID's (UID or group ID) of the current user or any other user in the server. List out all the groups a user belongs to. Display security context of the current user

```
(rootkali)~[~/college]
# id
uid=0(root) gid=0(root) groups=0(root),142(kaboxer)

(rootkali)~[~/college]
#
```

ps

The ps command is used to view currently running processes on the system. It helps us to determine which process is doing what in our system, how much memory it is using, how much CPU space it occupies, user ID, command name, etc .

The ps command may display different results for different systems because it displays information about the currently running process of a system.

Option	Function
<u>ps -ef/ ps -aux</u>	List currently running process in full format
<u>ps -ax</u>	List currently running process

<u>ps -u <username></u>	List process for specific user
<u>ps -C <command></u>	List process for given command
<u>ps -p <PID></u>	List process with given PID
<u>ps -ppid <PPID></u>	List process with given ppid
<u>pstree</u>	Show process in hierarchy
<u>ps -L</u>	List all threads for a particular process
<u>ps --sort pmem</u>	Find memory leak
<u>ps -eo</u>	Show security information
<u>ps -U root -u root u</u>	Show process running by root

```
(root@kali)~/college
# ps
  PID TTY          TIME CMD
 2592 pts/0    00:00:06 zsh
 3570 pts/0    00:00:00 ps

(root@kali)~/college
# ps -a
  PID TTY          TIME CMD
 3575 pts/0    00:00:00 ps

(root@kali)~/college
# ps -T
  PID   SPID  TTY          TIME CMD
 2592   2592 pts/0    00:00:06 zsh
 3580   3580 pts/0    00:00:00 ps

(root@kali)~/college
# ps -r
  PID TTY      STAT   TIME COMMAND
 3584 pts/0    R+      0:00   ps -r

(root@kali)~/college
# ps -L
  PID   LWP  TTY          TIME CMD
 2592   2592 pts/0    00:00:06 zsh
 3589   3589 pts/0    00:00:00 ps

(root@kali)~/college
# ps -eo
error: format specification must follow -o

Usage:
ps [options]

Try 'ps --help <simple|list|output|threads|misc|all>'
or 'ps --help <s|l|o|t|m|a>'
for additional help text.

For more details see ps(1).

(root@kali)~/college
# ps -ax
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss      0:03  /sbin/init splash
    2 ?        S        0:00  [kthreadd]
    3 ?        I<       0:00  [rcu_gp]
    4 ?        I<       0:00  [rcu_par_gp]
    6 ?        I<       0:00  [kworker/0:0H-events_highpri]
    9 ?        I<       0:00  [mm_percpu_wq]
   10 ?        S        0:00  [rcu_tasks_rude_]
   11 ?        S        0:00  [rcu_tasks_trace]
```

Top

The top command displays all the running process within the environment of your system. It helps in monitoring system usage and performances. It is mainly used to detect load on the server by system administrators.



```
top - 07:16:19 up 4:33, 2 users, load average: 0.07, 0.05, 0.01
Tasks: 157 total, 1 running, 156 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.3 sy, 0.0 ni, 99.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3000.4 total, 2083.0 free, 426.9 used, 490.5 buff/cache
MiB Swap: 975.0 total, 975.0 free, 0.0 used. 2418.1 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
557	root	20	0	1201620	112112	44312	S	1.0	3.6	0:59.92	Xorg
2589	root	20	0	1281688	79640	63488	S	1.0	2.6	0:14.55	qterminal
818	root	20	0	161152	2724	2224	S	0.3	0.1	0:34.54	VBoxClient
1	root	20	0	164012	10296	7732	S	0.0	0.3	0:03.08	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.03	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
9	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_rude_
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_trace
12	root	20	0	0	0	0	S	0.0	0.0	0:01.42	ksoftirqd/0
13	root	20	0	0	0	0	I	0.0	0.0	0:03.07	rcu_sched
14	root	rt	0	0	0	0	S	0.0	0.0	0:00.12	migration/0
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
16	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
17	root	rt	0	0	0	0	S	0.0	0.0	0:00.41	migration/1
18	root	20	0	0	0	0	S	0.0	0.0	0:00.02	ksoftirqd/1
20	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/1:0H-events_highpri
21	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/2
22	root	rt	0	0	0	0	S	0.0	0.0	0:00.42	migration/2
23	root	20	0	0	0	0	S	0.0	0.0	0:00.09	ksoftirqd/2
25	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/2:0H-events_highpri
26	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/3
27	root	rt	0	0	0	0	S	0.0	0.0	0:00.44	migration/3
28	root	20	0	0	0	0	S	0.0	0.0	0:00.02	ksoftirqd/3
30	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/3:0H-events_highpri
35	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
36	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
37	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kauditd
38	root	20	0	0	0	0	S	0.0	0.0	0:00.01	khungtaskd
39	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper
40	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	writeback
41	root	20	0	0	0	0	S	0.0	0.0	0:00.87	kcompactd0
42	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
43	root	39	19	0	0	0	S	0.0	0.0	0:00.34	khugepaged
45	root	20	0	0	0	0	I	0.0	0.0	0:02.49	kworker/3:1-events
62	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kintegrityd
63	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kblockd
64	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	blkcg_punt_bio
65	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	edac-poller
66	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	devfreq_wq
67	root	0	-20	0	0	0	I	0.0	0.0	0:00.06	kworker/3:1H-kblockd
71	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kswapd0
72	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kthrotld
73	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	acpi_thermal_pm