File handling utilities in Linux

Welcome to the world of Linux file handling utilities! In this presentation, we will explore essential commands and techniques for efficient file management in Linux.

AI by AI DS

LINUX COMMANDS CHEAT SHEET

SYSTEM

```
uname -a              =>Displaylinux system information
uname -r              =>Display kernel release information
uptime                =>Show how long the system has been running + load
hostname              =>Show system host name
hostname -i           =>Display the IP address of the host
last reboot           =>Show system reboot history
date                  =>Show the current date and time
cal                   =>Show this month calendar
w                     =>Display who is online
whoami                =>Who you are logged in as
finger user           =>Display information about user
```

HARDWARE

```
dmesg                 =>Detected hardware and boot messages
cat /proc/cpuinfo     =>CPU model
cat /proc/meminfo     =>Hardware memory
cat /proc/interrupts  =>Lists the number of interrupts per CPU per I/O device
lshw                  =>Displays information on hardware configuration of
                        the system
lsblk                 =>Displays block device related information in Linux
free -m               =>Used and free memory (-m for MB)
lspci -tv             =>Show PCI devices
lsusb -tv             =>Show USB devices
dmidecode             =>Show hardware info from the BIOS
hdparm -i /dev/sda    =>Show info about disk sda
hdparm -tT /dev/sda   =>Do a read speed test on disk sda
badblocks -s /dev/sda =>Test for unreadable blocks on disk sda
```

USERS

```
id                    =>Show the active user id with login and group
last                  =>Show last logins on the system
who                   =>Show who is logged on the system
groupadd admin        =>Add group "admin"
useradd -c "Sam Tomshi"    =>g admin -m sam #Create user "sam"
userdel sam           =>Delete user sam
adduser sam           =>Add user "sam"
usermod               =>Modify user information
```

FILE COMMANDS

```
ls –al                =>Display all information about files/ directories
pwd                   =>Show the path of current directory
mkdir directory-name  =>Create a directory
rm file-name          =>Delete file
rm -r directory-nam   =>Delete directory recursively
rm -f file-name       =>Forcefully remove file
rm -rf directory-name =>Forcefully remove directory recursively
cp file1 file2        =>Copy file1 to file2
cp -r dir1 dir2       =>Copy dir1 to dir2, create dir2 if it doesn't exist
mv file1 file2        =>Rename source to dest / move source to directory
ln –s /path/to/file-name link-name   #Create symbolic link to file-name
touch file            =>Create or update file
cat > file            =>Place standard input into file
more file             =>Output contents of file
head file             =>Output first 10 lines of file
tail file             =>Output last 10 lines of file
tail -f file          =>Output contents of file as it grows starting with the
                        last 10 lines
gpg -c file           =>Encrypt file
gpg file.gpg          =>Decrypt file
wc                    =>print the number of bytes, words, and lines in files
xargs                 =>Execute command lines from standard input
```

PROCESS RELATED

```
ps                    =>Display your currently active processes
ps aux | grep 'telnet'  =>Find all process id related to telnet process
pmap                  =>Memory map of process
top                   =>Display all running processes
kill pid              =>Kill process with mentioned pid id
killall proc          =>Kill all processes named proc
pkill process-name    =>Send signal to a process with its name
bg                    =>Resumes suspended jobs without bringing them to
                        foreground
fg                    =>Brings the most recent job to foreground
fg n                  =>Brings job n to the foreground
```

FILE PERMISSION RELATED

```
chmod octal file-name       =>Change the permissions of file to octal
Example
chmod 777 /data/test.c      =>Set rwx permission for owner,group,world
chmod 755 /data/test.c      =>Set rwx permission for owner,rx for group
                              and world
chown owner-user file       =>Change owner of the file
chown owner-user:owner-group file-name   =>Change owner and group
                                           owner of the file
chown owner-user:owner-group directory   =>Change owner and group
                                           owner of the directory
```

NETWORK

```
ip addr show          =>Display all network interfaces and ip address
                        (a iproute2 command,powerful than ifconfig)
ip address add 192.168.0.1 dev eth0   =>Set ip address
ethtool eth0          =>Linux tool to show ethernet status
mii-tool eth0         =>Linux tool to show ethernet status
ping host             =>Send echo request to test connection
whois domain          =>Get who is information for domain
dig domain            =>Get DNS information for domain
dig -x host           =>Reverse lookup host
host google.com       =>Lookup DNS ip address for the name
hostname –i           =>Lookup local ip address
wget file             =>Download file
netstat -tupl         =>Listing all active listening ports
```

COMPRESSION / ARCHIVES

```
tar cf home.tar home  =>Create tar named home.tar containing home/
tar xf file.tar       =>Extract the files from file.tar
tar czf file.tar.gz files  =>Create a tar with gzip compression
gzip file             =>Compress file and renames it to file.gz
```

INSTALL PACKAGE

```
rpm -i pkgname.rpm    =>Install rpm based package
rpm -e pkgname        =>Remove package
```

INSTALL FROM SOURCE

```
./configure
make
make install
```

SEARCH

```
grep pattern files    =>Search for pattern in files
grep -r pattern dir   =>Search recursively for pattern in dir
locate file           =>Find all instances of file
find /home/tom -name 'index*'   =>Find files names that start with "index"
find /home -size +10000k        =>Find files larger than 10000k in /home
```

LOGIN (SSH AND TELNET)

```
ssh user@host         =>Connect to host as user
ssh -p port user@host =>Connect to host using specific port
telnet host           =>Connect to the system using telnet port
```

FILE TRANSFER

```
scp
scp file.txt server2:/tmp     =>Secure copy file.txt to remote host /tmp folder
rsync
rsync -a /home/apps /backup/  =>Synchronize source to destination
```

DISK USAGE

```
df –h                 =>Show free space on mounted filesystems
df -i                 =>Show free inodes on mounted filesystems
fdisk -l              =>Show disks partitions sizes and types
du -ah                =>Display disk usage in human readable form
du -sh                =>Display total disk usage on the current directory
findmnt               =>Displays target mount point for all filesystem
mount device-path mount-point   =>Mount a device
```

DIRECTORY TRAVERSE

```
cd ..                 =>To go up one level of the directory tree
cd
cd /test
```

MORE DETAILED : HTTP://LINOXIDE.COM/GUIDE/LINUX-COMMAND-SHELF.HTML

Introduction to File Handling Commands

What is File Handling?

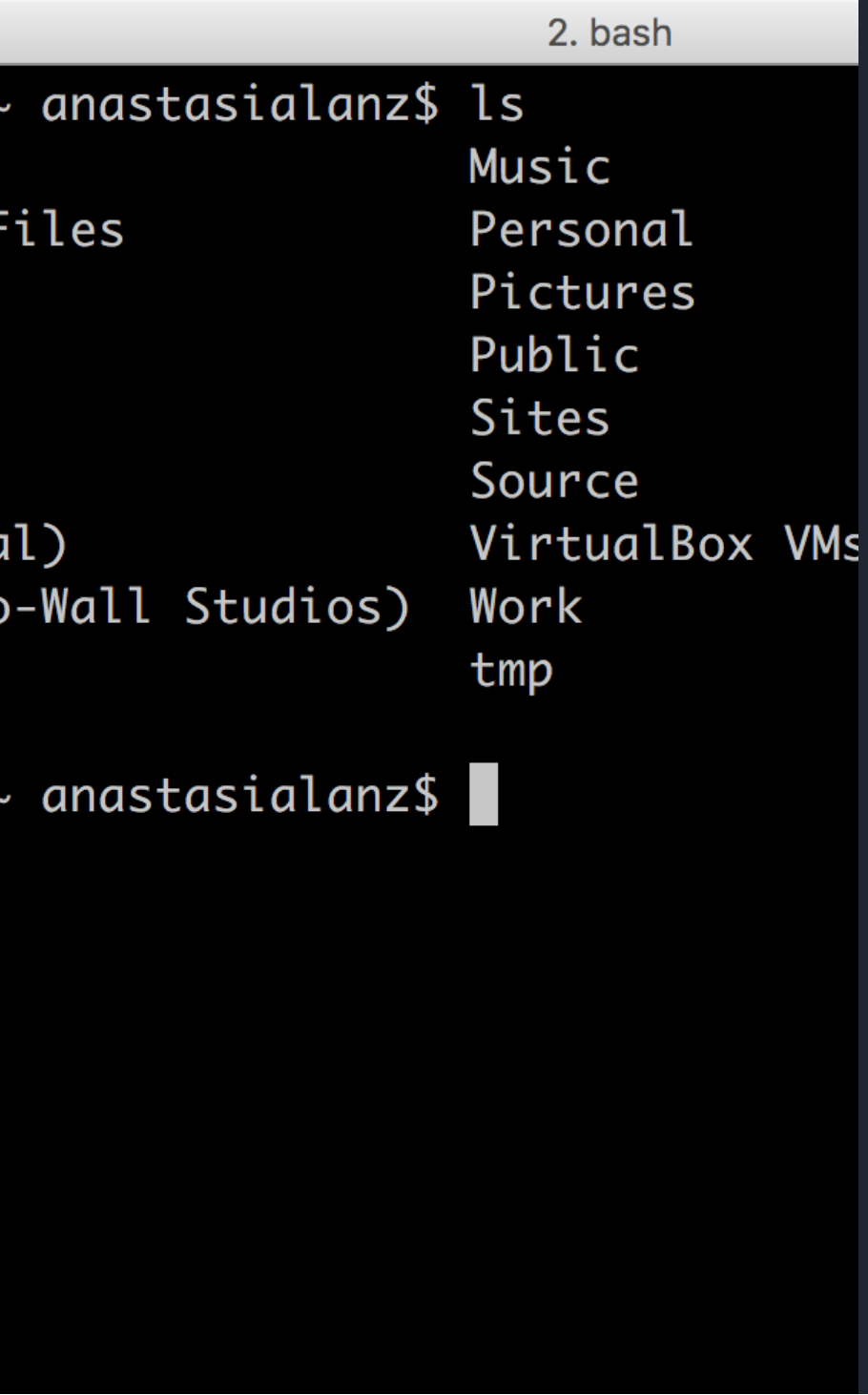
In Linux, file handling refers to the ability to create, manipulate, and organize files and directories using various commands.

Why is it Important?

Mastering file handling commands is crucial for efficient navigation, file transfer, data backup, and system administration tasks.

The Command Line Interface

File handling commands are executed in the Terminal, providing a powerful and flexible way to interact with the Linux operating system.



Common File Handling Commands

1 ls - List Files

Display the contents of a directory, including files and subdirectories, in a formatted manner.

2 cd - Change Directory

Navigate between directories and change your current working directory.

3 cp - Copy Files

Create copies of files or directories in Linux.

4 mv - Move Files

Move files or directories to a new location or rename them.

Creating and Manipulating Directories

1

mkdir - Make Directory

Create a new directory using the specified name.

2

rmdir - Remove Directory

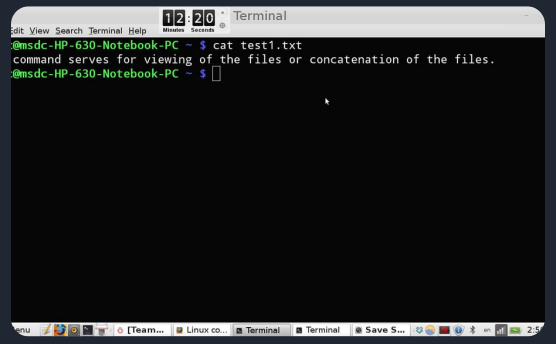
Delete an empty directory from the file system.

3

manipulating directories

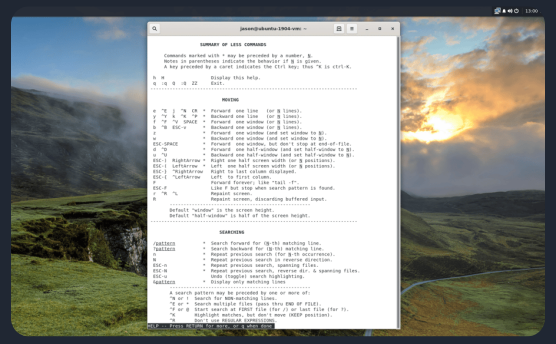
Explore various operations for managing directories, including renaming, moving, copying, and deleting.

Viewing File Contents



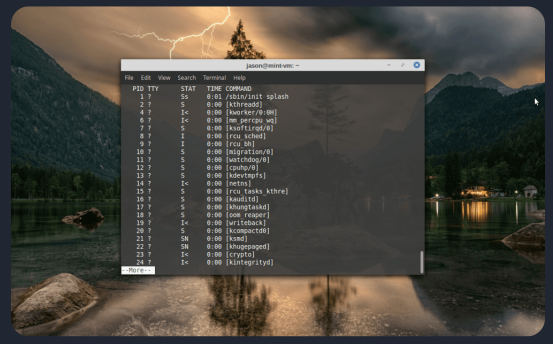
cat - Concatenate Files

Display the content of a file on the terminal. Can also be used to combine and create files.



less - Pager for File Viewing

View large files interactively, enabling smooth navigation and backward/forward scrolling.



more - Pager for Long Output

Display the content of a file one page at a time, allowing easy navigation through long outputs.

Redirecting Input and Output

> - Redirect Output

Save the output of a command to a file. Overwrites the file if it already exists.

>> - Append Output

Append the output of a command to a file. Creates the file if it doesn't exist, or appends to the existing file.

< - Read from a File

Redirect the input of a command from a file instead of the standard input stream.

Searching for Files

1

find - Search for Files

Recursively search for files and directories based on various criteria such as name, size, or type.

2

locate - Quickly Find Files

Quickly find files by searching through a database containing the names and locations of files on the system.

3

Search Strategies

Mastering efficient file search techniques is essential for effectively locating and managing files in Linux.

Archiving and Compressing Files

1 tar - Archiving Utility

Create an archive (tarball) consisting of one or more files and directories.

2 gzip - File Compression

Compress files using the gzip compression algorithm to reduce their size.

3 zip - Archive and Compression

Archive and compress files and directories using the ZIP compression format.

Syntax for Linux File Utilities Commands

Linux file utilities commands are powerful tools for managing files and directories. Below are some common commands:

List Files

To list files, use the 'ls' command followed by options like '-l' for long format or '-a' to show hidden files.

Example:

```
ls -l
```

Output:

```
total 8
-rw-r--r-- 1 user group 1234 Jan 1 10:00 file1.txt
-rw-r--r-- 1 user group 5678 Jan 1 11:00 file2.txt
```

Copy Files

To copy files, use the 'cp' command followed by the source file and destination directory.

Example:

```
cp file1.txt /path/to/destination
```

Output:

The file 'file1.txt' is copied to the directory '/path/to/destination'.

Move or Rename Files

To move or rename files, use the 'mv' command.

Example:

```
mv file1.txt /path/to/new/location/newname.txt
```

Output:

The file 'file1.txt' is moved to the directory '/path/to/new/location' and renamed as 'newname.txt'.

Remove Files or Directories

To remove files or directories, use the 'rm' command.

Example:

```
rm file1.txt
```

Output:

The file 'file1.txt' is removed.

Create Directories

To create directories, use the 'mkdir' command followed by the directory name.

Example:

```
mkdir new_directory
```

Output:

A new directory named 'new_directory' is created.

Find Files

To find files matching certain criteria, use the 'find' command followed by the search path and options.

Example:

```
find /path/to/search -name "*.txt"
```

Output:

Lists all files with the extension '.txt' in the specified search path.

Advanced Linux File Operations

Explore these advanced Linux file operations to enhance your file management skills:

Change Permissions

Use the 'chmod' command to change file permissions for read, write, and execute access.

Example:

```
chmod 755 file.txt
```

Output:

The file 'file.txt' now has read, write, and execute permissions for the owner, and read and execute permissions for group and others.

Compress and Extract

Compress files into archives using 'tar' command with options like '-czvf' for creating a gzip compressed archive.

Example:

```
tar -czvf archive.tar.gz file1.txt file2.txt
```

Output:

The files 'file1.txt' and 'file2.txt' are compressed into the archive 'archive.tar.gz'.

Search within Files

Use the 'grep' command to search for specific text patterns within files.

Example:

```
grep "pattern" file.txt
```

Output:

The command displays lines containing the specified pattern in the file 'file.txt'.

File Permissions

View file permissions and ownership using the 'ls' command with the '-l' option.

Example:

```
ls -l file.txt
```

Output:

The command shows the permissions, owner, group, and other details of the file 'file.txt'.

File Comparison

Compare the contents of two files using the 'diff' command.

Example:

```
diff file1.txt file2.txt
```

Output:

The command displays the differences between the contents of 'file1.txt' and 'file2.txt'.

Advanced C and Python File Operations

Take your file handling skills to the next level with these advanced commands:

Compile C Code

Compile C code using the 'gcc' command to create an executable file.

Example:

```
gcc -o program program.c
```

Output:

The C code in 'program.c' is compiled and an executable file named 'program' is created.

Execute C Program

Run a compiled C program using the './' followed by the program name.

Example:

```
./program
```

Output:

The compiled C program 'program' is executed and its output is displayed.

Run Python Script

Execute a Python script using the 'python' command followed by the script name.

Example:

```
python script.py
```

Output:

The Python script 'script.py' is executed and its output is displayed.

Debug C Code

Use the 'gdb' command to debug C code and analyze its execution.

Example:

```
gdb program
```

Output:

The 'gdb' debugger is launched for the C program 'program', allowing you to step through the code and inspect variables.

Python Code Profiling

Profile Python code using the 'cProfile' module to analyze its performance.

Example:

```
python -m cProfile script.py
```

Output:

The Python script 'script.py' is executed with profiling information displayed, helping identify performance bottlenecks.