



ARCHCOMPUTECH

FEBRERO 2025

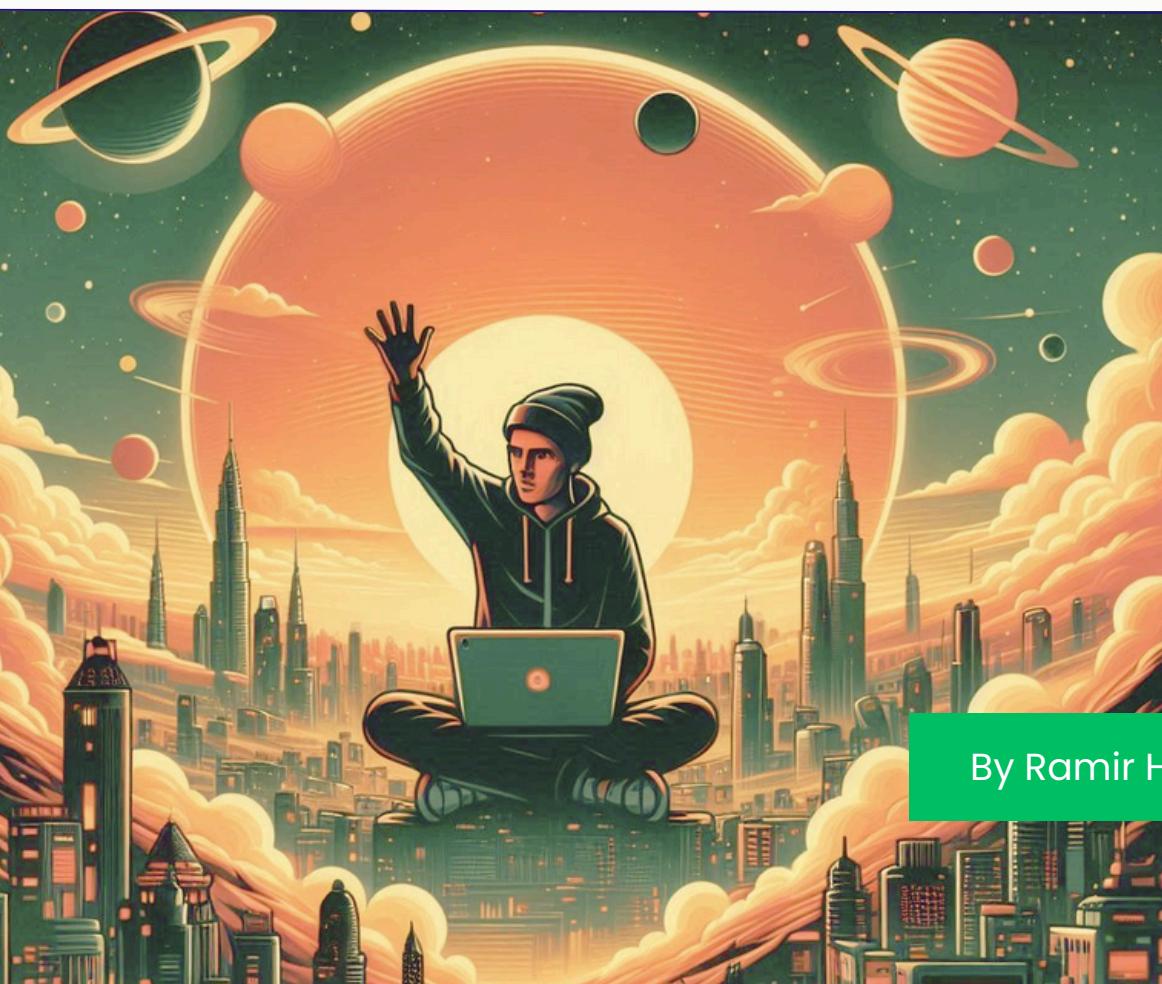
# Linux

# Basico

## VOL. 3

*Una Guía Fácil y Práctica*

Para usuarios que desean avanzar en la administración y configuración de sistemas operativos linux. Contiene información clave sobre gestión de usuarios, procesos, almacenamiento, redes, seguridad y automatización.



LINUX  
MANUAL

By Ramir Hernández

# 1. Gestión de Usuarios y Permisos Avanzados

## 1.1. Control de grupos y usuarios

- Crear usuarios con opciones avanzadas:

```
sudo useradd -m -s /bin/bash -G sudo usuario_nuevo
```

```
[kali㉿kali)-[~]
$ sudo useradd -m -s /bin/bash -G sudo usuario_nuevo
[sudo] password for kali:
```

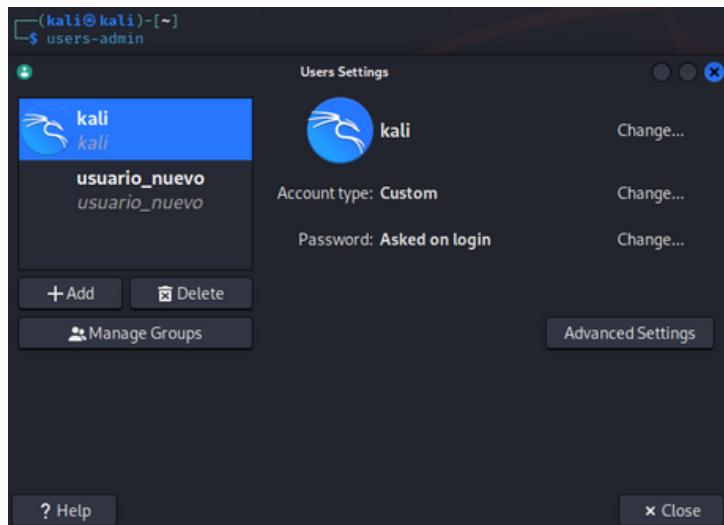
- Ver usuarios en el grupo "sudo" (usuarios con permisos de root):

```
getent group sudo   ó   grep '^sudo:.*$' /etc/group
```

```
[kali㉿kali)-[~]
$ getent group sudo
sudo:x:27:kali,usuario_nuevo
```

```
[kali㉿kali)-[~]
$ grep '^sudo:.*$' /etc/group
sudo:x:27:kali,usuario_nuevo
```

**user-admin**



- Modificar contraseña de un usuario:

```
sudo passwd usuario
```

```
[kali㉿kali)-[~]
$ sudo passwd usuario_nuevo
[sudo] password for kali:
New password:
Retype new password:
passwd: password updated successfully
```

- Crear y administrar grupos:

**sudo groupadd nombre\_grupo**

**getent group** → Mostrar grupos creados

```
└─(kali㉿kali)-[~]
$ sudo groupadd ejemplo_grupo

└─(kali㉿kali)-[~]
$ getent group | grep ejemplo
ejemplo_grupo:x:1003:
```

- agregar un usuario a un grupo

**sudo usermod -aG nombre\_grupo usuario**

- Ver grupos de un usuario:

**groups usuario**

```
└─(kali㉿kali)-[~]
$ sudo usermod -aG ejemplo_grupo usuario_nuevo

└─(kali㉿kali)-[~]
$ groups usuario_nuevo
usuario_nuevo : usuario_nuevo sudo ejemplo_grupo
```

- cambiar de usuario en la terminal

**su - usuario**

```
└─(kali㉿kali)-[~]
$ su - usuario_nuevo
Password:
(Message from Kali developers)

This is a minimal installation of Kali Linux, you likely
want to install supplementary tools. Learn how:
⇒ https://www.kali.org/docs/troubleshooting/common-minimum-setup/

(Run: "touch ~/.hushlogin" to hide this message)
└─(usuario_nuevo㉿kali)-[~]
$
```

## 1.2. Permisos Especiales (SUID, SGID, Sticky Bit)

- SUID: Permite ejecutar un programa con los permisos de su propietario.

**chmod u+s archivo**

- SGID: Permite heredar el grupo del directorio padre.

**chmod g+s directorio**

- Sticky Bit: Evita que usuarios eliminjen archivos de otros en un directorio compartido.

**chmod +t directorio**

```
(kali㉿kali)-[~/Desktop/ejemplopermisos]
└─$ ls
lsl.py

(kali㉿kali)-[~/Desktop/ejemplopermisos]
└─$ chmod u+s lsl.py
(kali㉿kali)-[~/Desktop/ejemplopermisos]
└─$ ls
lsl.py
```

## 2. Procesos y Monitorización del Sistema

### 2.1. Gestión de procesos

- Listar procesos en ejecución:

**ps aux**

```
(kali㉿kali)-[~]
└─$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
root      1  0.1  0.1 22928 13748 ?        Ss 15:08  0:02 /sbin/init splash
root      2  0.0  0.0     0   0 ?        S  15:08  0:00 [kthreadd]
root      3  0.0  0.0     0   0 ?        S  15:08  0:00 [pool_workqueue_release]
root      4  0.0  0.0     0   0 ?        I< 15:08  0:00 [kworker/R-rCU_gp]
root      5  0.0  0.0     0   0 ?        I< 15:08  0:00 [kworker/R-sync_wq]
root      6  0.0  0.0     0   0 ?        I< 15:08  0:00 [kworker/R-slub_flushwq]
root      7  0.0  0.0     0   0 ?        I< 15:08  0:00 [kworker/R-netns]
root      8  0.0  0.0     0   0 ?        T  15:08  0:00 [kworker/R-slab_alloc]
```

- Buscar un proceso específico:

**ps aux | grep nombre\_proceso**

```
(kali㉿kali)-[~]
└─$ ps aux | grep kworker
root      4  0.0  0.0    0   0 ?        I<  15:08  0:00 [kworker/R-rcu_gp]
root      5  0.0  0.0    0   0 ?        I<  15:08  0:00 [kworker/R-sync_wq]
root      6  0.0  0.0    0   0 ?        I<  15:08  0:00 [kworker/R-slub_flushwq]
root      7  0.0  0.0    0   0 ?        I<  15:08  0:00 [kworker/R-netns]
root      8  0.0  0.0    0   0 ?        I   15:08  0:00 [kworker/0:0-events]
root      9  0.0  0.0    0   0 ?        I   15:08  0:00 [kworker/0:1-events]
root     10  0.0  0.0    0   0 ?        I<  15:08  0:00 [kworker/0:0H-events_highpri]
```

- Matar procesos:

**kill -9 PID**

```
(kali㉿kali)-[~]
└─$ ps aux | grep firefox
kali      15609 73.3  3.7 11336936 332696 ?      Sl   15:33  0:10 /usr/lib/firefox-esr/firefox-es
kali      15701  1.0  0.4 217784 36624 ?      Sl   15:34  0:00 /usr/lib/firefox-esr/firefox-es
prefMapSize 236478 -appDir /usr/lib/firefox-esr/browser {fa30f845-6b26-439a-9c28-0bb471d2c34-
kali      15772  3.8  1.1 2428620 99972 ?      Sl   15:34  0:00 /usr/lib/firefox-esr/firefox-es
prefMapSize 236478 -jsInitLen 240916 -parentBuildID 20240826114832 -appDir /usr/lib/firefox-esr/b
ab
kali      15811  6.3  1.3 10844584 120616 ?      Sl   15:34  0:00 /usr/lib/firefox-esr/firefox-es
prefMapSize 236478 -jsInitLen 240916 -parentBuildID 20240826114832 -appDir /usr/lib/firefox-esr/b
ab
kali      15855  4.8  1.2 2435976 106772 ?      Sl   15:34  0:00 /usr/lib/firefox-esr/firefox-es
prefMapSize 236478 -jsInitLen 240916 -parentBuildID 20240826114832 -appDir /usr/lib/firefox-esr/b
ab
kali      15976  1.3  0.9 2402132 82664 ?      Sl   15:34  0:00 /usr/lib/firefox-esr/firefox-es
prefMapSize 236478 -jsInitLen 240916 -parentBuildID 20240826114832 -appDir /usr/lib/firefox-esr/b
ab
kali      16002  1.6  0.8 2402172 71640 ?      Sl   15:34  0:00 /usr/lib/firefox-esr/firefox-es
prefMapSize 236478 -jsInitLen 240916 -parentBuildID 20240826114832 -appDir /usr/lib/firefox-esr/b
ab
kali      16004  1.9  0.8 2402172 71820 ?      Sl   15:34  0:00 /usr/lib/firefox-esr/firefox-es
prefMapSize 236478 -jsInitLen 240916 -parentBuildID 20240826114832 -appDir /usr/lib/firefox-esr/b
ab
kali      16017  1.8  0.8 2402172 71080 ?      Sl   15:34  0:00 /usr/lib/firefox-esr/firefox-es
prefMapSize 236478 -jsInitLen 240916 -parentBuildID 20240826114832 -appDir /usr/lib/firefox-esr/b
ab
kali      16136  0.0  0.0    6396  2268 pts/3    S+   15:34  0:00 grep --color=auto firefox

(kali㉿kali)-[~]
└─$ kill -9 15609

(kali㉿kali)-[~]
└─$ ps aux | grep firefox
kali      16706  0.0  0.0    6396  2308 pts/3    S+   15:35  0:00 grep --color=auto firefox
```

- Ver procesos en tiempo real:

**top**

```
top - 15:39:39 up 30 min,  2 users,  load average: 0.02, 0.15, 0.14
Tasks: 172 total,   1 running, 171 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2.2 us,  4.6 sy,  0.0 ni, 92.3 id,  0.7 wa,  0.0 hi,  0.2 si,  0.0 st
MiB Mem :  8563.6 total,  6761.3 free,  1122.5 used,   940.8 buff/cache
MiB Swap:  1024.0 total,  1024.0 free,      0.0 used.  7441.1 avail Mem

          PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
        784 root      20   0  531960 221348  79252 S  5.0  2.5  0:50.24 Xorg
       1326 kali      20   0  458436 101440  87344 S  1.3  1.2  0:06.47 qterminal
       1166 kali      20   0  216000   3052  2656 S  0.7  0.0  0:06.82 VBoxClient
       1221 kali      20   0  985644 136300  79456 S  0.7  1.6  0:10.98 xfwm4
```

## 2.2. Administración de servicios con systemd

- Ver listado de servicios activos

**sudo systemctl list-units --type=service**

```
(root㉿kali)-[~/home/kali]
# systemctl list-units --type=service
UNIT                                     LOAD   ACTIVE SUB   DESCRIPTION
accounts-daemon.service                  loaded  active running Accounts Service
colord.service                           loaded  active running Manage, Install and Generate Color Profiles
console-setup.service                   loaded  active exited  Set console font and keymap
cron.service                            loaded  active running Regular background program processing daemon
dbus.service                            loaded  active running D-Bus System Message Bus
getty@tty1.service                      loaded  active running Getty on tty1
haveged.service                         loaded  active running Entropy Daemon based on the HAVEGE algorithm
ifupdown-pre.service                    loaded  active exited  Helper to synchronize boot up for ifupdown
keyboard-setup.service                 loaded  active exited  Set the console keyboard layout
kmod-static-nodes.service              loaded  active exited  Create List of Static Device Nodes
lightdm.service                          loaded  active running Light Display Manager
```

- Iniciar un servicio:

**sudo systemctl start servicio**

- Ver estado de un servicio:

**sudo systemctl status servicio**

```
(root㉿kali)-[~/home/kali]
# sudo systemctl start apache2.service
--(root㉿kali)-[~/home/kali]
# sudo systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled)
   Active: active (running) since Thu 2025-02-13 15:56:03 EST; 21s ago
     Invocation: 3273fc5788f41e3a8fb936a2bf85fb
       Docs: https://httpd.apache.org/docs/2.4/
      Process: 28834 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
    Main PID: 28850 (apache2)
      Tasks: 6 (limit: 10128)
     Memory: 19.9M (peak: 20.2M)
        CPU: 137ms
       CGroup: /system.slice/apache2.service
               ├─28850 /usr/sbin/apache2 -k start
               ├─28853 /usr/sbin/apache2 -k start
               ├─28854 /usr/sbin/apache2 -k start
               ├─28855 /usr/sbin/apache2 -k start
               ├─28856 /usr/sbin/apache2 -k start
               └─28857 /usr/sbin/apache2 -k start

Feb 13 15:56:02 kali systemd[1]: Starting apache2.service - The Apache HTTP Server ...
Feb 13 15:56:03 kali apachectl[28849]: AH00558: apache2: Could not reliably determine the server's fully qualified name, using 127.0.0.1 for Port 80
Feb 13 15:56:03 kali systemd[1]: Started apache2.service - The Apache HTTP Server.

Lines 1-21/21 (END)
```

- Detener un servicio:

**sudo systemctl stop servicio**

- Deshabilitar un servicio:

**sudo systemctl disable servicio**

- Habilitar un servicio en el arranque:

**sudo systemctl enable servicio**

## 3. Gestión del Almacenamiento y Discos

### 3.1. Ver información de discos

- Listar discos y particiones:

**lsblk**

- Ver espacio Libre

**df -h**

```
localhost:~# df -h
Filesystem      Size  Used Available Use% Mounted on
/dev/root       4.9G  2.3G    2.6G  47% /
devtmpfs        91.3M   0    91.3M  0% /dev
tmpfs          91.4M   8.0K   91.4M  0% /run
none           91.4M   0    91.4M  0% /dev/shm
```

- Ver el uso de tu espacio de un directorio

**du -sh "directorio"**

```
localhost:~# ls
bench.py     ejemplo      hello.c      hello.js      readme.txt
localhost:~# du -sh ejemplo/
4.0K  ejemplo/
localhost:~#
```

### 3.2 Montaje y desmontaje de unidades

- Montar una unidad manualmente:

**sudo mount /dev/sdX /mnt**

```
Disk /dev/sda: 16 GiB, 17179869184 bytes, 33554432 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x121cdc45

Device      Boot  Start    End  Sectors  Size Id Type
/dev/sda1      *     2048 320968963 320966816 15.3G 83 Linux
/dev/sda2         32096110 33552383 1462274 714M  5 Extended
/dev/sda5         32096112 33552383 1462272 714M 82 Linux swap / Solaris

Disk /dev/sdb: 7.6 GiB, 8103395328 bytes, 15826944 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x090b5ba7

Device      Boot  Start    End  Sectors  Size Id Type
/dev/sdb1      *     2048 15826943 15824896 7.6G 83 Linux
```

- Desmontar una unidad:

**sudo umount /mnt**

## 4. Redes y Conectividad

### 4.1. Configuración básica de red

- Ver configuración de red:

**ip a**

```
localhost:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 02:39:f1:ef:98:6e brd ff:ff:ff:ff:ff:ff
```

- Configurar una IP estática:

**sudo nano /etc/netplan/01-netcfg.yaml**

### 4.2. Diagnóstico de conectividad

- Probar conexión con un host

**ping 8.8.8.8**

```
localhost:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=116 time=191.100 ms
64 bytes from 8.8.8.8: seq=1 ttl=116 time=182.400 ms
```

- Resolver un dominio:

**nslookup google.com**

```
localhost:~# nslookup google.com
Server:          10.5.0.1
Address:         10.5.0.1:53

Non-authoritative answer:
Name:   google.com
Address: 2607:f8b0:4005:80e::200e
```

**traceroute google.com**

## 5. Seguridad Básica

### 5.1. Firewall con UFW

- Habilitar UFW:

Las siglas "UFW" significan "Uncomplicated Firewall" y hacen referencia a una aplicación que tiene como objetivo establecer reglas en "iptables", las tablas de firewall nativas en Linux.

**sudo ufw enable**

```
El cortafuegos está activo y habilitado en el arranque del sistema
solvetic@Solvetic:~$ sudo ufw status verbose
Estado: activo
Acceso: on (low)
Predeterminado: deny (entrantes), allow (salientes), disabled (enrutados)
Perfiles nuevos: skip

Hasta           Acción      Desde
----           ----      ----
19999/tcp       ALLOW IN   Anywhere
19999/tcp (v6)  ALLOW IN   Anywhere (v6)

solvetic@Solvetic:~$
```

**sudo ufw deny from 192.168.1.100**

## 6. Automatización con Cron y Scripts Básicos

### 6.1. Uso de Cron

En Linux, cron es un demonio que permite programar tareas para que se ejecuten automáticamente en momentos específicos. Se configura mediante el comando crontab, donde cada usuario puede definir sus propias tareas programadas.

En pocas palabras es una alarma que permite hacer cosas automáticas al sistema en un momento específico

- Editar el crontab:

**crontab -e**

```
# do daily/weekly/monthly maintenance
# min   hour   day    month   weekday command
*/15    *       *      *       *       run-parts /etc/periodic/15min
0       *       *      *       *       run-parts /etc/periodic/hourly
0       2       *      *       *       run-parts /etc/periodic/daily
0       3       *      *       6       run-parts /etc/periodic/weekly
0       5       1      *       *       run-parts /etc/periodic/monthly
```

- Crear una tarea programada:

**0 3 \* \* \* /ruta/script.sh**

## 6.2. Creación de scripts simples

Un script en Linux es un archivo de texto que contiene una serie de comandos que la computadora ejecuta en orden, como si los escribieras uno por uno en la terminal.

- Ejemplo de script básico:

```
#!/bin/bash
echo "Hola, este es un script de prueba."
```

```
GNU nano 4.9.3                               ejemplo.txt
#!/bin/bash
echo "hola, este es un script de prueba"
```

- Dar permisos de ejecución:

```
chmod +x script.sh
```

```
localhost:~# chmod +x ejemplo.txt
localhost:~# ls
bench.py      ejemplo.txt  hello.c      hello.js      readme.txt
localhost:~#
```

- Ejecutar el script

```
./script.sh
```

```
localhost:~# chmod +x ejemplo.txt
localhost:~# ls
bench.py      ejemplo.txt  hello.c      hello.js      readme.txt
localhost:~# ./ejemplo.txt
./ejemplo.txt: line 1: !#/bin/bash: not found
hola, este es un script de prueba
```



# Any questions?

Con este manual, tienes herramientas esenciales para administrar un sistema Linux con mayor eficiencia. Practicar estos comandos y conceptos te ayudará a mejorar tu fluidez en el entorno Linux.

[www.archcomputech.com](http://www.archcomputech.com)



archcomputech@gmail.com



+34 695618895



Torrejon de Ardoz, Madrid



**ARCHCOMPUTECH**