# Linux User and Group management

**Presentation** · June 2021

DOI: 10.13140/RG.2.2.29088.71687

**1 author:**

Naol Getachew
Mattu University
**27** PUBLICATIONS **0** CITATIONS

## Objectives:

- To know the Users and Groups Management
- To Add, Delete and Configure User and Groups

# Linux User and Group Management

## What's an account?

When a computer is used by many people it is usually necessary to differentiate between the users, for example, so that their private files can be kept private. This is important even if the computer can only be used by a single person at a time, as with most microcomputers. Thus, each user is given a unique username, and that name is used to log in. There's more to a user than just a name, however. An account is all the files, resources, and information belonging to one user. The term hints at banks, and in a commercial system each account usually has some money attached to it, and that money vanishes at different speeds depending on how much the user stresses the system. For example, disk space might have a price per megabyte and day, and processing time might have a price per second.

## Types of user account

There are three types of accounts on a Linux system:

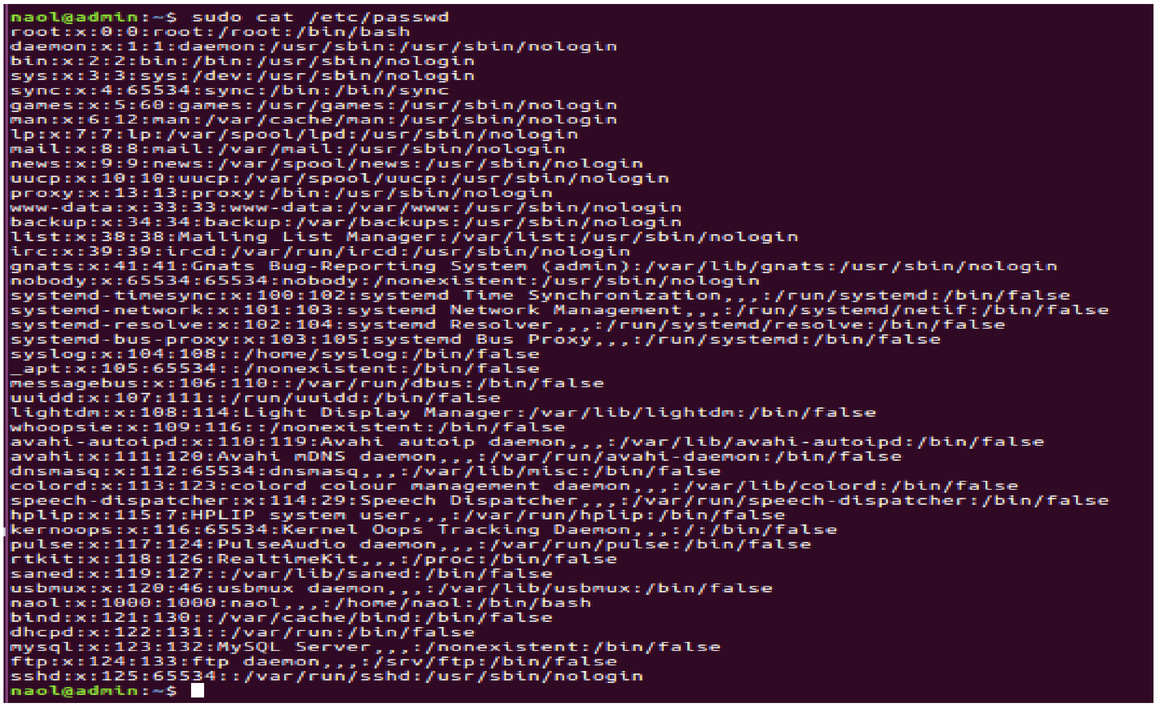
- Root account: This is also called super user and would have complete and unfettered control of the system. A super user can run any commands without any restriction. This user should be assumed as a system administrator.
- System accounts: System accounts are those needed for the operation of system specific components for example mail accounts and the sshd accounts. These accounts are usually needed for some specific function on your system, and any modifications to them could adversely affect the system.
- User accounts: User accounts provide interactive access to the system for users and groups of users. General users are typically assigned to these accounts and usually have limited access to critical system files and directories.

Linux supports a concept of Group Account which logically groups a number of accounts. Every account would be a part of any group account. Linux groups plays important role in handling file permissions and process management.

## Managing Users and Groups

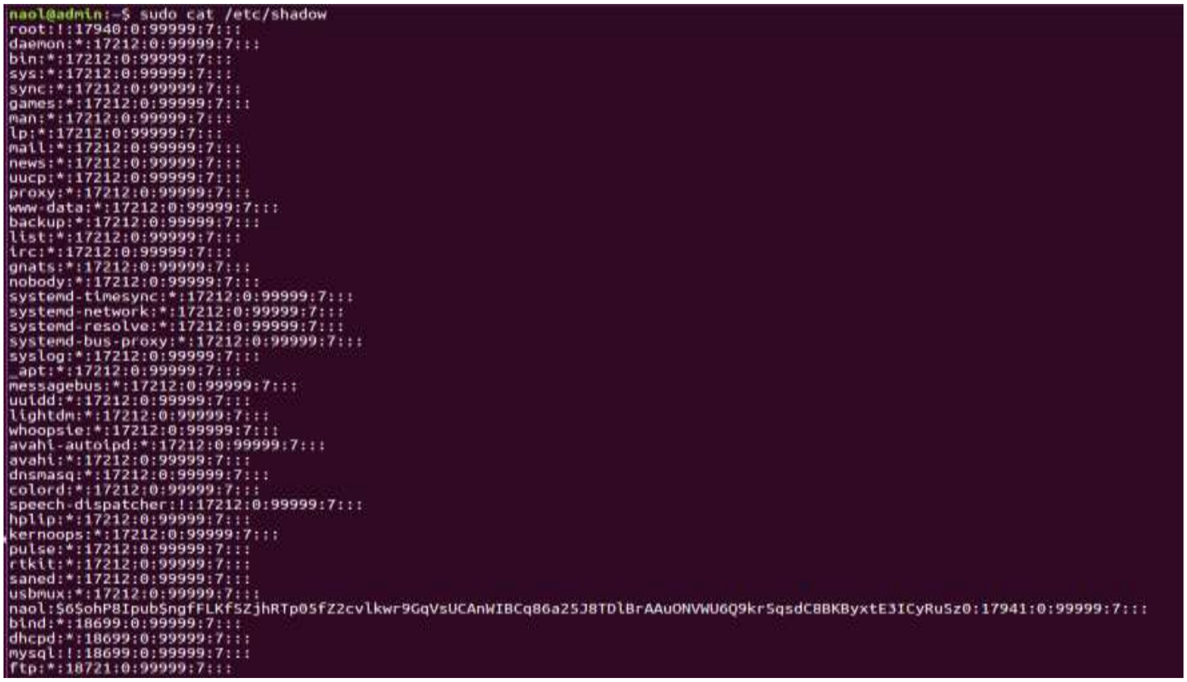
There are three main user administration files:

- /etc/passwd: Keeps user account and password information. This file holds the majority of information about accounts on the Linux system



```
naol@admin:~$ sudo cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
messagebus:x:106:110::/var/run/dbus:/bin/false
uuidd:x:107:111::/run/uuidd:/bin/false
lightdm:x:108:114:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:109:116::/nonexistent:/bin/false
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:111:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/bin/false
colord:x:113:123:colord colour management daemon,,,:/var/lib/colord:/bin/false
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
hplip:x:115:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127::/var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
naol:x:1000:1000:naol,,,:/home/naol:/bin/bash
bind:x:121:130::/var/cache/bind:/bin/false
dhcpd:x:122:131::/var/run:/bin/false
mysql:x:123:132:MySQL Server,,,:/nonexistent:/bin/false
ftp:x:124:133:ftp daemon,,,:/srv/ftp:/bin/false
sshd:x:125:65534::/var/run/sshd:/usr/sbin/nologin
naol@admin:~$
```

- /etc/shadow: Holds the encrypted password of the corresponding account. Not all the system support this file



```
naol@admin:~$ sudo cat /etc/shadow
root:!:17940:0:99999:7:::
daemon:*:17212:0:99999:7:::
bin:*:17212:0:99999:7:::
sys:*:17212:0:99999:7:::
sync:*:17212:0:99999:7:::
games:*:17212:0:99999:7:::
man:*:17212:0:99999:7:::
lp:*:17212:0:99999:7:::
mail:*:17212:0:99999:7:::
news:*:17212:0:99999:7:::
uucp:*:17212:0:99999:7:::
proxy:*:17212:0:99999:7:::
www-data:*:17212:0:99999:7:::
backup:*:17212:0:99999:7:::
list:*:17212:0:99999:7:::
irc:*:17212:0:99999:7:::
gnats:*:17212:0:99999:7:::
nobody:*:17212:0:99999:7:::
systemd-timesync:*:17212:0:99999:7:::
systemd-network:*:17212:0:99999:7:::
systemd-resolve:*:17212:0:99999:7:::
systemd-bus-proxy:*:17212:0:99999:7:::
syslog:*:17212:0:99999:7:::
_apt:*:17212:0:99999:7:::
messagebus:*:17212:0:99999:7:::
uuidd:*:17212:0:99999:7:::
lightdm:*:17212:0:99999:7:::
whoopsie:*:17212:0:99999:7:::
avahi-autoipd:*:17212:0:99999:7:::
avahi:*:17212:0:99999:7:::
dnsmasq:*:17212:0:99999:7:::
colord:*:17212:0:99999:7:::
speech-dispatcher:!:17212:0:99999:7:::
hplip:*:17212:0:99999:7:::
kernoops:*:17212:0:99999:7:::
pulse:*:17212:0:99999:7:::
rtkit:*:17212:0:99999:7:::
saned:*:17212:0:99999:7:::
usbmux:*:17212:0:99999:7:::
naol:$6$ohP8Ipub$ngfFLKfSZjhRTp05fZ2cvlkwr9GqVsUCAnWIBCq86a25J8TDlBrAAuONVWU6Q9krSqsdC8BKByxtE3ICyRuSz0:17941:0:99999:7:::
bind:*:18699:0:99999:7:::
dhcpd:*:18699:0:99999:7:::
mysql:!:18699:0:99999:7:::
ftp:*:18721:0:99999:7:::
```

- /etc/group: This file contains the group information for each account.
- /etc/gshadow: This file contains secure group account information.

Following are commands available on the majority of Linux systems to create and manage accounts and groups:

| Command | Description |
|---|---|
| useradd | Adds accounts to the system. |
| usermod | Modifies account attributes. |
| userdel | Deletes accounts from the system. |
| groupadd | Adds groups to the system. |
| groupmod | Modifies group attributes. |
| groupdel | Removes groups from the system. |

## Create a Group

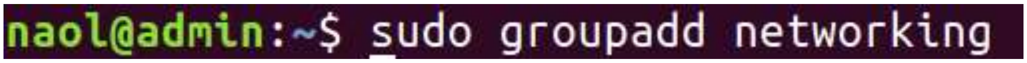
Would need to create groups before creating any account otherwise you would have to use existing groups at your system. You would have all the groups listed in /etc/groups file.

All the default groups would be system account specific groups and it is not recommended to use them for ordinary accounts. So following is the syntax to create a new group account:

Syntax: *groupadd [-g gid [-o]] [-r] [-f] groupname*

| Option | Description |
|---|---|
| -g GID | The numerical value of the group's ID. |
| -o | This option permits to add group with non-unique GID |
| -r | This flag instructs groupadd to add a system account |
| -f | This option causes to just exit with success status if the specified group already exists. With -g, if specified GID already exists, other (unique) GID is chosen. |
| Groupname | Actaul group name to be created. |

- If we do not specify any parameter then system would use default values. Following example would create Networking group with default values, which is very much acceptable for most of the administrators.



```
naol@admin:~$ sudo groupadd networking
```

## Modify a Group

To modify a group, use the groupmod syntax: *$groupmod –n newgroupname oldgroupame*

### Delete a Group

To delete an existing group, all you need are the groupdel command and the group name. To delete the ‘*networking’* group, the command is:

```
$goupdel  networking
```

### Create an Account

To create account we use the following command

```
$ sudo useradd -d homedir -g groupname -m -s shell -u userid accountname
```

| **Option** | **Description** |
|---|---|
| - d homedir | Specifies home directory for the account. |
| -g groupname | Specifies a group account for this account. |
| -m | Creates the home directory if it doesn't exist. |
| -s shell | Specifies the default shell for this account. |
| -u userid | You can specify a user id for this account. |
| accountname | Actual account name to be created |

### Adduser and useradd command

The *adduser and addgroup* commands add users and groups to the system according to command line options and configuration information in /etc/adduser.conf. They are friendlier front ends to the low level tools like useradd, groupadd and usermod programs, by default choosing Debian policy conformant UID and GID values, creating a home directory with skeletal configuration, running a custom script, and other features. A basic run of the adduser command is as follows:

```
$ sudo adduser username
```

This simple command will do a number of things:

1. Create the user named username.
2. Create the user's home directory (default is /home/username and copy the files from /etc/skelinto it.
3. Create a group with the same name as the user and place the user in it.
4. Prompt for a password for the user.
5. Prompt for additional information on the user.

The *useradd* program can most have accomplish most of this, however it does not do so by default and needs additional options. Some of the information requires more commands:

- useradd -m -U username
- passwd username
- chfn username

Another common use for adduser is to simplify the process of adding a user to a group. Here, the following command:

```
$ sudo adduser username newgroup
```

Replaces a more complex usermod command that requires the groups which the user is already a member of (and that you would like the user to remain a member) to be specified:

```
$ sudo usermod -G all,other,groups,user,is,in,newgroup
```

### Modify an Account

The usermod command enables you to make changes to an existing account from the command line. It uses the same arguments as the *useradd* command, plus the -l argument, which allows you to change the account name.

For example, to change the account name network to network2 and to change home directory accordingly, you would need to issue following command:

```
$ sudo usermod -d /home/network2 -m -l network network2
```

### Delete an Account

The *userdel* command can be used to delete an existing user. This is a very dangerous command if not used with caution. There is only one argument or option available for the command: **-r**, for removing the account's home directory and mail file.

For example, to remove account network2, you would need to issue following command:

```
$ sudo userdel -r network2
```

If you want to keep user’s home directory for backup purposes, omit the -r option. You can remove the home directory as needed at a later time.

### Changing user properties

There are a few commands for changing various properties of an account (i.e., the relevant field in /etc/passwd):

- chfn: to change the full name field.
- chsh: to change the login shell.
- passwd: to change the password.

The super−user may use these commands to change the properties of any account. Normal users can only change the properties of their own account. It may sometimes be necessary to disable these commands (with chmod) for normal users, for example in an environment with many novice users.

Other tasks need to be done by hand. For example, to change the username, you need to edit /etc/passwd directly (with vipw, remember). Likewise, to add or remove the user to more groups, you need to edit /etc/group (with vigr). Such tasks tend to be rare, however, and should be done with caution: for example, if you change the username, e−mail will no longer reach the user, unless you also create a mail alias.

### /etc/passwd and other informative files

The basic user database in a Unix system is the text file, /etc/passwd (called the password file), which lists all valid usernames and their associated information. The file has one line per username, and is divided into seven colon−delimited fields:

- Username
- Previously this was where the user's password was stored.
- Numeric user id.
- Numeric group id.
- Full name or other description of account.
- Home directory.
- Login shell (program to run at login).

Most Linux systems use shadow passwords. As mentioned, previously passwords were stored in the /etc/passwd file. This newer method of storing the password: the encrypted password is stored in a separate file, /etc/shadow, which only root can read. The /etc/passwd file only contains a special marker in the second field. Any program that needs to verify a user is set uid, and can therefore access the shadow password file. Normal programs, which only use the other fields in the password file, can't get at the password.

### Initial environment: /etc/skel

When the home directory for a new user is created, it is initialized with files from the /etc/skel directory. The system administrator can create files in /etc/skel that will provide a nice default environment for users. For example, he might create a /etc/skel/.profile that sets the EDITOR environment variable to some editor that is friendly towards new users. However, it is usually best to try to keep /etc/skel as small as possible, since it will be next to impossible to update existing users' files. For example, if the name of the friendly editor changes, all existing users would have to edit their .profile. The system administrator could try to do it automatically, with a script, but that is almost certain going to break someone's file.

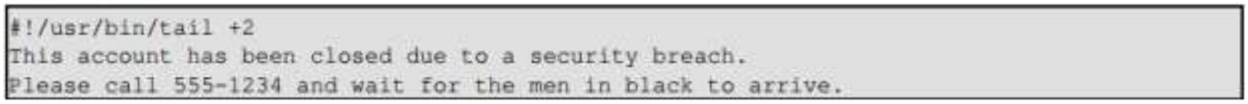
Whenever possible, it is better to put global configuration into global files, such as /etc/profile. This way it is possible to update it without breaking users' own setups.

### Disabling a user temporarily

It is sometimes necessary to temporarily disable an account, without removing it. For example, the user might not have paid his fees, or the system administrator may suspect that a cracker has got the password of that account.

The best way to disable an account is to change its shell into a special program that just prints a message. This way, whoever tries to log into the account, will fail, and will know why. The message can tell the user to contact the system administrator so that any problems may be dealt with.

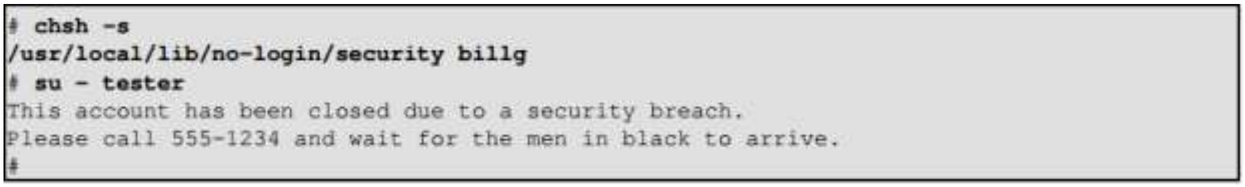
It would also be possible to change the username or password to something else, but then the user won't know what is going on. Confused users mean more work. A simple way to create the special programs is to write `tail scripts':



```
#!/usr/bin/tail +2
This account has been closed due to a security breach.
Please call 555-1234 and wait for the men in black to arrive.
```

The first two characters (`#!') tell the kernel that the rest of the line is a command that needs to be run to interpret this file. The tail command in this case outputs everything except the first line to the standard output.

If user billg is suspected of a security breach, the system administrator would do something like this:



```
# chsh -s
/usr/local/lib/no-login/security billg
# su - tester
This account has been closed due to a security breach.
Please call 555-1234 and wait for the men in black to arrive.
#
```

The purpose of the su is to test that the change worked, of course. Tail scripts should be kept in a separate directory, so that their names don't interfere with normal user commands.

### About passwd

Changes a user's password. Syntax:  passwd [options] [LOGIN]

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

The user is first prompted for his/her old password, if one is present. This password is then encrypted and compared against the stored password. The user has only one chance to enter the correct password. The superuser is permitted to bypass this step so that forgotten passwords may be changed.

After the password has been entered, password aging information is checked to see if the user is permitted to change the password at this time. If not, passwd refuses to change the password and exits. The user is then prompted twice for a replacement password. The second entry is compared against the first and both are required to match in order for the password to be changed. Then, the password is tested for complexity. As a general guideline, passwords should consist of 6 to 8 characters including one or more characters from each of the following sets:

- lower case letters
- digits 0 through 9
- punctuation marks

***passwd*** will reject any password which is not suitably complex.

The security of a password depends upon the strength of the encryption algorithm and the size of the key space. The legacy UNIX System encryption method is based on the NBS DES algorithm. More recent methods are now recommended when setting up password encryption on a system. The size of the key space depends upon the randomness of the password which is selected.

Compromises in password security normally result from careless password selection or handling. For this reason, you should not select a password which appears in a dictionary or which must be written down. The password should also not be a proper name, your license number, birth date, or street address. Any of these may be used as guesses to violate system security.

**Options:**

| | |
|---|---|
| -a, --all | This option can be used only with -S and causes show status for all users. |
| -d, --delete | Delete a user's password (make it empty). This is a quick way to disable a password for an account. It will set the named account passwordless. |
| -e, --expire | Immediately expire an account's password. This in effect can force a user to change his/her password at the user's next login. |
| -h, --help | Display a help message, and exit. |

Password complexity checking may vary from site to site. The user is urged to select a password as complex as he or she feels comfortable with. Users may not be able to change their password on a system if NIS is enabled and they are not logged into the NIS server.

***passwd*** exits with one of the following status codes, depending on what occurred:

| | |
|---|---|
| 0 | Success. |
| 1 | Permission denied. |
| 2 | Invalid combination of options. |
| 3 | Unexpected failure; nothing done. |
| 4 | Unexpected failure; passwd file missing. |
| 5 | passwd file busy; try again. |
| 6 | Invalid argument to one or more options. |

Examples

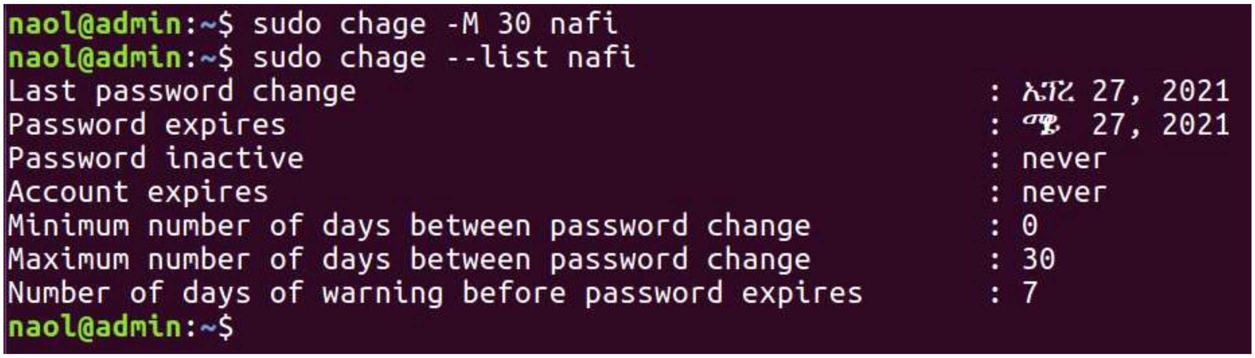
*passwd* : Change your own password.

*passwd username* : Change the password for the user named username.

## Set Password Expiry Date for an user using chage option -M

Root user (system administrators) can set the password expiry date for any user. In the following example, user dur password is set to expire 10 days from the last password change. Please note that option -M will update both ―Password expires‖ and ―Maximum number of days between password change‖ entries as shown below

Syntax: *# chage -M number-of-days username*



```
naol@admin:~$ sudo chage -M 30 nafi
naol@admin:~$ sudo chage --list nafi
Last password change                                    : ኤፕሪ 27, 2021
Password expires                                        : ሜይ  27, 2021
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 30
Number of days of warning before password expires       : 7
naol@admin:~$
```