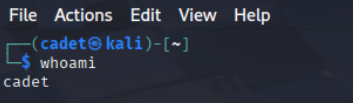
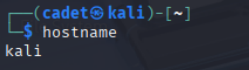
1. **System Commands:**
2. whoami:

* Shows the current logged-in username.



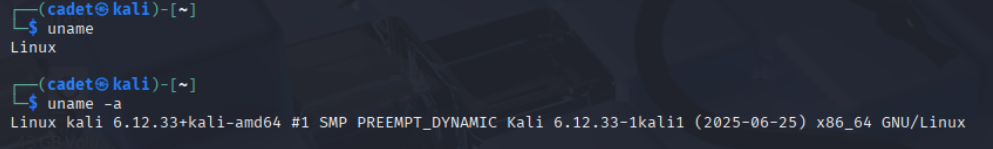
1. hostname:

* Displays the system’s hostname.



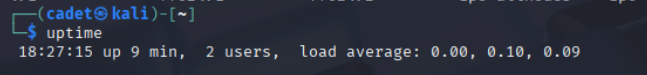
1. uname:

* Shows system information (use uname -a for all details).



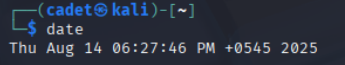
1. uptime:

* Displays how long the system has been running.

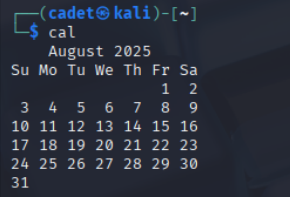


1. Date:

* Shows or sets the system date and time.

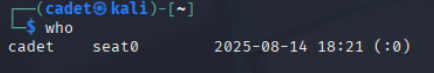


1. Cal:

* Displays a calendar.

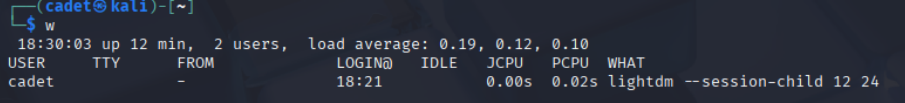
1. Who:

* Shows who is logged in.



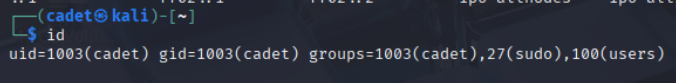
1. W:

* Displays logged-in users and their activities.



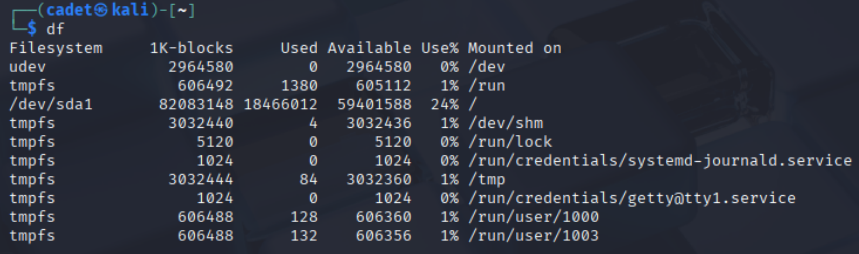
1. Id:

* Shows user identity (UID, GID).



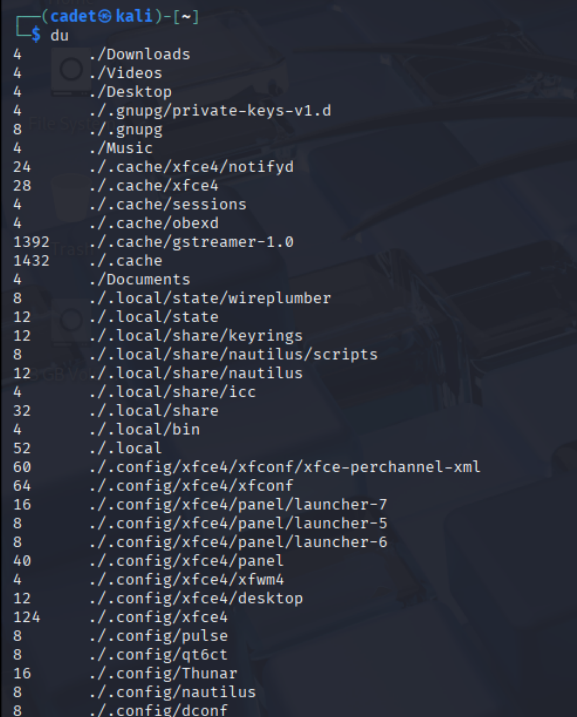
1. Df:

* Shows disk usage by filesystem.



1. Du:

* Shows disk usage of files and directories.



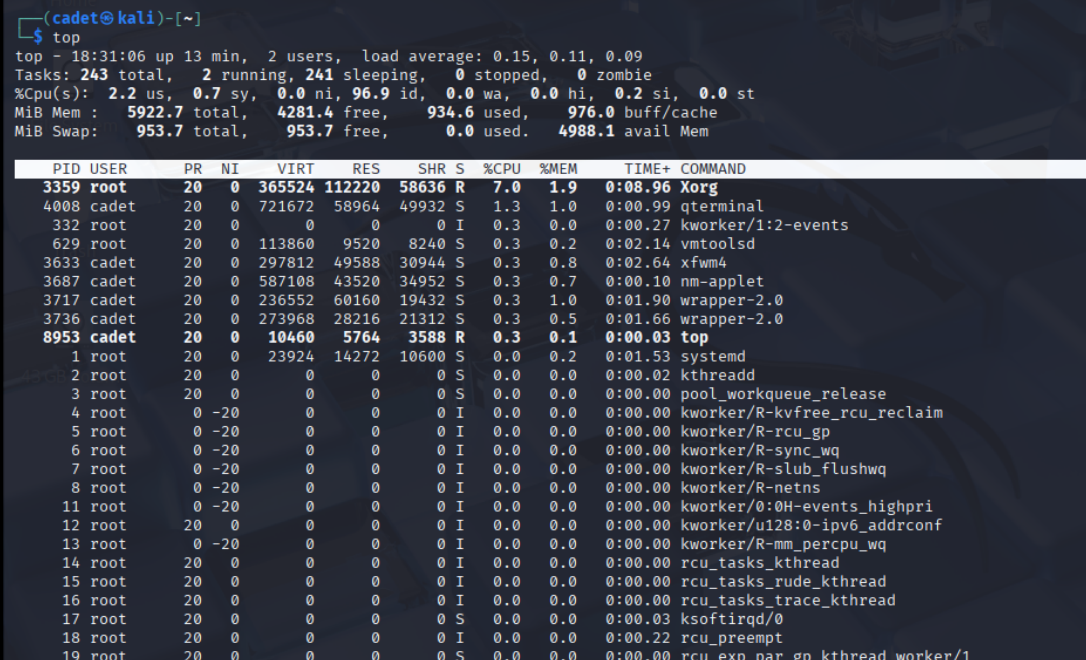
1. Free:

* Displays memory usage.



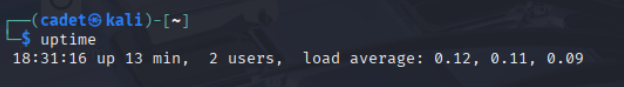
1. Top:

* Shows running processes and system resource usage.



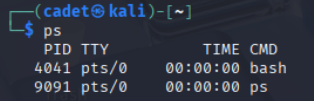
1. Uptime:

* Displays system running time and load average.



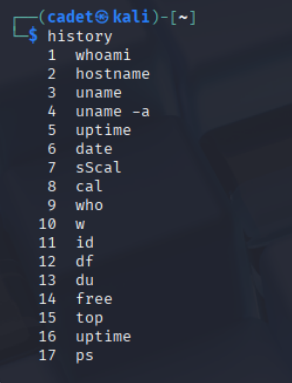
1. Ps:

* Displays running processes.



1. History:

* Shows command history.



1. Clear:

* Clears the terminal screen.

1. shutdown

* Shuts down or reboots the system.

1. reboot

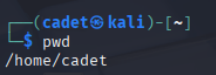
* Reboots the system.

1. **File commands:**

**Basic Navigation**

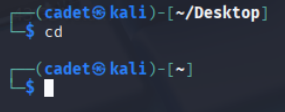
1. Pwd

* Shows the current working directory.



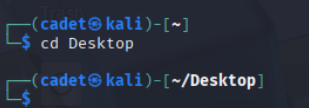
1. Cd

* Changes to the home directory.



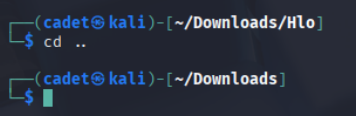
1. Cd/path/to/dir

* Changes to a specific directory.



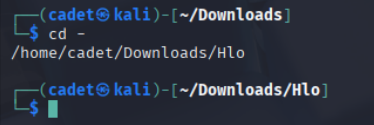
1. Cd..

* Moves up one directory level.



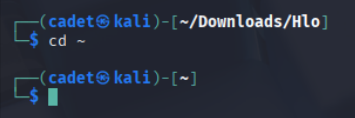
1. Cd –

* Returns to the previous directory.



1. Cd ~

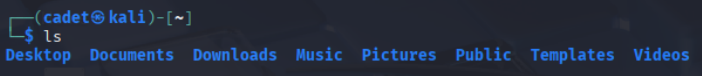
* Jumps to the home directory.



**Listing Files and Info:**

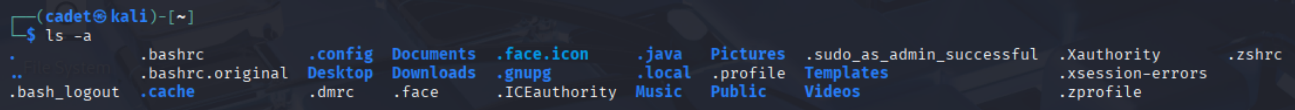
1. Ls

* Lists directory contents.



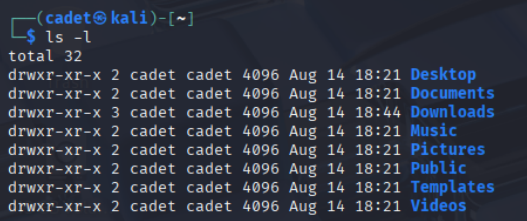
1. Ls -a

* Includes hidden files.



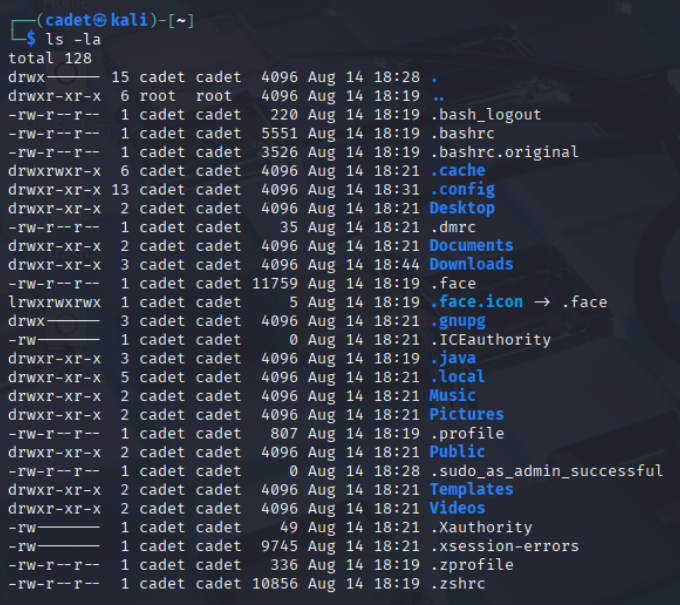
1. Ls -l

* Shows long listing with details.



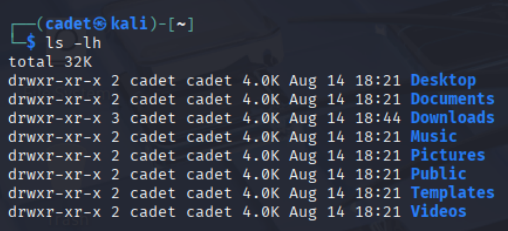
1. Ls -la

* Shows long listing including hidden files.



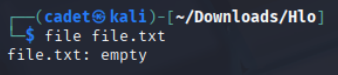
1. Ls -lh

* Displays long listing with human-readable sizes.



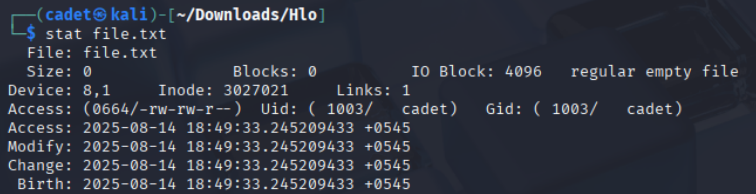
1. File filename

* Determines the file type.



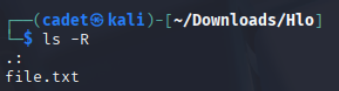
1. Stat file.txt

* Displays detailed file information.



1. Ls -R

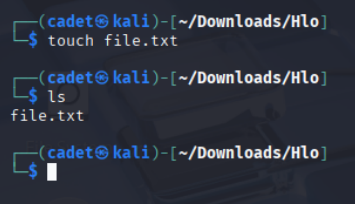
* Recursively lists directory contents.



**Creating, Copying, Moving:**

1. Touch file.txt

* Creates an empty file.



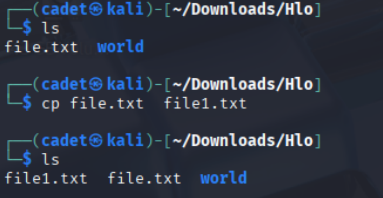
1. Mkdir newdir

* Creates a directory.



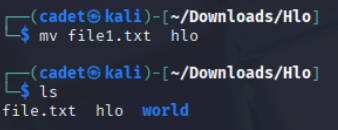
1. Cp file1 file2

* Copies file1 to file2.



1. Mv file1 file2

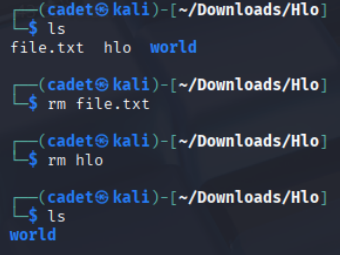
* Renames a file.



**Deleting:**

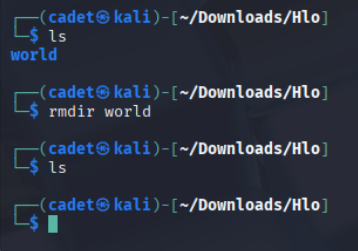
1. Rm file.txt

* Deletes a file.



1. Rmdir dir

* Removes an empty directory.

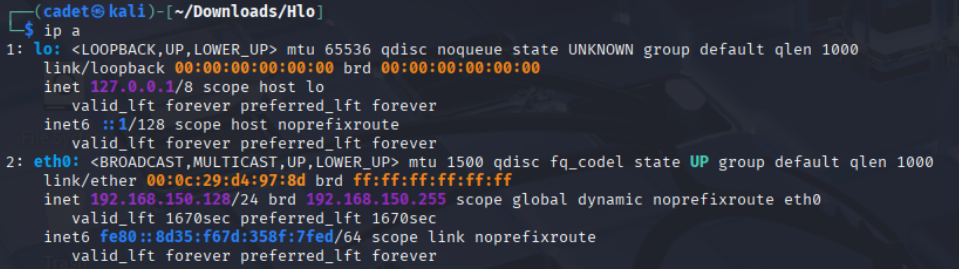


1. **Networking commands:**

**Basic Network Info:**

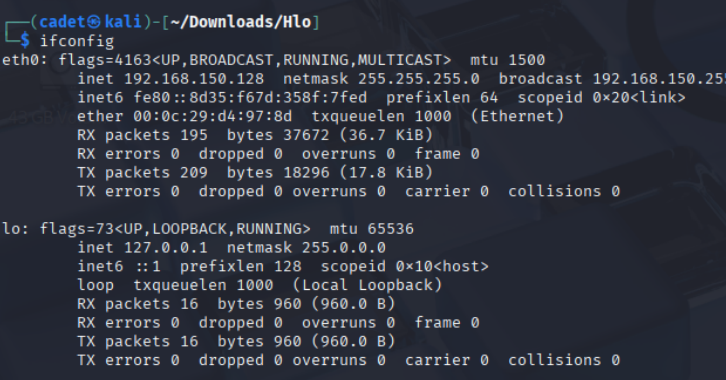
1. Ip a

* Shows IP addresses and network interfaces.



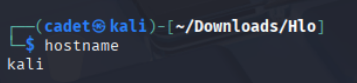
1. Ifconfig

* Displays network interface details.



1. Hostname

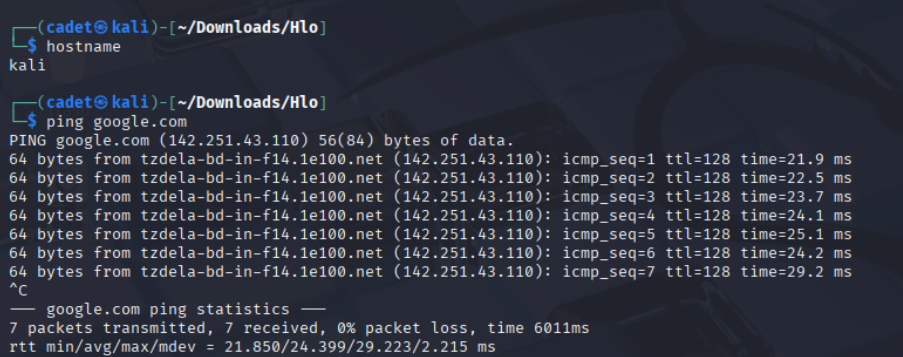
* Shows the system hostname.



**Connectivity and Diagnostics:**

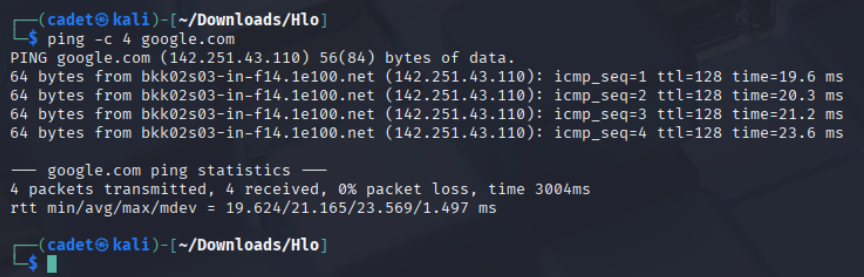
1. Ping host

* Tests connectivity to a host (e.g., ping google.com).



1. Ping -c 4 host

* Sends 4 ping requests.



1. **Basic shell scripting:**

**Shebang & Execution:**

1. #!/bin/bash

* Defines the script interpreter.

1. Chmod +x script.sh

* Makes the script executable.

1. ./script.sh

* Runs the script.

Script.sh:

#!/bin/bash

# =========================

# Basic Shell Scripting Examples (Corrected)

# =========================

# 1. Hello World and Variables

echo "Hello World"

name="Sulav"

echo "Hello, $name"

# 2. Arithmetic

a=10

b=5

sum=$((a + b))

echo "Sum = $sum"

# 3. User Input

read -p "Enter your name: " username

echo "Welcome, $username"

# 4. If-Else Condition (with input check)

read -p "Enter a number: " num

if [[ -z "$num" ]]; then

echo "No number entered. Using 0."

num=0

fi

if [ "$num" -gt 10 ]; then

echo "Number is greater than 10"

else

echo "Number is 10 or less"

fi

# 5. Even or Odd (with input check)

read -p "Enter another number: " n

if [[ -z "$n" ]]; then

echo "No number entered. Using 0."

n=0

fi

if (( n % 2 == 0 )); then

echo "The number is even"

else

echo "The number is odd"

fi

# 6. Simple Calculator

read -p "Enter first number: " n1

read -p "Enter second number: " n2

read -p "Choose operation (+ - \* /): " op

if [[ -z "$n1" ]]; then n1=0; fi

if [[ -z "$n2" ]]; then n2=0; fi

if [ "$op" = "+" ]; then

result=$((n1 + n2))

elif [ "$op" = "-" ]; then

result=$((n1 - n2))

elif [ "$op" = "\*" ]; then

result=$((n1 \* n2))

elif [ "$op" = "/" ]; then

if [ "$n2" -ne 0 ]; then

result=$((n1 / n2))

else

result="Error: Division by zero"

fi

else

result="Invalid operator"

fi

echo "Result: $result"

# 7. System Info

echo "Username: $(whoami)"

echo "Hostname: $(hostname)"

echo "OS Info: $(uname -a)"

# 8. File Exists Check

read -p "Enter a file name: " filename

if [ -e "$filename" ]; then

echo "$filename already exists."

else

echo "Creating $filename..."

echo "This is a new file." > "$filename"

echo "$filename created."

Fi



