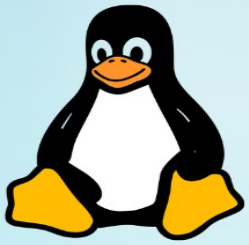

LINUX
USER & GROUP MANAGEMENT

Commands & Examples

DHANASEKAR T

Ebook Title : Linux User and Group Management Commands and Examples
https://www.linkedin.com/in/tkdhanasekar/

Table of Contents

1 /etc/passwd

/etc/passwd file is a fundamental configuration file on Unix and Linux systems that stores essential information about user accounts.

Purpose

It keeps user account information used by the system to identify and authenticate users.

It is world-readable, meaning any user can read it, but only the root user can modify it.

Passwords themselves are not stored here anymore (for security reasons) - they're in /etc/shadow.

File Format

Each line in /etc/passwd represents one user account, with 7 colon-separated fields:

`username:password:UID:GID:GECOS:home_directory:shell`

Breakdown of fields:

Field	Name	Description
1	**`username`**	The user's login name (e.g., root, doe, doe)
2	**`password`**	An x or * indicates the encrypted password is stored in /etc/shadow
3	**`UID`**	a unique number for each user
4	**`GID`**	corresponds to the user's primary group in /etc/group.
5	**`GECOS`**	Comment or full name - often used for the user's real name or contact info
6	**`home_directory`**	The path to the user's home directory
7	**`shell`**	The user's login shell (e.g., /bin/bash, /bin/false)

Example Entry

`root:x:0:0:root:/root:/bin/bash`

`frappe:x:1001:1001:,,,:/home/frappe:/bin/bash`

System users (like daemon, nobody, mail, etc.) are also listed here but often cannot log in.

To view it safely:

```
cat /etc/passwd
```

To list usernames only:

```
cut -d: -f1 /etc/passwd
```

To check a specific user’s entry:

```
grep username /etc/passwd
```

2 /etc/shadow

The /etc/shadow file is a critical security file on Unix and Linux systems that stores the encrypted (hashed) passwords and related password-aging information for user accounts.
It complements /etc/passwd, which holds general user info but no longer stores passwords for security reasons.

Purpose

Contains secure password hashes and password expiration details.

Only the root user (or processes with elevated privileges) can read it.

Provides stronger protection because /etc/passwd is world-readable, while /etc/shadow is not

File Format

Each line corresponds to one user account, with 9 colon-separated fields:

`username:password:last_change:min:max:warn:inactive:expire:reserved`

Field	Name	Description
1	**`username`**	Must match the username in /etc/passwd
2	**`password`**	Encrypted password hash (or a special value like ! or *)
3	**`last_change`**	Days since Jan 1, 1970, when the password was last changed
4	**`min`**	Minimum number of days before the password can be changed
5	**`max`**	Maximum number of days before the password must be changed
6	**`warn`**	Number of days before expiration to warn the user
7	**`inactive`**	Days after password expires until the account is disabled
8	**`expire`**	Date (days since Jan 1, 1970) when the account expires
9	**`reserverd`**	Reserved for future use (usually empty)

Example Entry

root:6Q5xHqYqG$9nKM2dD.4bdNZo3hTUE...:19500:0:99999:7:::

root → username

6Q5xHqYqG$9nKM2dD... → password hash (6 means SHA-512)

19500 → password last changed (in days since Jan 1, 1970)

0 → no minimum days before changing password

99999 → maximum 99999 days before it must be changed

7 → 7 days warning before expiration

::: → remaining fields unused

Special Password Field Values

Symbol	meaning
!	Account locked (password authentication disabled)
*****	Account cannot be logged into (no valid password)
Blank	No password required — security risk!
idsalt$hash	Encrypted password hash — $id indicates algorithm used

Common hash identifiers:

Prefix	algorithm
1	MD5
$2a$, $2b$, $2y$	Blowfish (bcrypt)
5	SHA-256
6	SHA-512

Viewing & Managing

Only root can read:

$ sudo cat /etc/shadow

Check password aging info of user doe:

```
$ sudo chage -l doe
```

Change password aging for user doe:

```
$ sudo chage doe
```

To Lock a user account doe:

```
$ sudo passwd -l doe
```

Security Notes

/etc/shadow must have strict permissions:

-rw------- 1 root root /etc/shadow

Tools like john or hashcat can attempt to crack hashes if this file is exposed.

3 /etc/group

The /etc/group file in Linux (and other Unix-like systems) defines groups on the system — collections of users that share permissions and privileges.
It's used by the OS to determine which users belong to which groups, controlling access to files, devices, and processes.

Purpose

Stores information about all user groups.

Each line corresponds to one group definition.

Used along with /etc/passwd to manage user permissions.

File Format

Each line contains 4 colon-separated fields:

`group_name:password:GID:user_list`

Field	Name	Description
1	**`group_name`**	The name of the group (e.g., sudo, wheel, developers)
2	**`password`**	Optional group password (usually x or empty; real passwords stored in /etc/gshadow)
3	**`GID`**	Group ID - a unique numeric identifier for the group
4	**`user_list`**	Comma-separated list of users that are members of the group (beyond their primary group)

Example Entries

`root:x:0:`

`daemon:x:1:`

`sudo:x:27:doe,bob`

`developers:x:1001:doe,charlie`

Explanation:

`root:x:0:` → Group root, GID 0, no additional users.

`sudo:x:27:doe,bob` → Group sudo with members doe and bob.

`developers:x:1001:doe,charlie` → Custom group for developers.

Relation to /etc/passwd

Each user in /etc/passwd has a primary group (via the GID field), and may also belong to supplementary groups listed in /etc/group

For example:

/etc/passwd

`doe:x:1001:1001:doe:/home/doe:/bin/bash`

/etc/group

`developers:x:1001:doe,charlie`

Here, doe's primary group is developers (GID 1001), but she could also be in other groups like sudo

Managing Groups

View all groups:

`$ cat /etc/group`

View groups for a user doe:

`$ groups doe`

Add a new group:

`$ sudo groupadd projectx`

Add user doe to group:

```
$ sudo usermod -aG projectx doe
```

Remove user doe from group:

```
$ sudo gpasswd -d doe projectx
```

4 /etc/gshadow

The /etc/gshadow file is the secure companion to /etc/group in Linux and other Unix-like systems. It stores encrypted group passwords and detailed group administration information - - things that are too sensitive to keep in /etc/group, which is world-readable.

Purpose

Holds secure group account information.

Ensures only privileged processes (like groupadd, gpasswd, or usermod) can modify or view sensitive group data.

Used by the system when verifying group membership or managing shared access.

Permissions

For security, only the root user and certain system processes can read or modify it:

-rw-r----- 1 root shadow /etc/gshadow

Readable only by root and the shadow group.

Never world-readable (unlike /etc/group)

File Format

Each line in /etc/gshadow corresponds to one group and has 4 colon-separated fields:

group_name:password:administrators:members

Field	Name	Description
1	**group_name**	Must match a group name in /etc/group
2	**password**	Encrypted group password (or a special symbol). Usually unused (! or *)
3	**administrators**	Comma-separated list of users who can manage the group (add/remove members)
4	**members**	Comma-separated list of users who are members of the group

Example Entries

```
root:::
sudo:!:doe,bob:charlie
developers:!::doe,charlie
projectx:$6$Gsh....$Fd8bsl3...:doe:bob,charlie
```

Explanation:

Field	Explanation
`root:::`	Group root, locked (no password)
`sudo:!:doe,bob:charlie`	Group sudo, locked password; doe,bob are admins; charlie is a member
`developers:!::doe,charlie`	Group developers with members doe and charlie
`projectx:$6$Gsh.... $Fd8bsl3...:doe:bob,charlie`	Group projectx with a SHA-512 password, doe is admin, bob and charlie are members

Password Field Values

Symbol	meaning
`!`	Locked group (cannot use password to join)
`*`	No password set / disabled
`Blank`	Group has no password (security risk)
`$id$salt$hash`	Encrypted password using algorithm indicated by $id

Group passwords are rarely used in modern Linux systems

Managing /etc/gshadow

Command	description
`sudo gpasswd groupname`	Set or change the group’s password
`sudo gpasswd -a user groupname`	Add a user to a group
`sudo gpasswd -d user groupname`	Remove a user from a group
`sudo gpasswd -A user groupname`	Set a user as group administrator
`sudo grpck`	Verify consistency between /etc/group and

	/etc/gshadow
`sudo grpconv`	Generate /etc/gshadow from /etc/group (if missing)
`sudo grpunconv`	Merge /etc/gshadow back into /etc/group

Relation Between Files

File	Purpose	Security level
`/etc/group`	Public group information	World-readable
`/etc/gshadow`	Secure group passwords & admin info	Restricted access

Consistency Example

/etc/group

`developers:x:1001:doe,charlie`

/etc/gshadow

`developers:!::doe,charlie`

Both entries define the same group, with matching members

If they get out of sync, use:

`sudo grpck`

`sudo grpconv`

5 adduser

adduser - add a user to the system

To install the adduser package in ubuntu, debian, mint distros

```
$ sudo apt install adduser
```

To install adduser on a system using dnf (like Fedora, Rocky Linux, AlmaLinux):

```
$ sudo dnf install shadow-utils
```

To add a new user doe (Interactive Mode)

```
$ adduser doe
```

What it does:

Creates a new user doe

Creates a home directory /home/doe

Prompts for a password

Prompts for user details (full name, etc.)

Adds default groups (like doe, sudo, users, depending on system)

To add a user doe with a different shell zsh

```
$ sudo adduser doe --shell /bin/zsh
```

To add a new user doe with a different configuration file

```
$ sudo adduser doe --conf custom_config.conf
```

To add a user doe with different home directory thirstyminds

```
$ sudo adduser doe --home /home/thirstyminds/
```

To get the version of the adduser command

$ sudo adduser --version

To display the help section of the adduser command

$ sudo adduser -h

Non-interactive User Creation for user doe (for scripts)

$ sudo adduser --disabled-password --gecos "" doe

Creates user , Skips all prompts (no password, no questions)

Add User doe with Custom UID/GID

$ sudo adduser --uid 1050 --gid 1050 doe

Disable Password Login for user john (e.g., for SSH key only)

$ sudo adduser --disabled-password john

creates user john but disables password login (you can still use SSH keys).

To Add a existing user doe to a Specific Group devops

$ sudo adduser doe devops

To Add a User doe Without a Home Directory

$ sudo adduser --no-create-home doe

To Add a System User backupuser

$ sudo adduser --system backupuser

Creates a system account (UID < 1000 typically)

No password login

Used for services or daemons

6 addgroup

addgroup - add group to the system

addgroup command is primarily associated with Debian/Ubuntu-based systems. On CentOS and other RHEL-based distributions, the equivalent command for creating a new group is groupadd

Basic Syntax

```
$ sudo addgroup [options] <groupname>
```

install addgroup package

```
$ sudo apt install addgroup
```

To add a new group devops

```
$ sudo addgroup devops
```

To Create a System Group

```
$ sudo addgroup --system nginx
$ sudo addgroup --system apache
```

it will Create a system group (with a lower GID, typically < 1000) for daemons/services

To Add an Existing User doe to a Group devops

```
$ sudo adduser doe devops
```

To View Group Information

```
$ getent group devops
```

or

```
$ grep devops /etc/group
```

To add a new group devops with specified group id 6789

```
$ sudo addgroup devops --gid 6789
```

To create a group with a specific shell

```
$ sudo addgroup devops --shell /bin/sh
```

To enter verbose mode

```
$ sudo addgroup devops --debug
```

To display help related to addgroup command.

```
$ addgroup –help
```

7 deluser

deluser - remove a user from the system

Basic Syntax

```
$ sudo deluser [options] <username>
```

To delete an user account doe and Keep Home Directory

```
$ sudo deluser doe
```

To delete a user doe and their home directory

```
$ sudo deluser --remove-home doe
```

To remove a user doe and all associated files

```
$ sudo deluser --remove-all-files doe
```

To remove a user doe from a specific group devops

```
$ sudo deluser doe devops
```

To delete user account doe even while the user logged in

```
$ sudo deluser --force doe
```

To delete user account doe and backup the doe’s home directory into /backup_dir

```
$ sudo deluser --backup-to /backup_dir doe
```

To view Verbose Mode of removing a user doe

```
$ sudo deluser --verbose doe
```

To Simulate (Dry Run) deleting a user doe

```
$ sudo deluser --dry-run doe
```

To remove user doe but keep their files in /backup directory

```
$ sudo deluser --remove-home --backup-to /backup/ doe
```

To remove a system user

```
$ sudo deluser --system nginx
```

To delete a group

```
$ sudo deluser --group devops
```

To view help options

```
$ deluser --help
```

8 delgroup

delgroup - remove a group from the system

Basic Syntax

```
$ sudo delgroup [options] <group_name>
```

To remove a group devops from the system

```
$ sudo delgroup devops
```

To Delete a System Group

```
$ sudo delgroup --system nginx
$ sudo delgroup --system apache
```

To view Verbose Mode

```
$ sudo delgroup --verbose devops
```

To Simulate Group Deletion of devops

```
$ sudo delgroup --dry-run devops
```

To Delete a Group Used by a System User

```
$ sudo delgroup --only-if-empty devops
```

To Check if Group Exists Before Deletion

```
$ getent group devops && sudo delgroup devops
```

To view help options

```
$ delgroup --help
```

9 useradd

useradd - create a new user or update default new user information

Basic Syntax

```
$ useradd [options] <username>
```

To add a new user doe

```
$ sudo useradd doe
```

To set a password for account doe

```
$ sudo passwd doe
```

To Create a user doe with a home directory */home/doe*

```
$ sudo useradd -m doe
```

To create a User doe with Different Home Directory

```
$ sudo useradd -d /data/myprojects doe
```

To view user doe related info

```
$ sudo cat /etc/passwd | grep doe
```

To create a User doe with a Specific User ID 1007

```
$ sudo useradd -u 1007 doe
```

Create a User with a Specific Group ID

```
$ sudo useradd -u 1007 -g devops doe
```

To verify the user’s GID

```
$ id -gn doe
```

To Add a User doe to Multiple Groups

```
$ sudo groupadd admins
$ sudo groupadd devops
$ sudo groupadd cloud
$ sudo usermod -a -G admins,devops,cloud doe
```

or

```
$ sudo useradd -m -G sudo,developers doe
```

To verify

```
$ id doe
```

To Create a user doe with a specific group developers

```
$ sudo useradd -g developers doe
```

To Add a User doe without Home Directory

```
$ sudo useradd -M doe
```

to check

```
$ ls -l /home/doe
```

To Create a user doe with a specific shell

```
$ sudo useradd -m -s /bin/bash doe
```

To Create a User doe with Account Expiry Date

```
$ sudo useradd -e 2025-12-31 doe
```

To verify the age of the account and password of the user doe

```
$ chage -l doe
```

To Create a User doe with Password Expiry Date

```
$ sudo useradd -E 2025-12-31 doe
```

or

```
$ sudo chage -M 100 doe
```

-M 100 → password will expire after 100 days.

To verify

```
$ sudo chage -l doe
```

To Add a User doe with Custom Comments

```
$ sudo useradd -c "Welcome to foss world” doe
```

To verify

```
$ sudo tail -1 /etc/passwd
```

To Create a System User apache Without Login shell

```
$ sudo useradd -r -s /sbin/nologin -m apache
```

To check

```
$ grep apache /etc/passwd
```

To Create a system user (for services or daemons)

```
$ sudo useradd -r nginx
```

10 userdel

userdel - delete a user account and related files

syntax

```
$ userdel [OPTIONS] <username>
```

To delete a user account doe

```
$ sudo userdel doe
```

To remove the user's and home directory and mail spool of user doe

```
$ sudo userdel -r doe
```

To forcefully remove the user account doe

```
$ sudo userdel -f doe
```

To display help options

```
$ sudo userdel --help
```

To remove any SELinux(Security-Enhanced Linux) user mapping for the user's login.

```
$ sudo userdel -Z doe
```

11 usermod

usermod - modify a user account

syntax

$ usermod [options] username

To add Information to user account doe

$ sudo usermod -c "This is test message" doe

To change user doe home directory

$ sudo usermod -d /home/john/ doe

To set user account expiry date

$ sudo usermod -e 2025-12-31 doe

To change user primary group

$ sudo usermod -g devops doe

To add group to an existing user

$ sudo usermod -G web doe

To add supplementary and primary group to user

$ sudo usermod -aG wheel doe (for RHEL based systems)

$ sudo usermod -aG sudo doe (for Debian/Ubuntu based systems)

To change user login name

```
$ sudo usermod -l new_username old_username
$ sudo usermod -l john doe
```

To lock user account

```
$ sudo usermod -L doe
```

To unlock user account

```
$ sudo usermod -U doe
```

To move user home directory to new location from /home/doe to /var/doe

```
$ sudo usermod -d /var/doe/ -m doe
```

To create unencrypted password for user

```
$ sudo usermod -p passcode123 doe
```

To change user shell

```
$ sudo usermod -s /bin/sh doe
```

To change user ID

```
$ sudo usermod -u 666 doe
```

To modify the UID and GID.

```
$ sudo usermod -u 555 -g 665 doe
```

To Force User to Change Password at Next Login

```
$ sudo passwd -e doe
```

12 groupadd

groupadd - used to create a new group on the system

Basic Syntax

```
$ sudo groupadd [options] <groupname>
```

To create a group devops

```
$ sudo groupadd devops
```

To create a group devops with specific groupid

```
$ sudo groupadd devops -g 1234
```

To create a system group

```
$ sudo groupadd -r sysadmin
```

To set a custom group password

```
$ sudo groupadd devops
$ sudo gpasswd devops
```

To Create a Group with a Specific GID Range

```
$ sudo groupadd -g 5000 devops
```

To Verify Group Creation

```
$ getent group devops
```

or

```
$ grep devops /etc/group
```

To List All Groups on the System

```
$ getent group
```

or

```
$ cat /etc/group
```

To create a new group devops with group ID from 5000 to 7000

```
$ sudo groupadd devops -K GID_MIN=5000 -K GID_MAX=7000
```

To use an encrypted password for the group

```
$ sudo groupadd devops -p pa55code123!@#
```

13 groupdel

groupdel - used to delete an existing group from the system.

Basic Syntax

```
$ sudo groupdel [options] <group_name>
```

To delete a group

```
$ sudo groupdel developers
```

To verify group deletion

```
$ getent group developers
```

14 groupmod

groupmod - used to modify the properties of an existing group on the system

To change the group “devops” to “developers”

```
$ sudo groupmod -n developers devops
```

To change groupid of a group devops to 1234

```
$ sudo groupmod -g 1234 devops
```

To change the group ID with non-unique

```
$ sudo groupmod -o 0 devops
```

To change the group password of group devops

```
$ sudo groupmod -p pa55@123 devops
```

To verify group changes

```
$ getent group devops
```

or

```
$ grep devops /etc/group
```

15 groupmems

groupmems - used to add or remove users from a specific group

syntax

```
$ sudo groupmems -g GROUPNAME [options]
```

To list all members of a group developers

```
$ sudo groupmems -g developers -l
```

To make the user doe a member of the group devops

```
$ sudo groupmems -g devops -a doe
```

To delete/remove a user doc from a group devops

```
$ sudo groupmems -g devops -d doe
```

To remove multiple users from the group devops

```
$ sudo groupmems -g devops -d alice -d doe
```

To check membership after changes

```
$ sudo groupmems -g devops -l
```

To use with system groups

```
$ sudo groupmems -g sudo -a doe
```

16 gpasswd

gpasswd - administer /etc/group and /etc/gshadow

syntax

```
$ sudo gpasswd [option] group
```

To set a group password

```
$ sudo gpasswd developers
```

To add user alice to the group developers

```
$ sudo gpasswd -a alice developers
```

to check

```
$ groups alice
```

To give user alice administrative rights to the group developers

```
$ sudo gpasswd -A alice developers
```

To add multiple administrators to a group developers

```
$ sudo gpasswd -A bob,doe developers
```

To remove user alice from the group developers

```
$ sudo gpasswd -d alice developers
```

To add multiple members at once to a group developers

```
$ sudo gpasswd -M alice,bob,doe developers
```

To lock a group developers

```
$ sudo gpasswd -r developers
```

To View current group developers info

```
$ grep developers /etc/group
$ grep developers /etc/gshadow
```

17 passwd

passwd - used to change a user's password on the system

syntax

```
$ sudo passwd [options] username
```

To change system user's password

```
$ passwd
```

To change password for root

```
$ sudo passwd root
```

To set password for doe

```
$ sudo passwd doe
```

To display user status Information of doe

```
$ sudo passwd -S doe
```

To display information of all users

```
$ sudo passwd -Sa
```

To delete user's password

```
$ sudo -d doe
```

To force expire the password to the user doe , force the user to change the password in the next login

```
$ sudo passwd -e doe
```

To lock a user password

$ sudo passwd -l doe

to check

$ sudo passwd -S doe

To unlock user password

$ sudo passwd -u doe

To set Inactive days after password expiry

$ sudo -i 10 doe

to check

$ sudo passwd -S doe

To force system users to change its password in 100 number of days

$ sudo passwd -n 100 doe

To set warning days before password expiry

$ sudo passwd -w 15 doe

to check

$ sudo passwd -S doe

To set password expiry rules

$ sudo passwd -x 90 -n 10 -w 7 doe

Sets password policy for doe

-x 90: Max 90 days before password must be changed

-n 10: Min 10 days before user can change again

-w 7: Warn user 7 days before expiry

To view help options

```
$ passwd –help
```

18 chage

chage - used to view or modify user password expiration and aging settings

To view the list of options

```
$ chage -h
```

To view the account aging information of user doe

```
$ chage -l doe
```

To set the last password change date to your specified date for user doe

```
$ chage -d 2025-12-31 doe
```

To set the date when the account should expire

```
$ chage -E 2025-12-31 doe
```

To remove expiration

```
$ sudo chage -E -1 doe
```

To specify the maximum number of days between password change

```
$ chage -M 90 doe
```

Password for doe expires every 90 days

To give prior warning 7days before the password expires

```
$ chage -W 7 doe
```

To make the user account to be locked after X number of inactivity days

```
$ chage -I 10 doe
```

To set minimum password age for user doe

$ sudo chage -m 7 doe

doe must wait 7 days before changing his password again.

To Force User to Change Password at Next Login

$ sudo chage -d 0 doe

To make passwords never expire for user doe

$ sudo chage -M 99999 doe

To set both min and max together

$ sudo chage -m 3 -M 60 doe

doe must wait at least 3 days between password changes, and must change his password every 60 days

Interactive Mode

$ sudo chage doe

To enforce strict password rules for a new user doe

$ sudo chage -m 7 -M 60 -W 10 -E 2025-12-31 doe

wait 7 days between password changes

change password every 60 days

will be warned 10 days before expiry

account will expire at the end of 2025

To view all users password expiration dates

$ sudo chage -l $(cut -d: -f1 /etc/passwd)

19 chpasswd

chpasswd - update passwords in batch mode

```
$ sudo chpasswd
doe: p@ssword1
alice: p@ssword2
```

CTRL+D

storing username and password in a file and give input to chpasswd

```
$ cat > password.txt
```

doe: p@ssword1

alice: p@ssword2

then,

```
$ sudo chpasswd < password.txt
```

or

```
$ sudo cat password.txt | chpasswd
```

To apply encryption algorithm on password

```
$ sudo chpasswd -c SHA512
$ sudo chpasswd -c SHA256
$ sudo chpasswd --md5
```

To change a single users password

```
$ echo "alice:NewPassword123" | sudo chpasswd
```

To change multiple users password

```
$ echo -e "doe:Passw0rd!\nalice:MySecret123\njohn:Hello2025" |
sudo chpasswd
```

To enforce password hashing algorithms

```
$ echo "doe:Secure@123" | sudo chpasswd -c SHA512
```

To update passwords in script

```
#!/bin/bash
cat <<EOF | sudo chpasswd
alice:Password1
bob:Password2
doe:Password3
EOF
```

20 pwconv

pwconv - used to create shadow password files by moving password hashes from /etc/passwd to /etc/shadow

Converts /etc/passwd → /etc/shadow (enables shadow passwords)

If /etc/shadow was accidentally deleted or damaged

$ sudo pwconv

Recreates it using /etc/passwd entries

Reads /etc/passwd

Creates /etc/shadow if missing

Moves encrypted passwords from /etc/passwd → /etc/shadow

Replaces the password field in /etc/passwd with an x

from a sudo user delete /etc/shadow

$ sudo rm /etc/shadow

check with

$ cat /etc/shadow

To recover /etc/shadow file from the /etc/passwd file

$ sudo pwconv

and check with

$ cat /etc/shadow

21 pwunconv

pwunconv - used to remove shadow password files by moving password hashes back from /etc/shadow to /etc/passwd

Converts /etc/shadow → /etc/passwd (disables shadow passwords)

To disable shadow passwords

$ sudo pwunconv

Moves password hashes from /etc/shadow → /etc/passwd, then deletes /etc/shadow

To recover corrupted /etc/shadow

$ sudo pwunconv

22 finger

finger - used to display information about system users, such as login name, home directory, and last login time.

```
$ sudo apt install finger      # Debian/Ubuntu
$ sudo dnf install finger      # Fedora/RHEL/CentOS
```

To display information about system user and user doe

```
$ finger user_name
$ finger doe
```

To get idle status and login details of a user

```
$ finger -s doe
```

To avoid printing PGP key, plan and project details

```
$ finger -p doe
```

To show all logged-in users

```
$ finger
```

To show multiple users at once

```
$ finger doe alice john
```

To check information about a remote user

```
$ finger doe@remote_server
```

To show long details output

```
$ finger -l
```

To extract fields from finger output

```
$ finger -s | awk '{print $1, $3, $4}'
```

23 pinky

pinky - It is a lightweight version of finger that displays brief user information.

To display the information about the users

```
$ pinky
```

To Show Information for a Specific User doe

```
$ pinky doe
```

To produce the long format output

```
$ pinky -l doe
```

To Omit Column Headings

```
$ pinky -f
```

To remove the name and where column

```
$ pinky -i
```

To remove the name, idle and where columns

```
$ pinky -q
```

To Show Only Usernames

```
$ pinky -f
```

To Check Information About Users from /etc/passwd

```
$ pinky -l root
```

To Use in a Script to Count Active Users

```
$ pinky | tail -n +2 | wc -l
```

24 id

id - It displays the user ID (UID), group ID (GID), and group memberships of a user.

To print your own id without any options

```
$ id
```

To find a specific user doe id

```
$ id -u doe
```

To find a specific users GID

```
$ id -g doe
```

To find out UID and all groups associated with a username

```
$ id doe
```

To find out all the groups a user doe belongs

```
$ id -G doe
```

To display a name instead of numbers

```
$ id -nG doe
```

To display real id instead of effective id

```
$ id -r -g doe
$ id -r -G doe
```

To view information for system users

```
$ id root
```

To Check Effective User and Group IDs

```
$ id -r
```

To show another users UID only

```
$ id -u doe
```

To Display All Groups (Names Only)

```
$ id -Gn
```

To Show only user ID

```
$ id -u
```

25 who

who - It shows who is currently logged into the system

To print who command output without options

```
$ who
```

To print the heading of the columns displayed

```
$ who -H
```

To print the login names and total number of logged on users

```
$ who -q
```

To show only hostname and user associated with stdin

```
$ who -m
```

To add user's message status as +, - or ?

```
$ who -T
```

To view the time of last system boot

```
$ who -b
```

To check the current runlevel

```
$ who -r
```

To print dead processes

```
$ who -d
```

To see active processes spawned by init

```
$ who -p
```

To print last system clock change

```
$ who -t
```

To list users logged in and their processes

```
$ who -u
```

To print all available information

```
$ who -a
```

To Show only logged-in users’ names

```
$ who | awk '{print $1}' | sort | uniq
```

26 whoami

whoami - print effective userid

syntax

```
$ whoami [option]
```

To show the name of the currently logged-in user

```
$ whoami
```

To check an account for sudo permissions

```
$ sudo whoami
```

27 logname

logname - print the name of the current login user

To display user’s login name

```
$ logname
```

28 last

last - list of the most recent logins and logouts of users, as recorded in the /var/log/wtmp file

Basic usage

```
$ last
```

To list last five users logged in

```
$ last -5
```

To display without the host-name field

```
$ last -R user_name
```

To display the login and logout time including the dates

```
$ last -F
```

To display the host-name in the last column

```
$ last -a
```

To display within a specific time period.(-s) since and (-t) until

```
$ last -s yesterday -t today
```

To display information like system down entries and run level changes

```
$ last -x
```

To Show specific user doe login history

```
$ last doe
```

To Show the system reboot history

```
$ last reboot
```

To Show shutdown records

```
$ last shutdown
```

To Limit the number of results

```
$ last -n 5
```

To Display login records since a certain time

```
$ last -s 2025-11-01
```

To specify ranges

```
$ last -s 2025-11-01 -t 2025-11-07
```

29 w

w - Show who is logged on and what they are doing

Basic usage

```
$ w
```

To Display the Short Format

```
$ w -s
```

To List the w Command Output Without Printing the Header

```
$ w -h
```

To ignore usernames when calculating the current process and CPU times

```
$ w -u
```

To Display the IP Address

```
$ w -i
```

To Display in Old Style

```
$ w -o
```

To show remote hostname field

```
$ w -f
```

To Combine with grep to find a user doe

```
$ w | grep doe
```

30 users

users - print the user names of users currently logged in to the current host

Basic usage

```
$ users
```

To print the user name currently logged in the host

```
$ users
```

To Show unique users (remove duplicates)

```
$ users | tr ' ' '\n' | sort | uniq
```

To Combine with wc to count logged-in users

```
$ users | wc -w
```

31 chgrp

chgrp - change group ownership

Basic syntax

```
$ chgrp [OPTIONS] GROUP FILE...
```

To Change group ownership of a file

```
$ sudo chgrp devops project.txt
```

To change a directory example_dir group ownership

```
$ sudo chgrp devops example_dir
```

To recursively change group ownership

```
$ sudo chgrp -R devops example_dir
```

To change the group of a file to match the group of another, reference file

To change the group ownership of the file abc.file to be the same as that of the test.file

```
$ sudo chgrp --reference=test.file abc.file
```

To list the changes that happened in our example_dir directory

```
$ sudo chgrp -c -R devops example_dir
```

To describe the action or non-action taken for every File

```
$ sudo chgrp -v devops project.txt
$ sudo chgrp -v devops example_dir
```

To change the group name of link files

```
$ sudo chgrp --dereference devops symbolic_link
```

To suppress potential error messages when executing the chgrp command

```
$ sudo chgrp -f [GROUP_NAME] [DIRECTORY/FILE_NAME]
$ sudo chgrp -f devops example_dir
$ sudo chgrp -f devops project.txt
```

To Change group for multiple files

```
$ chgrp devops file1.txt file2.txt test.sh
```

To Change group of a directory

```
$ chgrp devops /var/www/html
```

To Use GID instead of group name

```
$ chgrp 1002 project.txt
```

32 setfacl

setfacl - set file access control lists

syntax

```
$ setfacl option acl_spec FILE…
```

To modify ACLs of file to give read and write permission to doe

```
$ setfacl -m u:doe rw file.txt
```

to check

```
$ getfacl file.txt
```

To remove all extended ACL entries

```
$ setfacl -b file.txt
```

To Give a group write access

```
$ setfacl -m g:devops:rw file.txt
```

To Remove a user’s ACL entry

```
$ setfacl -x u:doe file.txt
```

To remove entries from the ACL of file, To remove group 'devops' from a file's ACL

```
$ setfacl -x g:devops file.txt
```

To remove the default ACL

```
$ setfacl -k file.txt
```

To apply operations to all files and directories recursively

```
$ setfacl -m g:devops:rw -R example_directory
```

To restore a permission backup

```
$ setfacl --restore=file
```

To copy the ACL of one file to another

```
$ getfacl example.txt | setfacl --set-file=- f sample.txt
```

To Give default ACLs for new files in a directory

```
$ setfacl -d -m g:devops:rwx /project/shared
```

To view verbose mode

```
$ setfacl -v -m u:doe:rwx test.txt
```

33 getfacl

getfacl - getfacl displays the file name, owner, the group, and the Access Control List (ACL)

Basic Syntax

```
$ getfacl [OPTIONS] FILE...
```

To get the ACL's of a file

```
$ getfacl file_name.txt
```

To display the file access control list

```
$ getfacl -a file.txt
```

To View ACL of a directory

```
$ getfacl /var/www/html
```

To Display ACLs for multiple files

```
$ getfacl file1.txt file2.txt
```

To display the default access control list

```
$ getfacl -d file.txt
```

To avoid displaying comment header

```
$ getfacl --omit-header file.txt
```

To Print all effective rights comments

```
$ getfacl -e file.txt
```

To skip files that only have the base ACL entries

```
$ getfacl -s file.txt
```

To list the ACL’s recursively

```
$ getfacl -R /dir_name
```

To get the tabular output format

```
$ getfacl -t /home/doe/file.txt
```

To list the numeric user and group IDs

```
$ getfacl -n file.txt
```

To Combine with grep to find specific user permissions

```
$ getfacl file.txt | grep doe
```

To Save ACLs to a backup file

```
$ getfacl -R /project > project_acls.txt
```

Useful for backing up ACLs before making changes

later restore them with

```
$ setfacl --restore=project_acls.txt
```

34 sg

sg - execute command as different group ID

To Run a command as another group

```
$ sg developers "touch /shared/dev_notes.txt"
```

Switches to the developers group temporarily

Runs the command touch /shared/dev_notes.txt

The created file will have developers as its group owner

To execute command as different group ID

```
$ sg group-name -c 'command'
$ sg dev-group -c 'sleep 100'
$ sg admin-group -c 'ping 8.8.8.8'
```

35 login

login - used to start a new user session by authenticating the user on the system.

syntax

```
# login [OPTIONS] [USERNAME]
```

To log in to the system

```
# login
```

To log in to the system as user doe , preserves environment variables from the previous session

```
# login -p doe
```

To login to a domain

```
# login thirstyminds.in
```

To skip the second login authentication

```
# login -f -h host_name -f user_name
# login -f -h thirstyminds -f doe
```

To Switch user in a console session

```
# login alice
```

To Use with -f option (automatic login)

```
# login -f alice
```

To display help

```
# login --help
```

36 logout

logout - used to end the current user's login session and exit the shell

To logout the user from the current session from logon shell

```
$ logout
```

37 loginctl

loginctl - used to introspect and manage user logins and sessions on systems running systemd

To Show all sessions and attributes

```
$ loginctl -a
```

To display session configuration message

```
$ loginctl show-session
```

To List all current sessions

```
$ loginctl list-sessions
```

To list currently logged in users

```
$ loginctl list-users
```

To show concise runtime status information about one or more logged in users

```
$ loginctl user-status doe
```

To show properties of user doe

```
$ loginctl show-user doe
```

To Show information about a specific session

```
$ loginctl show-session 1
```

38 no-login

no-login - used to prevent a user from logging into the system by denying shell access

To Prevent a user from logging in

```
$ sudo usermod -s /sbin/nologin ftpuser
$ sudo usermod -s /sbin/nologin john
```

To check

```
$ grep ftpuser /etc/passwd
$ grep john /etc/passwd
```

create a custom message file shown by nologin

```
$ echo "Access to this system is restricted." | sudo tee /etc/nologin.txt
```

someone with /sbin/nologin tries to log in the msg will be displayed

To temporarily prevent all non-root users from logging in

```
$ sudo touch /etc/nologin
```

All non-root users will be blocked from logging in.

Root can still log in.

The message from /etc/nologin.txt (if it exists) will be displayed.

To restore normal access

```
$ rm /etc/nologin
```

To check which users are disabled

```
$ grep nologin /etc/passwd
```

39 runuser

runuser - runs a command as another user without requiring a password

syntax:

```
# runuser - username -- [commands...]
```

To run a command as another user alice without password

```
$ sudo runuser -u alice -- whoami
$ sudo runuser -u alice -- df -Th
```

To Run a script as another user

```
$ sudo runuser -u alice -- /home/alice/install.sh
```

To run multiple commands as another user

```
$ sudo runuser -u alice -- free -h; df -Th; ls -lh; uptime
```

40 su

su - used to switch to another user account, typically the superuser, by providing the user's password

To Switch to the root user

```
$ su
```

su command to switch users without a password

```
# su - alice
```

To switch to root user

```
$ su -
```

To print help options

```
$ su -h
```

To switch to a Different User

```
$ su -l doe
```

To Use su with sudo command

```
$ sudo su - alice
```

To run specific command as a different user

```
$ su -c pwd alice
$ su -c uptime alice
```

To use a different shell

```
$ su -s /usr/bin/zsh
```

To use a different user in the same environment

```
$ su -p alice
$ su -m alice
```

To switch to another user

```
$ su charlie
```

To Simulate a full login (load user environment)

```
$ su - charlie
```

or

```
$ su --login charlie
```

41 sudo

sudo - allows a permitted user to run commands with superuser or other user privileges

To run command as a root user

```
$ sudo <command>
$ sudo apt update
$ sudo dnf update
```

Edit a file that requires root privileges

```
$ sudo vim /etc/ssh/sshd_config
```

To run command as a different user

```
$ sudo -u user3 <command>
$ sudo -u user3 whoami
```

To un a root shell

```
$ sudo -i
```

Opens an interactive root shell.

Loads root’s environment (like su -).

To List user privileges with sudo

```
$ sudo -l
```

To Display privileges for another user with sudo

```
$ sudo -l -U user3
```

To add a user to the sudo group

```
$ sudo usermod -aG sudo user3
```

To add users to the sudoers file

```
$ sudo visudo
```

To run command in the background

```
$ sudo -b <command>
```

To update sudoers files

```
$ sudo -e <file>
```

To update the user’s cached credentials

```
$ sudo -v
```

To invalidate user’s cached credentials and revoke sudo access immediately

```
$ sudo -k
```

To View sudo command history

```
$ sudo cat /var/log/auth.log | grep sudo
```

41 sudoreplay

sudoreplay - replay sudo session logs

To List recorded sessions

$ sudo sudoreplay -l

Before running this command

add these lines to the end of the sudoers file

$ sudo vim /etc/sudoers

Defaults log_output

Defaults!/usr/bin/sudoreplay !log_output

Defaults!/sbin/reboot !log_output

:x Save and exit

To list sessions run by user <your_username>

$ sudo sudoreplay -l user your_username

To list sessions run by user your_user with a command containing the string vim

$ sudo sudoreplay -l user your_username command vim

Default log directories:

/var/log/sudo-io

42 setsid

setsid - runs a program in a new session, detaching it from the controlling terminal

syntax

```
$ setsid [options] command [arguments…]
```

To Run myscript.sh in the background.Even if you close the terminal, it keeps running because it’s in a separate session.

```
$ setsid myscript.sh &
```

To Run a long-running server process ex.Starts your Python server in a new session so it won’t stop when you log out.

```
$ setsid python3 server.py &
```

Run a GUI app detached from the terminal, ex. Launches Gedit independently from the terminal; closing the terminal won’t kill it.

```
$ setsid gedit &
```

43 faillog

faillog - displays or resets the failed login attempt records for users

syntax

```
$ faillog [options] [username]
```

To Show all failed login attempts

```
$ sudo faillog
```

To display the faillog records for all the users

```
$ sudo faillog -a
```

To Show failed login info for a specific user doe

```
$ sudo faillog -u doe
```

To lock an account doe for 2 minute / 120 seconds after failed login

```
$ sudo faillog -l 60 -u doe
```

To set the maximum number of login failures for user doe

```
$ sudo faillog -m 5 -u doe
```

To reset the counters of login failures of user doe

```
$ sudo faillog -r -u doe
```

To display faillog records more recent than days

```
# faillog -t 5 username
# faillog --time DAYS username
```

To display faillog record or maintains failure counters and limits

```
$ sudo faillog -u username
```

To Apply a maximum fail limit to all users

```
$ sudo faillog -m 3
```

To Unlock an account doe that’s been locked

```
$ sudo faillog -r -u doe
```

44 chfn

chfn - used to change a user's finger information like full name, office, and phone number

To Change your own finger information interactively

$ chfn

Changing the user information for klug

Enter the new value, or press ENTER for the default

Full Name: klug

Room Number [123]: 456

Work Phone [9898]: 2323

Home Phone [9999]: 4545

To Change another user’s information (as root)

$ sudo chfn doe

To change the full name on the account from alice to doe

$ sudo chfn -f doe alice

To change the work phone number on the account doe

$ sudo chfn -w 9999988888 doe

To change the room number on the account doe

$ sudo chfn -r 8888 doe

To change the home phone number on the account doe

$ sudo chfn -h 7777 doe

To change any other detail on the account doe

```
$ sudo chfn -o "7th floor room 55555" doe
```

45 mkhomedir_helper

mkdhomedir_helper - used to create a user's home directory with correct permissions during login

syntax

```
$ mkhomedir_helper [options] username
```

To Create a home directory for a user doe

```
$ sudo mkhomedir_helper doe
```

to verify

```
$ ll /home/doe/
```

To Create a home directory with a custom skeleton directory

```
$ sudo mkhomedir_helper -s /etc/custom_skel doe
```

To make a dry run

```
$ sudo mkhomedir_helper -n alice
```

46 newusers

newusers - update and create new users in batch

create users details in a file

$ sudo vim users.txt

alice:6abcd1234$xyz:1001:1001:Alice:/home/alice:/bin/bash

doe:6efgh5678$xyz:1002:1002:Doe:/home/doe:/bin/bash

:x save and exit

set the required permissions

$ sudo chmod 0600 users.txt

run the newusers command to add the users in the users.txt

$ sudo newusers users.txt

to verify check for the users added

$ cat /etc/passwd

47 grpck

grpck - used to check the integrity of the group files like /etc/group and /etc/gshadow

syntax

```
$ grpck [option] [files]
```

To Check the default group files

```
$ sudo grpck
```

To Check a group file in read-only mode

```
$ sudo grpck -r
```

To verify the group account file

```
# grpck /etc/group
```

To verify the shadow file

```
# grpck /etc/gshadow
```

Exit Codes

0: Success.

1: Syntax error.

2: One or more bad group entries found.

3: Could not open group files.

4: Could not lock group files.

5: Could not write group files.

48 vigr

vigr - edit the password, group, shadow-password or shadow-group file

To open /etc/group in an editor.

```
$ sudo vigr
```

To open /etc/gshadow (shadow group file) in an editor.

```
$ sudo vigr -s
```

49 vipw

vipw - edit the password, group, shadow-password or shadow-group file

To open /etc/passwd for editing

```
$ sudo vipw
```

To open /etc/shadow (encrypted passwords) for editing.

```
$ sudo vipw -s
```

50 wall

wall - sends a message to all logged-in users on the system

syntax

```
$ wall [OPTIONS] [<FILE>|<MESSAGE>]
```

To send a message on the terminals of all logged-in users

```
$ wall "The system will be rebooted in 20 minutes."
```

To Send a multi-line message from standard input

```
$ sudo wall
```

The server will reboot in 15 minutes.

Please log out safely.

CTRL+D

To suppress the banner and show only the text to the logged-in users

```
$ wall -n "The system will be rebooted in 20 minutes."
```

To Send a message from a file

```
$ cat /path/to/message.txt
```

The server will reboot in 15 minutes.

Please log out safely.

```
$ wall /path/to/message.txt
```

To pipe the output of another command to wall

```
$ echo "The system will be rebooted in 20 minutes. \nPlease save your work." | wall
```

To broadcast a message to a group

```
$ wall -g devops "The system will be rebooted in 20 minutes."
```

51 write

write - allows you to send a message directly to another logged-in user’s terminal

syntax

```
$ write <user> <tty name>
```

To write a message to user2 from user1

```
$ write user2
```

this is test message

^D

To pipe a message to write

```
$ echo "Hello from user1" | write user2
```

To write a message from file

```
$ cat msg.txt
```

this is test msg

```
$ write user2 < msg.txt
```

To write to specified terminal

```
$ write user2 pts/0
```

52 mesg

mesg - it allows to control write access to your terminal by other users.

To display the current write status of your terminal

```
$ mesg
```

To disallow write access to your terminal

```
$ mesg n
```

To allow write access to your terminal

```
$ mesg y
```