

Exploring Linux File Handling Utilities

Discover the power and efficiency of Linux file handling utilities, essential tools for managing and manipulating files seamlessly.

File Operations	
ls	Lists files - both regular & hidden files and their permissions as well.
pwd	Displays the current directory path
mkdir 'directory_name'	Creates a new directory
rm file_name	Removes a file
rm -f filename	Forcefully removes a file
rm -r directory_name	Removes a directory recursively
cp file1 file2	Copies the contents of file1 to file2
cp -r dir1 dir2	Recursively Copies dir1 to dir2. dir2 is created if it does not exist
mv file1 file2	Renames file1 to file2
ln -s /path/to/file_name link_name	Creates a symbolic link to file_name
touch file_name	Creates a new file
cat > file_name	Places standard input into a file
more file_name	Outputs the contents of a file
head file_name	Displays the first 10 lines of a file
tail file_name	Displays the last 10 lines of a file
gpg -c file_name	Encrypts a file
gpg file_name.gpg	Decrypts a file
wc	Prints the number of bytes, words and lines in a file
xargs	Executes commands from standard input
Process Related	
ps	Display currently active processes
ps aux grep 'telnet'	Searches for the id of the process 'telnet'
pmap	Displays memory map of processes
top	Displays all running processes
kill pid	Terminates process with a given pid
killall proc	Kills / Terminates all processes named proc
pskill process-name	Sends a signal to a process with its name
bg	Resumes suspended jobs in the background
fg	Brings suspended job to the foreground
File Permissions	
chmod octal filename	=> Change file permissions of the file to octal
Example	
chmod 777 /data/test.c	=> Set rwx permissions to owner, group and everyone (everyone else who has access to the server)
chmod 755 /data/test.c	=> Set rwx to the owner and r_x to group and everyone
chmod 766 /data/test.c	=> Sets rwx for owner, rw for group and everyone
chown owner:group file_name	=> Change ownership of the file
chown owner:owner-group file_name	=> Change owner and group owner of the file
chown owner:owner-group:owner-group directory	=> Change owner and group owner of the directory
Network	
ip addr show	=> Displays IP addresses and all the network interfaces
ip address add 192.168.0.1/24 dev eth0	=> Assigns IP address 192.168.0.1 to interface eth0
ifconfig	=> Displays IP addresses of all network interfaces
ping host	=> ping command sends an ICMP echo request to establish a connection to server / PC
whois domain	=> Retrieves more information about a domain name
dig domain	=> Retrieves DNS information about the domain
dig +x host	=> Performs reverse lookup on a domain
host google.com	=> Performs an IP lookup for the domain name
hostname -i	=> Displays local IP address
wget file_name	=> Downloads a file from an online source
netstat -ntu	=> Displays all active listening ports
Compression / Archives	
tar -cf home.tar home	=> Creates archive file called 'home.tar' from file 'home'
tar -xf files.tar	=> Extract archive file 'files.tar'
tar -zcvf home.tar.gz source-folder	=> Creates gzipped tar archive file from source folder
gzip file	=> Compression a file with .gz extension
Install Packages	
rpm -i pkg_name rpm	=> Install an rpm package
rpm -e pkg_name	=> Removes an rpm package
dnf install pkg_name	=> Install package using dnf utility
Install Source (Compilation)	
/configure	
make	
make install	
Search	
grep 'pattern' files	=> Search for a given pattern in files
grep -r 'pattern' dir	=> Search recursively for a pattern in a given directory
locate file	=> Find all instances of the file
find /home/ -name "index"	=> Find file names that begin with 'index' in /home folder
find /home/ -size +10000k	=> Find files greater than 10000k in the home folder
Login	
ssh user@host	=> Securely connect to host as user
ssh -p port_number user@host	=> Securely connect to host using a specified port
ssh host	=> Securely connect to the system via SSH default port 22
telnet host	=> Connect to host via telnet default port 23
File Transfer	
scp file1.txt server2/tmp	=> Securely copy file1.txt to server2 in /tmp directory
rsync -a /home/apps /backup/	=> Synchronize contents in /home/apps directory with /backup directory
Disk Usage	
df -h	=> Displays free space on mounted systems
df -i	=> Displays free inodes on filesystems
fdisk -l	=> Shows disk partitions, sizes, and types
du -sh	=> Displays disk usage in the current directory in a human-readable format
findmnt	=> Displays target mount point for all filesystems
mount device-path	=> Mount a device

History of Linux File Handling Utilities

1 Development and Evolution

Trace the origins and growth of Linux file handling utilities, showcasing key milestones and versions.

2 Revolutionary Functionality

Learn how Linux utilities have revolutionized file management on the command line.

3 Widespread Adoption

Explore the global impact and popularity of Linux file handling utilities in the tech community.

Key Features of Linux File Handling Utilities

cp (copy)

Easily duplicate files or entire directories while preserving their structure.

mv (move)

Relocate files and directories with ease, maintaining file integrity and permissions.

rm (remove)

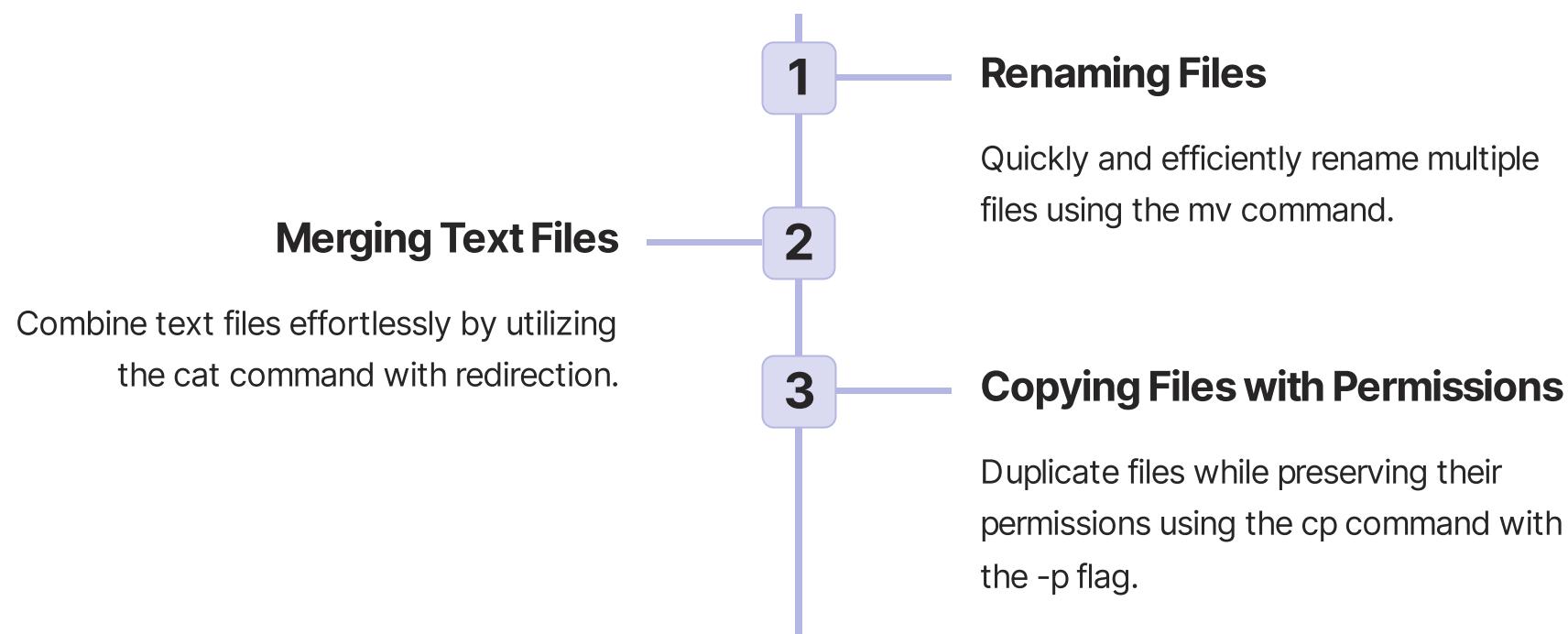
Delete files and directories securely, protecting against accidental data loss.

cat (concatenate)

Combine files or display their contents, providing flexible file viewing options.



Practical Examples of Linux File Handling Utilities



Best Practices for Efficient File Handling

Organize: Maintain a Well-Structured Directory

Create a logical directory structure to facilitate easy file navigation and retrieval.

Backup: Safeguard Your Data

Regularly back up critical files to prevent data loss in case of unexpected events.

Automation: Harness the Power of Scripts

Automate repetitive tasks with scripting, saving time and effort in file handling.

Version Control: Track File Changes

Utilize version control systems to track changes and collaborate effectively.

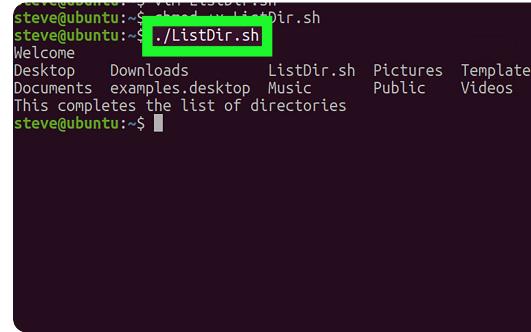


Advanced Features of Linux File Handling Utilities



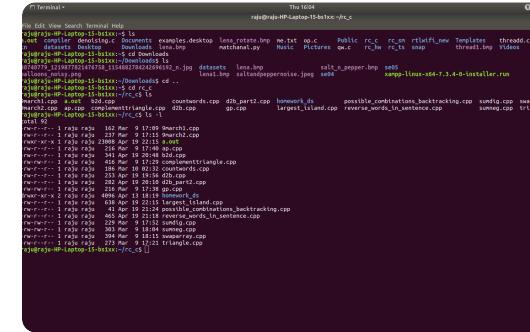
Regular Expressions

Master the art of pattern matching and advanced file selection using regular expressions.



Shell Scripting

Unlock the true potential of Linux file handling utilities with powerful scripts and automation.



Piping and Redirection

Combine commands and redirect output to create custom file handling workflows.

Conclusion

1 Enhanced Productivity

Supercharge your file management skills with Linux file handling utilities, saving time and boosting efficiency.

2 Versatility and Flexibility

From basic file operations to advanced automation, Linux utilities offer endless possibilities.

3 Continuous Learning

Embrace the Linux philosophy and keep exploring new ways to optimize your file handling workflows.

