

System Administration 1

Prepared by Dr. Ahmed Omar

IT391

System Administration 1

Lecture 9



Managing Users and Permissions in Linux

Lecture 9

IT391

System Administration 1

Prepared by
Dr. Ahmed Omar



Agenda

- ☐ **Links, and Command Line Tools**
- ☐ **Linux Users and Groups**
- ☐ **Types of users**
- ☐ **Types of groups**
- ☐ **Gain Superuser Access**

History Command

The **history** command in Linux displays the command history for the current user session.

History

history 10 *# Show last 10 commands*

history 5 *# Show last 5 commands*

```
ahmed@debiansrv1:~$ history
```

```
1  pwd
2  cls
3  clear
4  root
5  clear
6  ls
7  tty
8  date
9  clear
10 date
11 clear
12 cal
13 calendar
14 Cal
```

History Command

The **!** (bang) operator is a powerful way to execute commands from your history.

Execute by Line Number

!1

!! *# Runs last command in history*

```
ahmed@debiansrv1:~$ history
```

```
1  pwd
2  cls
3  clear
4  root
5  clear
6  ls
7  tty
8  date
9  clear
10 date
11 clear
12 cal
13 calendar
14 Cal
```

```
ahmed@debiansrv1:~$ !1
```

```
pwd
/home/ahmed
```

```
ahmed@debiansrv1:~$
```

History Command

Search in History

When you press **Ctrl + R**, you enter reverse search mode where you can type partial text to find matching commands from your history.

Press Ctrl + R (reverse-i-search)`:

Start typing to search (reverse-i-search)

`grep': grep "^ahmed:" /etc/passwd

Press Enter to execute, or Ctrl + C to cancel

```
(reverse-i-search)`gre': grep ahmed /etc/group
```

Shell shortcuts

The **Tab** key is one of the most essential shortcuts in Linux terminals, providing **auto-completion** and **suggestion** features that dramatically improve efficiency.

da[tab] [tab] #all available options

/h[tab]/[tab] # complete the path

```
ahmed@debiansrv1:~$ da
dash date
ahmed@debiansrv1:~$ da
```

```
ahmed@debiansrv1:~$ /home/ahmed/
.cache/   Desktop/   dir3/      Downloads/ .local/    Music/     Public/    Templates/ work/
.config/  dir1/      Documents/ .gnupg/   .mozilla/ Pictures/  .ssh/     Videos/
ahmed@debiansrv1:~$ /home/ahmed/
```

Shell shortcuts

ctrl+A	Jump to the beginning of the command line.
ctrl+E	Jump to the end of the command line.
ctrl+U	Clear from the cursor to the beginning of the command line.
ctrl+K	Clear from the cursor to the end of the command line.
ctrl+LeftArrow	Jump to the beginning of the previous word on the command line.
ctrl+RightArrow	Jump to the end of the next word on the command line.
ctrl+R	Search the history list of commands for a pattern.

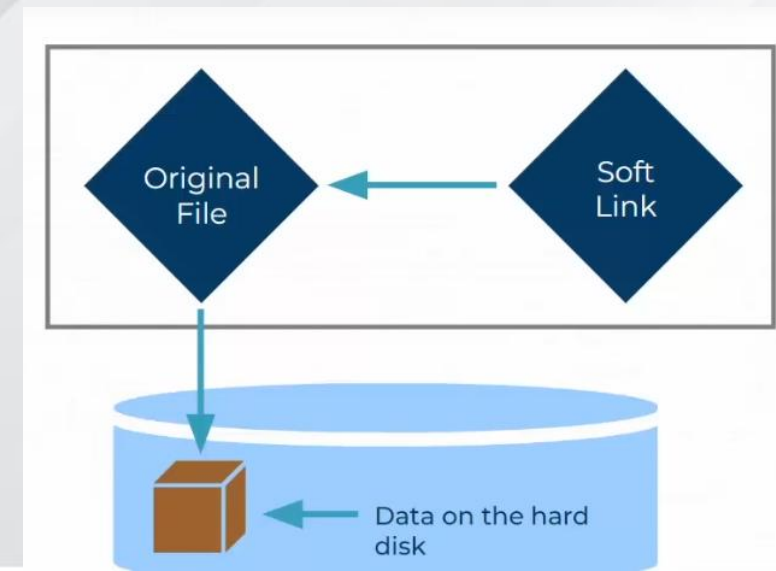
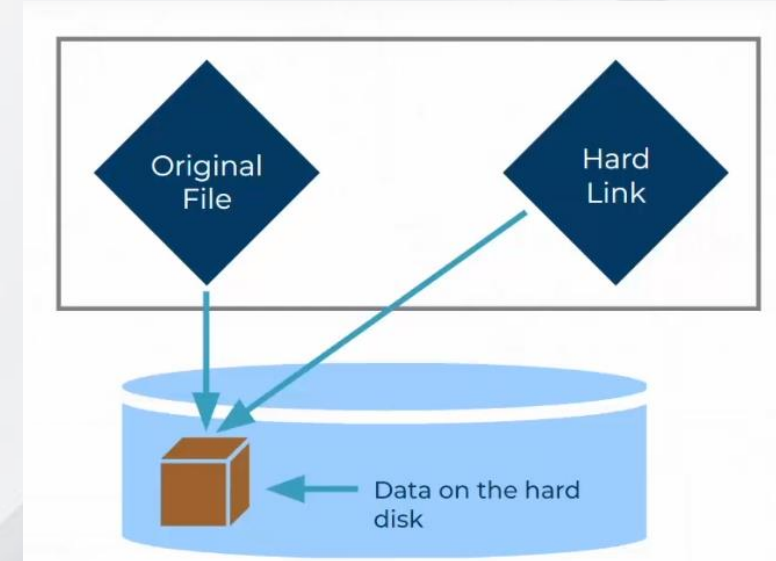
Introduction to Links in Linux

What Are Links?

Definition: Links are references to files or directories in the Linux filesystem

Purpose: They allow multiple access paths to the same file or directory

Types: Linux supports two types of links - *hard links* and *soft links* (symbolic links)



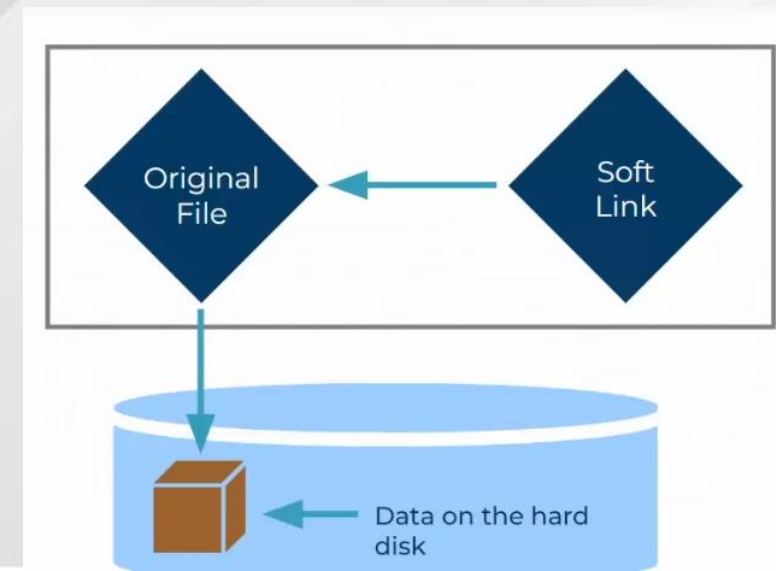
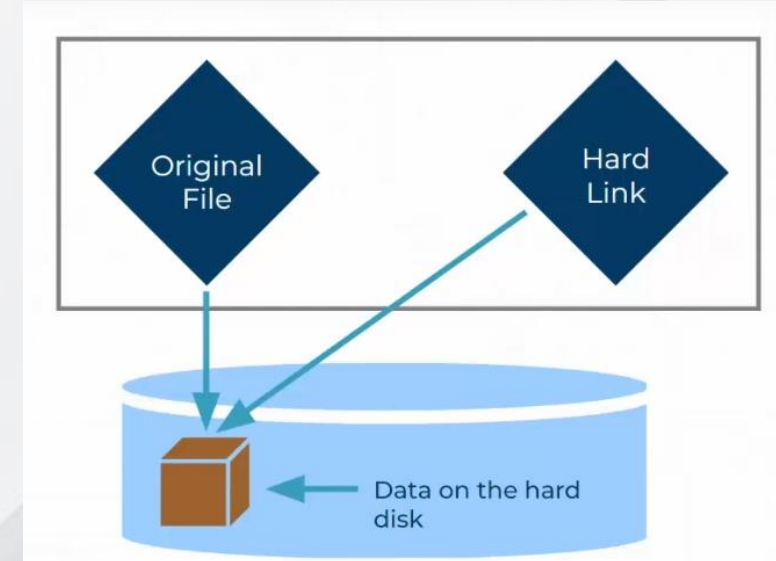
Introduction to Links in Linux

❑ Hard Link

A **hard link** is a direct reference to the file's data on the disk (inode). Different name for the same data with the same inode

❑ Soft Links (Symbolic Links)

A soft link is a special file that contains a path to another file. As shortcut in windows



Introduction to Links in Linux

❑ Create hard link

In file1 hard-link-file1

```
ahmed@debiansrv1:~$ ln file1 hard-link-file1
```

```
ahmed@debiansrv1:~$ ls -li
total 52
2007664 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Desktop
1876712 drwxr-xr-x 2 ahmed ahmed 4096 Jul 29 22:10 dir1
1876713 drwxr-xr-x 3 ahmed ahmed 4096 Jul 29 22:15 dir3
2119707 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Documents
2007665 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Downloads
1876878 -rw-r--r-- 2 ahmed ahmed 24 Jul 31 08:45 file1
1876878 -rw-r--r-- 2 ahmed ahmed 24 Jul 31 08:45 hard-link-file1
2119708 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Music
2119709 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Pictures
2119706 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Public
2007666 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Templates
2119710 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Videos
1876715 drwxr-xr-x 5 ahmed ahmed 4096 Jul 30 20:45 work
ahmed@debiansrv1:~$
```

Introduction to Links in Linux

❑ Create soft link

`ln -s file1 softlink`

Symbolic link (l at start)

```
ahmed@debiansrv1:~$ ln -s file1 softlink
```

```
ahmed@debiansrv1:~$ ls -li
```

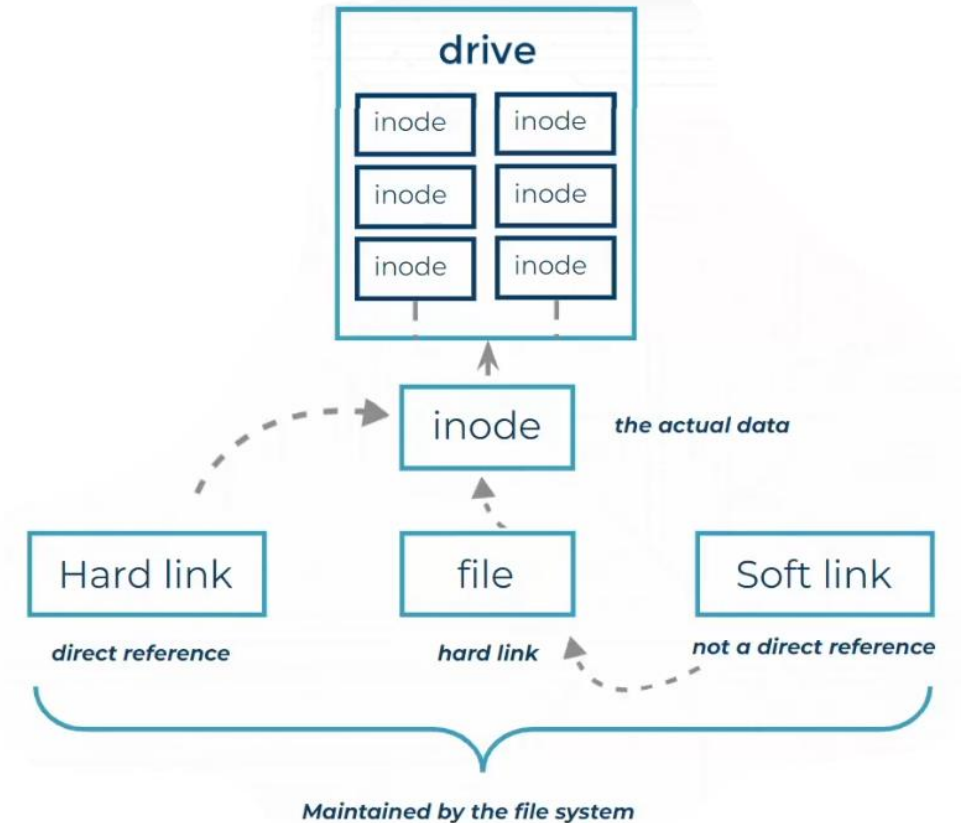
```
total 52
2007664 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Desktop
1876712 drwxr-xr-x 2 ahmed ahmed 4096 Jul 29 22:10 dir1
1876713 drwxr-xr-x 3 ahmed ahmed 4096 Jul 29 22:15 dir3
2119707 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Documents
2007665 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Downloads
1876878 -rw-r--r-- 2 ahmed ahmed 24 Jul 31 08:45 file1
1876878 -rw-r--r-- 2 ahmed ahmed 24 Jul 31 08:45 hard-link-file1
2119708 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Music
2119709 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Pictures
2119706 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Public
1876879 lrwxrwxrwx 1 ahmed ahmed 5 Aug 4 12:31 softlink -> file1
2007666 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Templates
2119710 drwxr-xr-x 2 ahmed ahmed 4096 Jul 21 19:10 Videos
1876715 drwxr-xr-x 5 ahmed ahmed 4096 Jul 30 20:45 work
```

Introduction to Links in Linux

Soft Link	Hard Link
Soft link is an alias to the original file similar to the shortcut feature in Windows OS.	Hard link is the exact replica of the original file it is pointing to.
It just contain the location to the original file but not the actual data.	It contains the actual content of the file.
Soft links have different Inode values pointing to the original value.	Hard links share the same Inode value pointing to the same file location.
Links can be created across filesystem.	Links cannot be created outside the filesystem.
The link becomes inaccessible when the original file is removed.	Changes in the hard linked file will reflect in the other files.
Soft links can link both to a file or a directory.	Hard links can only link to a file, not a directory.

What is an Inode?

An **inode** (index node) is a fundamental data structure in Unix-like filesystems that stores **metadata** about files and directories. Think of it as a "**file information card**" that the filesystem uses to manage files.



What is an Inode?

Metadata stored in inodes:

- File type (regular file, directory, symlink, etc.)
- File permissions and ownership (user, group)
- File size in bytes
- Timestamps (access, modify, change times)
- Number of hard links pointing to the file
- Pointers to data blocks where file content is stored
- Device ID and inode number

What Inodes **DON'T** Store:

- ☐ Filename (stored in directory entries)
- ☐ File content (stored in data blocks)

What is an Inode?

Show inode

ls -i

```

ahmed@debiansrv1: ~
ahmed@debiansrv1:~$ ls -i
2007664 Desktop 2119707 Documents 1876878 hard-link-file1 2119706 Public 2119710 Videos
1876712 dir1 2007665 Downloads 2119708 Music 1876879 softlink 1876715 work
1876713 dir3 1876878 file1 2119709 Pictures 2007666 Templates
ahmed@debiansrv1:~$

```

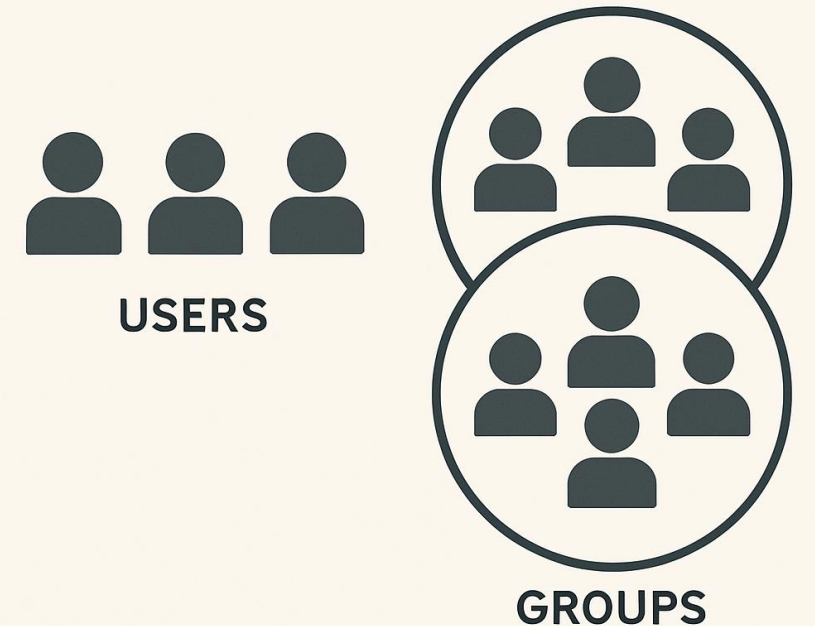
Linux Users and Groups

- ❑ Linux is a **multi-user** operating system that provides robust mechanisms for managing ***user access, permissions, and security*** through its user and group system. This fundamental aspect of Linux administration ensures that multiple users can ***safely*** and ***securely*** share system resources while maintaining appropriate levels of access control and data protection.

Linux Users and Groups

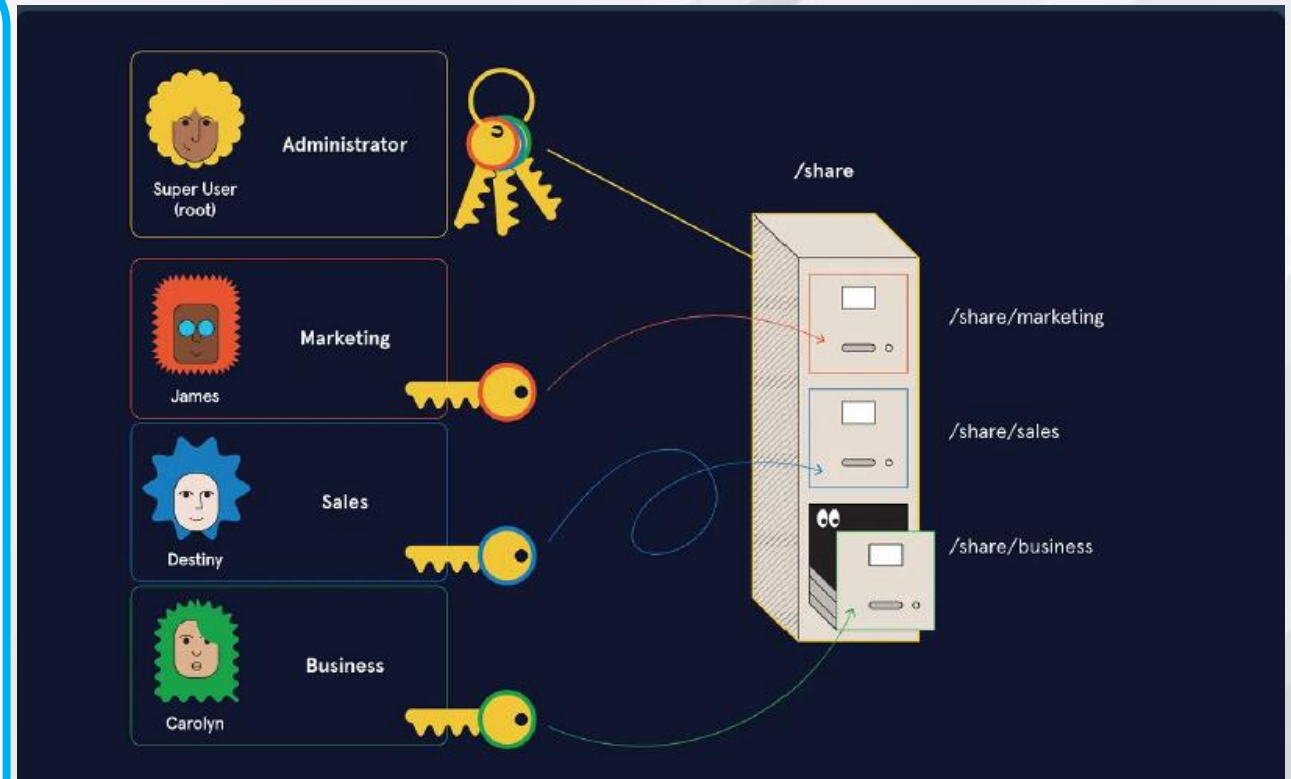
Fundamental entities for identity and access management in Linux

- ❑ **Users** represent individuals or services that interact with the system
- ❑ **Groups** organize users to simplify permission management



Linux Users and Groups

- ❑ In Linux, **users** and **groups** are fundamental concepts for managing **access control** and **permissions**.



Linux Users and Groups

The concept of users and groups in Linux serves several critical purposes.

- ❑ First, it provides **identity management** by uniquely *identifying* each person or process that interacts with the system.
- ❑ Second, it enables **access control** by determining what **resources** each user can access and what operations they can perform.
- ❑ Third, it **facilitates resource allocation** by allowing administrators to assign system resources to specific users or groups of users.
- ❑ Finally, it supports **accountability** by maintaining logs and audit trails that track user activities on the system.

Linux Users and Groups

Users

A user in Linux is an account that can log into the system and perform operations. Each user has:

- **Username:** Human-readable identifier
- **User ID (UID):** Numeric identifier assigned by the system
- **Home directory:** Personal directory (usually **/home/username**)
- **Shell:** Command interpreter (like **/bin/bash**)
- **Password:** Authentication credential

Linux Users and Groups

Types of Users

- **Root user:** Superuser with UID **0**, has complete system access
- **System users:** Used by system **services** and daemons (UIDs typically **1-999**)
- **Regular users:** Normal user accounts for people (UIDs typically **1000+**)

Linux Users and Groups

System accounts

- also known as **service accounts**, are created for system *processes*, *daemons*, and *services* that need to run with specific privileges. These accounts typically have UIDs below 1000 and are not intended for human login.
- Examples include the ***www-data*** user for web servers, the ***mysql*** user for database services, and the mail user for **mail services**. System accounts often have restricted shells or no shell at all, preventing direct login while still allowing processes to run under their identity.

Linux Users and Groups

Groups

Group accounts in Linux provide a powerful and flexible mechanism for managing permissions and access control for multiple users simultaneously.

Instead of assigning permissions to individual users, which can become cumbersome in environments with many users, groups allow administrators to define permissions once for a group and then assign users to that group. This simplifies administration, enhances security, and improves scalability.

Linux Users and Groups

Groups

A group is a collection of users that share certain permissions. Each group has:

- **Group name:** Human-readable identifier
- **Group ID (GID):** Numeric identifier
- **Member list:** Users belonging to the group

Types of Groups

- **Primary group:** Every user has one primary group (usually matches username)
- **Secondary groups:** Additional groups a user can belong to

Linux Users and Groups

Key Files

- **/etc/passwd**: Contains user account information
- **/etc/group**: Contains group information
- **/etc/shadow**: Contains encrypted passwords (restricted access)

Linux Users and Groups

Get user

id

id username

```
ahmed@debiansrv1:~$ id
uid=1000(ahmed) gid=1000(ahmed) groups=1000(ahmed),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),
44(video),46(plugdev),100(users),106(netdev),111(bluetooth),113(lpadmin),116(scanner)
ahmed@debiansrv1:~$ id root
uid=0(root) gid=0(root) groups=0(root)
ahmed@debiansrv1:~$ █
```

Linux Users and Groups

Show all users

`cat /etc/passwd`

```
ahmed@debiansrv1:~$ 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:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
tss:x:100:107:TPM software stack,,,:/var/lib/tpm:/bin/false
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:101:108::/nonexistent:/usr/sbin/nologin
usbmux:x:102:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:999:65534:dnsmasq:/var/lib/misc:/usr/sbin/nologin
avahi:x:103:112:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:104:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
fwupd-refresh:x:105:115:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
saned:x:106:117::/var/lib/saned:/usr/sbin/nologin
geoclue:x:107:118::/var/lib/geoclue:/usr/sbin/nologin
polkitd:x:996:996:polkit:/nonexistent:/usr/sbin/nologin
rtkit:x:108:119:RealtimeKit,,,:/proc:/usr/sbin/nologin
colord:x:109:120:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
gnome-initial-setup:x:110:65534::/run/gnome-initial-setup:/bin/false
Debian-gdm:x:111:121:Gnome Display Manager:/var/lib/gdm3:/bin/false
ahmed:x:1000:1000:Ahmed Omar,,,:/home/ahmed:/bin/bash
```

Linux Users and Groups

Show all users

ahmed:x:1000:1000:Ahmed Omar,,,:/home/ahmed:/bin/bash

- **ahmed** - Username/login name
- **x** - Password placeholder (actual encrypted password is in /etc/shadow)
- **1000** - User ID (UID) **1000** - Primary Group ID (GID)
- **Ahmed Omar,,,** - GECOS field (full name and optional contact info)
- **/home/ahmed** - Home directory path
- **/bin/bash** - Default shell

Linux Users and Groups

Show specific user

`grep ahmed /etc/passwd`

```
ahmed@debiansrv1:~$ grep ahmed /etc/passwd
ahmed:x:1000:1000:Ahmed Omar,,,:/home/ahmed:/bin/bash
ahmed@debiansrv1:~$ grep ali /etc/passwd
ahmed@debiansrv1:~$
```

Linux Users and Groups

Show specific user

Case-Insensitive Search

```
grep -i ahmed /etc/passwd
```

#This finds "ahmed", "Ahmed", "AHMED", etc.

```
grep -E "^(ahmed|root):" /etc/passwd # Multiple users at once
```

Linux Users and Groups

Show all groups

`cat /etc/group`

```
ahmed@debiansrv1:~$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
polkitd:x:996:
rtkit:x:119:
colord:x:120:
Debian-gdm:x:121:
ahmed:x:1000:
gnome-initial-setup:x:995:
```

Linux Users and Groups

Show specific group

#group that specific contain user

```
grep ahmed /etc/group
```

#get a specific group name

```
getent group ahmed
```

```
ahmed@debiansrv1:~$ grep ahmed /etc/group
```

```
cdrom:x:24:ahmed
```

```
floppy:x:25:ahmed
```

```
sudo:x:27:ahmed
```

```
audio:x:29:ahmed
```

```
dip:x:30:ahmed
```

```
video:x:44:ahmed
```

```
plugdev:x:46:ahmed
```

```
users:x:100:ahmed
```

```
netdev:x:106:ahmed
```

```
bluetooth:x:111:ahmed
```

```
lpadmin:x:113:ahmed
```

```
scanner:x:116:saned,ahmed
```

```
ahmed:x:1000:
```

```
ahmed@debiansrv1:~$ getent group ahmed
```

```
ahmed:x:1000:
```

```
ahmed@debiansrv1:~$
```

Gain Superuser Access

Gaining Superuser Access: means obtaining the **privileges** of the **root** user, with complete access to all commands and files.

This level of access is necessary for administrative tasks but also carries significant **risks**.

Gain Superuser Access

Privilege Separation in Linux

- ❑ Linux implements a multi-user security model with strict permission controls
- ❑ Regular users operate with limited privileges to protect system integrity
- ❑ The principle of least privilege ensures users only have access to what they need

When Elevated Privileges Are Required

- System configuration changes (network, hardware, services)
- Software installation or removal system-wide
- Access to protected system files and directories
- System maintenance and security operations

Gain Superuser Access

Running commands with sudo

In some cases, the root user's account may not have a valid password at all for security reasons.

In this case, users cannot log in to the system as root directly with a password, and **su** cannot be used to get an interactive shell. One tool that can be used to get root access in this case is **sudo**.

Unlike **su**, **sudo** normally requires users to enter their **own password** for authentication, not the password of the user account they are trying to access. That is, users who use **sudo** to run commands as root do not need to know the root password. Instead, they use their own passwords to authenticate access.

Gain Superuser Access

What is sudo?

sudo = superuser **do**

- ☐ Executes commands with temporary root privileges
- ☐ Uses current user's password, not root's
- ☐ Provides granular control over privileged operations

Security Advantages

- ☐ **Audit trail** - All sudo commands are logged
- ☐ **Password timeout** - Temporary access with automatic expiration
- ☐ **Least privilege** - Limit access to specific commands only
- ☐ **Command control** - Restrict which commands can be executed

Gain Superuser Access

Basic Syntax and Usage

- ❑ Basic syntax: **sudo command**
- ❑ Run as specific user:
sudo -u username command
- ❑ Open root shell: **sudo -i** or **sudo -s** #
without using root password
- ❑ Check available permissions: **sudo -l**

```
ahmed@debiansrv1:~$ sudo -i
[sudo] password for ahmed:
ahmed is not in the sudoers file.
ahmed@debiansrv1:~$ sudo -l
[sudo] password for ahmed:
Sorry, user ahmed may not run sudo on debiansrv1.
```

Gain Superuser Access

Add sudo permission

- Switch to root user `su -`
- Open `nano /etc/sudoers` # contains sudo permissions
- As shown sudo permissions is given for root user and sudo group

1. Write username below root user

Ahmed `ALL=(ALL:ALL) ALL`

Group privileges:

`%wheel ALL=(ALL:ALL) ALL`

2. Add the user to the sudo group

```
GNU nano 7.2 /etc/sudoers
# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@include /etc/sudoers.d
█

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

Gain Superuser Access

☐ Show sudo group users

`getent group sudo`

☐ Add user to group

`gpasswd -a ahmed sudo`

☐ Remove user from group

`gpasswd -d ali sudo`

Logout is required to confirm the updates

```
root@debiansrv1:~# getent group sudo
sudo:x:27:
root@debiansrv1:~#
```

```
root@debiansrv1:~# gpasswd -a ahmed sudo
Adding user ahmed to group sudo
root@debiansrv1:~# getent group sudo
sudo:x:27:ahmed
root@debiansrv1:~#
```

Gain Superuser Access

❑ create a new user account
`sudo useradd [options] username`
`sudo useradd ali`

```
ahmed@debiansrv1:~$ sudo useradd ali
[sudo] password for ahmed:
ahmed@debiansrv1:~$ sudo passwd ali
New password:
Retype new password:
passwd: password updated successfully
```

Gain Superuser Access

- ☐ create a new user account

- ☐ Add password

`sudo passwd ali`

This warning occurs because the user **ali** was created without a *home* directory.

```
ahmed@debiansrv1:~$ su - ali
```

```
Password:
```

```
su: warning: cannot change directory to /home/ali: No such file or directory
```

```
$ █
```

Gain Superuser Access

Create A New User Account

Common Options

- ☐ **-m** or **--create-home**: Create the user's home directory
- ☐ **-s** or **--shell**: Specify the user's login shell
- ☐ **-G** or **--groups**: Add user to supplementary groups
- ☐ **-d** or **--home-dir**: Specify home directory path
- ☐ **-c** or **--comment**: Add a comment (usually full name)
- ☐ **-u** or **--uid**: Specify user ID
- ☐ **-g** or **--gid**: Specify primary group ID

`sudo useradd -m -s /bin/bash omar`

Gain Superuser Access

Create A New User Account

```
sudo useradd -m -s /bin/bash
omar
```

```
ahmed@debiansrv1:~$ sudo useradd -m -s /bin/bash omar
ahmed@debiansrv1:~$ sudo passwd omar
New password:
Retype new password:
passwd: password updated successfully
```

```
ahmed@debiansrv1:~$ 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
```

```
ahmed:x:1000:1000:Ahmed Omar,,,:/home/ahmed:/bin/bash
ali:x:1001:1001::/home/ali:/bin/sh
omar:x:1002:1002::/home/omar:/bin/bash
```

Gain Superuser Access

Create A New User Account

The **adduser** command is a more user-friendly, interactive way to create users compared to **useradd**. Here's how to use it:

`sudo adduser username`

```
ahmed@debiansrv1:~$ sudo adduser user1
[sudo] password for ahmed:
Adding user `user1' ...
Adding new group `user1' (1003) ...
Adding new user `user1' (1003) with group `user1 (1003)' ...
Creating home directory `/home/user1' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for user1
Enter the new value, or press ENTER for the default
    Full Name []: omar ali
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
Adding new user `user1' to supplemental / extra groups `users' ...
Adding user `user1' to group `users' ...
ahmed@debiansrv1:~$
```

Gain Superuser Access

Modify User Account

The **usermod** command is used to modify existing user accounts

sudo usermod [options] username

Account Information

- ☐ **-c** : Change user's full name/comment
- ☐ **-d** : Change home directory
- ☐ **-m** : Move contents to new home directory (use with -d)
- ☐ **-s** : Change login shell
- ☐ **-u** : Change user ID
- ☐ **-g** : Change primary group

Gain Superuser Access

Modify User Account

The **usermod** command is used to modify existing user accounts

sudo usermod [options] username

Account Control

- ☐ **-L** : Lock the account
- ☐ **-U** : Unlock the account
- ☐ **-e** : Set account expiration date

Summary

- ☐ **Links, and Command Line Tools**
- ☐ **Linux Users and Groups**
- ☐ **Types of users**
- ☐ **Types of groups**
- ☐ **Gain Superuser Access**



Thank You