

Assignment no 4

AIM-Study of user management commands in Unix Theory:

User management in Unix involves commands for adding, modifying, and deleting users and groups. The `useradd` command creates new users, while `usermod` modifies user properties such as group membership and home directory. To remove a user, `userdel` is used. Groups are managed using `groupadd`, `groupmod`, and `groupdel`. The `passwd` command sets or changes user passwords, and `chage` manages password expiration policies. Permissions and ownership of files can be controlled using `chown` (change owner) and `chmod` (change permissions). The `id` command displays user and group IDs, while `who` and `w` show logged-in users. These commands help administrators efficiently manage user accounts and access rights.

Commands:

whoami:

Prints the username of the current user.

```
$ whoami  
webmaster
```

2.id

- **Purpose:** Displays the user ID (UID), group ID (GID), and the groups that the user belongs to.

Syntax: □ `id [username]`

```
webmaster@4647c6356dee:/home/cg/root/67a8d92773029$id

uid=1000(webmaster) gid=1000(webmaster) groups=1000(webmaster)
webmaster@4647c6356dee:/home/cg/root/67a8d92773029$ webmaster@4647c6356dee
:/home/cg/root/67a8d92773029$

webmaster@4647c6356dee:/home/cg/root/67a8d92773029$ export "PS1=$ "
```

usermod

Modifies an existing user account.

```
root@LAPTOP-00G4J1FA:~# usermod
Usage: usermod [options] LOGIN

Options:
  -a, --append                append the user to the supplemental GROUPS
                              mentioned by the -G option without removing
                              the user from other groups
  -b, --badname               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
  -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
  -P, --prefix PREFIX_DIR    prefix directory where are located the /etc/* files
  -r, --remove                remove the user from only the supplemental GROUPS
                              mentioned by the -G option without removing
                              the user from other groups
  -R, --root CHROOT_DIR      directory to chroot into
  -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
```

Variations:

- `usermod -l newname oldname`

Changes username

```
root@LAPTOP-00G4J1FA:~# usermod -l oldUser newUser
root@LAPTOP-00G4J1FA:~# su oldUser
$ id -un
oldUser
```

- `usermod -d /new/home username`

Changes home directory.

```
root@LAPTOP-00G4J1FA:~# usermod -d /new/home oldUser
root@LAPTOP-00G4J1FA:~# eval echo ~oldUser
/new/home
```

userdel

Deletes a user account.

Variation:

- `userdel -r username`

Removes user and home directory.

```
root@LAPTOP-00G4J1FA:~# userdel -r oldUser
userdel: oldUser mail spool (/var/mail/oldUser) not found
userdel: oldUser home directory (/new/home) not found
root@LAPTOP-00G4J1FA:~# su oldUser
su: user oldUser does not exist or the user entry does not contain all the required fields
```

groupadd

Creates a new group.

```
root@LAPTOP-00G4J1FA:~# sudo groupadd new_group
root@LAPTOP-00G4J1FA:~# getent group new_group
new_group:x:1002:
```

groupmod:

Modifies an existing group.

```
root@LAPTOP-00G4J1FA:~# sudo groupmod options new_group
Usage: groupmod [options] GROUP

Options:
  -a, --append                append the users mentioned by -U option to the group
                              without removing existing user members
  -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
  -U, --users USERS           list of user members of this group
```

Variation:

- `groupmod -n newname oldname`

Renames a group.

```
root@LAPTOP-00G4J1FA:~# groupmod -n old_group new_group
root@LAPTOP-00G4J1FA:~# getent group old_group
old_group:x:1002:
```

- `groupmod -g newGID groupname`

Changes the group ID (GID).

```
root@LAPTOP-00G4J1FA:~# groupmod -g 777 old_group
root@LAPTOP-00G4J1FA:~# getent group old_group
old_group:x:777:
```

groupdel:

Deletes a group.

```
root@LAPTOP-00G4J1FA:~# sudo groupdel old_group
root@LAPTOP-00G4J1FA:~# getent group old_group
root@LAPTOP-00G4J1FA:~# |
```

gpasswd:

Administers the /etc/group file

```
root@LAPTOP-00G4J1FA:~# gpasswd
Usage: gpasswd [option] GROUP

Options:
  -a, --add USER           add USER to GROUP
  -d, --delete USER        remove USER from GROUP
  -h, --help               display this help message and exit
  -Q, --root CHROOT_DIR    directory to chroot into
  -r, --remove-password    remove the GROUP's password
  -R, --restrict            restrict access to GROUP to its members
  -M, --members USER,...  set the list of members of GROUP
                          --extrausers    use the extra users database
  -A, --administrators ADMIN,... set the list of administrators for GROUP

Except for the -A and -M options, the options cannot be combined.
```

chown:

Changes file ownership.

```
root@LAPTOP-00G4J1FA:~# touch myfile.txt
root@LAPTOP-00G4J1FA:~# ls -l myfile.txt
-rw-r--r-- 1 root root 0 Feb  8 18:04 myfile.txt
root@LAPTOP-00G4J1FA:~# sudo chown deepmalika25 myfile.txt
root@LAPTOP-00G4J1FA:~# ls -l myfile.txt
-rw-r--r-- 1 deepmalika25 root 0 Feb  8 18:04 myfile.txt
```

Variation:

- chown user:group filename Changes

file owner and group.

```
root@LAPTOP-00G4J1FA:~# sudo chown test:newGroup myfile.txt
root@LAPTOP-00G4J1FA:~# ls -l myfile.txt
-rw-r--r-- 1 test newGroup 0 Feb  8 18:04 myfile.txt
```

chgrp:

Changes the group of a file.

```
root@LAPTOP-00G4J1FA:~# sudo chgrp oldGroup myfile.txt
root@LAPTOP-00G4J1FA:~# ls -l myfile.txt
-rw-r--r-- 1 test oldGroup 0 Feb  8 18:04 myfile.txt
```

chage:

Manages user password expiration.

```
root@LAPTOP-00G4J1FA:~# chage
Usage: chage [options] LOGIN

Options:
  -d, --lastday LAST_DAY      set date of last password change to LAST_DAY
  -E, --expiredate EXPIRE_DATE set account expiration date to EXPIRE_DATE
  -h, --help                  display this help message and exit
  -i, --iso8601               use YYYY-MM-DD when printing dates
  -I, --inactive INACTIVE     set password inactive after expiration
                              to INACTIVE
  -l, --list                  show account aging information
  -m, --mindays MIN_DAYS      set minimum number of days before password
                              change to MIN_DAYS
  -M, --maxdays MAX_DAYS     set maximum number of days before password
                              change to MAX_DAYS
  -R, --root CHROOT_DIR       directory to chroot into
  -W, --warndays WARN_DAYS    set expiration warning days to WARN_DAYS
```

Variations:

- chage -l username

Displays password aging info.

```
root@LAPTOP-00G4J1FA:~# chage -l test
Last password change           : Feb 08, 2025
Password expires                : never
Password inactive              : never
Account expires                : never
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
```

- chage -M days username

Sets maximum days before password change.

```
root@LAPTOP-00G4J1FA:~# chage -M 300 test
root@LAPTOP-00G4J1FA:~# chage -l test
Last password change           : Feb 08, 2025
Password expires                : Dec 05, 2025
Password inactive              : never
Account expires                : never
Minimum number of days between password change : 0
Maximum number of days between password change : 300
Number of days of warning before password expires : 7
```

chfn:

```
$ chfn -h
chfn: option requires an argument -- 'h'
Usage: chfn [options] [LOGIN]

Options:
  -f, --full-name FULL_NAME    change user's full name
  -h, --home-phone HOME_PHONE  change user's home phone number
  -o, --other OTHER_INFO       change user's other GECOS information
  -r, --room ROOM_NUMBER       change user's room number
  -R, --root CHROOT_DIR        directory to chroot into
  -u, --help                    display this help message and exit
  -w, --work-phone WORK_PHONE  change user's office phone number
      --extrausers              Use the extra users database
```

Variations:

- `chfn -f "New Full Name" username`

Updates user's full name.

```
root@LAPTOP-00G4J1FA:~# chfn -f "NewFullName" test
root@LAPTOP-00G4J1FA:~# getent passwd test
test:x:1001:1002:NewFullName,,,:/home/test:/bin/sh
```

exit

Exits the current shell session.

Conclusion:

From this assignment, I understand that user management in Unix is essential for controlling access, security, and system administration. Commands like `useradd`, `usermod`, and `userdel` help create and manage users, while `groupadd` and `groupdel` handle group assignments. Permissions and ownership are controlled using `chown` and `chmod`, ensuring secure file access. Additionally, tools like `passwd` and `chage` enforce password policies.