



Linux Hands-on Practice Guide

What is Linux?

Linux is primarily the kernel of an operating system. It was created by Linus Torvalds in the 1990's .

Linux is a free and open-source operating system based on UNIX. It serves as the interface between user applications and computer hardware. Designed to be stable, secure, and flexible, Linux is widely used in servers, cloud infrastructure, mobile devices, networking equipment, and embedded systems.

Why Linux?

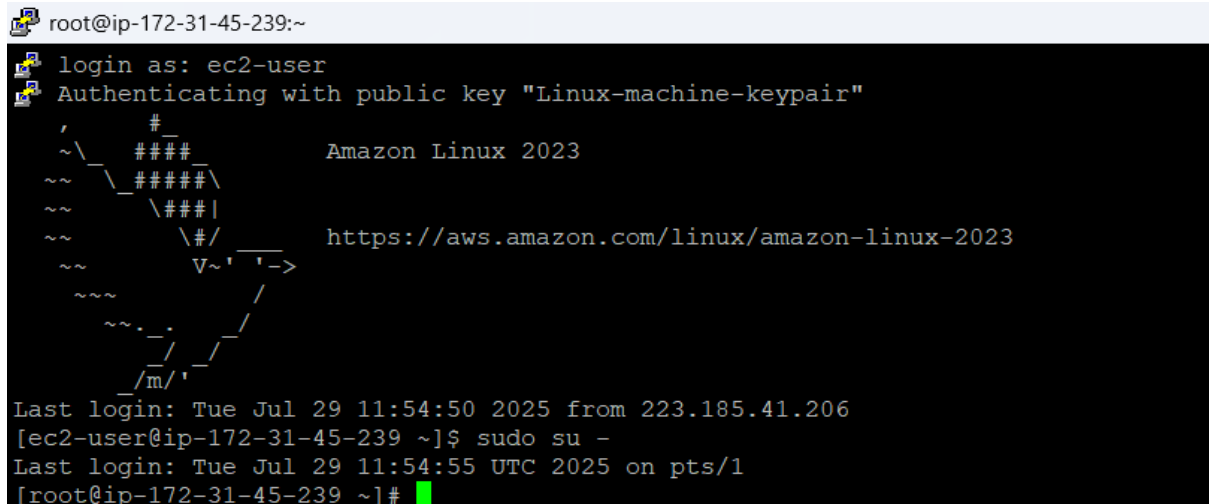
- **Security:** Built with strong permission and access control mechanisms
- **Free & Open Source:** No licensing cost; customizable source code
- **Stability & Performance:** Ideal for long-running, mission-critical systems
- **Package Ecosystem:** Offers thousands of tools, libraries, and services
- **Community & Enterprise Support:** Backed by both global communities and vendors like Red Hat, Canonical, etc.
- **Powerful CLI:** Efficient for automation, scripting, and remote administration

How Linux Works?

1. **User Interface:** Users interact via Graphical (GUI) or Command Line (CLI).
2. **Shell:** Receives user commands, interprets them, and communicates with the kernel.
3. **Kernel:** Core component that manages system resources (CPU, memory, I/O) and handles process control, file management, and hardware communication.
4. **Hardware:** The physical layer including memory, processors, disk drives, etc.

◆ Linux Overview

- Linux is a free, open-source, multi-user, multi-tasking operating system.
- It supports both GUI (Graphical User Interface) and CLI (Command Line Interface).
- Case-sensitive and security-focused with fine-grained control over user access and file permissions.



◆ Core Shell & Kernel Interaction

- `echo $SHELL` — Shows default shell
- `echo $0` — Shows current shell
- `sh`, `bash`, `csh`, `tcsh` — Switch between shell types

- `uname -r` — View kernel version

```
[root@ip-172-31-45-239 ~]# echo $SHELL
/bin/bash
[root@ip-172-31-45-239 ~]# echo $0
bash
[root@ip-172-31-45-239 ~]# uname -r
6.1.144-170.251.amzn2023.x86_64
[root@ip-172-31-45-239 ~]#
```

◆ File and Directory Operations

- Jayshri Ashok Landge



Bulk creation examples:

- touch file{1..100} — Create 100 files
- mkdir dir{1..5} — Create multiple directories

```
[root@ip-172-31-45-239 ~]# touch file{1..3}
[root@ip-172-31-45-239 ~]# mkdir dir1 dir2
[root@ip-172-31-45-239 ~]# ls -l
total 0
drwxr-xr-x. 2 root root 6 Jul 29 12:12 dir1
drwxr-xr-x. 2 root root 6 Jul 29 12:12 dir2
-rw-r--r--. 1 root root 0 Jul 29 12:12 file1
-rw-r--r--. 1 root root 0 Jul 29 12:12 file2
-rw-r--r--. 1 root root 0 Jul 29 12:12 file3
[root@ip-172-31-45-239 ~]# rm -rf *
[root@ip-172-31-45-239 ~]# ls -l
total 0
```

Screenshot 3: File & Directory Creation and Deletion Examples

◆ Permissions and Ownership

Viewing Permissions:

- ls -l — Long listing of files
- ls -lh — Human-readable format

Modify Permissions:

- chmod 755 file.sh — rwxr-xr-x
- chmod -R 700 dir/ — Recursive permission set

Ownership:

- chown user:group file — Change owner and group
- chgrp developers file — Change group only

Special Permissions:

- chmod g+s shared_folder/ — Set group ID on directory
- chmod +t /tmp — Sticky bit for secure shared directories



```
[root@ip-172-31-45-239 ~]# useradd jayshri
[root@ip-172-31-45-239 ~]# touch file1
[root@ip-172-31-45-239 ~]# ls -l
total 0
-r----- 1 root root 0 Jul 29 12:22 file1
[root@ip-172-31-45-239 ~]# chmod 644 file1
[root@ip-172-31-45-239 ~]# ls -l
total 0
-rw-r--r-- 1 root root 0 Jul 29 12:22 file1
[root@ip-172-31-45-239 ~]# chown jayshri file1
[root@ip-172-31-45-239 ~]# ls -l
total 0
-rw-r--r-- 1 jayshri root 0 Jul 29 12:22 file1
[root@ip-172-31-45-239 ~]#
```

Screenshot 4: Permissions with chmod/chown Demonstration

◆ Linux File System Hierarchy

- / — Root directory
- /home — User home directories
- /etc — System config files
- /var — Logs, variable data
- /tmp — Temporary files
- /boot — Bootloader files
- /usr — User utilities

Navigation:

- pwd — Present working directory
- cd /path — Absolute path navigation
- cd .. — Go up one directory
- cd - — Switch to previous directory

```
[root@ip-172-31-35-183 etc]# cd /
[root@ip-172-31-35-183 /]# ls
bin  dev  home  lib64  media  opt  root  sbin  sys  usr
boot  etc  lib  local  mnt  proc  run  srv  var
[root@ip-172-31-35-183 /]# cd /home
[root@ip-172-31-35-183 home]# ls
ec2-user
[root@ip-172-31-35-183 home]# cd /tmp
[root@ip-172-31-35-183 tmp]# ls
systemd-private-ed1d5c6cd7493eb7e78b691e04afd5-chronyd.service-v1HCZp
systemd-private-ed1d5c6cd7493eb7e78b691e04afd5-dbus-broker.service-CmqaVl
systemd-private-ed1d5c6cd7493eb7e78b691e04afd5-policy-routes$enX0.service-BdlLUC
systemd-private-ed1d5c6cd7493eb7e78b691e04afd5-refresh-policy-routes$enX0.service-h9J021
systemd-private-ed1d5c6cd7493eb7e78b691e04afd5-systemd-logind.service-t74jF
systemd-private-ed1d5c6cd7493eb7e78b691e04afd5-systemd-resolved.service-15686x
[root@ip-172-31-35-183 tmp]# cd /boot
[root@ip-172-31-35-183 boot]# ls
System.map-6.1.144-170.251.amzn2023.x86_64  initramfs-6.1.144-170.251.amzn2023.x86_64.img
config-6.1.144-170.251.amzn2023.x86_64      loader
efi                                          symvers-6.1.144-170.251.amzn2023.x86_64.gz
grub2                                       vmlinuz-6.1.144-170.251.amzn2023.x86_64
[root@ip-172-31-35-183 boot]# cd /var
[root@ip-172-31-35-183 var]# ls
account  cache  empty  games  lib  lock  mail  opt  run  var
adm  db  ftp  kerberos  local  log  nis  preserve  spool  yp
[root@ip-172-31-35-183 var]# cd /usr
[root@ip-172-31-35-183 usr]# ls
bin  games  include  lib  lib64  libexec  local  sbin  share  src  tmp
[root@ip-172-31-35-183 usr]# cd /etc
[root@ip-172-31-35-183 etc]# ls
DIR_COLORS  filesystems  machine-id  rsyncd.conf
DIR_COLORS.lightbgcolor  fstab  magic  rsyslog.d
```

Screenshot 5: Exploring File System with ls and cd



◆ User & Group Management

User Operations:

- `useradd devuser` — Add user
- `passwd devuser` — Set password
- `userdel -r devuser` — Remove user with home dir
- `usermod -u 1101 devuser` — Modify UID

Group Management:

- `groupadd developers`
- `usermod -aG developers devuser` — Add to group
- `gpasswd -M user1,user2 devgroup` — Set group members

Check Membership:

- `id devuser`
- `groups devuser`

Sudo Access:

- `visudo` → Add: `devuser ALL=(ALL) ALL`

```
[root@ip-172-31-43-49 ~]# useradd rohan
[root@ip-172-31-43-49 ~]# groupadd aws
[root@ip-172-31-43-49 ~]# id rohan
uid=1001(rohan) gid=1001(rohan) groups=1001(rohan)
[root@ip-172-31-43-49 ~]# useradd -G aws rohan
useradd: user 'rohan' already exists
[root@ip-172-31-43-49 ~]# usermod -G aws rohan
[root@ip-172-31-43-49 ~]# id rohan
uid=1001(rohan) gid=1001(rohan) groups=1001(rohan),1002(aws)
[root@ip-172-31-43-49 ~]# groups rohan
rohan : rohan aws
[root@ip-172-31-43-49 ~]#
```

Screenshot 6: Creating Users and Adding to Groups

◆ Text Editors (vi/vim)

Open File:

- `vi file.txt`
- `vim file.txt`

Modes:

- Press `i` for insert mode
- Press `Esc` then `:wq` to save and quit



Navigation/Editing:

- gg — Start of file
- G — End of file
- :5 — Line 5
- dd, 2dd — Delete lines
- yy, p — Copy and paste
- :%s/old/new/g — Replace text

```
root@ip-172-31-43-49:~  
HELLO ALL  
This IS JAYSHRI  
~  
~  
~  
~  
~  
-- INSERT --
```

Screenshot 7: vi Editor Usage

◆ Monitoring and Diagnostics

- top, htop — Real-time resource monitoring
- ps aux — List processes
- df -h — Disk space
- du -sh folder/ — Folder size
- free -m — Memory usage
- uptime — System load

```
[root@ip-172-31-43-49 ~]# df -h  
Filesystem      Size  Used Avail Use% Mounted on  
devtmpfs        4.0M   0  4.0M   0% /dev  
tmpfs           381M   0  381M   0% /dev/shm  
tmpfs           153M  5.1M  148M   4% /run  
/dev/xvda4      8.8G  1.9G  7.0G  21% /  
/dev/xvda3     960M  184M  777M  20% /boot  
/dev/xvda2     200M  7.1M  193M   4% /boot/efi  
tmpfs           77M    0   77M   0% /run/user/1000  
[root@ip-172-31-43-49 ~]# top  
top - 16:44:10 up 15 min,  1 user,  load average: 0.07, 0.09, 0.04  
Tasks: 110 total,  1 running, 109 sleeping,  0 stopped,  0 zombie  
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni, 96.5 id,  0.0 wa,  0.0 hi,  0.3 si,  3.2 st  
MiB Mem :  761.7 total,  221.4 free,  270.8 used,  390.6 buff/cache  
MiB Swap:   0.0 total,   0.0 free,   0.0 used,  490.9 avail Mem  


| PID  | USER     | PR | NI | VIRT   | RES   | SHR   | S | %CPU | %MEM | TIME+   | COMMAND |
|------|----------|----|----|--------|-------|-------|---|------|------|---------|---------|
| 4234 | ec2-user | 20 | 0  | 20392  | 6828  | 4992  | S | 0.3  | 0.9  | 0:00.06 | sshd    |
| 1    | root     | 20 | 0  | 174676 | 17700 | 11024 | S | 0.0  | 2.3  | 0:01.44 | systemd |


```

Screenshot 8: Output from top and df -h



◆ Package Management

RPM:

- `rpm -ivh pkg.rpm` — Install package
- `rpm -qa` — List installed packages

YUM/DNF:

- `yum install httpd -y`
- `yum remove git -y`
- `yum list installed`

```
[root@ip-172-31-43-49 ~]# yum install httpd -y
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.
Red Hat Enterprise Linux 9 for x86_64 - AppStream from RHUI (RPMs) 69 MB/s | 65 MB 00:00
Red Hat Enterprise Linux 9 for x86_64 - BaseOS from RHUI (RPMs) 70 MB/s | 69 MB 00:00
Red Hat Enterprise Linux 9 client Configuration 32 KB/s | 2.9 KB 00:00
Dependencies resolved.
=====
Package Architecture Version Repository Size
-----
Installing:
httpd x86_64 2.4.62-4.el9 rhel-9-appstream-rhui-rpms 51 k
Installing dependencies:
apr x86_64 1.7.0-12.el9_3 rhel-9-appstream-rhui-rpms 126 k
apr-util x86_64 1.6.1-23.el9 rhel-9-appstream-rhui-rpms 97 k
apr-util-bdb x86_64 1.6.1-23.el9 rhel-9-appstream-rhui-rpms 14 k
httpd-core x86_64 2.4.62-4.el9 rhel-9-appstream-rhui-rpms 1.5 M
httpd-filesystem x86_64 2.4.62-4.el9 rhel-9-appstream-rhui-rpms 15 k
httpd-tools x86_64 2.4.62-4.el9 rhel-9-appstream-rhui-rpms 86 k
mailcap 2.1.49-5.el9 rhel-9-baseos-rhui-rpms 35 k
redhat-logos-httpd noarch 90.5-1.el9_6.1 rhel-9-appstream-rhui-rpms 16 k
Installing weak dependencies:
apr-util-openssl x86_64 1.6.1-23.el9 rhel-9-appstream-rhui-rpms 17 k
mod_http2 x86_64 2.0.26-4.el9 rhel-9-appstream-rhui-rpms 167 k
mod_lua x86_64 2.4.62-4.el9 rhel-9-appstream-rhui-rpms 60 k
Transaction Summary
-----
Install 12 Packages
Total download size: 2.2 M
```

Screenshot 9: Installing Packages with YUM

◆ Cron Jobs (Scheduling)

- `crontab -e` — Edit crontab
- `crontab -l` — List jobs

Syntax:

```
# _____ minute (0 - 59)
# | _____ hour (0 - 23)
# | | _____ day of month (1 - 31)
# | | | _____ month (1 - 12)
# | | | | _____ day of week (0 - 6) (Sunday=0)
# | | | | |
# * * * * * command_to_execute
```

- Example: `05 11 * * * /home/user/script.sh`



```
[root@ip-172-31-43-49 /]# crontab -e
crontab: installing new crontab
[root@ip-172-31-43-49 /]# crontab -l
* * * * * touch /file1
[root@ip-172-31-43-49 /]# cd /
[root@ip-172-31-43-49 /]# ls
afs bin boot dev efi etc file1 home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
[root@ip-172-31-43-49 /]# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
```

Screenshot 10: Viewing and Creating Cron Jobs

◆ Hard vs Soft Links

- `ln file1 file_hard` — Hard link (same inode)
- `ln -s file1 file_soft` — Soft link (symbolic)

Check: `ls -li`

Feature	Hard Link	Soft Link (Symbolic Link)
Definition	A direct reference to the physical data (inode) on disk.	A pointer (shortcut) to the file name, not the data itself.
Works Across File Systems?	No – must be on the same filesystem.	Yes – can point across filesystems or partitions.
Points To	The file's inode.	The file's path/name.
Effect if Original File is Deleted	Data still accessible (until all hard links are removed).	Link breaks (becomes a dangling link).
Can Link to Directories?	Not usually (restricted to avoid loops).	Yes (commonly used for directories).
File Permissions	Same as the original file.	Independent of the original file's permissions.
Inode Number	Same inode as the original file.	Different inode (points to the file name).
Creation Command	<code>ln source target</code>	<code>ln -s source target</code>
Example	<code>ln file1 file2</code>	<code>ln -s file1 file2</code>

Screenshot 11: Hard vs Soft Link Comparison



◆ Networking Basics

- ip a — Show IP addresses
- hostname — Show hostname
- hostnamectl set-hostname newname
- ping google.com — Connectivity test

```
[root@ip-172-31-35-183 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enX0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 0e:43:86:4c:01:27 brd ff:ff:ff:ff:ff:ff
    altname eni-086972c2d0898c37b
    altname device-number-0.0
    inet 172.31.35.183/20 metric 512 brd 172.31.47.255 scope global dynamic enX0
        valid_lft 2984sec preferred_lft 2984sec
    inet6 fe80::c43:86ff:fe4c:127/64 scope link proto kernel_l1
        valid_lft forever preferred_lft forever
[root@ip-172-31-35-183 ~]# hostnamectl set-hostname jayshri
[root@ip-172-31-35-183 ~]# hostname
jayshri
[root@ip-172-31-35-183 ~]# ping google.com
PING google.com (216.58.215.46) 56(84) bytes of data.
64 bytes from par21s17-in-f14.1e100.net (216.58.215.46): icmp_seq=1 ttl=118 time=0.695 ms
64 bytes from par21s17-in-f14.1e100.net (216.58.215.46): icmp_seq=2 ttl=118 time=0.739 ms
64 bytes from par21s17-in-f14.1e100.net (216.58.215.46): icmp_seq=3 ttl=118 time=0.700 ms
64 bytes from par21s17-in-f14.1e100.net (216.58.215.46): icmp_seq=4 ttl=118 time=0.680 ms
```

Screenshot 12: Output of ip a and hostname

◆ ACL (Access Control List)

Set/Modify Permissions:

- setfacl -m u:devuser:rwx dir/
- setfacl -x u:devuser dir/ — Remove
- getfacl file/ — View ACL

```
[root@ip-172-31-35-183 ~]# ls -l
total 0
-rw-r--r--. 1 root root 0 Jul 29 14:00 file1
[root@ip-172-31-35-183 ~]# setfacl -m u:rohan:rwx file1
[root@ip-172-31-35-183 ~]# setfacl -m g:aws:rwx file1
[root@ip-172-31-35-183 ~]# getfacl file1
# file: file1
# owner: root
# group: root
user::rw-
user:rohan:rwx
group::r--
group:aws:rwx
mask::rwx
other::r--

[root@ip-172-31-35-183 ~]# ls -l
total 0
-rw-rwxr--+ 1 root root 0 Jul 29 14:00 file1
[root@ip-172-31-35-183 ~]# setfacl -x u:rohan file1
[root@ip-172-31-35-183 ~]# ls -l
total 0
-rw-rwxr--+ 1 root root 0 Jul 29 14:00 file1
[root@ip-172-31-35-183 ~]# getfacl file1
# file: file1
# owner: root
# group: root
user::rw-
group::r--
group:aws:rwx
mask::rwx
other::r--

[root@ip-172-31-35-183 ~]# setfacl -b file1
[root@ip-172-31-35-183 ~]# ls -l
total 0
-rw-r--r--. 1 root root 0 Jul 29 14:00 file1
[root@ip-172-31-35-183 ~]#
```

Screenshot 13: getfacl and setfacl Output



◆ Archiving and Compression

- `tar -cvf archive.tar file1 file2`
- `tar -xvf archive.tar`
- `tar -czvf archive.tar.gz dir/`
- `tar -xzvf archive.tar.gz`

```
[root@ip-172-31-35-183 ~]# touch file1 file2 file3
[root@ip-172-31-35-183 ~]# tar -cvf file4.tar file1 file2 file3
file1
file2
file3
[root@ip-172-31-35-183 ~]# ls -lh
total 12K
-rw-r--r-- 1 root root 0 Jul 29 13:29 file1
-rw-r--r-- 1 root root 0 Jul 29 13:29 file2
-rw-r--r-- 1 root root 0 Jul 29 13:29 file3
-rw-r--r-- 1 root root 10K Jul 29 13:30 file4.tar
[root@ip-172-31-35-183 ~]# gzip file4.tar
[root@ip-172-31-35-183 ~]# ls
file1 file2 file3 file4.tar.gz
[root@ip-172-31-35-183 ~]# ls -lh
total 4.0K
-rw-r--r-- 1 root root 0 Jul 29 13:29 file1
-rw-r--r-- 1 root root 0 Jul 29 13:29 file2
-rw-r--r-- 1 root root 0 Jul 29 13:29 file3
-rw-r--r-- 1 root root 142 Jul 29 13:30 file4.tar.gz
[root@ip-172-31-35-183 ~]# rm -rf file1 file2 file3
[root@ip-172-31-35-183 ~]# ls
file4.tar.gz
[root@ip-172-31-35-183 ~]# gunzip file4.tar.gz
[root@ip-172-31-35-183 ~]# ls
file4.tar
[root@ip-172-31-35-183 ~]# tar -tvf file4.tar
-rw-r--r-- root/root 0 2025-07-29 13:29 file1
-rw-r--r-- root/root 0 2025-07-29 13:29 file2
-rw-r--r-- root/root 0 2025-07-29 13:29 file3
[root@ip-172-31-35-183 ~]# ls
file4.tar
[root@ip-172-31-35-183 ~]# tar -xvf file4.tar
file1
file2
file3
[root@ip-172-31-35-183 ~]# ls
file1 file2 file3 file4.tar
[root@ip-172-31-35-183 ~]#
```

Screenshot 14: tar Archive and Extraction Commands

◆ File Copying and Moving

- `cp file1 file2`
- `cp -r dir1/ dir2/`
- `mv old.txt new.txt` — Rename
- `mv file.txt /tmp/` — Move file

```
[root@ip-172-31-35-183 etc]# touch file1
[root@ip-172-31-35-183 etc]# cat > file1
Hello All
My Name is Jayshri
[root@ip-172-31-35-183 etc]# cp file1 file2
[root@ip-172-31-35-183 etc]# cat file2
Hello All
My Name is Jayshri
[root@ip-172-31-35-183 etc]# mv file1 /tmp
[root@ip-172-31-35-183 etc]# cd /
[root@ip-172-31-35-183 /]# cd /tmp
[root@ip-172-31-35-183 tmp]# ls
file1
systemd-private-ed1d5c6c6cd7493eb7e78b691e04afd5-chronyd.service-vlHCZp
systemd-private-ed1d5c6c6cd7493eb7e78b691e04afd5-dbus-broker.service-CmqaV1
systemd-private-ed1d5c6c6cd7493eb7e78b691e04afd5-systemd-logind.service-t74jjfF
systemd-private-ed1d5c6c6cd7493eb7e78b691e04afd5-systemd-resolved.service-I5686x
[root@ip-172-31-35-183 tmp]# mv file1 newfile1.txt
[root@ip-172-31-35-183 tmp]# ls
newfile1.txt
systemd-private-ed1d5c6c6cd7493eb7e78b691e04afd5-chronyd.service-vlHCZp
systemd-private-ed1d5c6c6cd7493eb7e78b691e04afd5-systemd-logind.service-t74jjfF
systemd-private-ed1d5c6c6cd7493eb7e78b691e04afd5-systemd-resolved.service-I5686x
[root@ip-172-31-35-183 tmp]#
```

Screenshot 15: Copy and Move File Commands Output



Conclusion

In conclusion, this guide encapsulates the essential Linux system administration competencies I have developed and validated through RHCSA certification. From managing users and groups to implementing permissions and Access Control Lists (ACLs), I have demonstrated the ability to maintain secure and organized systems. My clear understanding of hard and soft links, along with proficiency in copying and moving files and directories, reflects strong command over core file operations. Additionally, my knowledge of package management using YUM and RPM, job scheduling with crontab, and expertise in file compression and archiving techniques illustrate a comprehensive grasp of practical Linux administration tasks.

These skills collectively underscore my readiness to take on advanced server management, scripting, and automation responsibilities—establishing a strong, reliable foundation for future growth in professional Linux and DevOps environments.