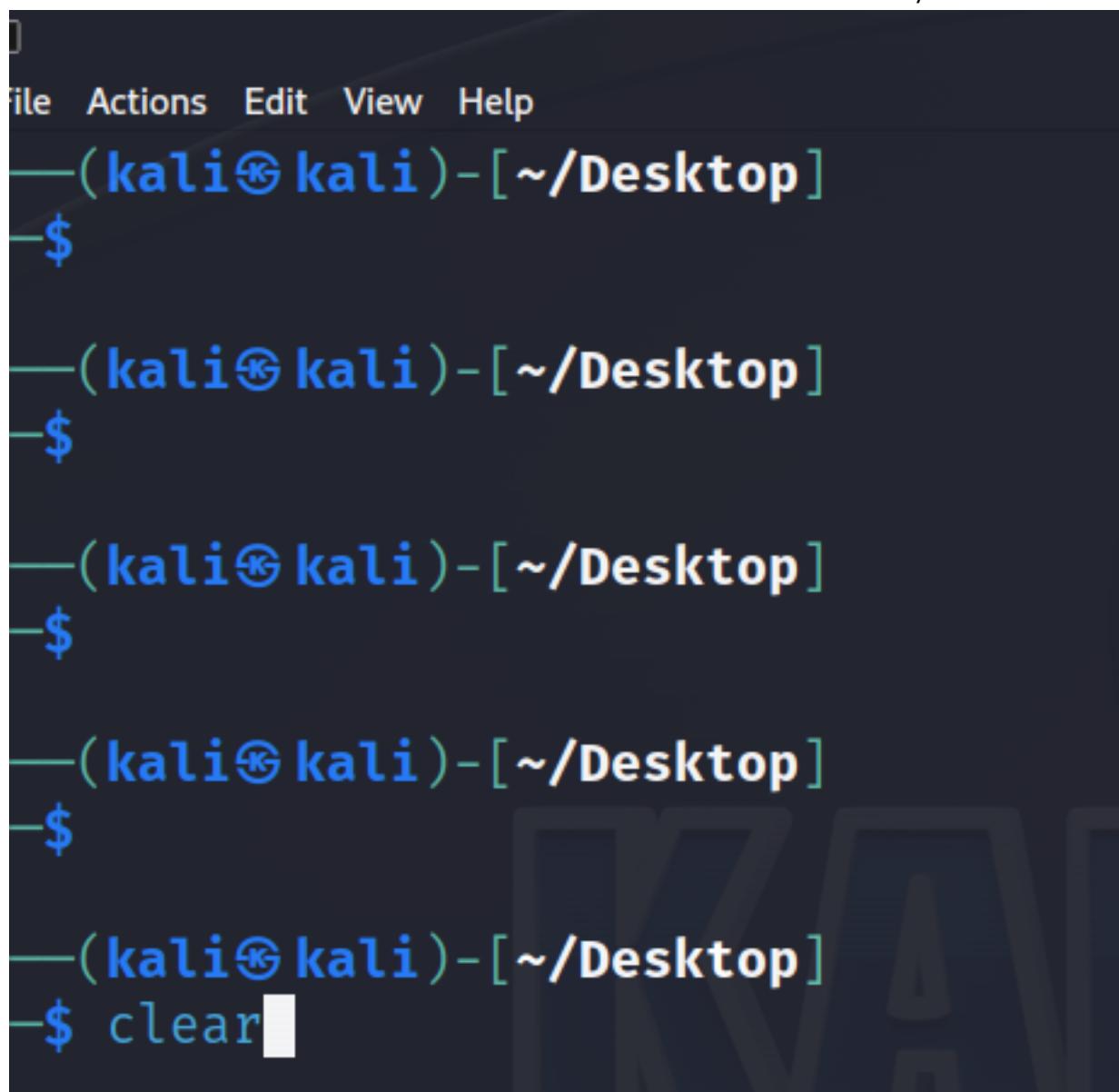


Kali-Linux Commands

Basic

clear command

The clear command is a standard command to clear the terminal screen. Here is the syntax: # clear



A screenshot of a terminal window titled "File Actions Edit View Help". The window shows a command-line interface with a blue background and white text. It displays five identical terminal sessions stacked vertically. Each session starts with the prompt "(kali㉿kali)-[~/Desktop]" followed by a blue dollar sign (\$) indicating where input can be entered. In the bottom session, the user has typed the command "clear" and is awaiting the output.

```
[]
File Actions Edit View Help
—(kali㉿kali)-[ ~/Desktop ]
-$

—(kali㉿kali)-[ ~/Desktop ]
-$

—(kali㉿kali)-[ ~/Desktop ]
-$

—(kali㉿kali)-[ ~/Desktop ]
-$

—(kali㉿kali)-[ ~/Desktop ]
-$ clear |
```



File Actions Edit View Help

(kali㉿kali)-[~/Desktop]

\$ █

KALI

date command

In Kali Linux, the "date" command is used to display the system date and time. In order to display the date, here is the syntax:

```
# date
```

(kali㉿kali)-[~/Desktop]
\$ date

Sat Oct 28 02:21:41 PM EDT 2023

(kali㉿kali)-[~/Desktop]
\$ █

pwd command

In Kali Linux, the 'Pwd' command is used to print working directory. It gives us information about the directory we are now in. This is especially useful if we need to access the directory while in the middle of a complicated process. Here is the syntax:

```
#pwd
```

```
[└(kali㉿kali)-[~/Desktop]]
```

```
└$ pwd
```

```
/home/kali/Desktop
```

```
[└(kali㉿kali)-[~/Desktop]]
```

```
└$ ┌
```

cd command

The 'cd' command is also called chdir (Change Directory). We use this command to change or switch the current working directory. Here is the syntax: # cd /directory/folder/path

Here are some of the "cd" navigation shortcuts:

cd ~[username] – goes to another user's home directory.

cd .. – moves one directory up.

cd- – switches to the previous directory.

```
[└(kali㉿kali)-[~/Desktop]]
```

```
[└$ pwd]
```

```
/home/kali/Desktop
```

```
[└(kali㉿kali)-[~/Desktop]]
```

```
[└$ cd ~]
```

```
[└(kali㉿kali)-[~]]
```

```
[└$ pwd]
```

```
/home/kali
```

```
[└(kali㉿kali)-[~]]
```

```
[└$ cd ..]
```

```
[└(kali㉿kali)-[/home]]
```

```
[└$ pwd]
```

```
/home
```

```
[└(kali㉿kali)-[/home]]
```

```
[└$ cd -]
```

```
~
```

```
[└(kali㉿kali)-[~]]
```

```
[└$ pwd]
```

```
/home/kali
```

ls command

One of the most useful commands in Kali Linux is the 'ls' command. The ls command lists the directory contents of files and directories. With the help of the ls command, we can easily list out every hidden file of a directory with the -a attribute, and for more detailed output, we can use the -l attribute.

-R – lists all the files in the subdirectories.

-a – shows all files, including hidden ones.

-lh – converts sizes to readable formats, such as MB, GB, and TB.

Here is the syntax: # ls -al

```
(kali㉿kali)-[~]
└─$ ls
Desktop Documents Downloads go Music Pictures Public Templates veracrypt Videos

(kali㉿kali)-[~]
└─$ ls -la
total 192
drwxr-xr-x 23 kali kali 4096 Oct 28 14:19 .
drwxr-xr-x  3 root root 4096 Aug  8 2022 ..
-rw-r--r--  1 kali kali 220 Aug  8 2022 .bash_logout
-rw-r--r--  1 kali kali 5604 Aug 22 03:00 .bashrc
-rw-r--r--  1 kali kali 3526 Aug  8 2022 .bashrc.original
drwx----- 14 kali kali 4096 Oct 27 22:51 .cache
drwxr-xr-x 20 kali kali 4096 Oct 27 22:51 .config
drwxr-xr-x  4 kali kali 4096 Oct 26 04:31 Desktop
```

mkdir command

The 'mkdir' command is used to create directories. For example, if we wish to create a directory named 'testing' under the 'Documents' directory, then we have to open a terminal and here is the syntax:

```
# cd Documents
# mkdir testing
```

```
(kali㉿kali)-[~]
└─$ cd Documents

(kali㉿kali)-[~/Documents]
└─$ mkdir Testing

(kali㉿kali)-[~/Documents]
└─$ ls
Testing

(kali㉿kali)-[~/Documents]
```

uptime command

The 'uptime' command displays the amount of time the system has been running. Uptime's basic usage is simple. Use the -p command-line option if we merely want to know how long the system has been up for and in a more human-readable format. Here is the syntax: # uptime

```
└─(kali㉿kali)-[~/Documents]
$ uptime
15:08:35 up 54 min, 1 user, load average: 0.55, 0.44, 0.40

└─(kali㉿kali)-[~/Documents]
$ uptime -p
up 55 minutes

└─(kali㉿kali)-[~/Documents]
$ ┌─
```

whoami command

The 'whoami' command is used to print the effective user ID whereas the who command prints information regarding users who are presently logged in.

```
# whoami
```

```
└─(kali㉿kali)-[~/Documents]
$ whoami
kali

└─(kali㉿kali)-[~/Documents]
$ ┌─
```

&& command

This combines the given commands and executes them one after the other. Here is the syntax: first_command && second_command && third_comand

NB:

One can do as many "&&" as they need.

```
└─(kali㉿kali)-[~/Downloads]
$ mkdir Kampala && cd Kampala && pwd
/home/kali/Downloads/Kampala

└─(kali㉿kali)-[~/Downloads/Kampala]
$ ┌─
```

uname Command

The 'uname' command displays the current system's information. We can view system information about our Linux environment with the uname command in Linux. With the uname -a command, we can learn more about our system, including Kernel Name, Node Name, Kernel Release, Kernel Version, Hardware Platform, Processor and Operating System. Here is the syntax:

```
# uname
```

```
(kali㉿kali)-[~/Documents]
└─$ uname
Linux

(kali㉿kali)-[~/Documents]
└─$ uname -a
Linux kali 6.1.0-kali7-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.20-2kali1 (2023-04-18) x86_64 GNU/Linux

(kali㉿kali)-[~/Documents]
└─$ █
```

history Command

The 'history' command is one of Kali Linux's most commonly used commands. The history command in the bash shell saves a history of commands entered that can be used to repeat commands.

We can run the history command by itself, and it will just print the current user's bash history on the screen, as shown below:

Here is the syntax: # history

```
281  uname
282  uname -a
283  users
284  clear
285  ls
286  touch level1.txt
287  touch level2.txt
288  ls
289  rm -i level1.txt
290  ls
291  clear
292  ls
293  rm -rf level2.txt
294  ls
295  clear
```

```
(kali㉿kali)-[~/Documents]
└─$ █
```

echo command

This command directly prints the typed content on the terminal, here is the syntax: # echo "text"

```
(kali㉿kali)-[~/Documents]
$ echo "Hello"
Hello
```

Here is more advanced usage;

echo "text" > filename.txt → creates a file named filename

```
(kali㉿kali)-[~/Documents]
$ echo "Hello" > hello.txt
```

```
(kali㉿kali)-[~/Documents]
$ ls
hello.txt  Testing
```

echo "text" >> filename.txt → appends content to an existing file.

```
(kali㉿kali)-[~/Documents]
$ echo "Hello2" >> hello.txt
```

cat command

Concatenate or cat is one of the most used Linux commands. It lists, combines and writes file content to the standard output. # cat filename

```
(kali㉿kali)-[~/Documents]
$ cat hello.txt
Hello
Hello2
```

```
(kali㉿kali)-[~/Documents]
$ █
```

There are various ways to use the cat command, here's the syntax:

```
# cat > file.txt – creates a new file. Press "Control + C" to quite  
# cat filename.txt - reads the file contents
```

```
└─(kali㉿kali)-[~/Documents]  
└─$ cat > file1.txt  
sdsd  
sd  
^C
```

```
└─(kali㉿kali)-[~/Documents]  
└─$ cat file1.txt  
sdsd  
sd  
sda
```

```
# cat file1.txt file2.txt > file3.txt – merges file1.txt with file2.txt and stores the output in filename3.txt.
```

```
└─(kali㉿kali)-[~/Documents]  
└─$ cat file1.txt hello.txt > newfile.txt  
  
└─(kali㉿kali)-[~/Documents]  
└─$ cat newfile.txt  
sdsd  
sd  
Hello  
Hello2
```

```
# tac file.txt – displays content in reverse order.
```

```
(kali㉿kali)-[~/Documents]
└─$ tac newfile.txt
Hello2
Hello
sdsd
sdsd
```

```
(kali㉿kali)-[~/Documents]
└─$ ┌─[
```

cp command

Use the cp command to copy files or directories, including their content, from your current location to another. It has various use cases, such as:

Copying one file from the current directory to another folder. Specify the file name and target path: # cp filename.txt /home/username/location

```
(kali㉿kali)-[~/Documents]
└─$ ls
alphabets.txt  file1.txt  hello.txt  level10.txt  newfile.txt  Testing

(kali㉿kali)-[~/Documents]
└─$ cp alphabets.txt /home/kali/Downloads && cd /home/kali/Downloads && ls
127.0.0.3.jpg
alphabets.txt
guese.txt
Nessus-10.6.1-ubuntu1404_amd64.deb
output
senselearner.com.png
Steghide-Brute-Force-Tool
veracrypt-1.25.9-Debian-11-amd64.deb
veracrypt-console-1.25.9-Debian-11-amd64.deb

(kali㉿kali)-[~/Downloads]
└─$ ┌─[
```

Duplicating multiple files to a directory. Enter the file names and the destination path: cp filename1.txt filename2.txt filename3.txt /home/username/location

```
└─(kali㉿kali)-[~/Documents]
$ ls
alphabets.txt  file1.txt  hello.txt  level10.txt  newfile.txt  Testing

└─(kali㉿kali)-[~/Documents]
$ cp alphabets.txt file1.txt hello.txt level10.txt newfile.txt /home/kali/Desktop

└─(kali㉿kali)-[~/Documents]
$ ls
alphabets.txt  file1.txt  hello.txt  level10.txt  newfile.txt  Testing

└─(kali㉿kali)-[~/Documents]
$ cd /home/kali/Desktop && ls
alphabets.txt
file1.txt
hello.txt
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx~
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx~~
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx~~~ able to h
level10.txt
newfile.txt
StegSeek
uganda.txt
work

└─(kali㉿kali)-[~/Desktop]
$
```

Copying a file's content to another within the same directory. Enter the source and the destination file: cp filename1.txt
filename2.txt

```
└─(kali㉿kali)-[~/Desktop]
└─$ cat file1.txt
sdsd
sdsa
```

```
└─(kali㉿kali)-[~/Desktop]
└─$ cat hello.txt
Hello
```

```
└─(kali㉿kali)-[~/Desktop]
└─$ cp file1.txt hello.txt
```

```
└─(kali㉿kali)-[~/Desktop]
└─$ cat hello.txt
sdsd
sdsa
```

```
└─(kali㉿kali)-[~/Desktop]
└─$ █
```

Duplicating an entire directory. Pass the -R flag followed by the source and destination directory: cp -R /home/username/location /home/username/Docum_backup

```
└─(kali㉿kali)-[~/Documents]
└─$ cp -R /home/kali/Documents /home/kali/Music/ && cd /home/kali/Music && ls
Documents
```

```
└─(kali㉿kali)-[~/Music]
└─$ █
```

rm command

In Kali Linux, the 'rm' command is used to delete files. It can be used to delete directories when we use them recursively. To modify the command, add the following options:

-i – prompts a confirmation before deletion.

```
└─(kali㉿kali)-[~/Documents]
$ ls
file1.txt  hello.txt  level1.txt  level2.txt  newfile.txt  Testing

└─(kali㉿kali)-[~/Documents]
$ rm -i level1.txt
rm: remove regular empty file 'level1.txt'? yes

└─(kali㉿kali)-[~/Documents]
$ ls
file1.txt  hello.txt  level2.txt  newfile.txt  Testing

└─(kali㉿kali)-[~/Documents]
$ ┌
```

-f – allows file removal without a confirmation.

-r – deletes files and directories recursively.

```
└─(kali㉿kali)-[~/Documents]
$ ls
file1.txt  hello.txt  level2.txt  newfile.txt  Testing

└─(kali㉿kali)-[~/Documents]
$ rm -rf level2.txt

└─(kali㉿kali)-[~/Documents]
$ ls
file1.txt  hello.txt  newfile.txt  Testing

└─(kali㉿kali)-[~/Documents]
$ ┌
```

The removal process separates a file name from its associated data in a file system and identifies that space in the storage device as available for future writes/ use. In other words, when we erase a file, the data inside it remains unchanged, but it is no longer linked to a filename.

mv command

Use the mv command to move or rename files and directories. To move items, enter the file name followed by the destination directory: # mv filename.txt /home/username/Documents

```
(kali㉿kali)-[~/Documents]
$ ls
alphabets.txt  hello.txt    newfile.txt  uganda.txt
file1.txt      level10.txt  Testing

(kali㉿kali)-[~/Documents]
$ pwd
/home/kali/Documents

(kali㉿kali)-[~/Documents]
$ mv uganda.txt /home/kali/Desktop

(kali㉿kali)-[~/Documents]
$ ls
alphabets.txt  file1.txt  hello.txt  level10.txt  newfile.txt  Testing

(kali㉿kali)-[~/Desktop]
$ cd /home/kali/Desktop

(kali㉿kali)-[~/Desktop]
$ ls
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx~
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx~
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx~~~
StegSeek
uganda.txt
work
```

Meanwhile, use the following syntax to rename a file in Linux with the mv command: # mv old_filename.txt new_filename.txt

```
└─(kali㉿kali)-[~/Documents]
$ ls
file1.txt  letters.txt  newfile.txt  uganda.txt
hello.txt  level10.txt  Testing
```

```
└─(kali㉿kali)-[~/Documents]
$ mv letters.txt alphabets.txt
```

```
└─(kali㉿kali)-[~/Documents]
$ ls
alphabets.txt  hello.txt  newfile.txt  uganda.txt
file1.txt       level10.txt  Testing
```

```
└─(kali㉿kali)-[~/Documents]
```

less command

In Kali Linux, the 'less' command is used to view files instead of opening the file. The less command is a more powerful variant of the "more" command which is used to show information one page at a time to the terminal. We can view any text file with the help of the "less" command simply by typing the following command into a terminal window:

```
└─(kali㉿kali)-[/]
$ ls -la | less
```

Here is the output of the command. Press "Q" to exit/ quite

```

total 1048656
drwxr-xr-x 18 root root          4096 May 11 02:23 .
drwxr-xr-x 18 root root          4096 May 11 02:23 ..
-rw-r--r-- 1 root root          0 Aug  8 2022 0
lrwxrwxrwx 1 root root          7 Aug  8 2022 bin → usr/bin
drwxr-xr-x  3 root root          4096 May 11 02:57 boot
drwxr-xr-x 17 root root          3400 Oct 28 14:13 dev
drwxr-xr-x 178 root root         12288 Oct 28 14:13 etc
drwxr-xr-x  3 root root          4096 Aug  8 2022 home
lrwxrwxrwx  1 root root          33 May 11 02:23 initrd.img → boot/initrd.img-6.1.0-kali7-amd64
lrwxrwxrwx  1 root root          34 Aug  8 2022 initrd.img.old → boot/initrd.img-5.18.0-kali5-amd64
lrwxrwxrwx  1 root root          7 Aug  8 2022 lib → usr/lib
lrwxrwxrwx  1 root root          9 Aug  8 2022 lib32 → usr/lib32
lrwxrwxrwx  1 root root          9 Aug  8 2022 lib64 → usr/lib64
lrwxrwxrwx  1 root root          10 Aug  8 2022 libx32 → usr/libx32
drwx———  2 root root          16384 Aug  8 2022 lost+found
drwxr-xr-x  2 root root          4096 Aug  8 2022 media
drwxr-xr-x  2 root root          4096 Aug  8 2022 mnt
drwxr-xr-x 13 root root          4096 Oct 27 22:48 opt
dr-xr-xr-x 256 root root          0 Oct 28 14:13 proc
drwx———  7 root root          4096 Sep 26 11:40 root
drwxr-xr-x 33 root root          820 Oct 28 14:14 run
lrwxrwxrwx  1 root root          8 Aug  8 2022 sbin → usr/sbin
drwxr-xr-x  3 root root          4096 Aug  8 2022 srv
-rw———  1 root root 1073741824 Aug  8 2022 swapfile
dr-xr-xr-x 13 root root          0 Oct 28 14:13 sys
drwxrwxrwt 14 root root          4096 Oct 28 15:39 tmp
drwxr-xr-x 16 root root          4096 Aug  7 2020 usr
drwxr-xr-x 12 root root          4096 Aug  8 2022 var
lrwxrwxrwx  1 root root          30 May 11 02:23 vmlinuz → boot/vmlinuz-6.1.0-kali7-amd64
lrwxrwxrwx  1 root root          31 Aug  8 2022 vmlinuz.old → boot/vmlinuz-5.18.0-kali5-amd64
(END)

```

more command

The "more" command permits us to show output in the terminal one page at a time. This is particularly beneficial when using a command that requires a lot of scrolling, such as the 'ls' command or the 'du' commands. The 'more' command works with any applications that output to the screen. A good way to test this is to type the following command into a terminal window:



```

(kali㉿kali)-[~/]
$ ls -la | more

```

This is how the content is displayed. Press "Q" to quite/ exit

```

total 1048656
drwxr-xr-x 18 root root        4096 May 11 02:23 .
drwxr-xr-x 18 root root        4096 May 11 02:23 ..
-rw-r--r--  1 root root         0 Aug  8 2022 0
lrwxrwxrwx  1 root root         7 Aug  8 2022 bin → usr/bin
drwxr-xr-x  3 root root        4096 May 11 02:57 boot
drwxr-xr-x 17 root root       3400 Oct 28 14:13 dev
drwxr-xr-x 178 root root      12288 Oct 28 14:13 etc
drwxr-xr-x  3 root root        4096 Aug  8 2022 home
lrwxrwxrwx  1 root root        33 May 11 02:23 initrd.img → boot/initrd.img-6.1.0-kali7-amd64
lrwxrwxrwx  1 root root        34 Aug  8 2022 initrd.img.old → boot/initrd.img-5.18.0-kali5-amd64
lrwxrwxrwx  1 root root         7 Aug  8 2022 lib → usr/lib
lrwxrwxrwx  1 root root         9 Aug  8 2022 lib32 → usr/lib32
lrwxrwxrwx  1 root root         9 Aug  8 2022 lib64 → usr/lib64
lrwxrwxrwx  1 root root        10 Aug  8 2022 libx32 → usr/libx32
drwx----- 2 root root       16384 Aug  8 2022 lost+found
drwxr-xr-x  2 root root        4096 Aug  8 2022 media
drwxr-xr-x  2 root root        4096 Aug  8 2022 mnt
drwxr-xr-x 13 root root       4096 Oct 27 22:48 opt
dr-xr-xr-x 255 root root      0 Oct 28 14:13 proc
drwx----- 7 root root       4096 Sep 26 11:40 root
drwxr-xr-x 33 root root       820 Oct 28 14:14 run
lrwxrwxrwx  1 root root         8 Aug  8 2022 sbin → usr/sbin
drwxr-xr-x  3 root root        4096 Aug  8 2022 srv
-rw----- 1 root root      1073741824 Aug  8 2022 swapfile
dr-xr-xr-x 13 root root         0 Oct 28 14:13 sys
drwxrwxrwt 14 root root       4096 Oct 28 15:39 tmp
drwxr-xr-x 16 root root       4096 Aug  7 2020 usr
drwxr-xr-x 12 root root       4096 Aug  8 2022 var
lrwxrwxrwx  1 root root        30 May 11 02:23 vmlinuz → boot/vmlinuz-6.1.0-kali7-amd64
lrwxrwxrwx  1 root root        31 Aug  8 2022 vmlinuz.old → boot/vmlinuz-5.18.0-kali5-amd64

```

(END)

free Command

In Kali Linux, the 'free' command provides us the useful information about the amount of RAM available on a Linux machine. It also displays the entire amount of physical memory used and available space, as well as swap memory with kernel buffers. Here is the syntax: # free

If we use the free command with the -t option, it would list the total line at the end. Here is the syntax: # free -t

```

└─(kali㉿kali)-[~/Documents]
$ free
              total        used        free      shared  buff/cache   available
Mem:      1998472     817164     767844        6364      570652    1181308
Swap:     1048572          0     1048572

└─(kali㉿kali)-[~/Documents]
$ free -t
              total        used        free      shared  buff/cache   available
Mem:      1998472     821984     762992        6372      570680    1176488
Swap:     1048572          0     1048572
Total:    3047044     821984     1811564

└─(kali㉿kali)-[~/Documents]
$ 

```

sort command

Using the 'sort' command, we can sort the content of the text file, line by line. Sort is a standard command-line program which prints the lines of its input or concentration of all files listed in its argument list in sorted order. Here is the syntax: # sort file

```
└─(kali㉿kali)-[~/Documents]
└─$ cat letters.txt
```

E
X
A
M
P
L
E

```
└─(kali㉿kali)-[~/Documents]
└─$ sort letters.txt
```

A
E
E
L
M
P
X

"the quieter you become

```
└─(kali㉿kali)-[~/Documents]
└─$ █
```

We can reverse the order of any file's contents by using the -r sort. Here is the syntax: # sort -r

```
└─(kali㉿kali)-[~/Documents]
└─$ sort -r letters.txt
```

X
P
M
L
E
E
A

"the quieter you be

```
└─(kali㉿kali)-[~/Documents]
└─$
```

touch command

The touch command lets you create an empty file in a specific directory path. Here's the syntax: touch [option] /home/directory/path/file.txt

If you omit the path, the command will create the item in the current folder. You can also use touch to generate and modify a timestamp in the Linux command line.

```
└─(kali㉿kali)-[~/Documents]
└─$ touch uganda.txt
```

```
└─(kali㉿kali)-[~/Documents]
└─$ ls -la
total 24
drwxr-xr-x  3 kali kali 4096 Oct 28 15:33 .
drwxr-xr-x 23 kali kali 4096 Oct 28 14:19 ..
-rw-r--r--  1 kali kali   10 Oct 28 14:55 file1.txt
-rw-r--r--  1 kali kali   13 Oct 28 14:49 hello.txt
-rw-r--r--  1 kali kali     0 Oct 28 15:31 level10.txt
-rw-r--r--  1 kali kali   23 Oct 28 15:01 newfile.txt
drwxr-xr-x  2 kali kali 4096 Oct 28 14:34 Testing
-rw-r--r--  1 kali kali     0 Oct 28 15:33 uganda.txt
```

```
└─(kali㉿kali)-[~/Documents]
└─$ touch uganda.txt
```

```
└─(kali㉿kali)-[~/Documents]
└─$ ls -la
total 24
drwxr-xr-x  3 kali kali 4096 Oct 28 15:33 .
drwxr-xr-x 23 kali kali 4096 Oct 28 14:19 ..
-rw-r--r--  1 kali kali   10 Oct 28 14:55 file1.txt
-rw-r--r--  1 kali kali   13 Oct 28 14:49 hello.txt
-rw-r--r--  1 kali kali     0 Oct 28 15:31 level10.txt
-rw-r--r--  1 kali kali   23 Oct 28 15:01 newfile.txt
drwxr-xr-x  2 kali kali 4096 Oct 28 14:34 Testing
-rw-r--r--  1 kali kali     0 Oct 28 15:35 uganda.txt
```

```
└─(kali㉿kali)-[~/Documents]
└─$
```

The basic commands are more and not limited to the ones demonstrated here. Please go ahead and explore by making further research.

Advanced

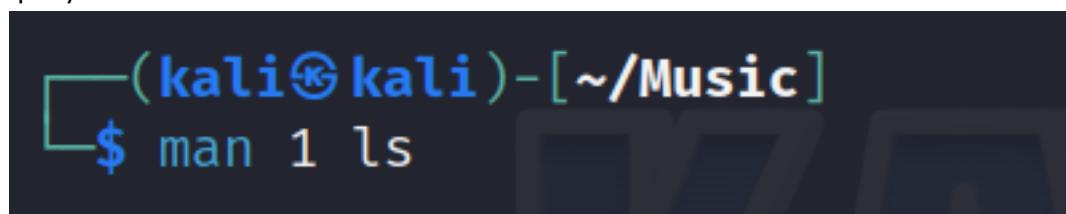
man command

The man command provides a user manual of any Linux Terminal utilities, including their names, descriptions, and options. It consists of nine sections:

- Executable programs or shell commands
- System calls
- Library calls
- Games
- Special files
- File formats and conventions
- System administration commands
- Kernel routines
- Miscellaneous

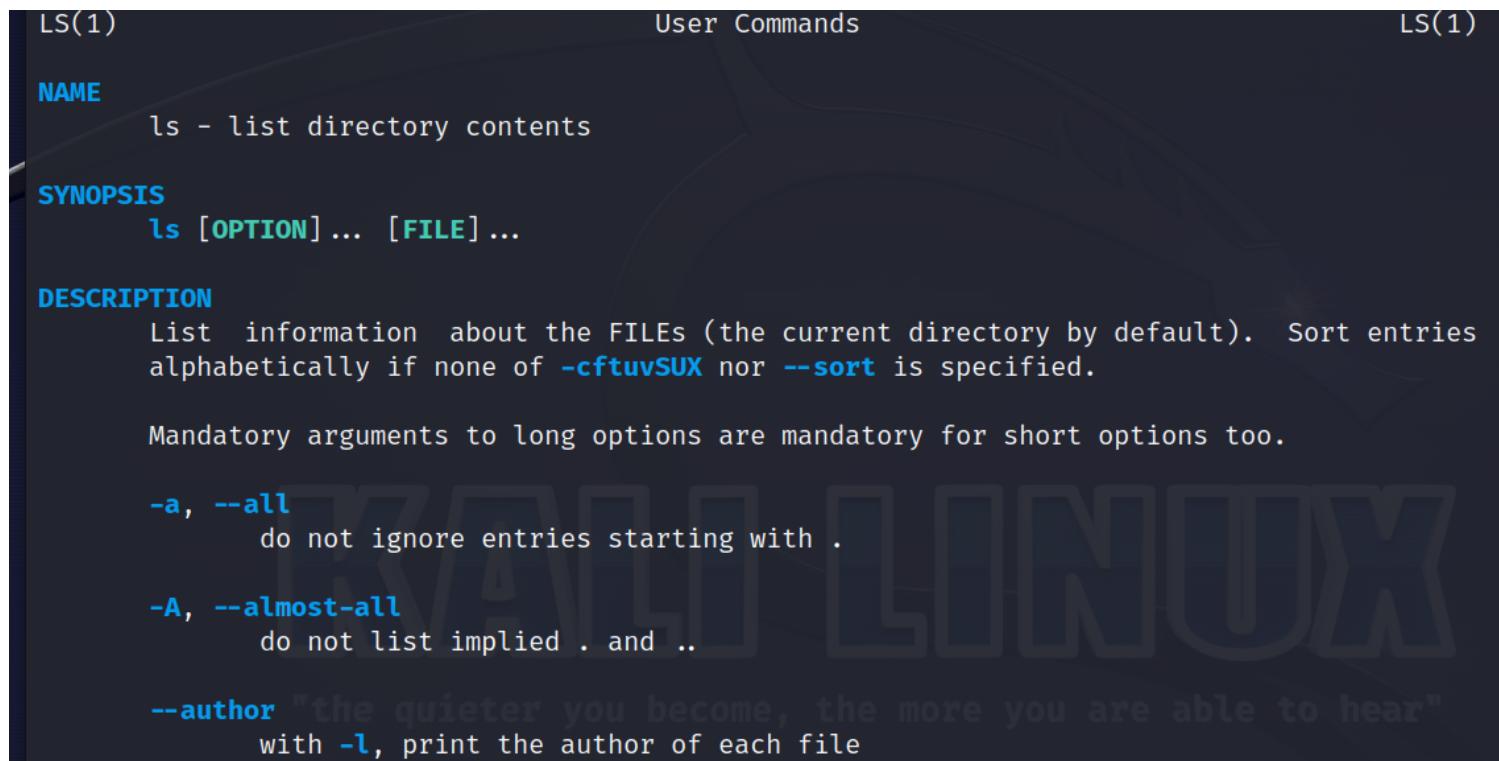
Here's the command syntax: `man [option] [section_number] command_name`

If you only use the command name as the parameter, Terminal displays the full user manual. Here's an example command to query section 1 of the ls command manual: `# man 1 ls`



A screenshot of a terminal window titled '(kali㉿kali)-[~/Music]'. The prompt shows '\$ man 1 ls'. The window has a dark background with light-colored text.

And the man ls man page is displayed



LS(1) User Commands LS(1)

NAME
`ls` - list directory contents

SYNOPSIS
`ls [OPTION] ... [FILE] ...`

DESCRIPTION
List information about the FILEs (the current directory by default). Sort entries alphabetically if none of `-cftuvSUX` nor `--sort` is specified.
Mandatory arguments to long options are mandatory for short options too.

-a, --all
do not ignore entries starting with `.`

-A, --almost-all
do not list implied `.` and `..`

--author "the quieter you become, the more you are able to hear"
with `-l`, print the author of each file

file command

The file command lets you check a file type – whether it is a text, image, or binary. Here's the syntax: `# file filename.txt`

```
└─(kali㉿kali)-[~/Documents]
$ ls
alphabets.txt  file1.txt  hello.txt  level10.txt  newfile.txt  Testing
```

```
└─(kali㉿kali)-[~/Documents]
$ file alphabets.txt
alphabets.txt: ASCII text
```

```
└─(kali㉿kali)-[~/Documents]
$ ┌
```

To bulk-check multiple files, list them individually or use their path if they are in the same directory.

```
└─(kali㉿kali)-[~/Documents]
$ file alphabets.txt hello.txt level10.txt
alphabets.txt: ASCII text
hello.txt:      ASCII text
level10.txt:    empty
```

```
└─(kali㉿kali)-[~/Documents]
$ ┌
```

Add the -k option to display more detailed information in case it is available.

```
└─(kali㉿kali)-[~/Documents]
$ file -k alphabets.txt hello.txt level10.txt
alphabets.txt: ASCII text
hello.txt:      ASCII text
level10.txt:    empty
```

```
└─(kali㉿kali)-[~/Documents]
```

And -i to show the file's MIME type.

```
(kali㉿kali)-[~/Documents]
$ file -i alphabets.txt hello.txt level10.txt
alphabets.txt: text/plain; charset=us-ascii
hello.txt:      text/plain; charset=us-ascii
level10.txt:    inode/x-empty; charset=binary
```

```
(kali㉿kali)-[~/Documents]
$
```

zip, unzip commands

The zip command lets you compress items into a ZIP file with the optimal compression ratio. Here's the syntax: # zip [options] zipfile file1 file2....

For example, this command compresses file1.txt into file1.zip in the current working directory: zip file1.zip file1.txt

```
(kali㉿kali)-[~/Documents]
$ ls
alphabets.txt  file1.txt  hello.txt  level10.txt  newfile.txt  Testing

(kali㉿kali)-[~/Documents]
$ zip file1.zip file1.txt
adding: file1.txt (stored 0%)

(kali㉿kali)-[~/Documents]
$ ls
alphabets.txt  file1.txt  file1.zip  hello.txt  level10.txt  newfile.txt  Testing

(kali㉿kali)-[~/Documents]
$
```

Use the unzip command to extract the compressed file. Here's the syntax: # unzip [option] file_name.zip

```

└─(kali㉿kali)-[~/Documents]
$ ls
alphabets.txt  file1.txt  file1.zip  hello.txt  level10.txt  newfile.txt  Testing

└─(kali㉿kali)-[~/Documents]
$ rm -rf file1.txt && ls
alphabets.txt  file1.zip  hello.txt  level10.txt  newfile.txt  Testing

└─(kali㉿kali)-[~/Documents]
$ unzip file1.zip
Archive:  file1.zip
extracting: file1.txt

└─(kali㉿kali)-[~/Documents]
$ ls
alphabets.txt  file1.txt  file1.zip  hello.txt  level10.txt  newfile.txt  Testing

└─(kali㉿kali)-[~/Documents]
$ 

```

tar command

The tar command archives multiple items into a TAR file – a format similar to ZIP with optional compression. Here's the syntax: # tar [options] [archive_file] [target file or directory]

For instance, enter the following to create a new alphabets.tar archive in the /home/user/location directory: tar -cvf alphabets.tar /home/user/location

```

└─(kali㉿kali)-[~/Documents]
$ ls
alphabets.txt  file1.txt  file1.zip  hello.txt  level10.txt  newfile.txt  Testing

└─(kali㉿kali)-[~/Documents]
$ tar -cvf alphabets.tar /home/kali/Documents
tar: Removing leading `/' from member names
/home/kali/Documents/
/home/kali/Documents/level10.txt
/home/kali/Documents/Testing/
/home/kali/Documents/file1.txt
tar: /home/kali/Documents/alphabets.tar: file is the archive; not dumped
/home/kali/Documents/alphabets.txt
/home/kali/Documents/file1.zip
/home/kali/Documents/hello.txt
/home/kali/Documents/newfile.txt

└─(kali㉿kali)-[~/Documents]
$ ls
alphabets.tar  file1.txt  hello.txt  newfile.txt
alphabets.txt  file1.zip  level10.txt  Testing

└─(kali㉿kali)-[~/Documents]
$ 

```

nano, vi, jed, mousepad commands

Linux lets users edit files using a text editor like nano, vi, jed or mousepad. While most distributions include nano and vi, users must install jed or mousepad manually. All these tools have the same command syntax:

nano filename.

```
(kali㉿kali)-[~/Documents]
$ nano file1.txt
```

Press "Control+X" to exit

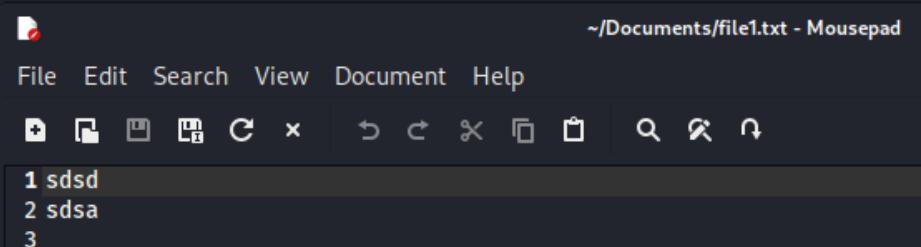
```
GNU nano 7.2
sdsd
sdsa
```

vi filename

```
File Actions Edit View Help
sdsd
sdsa
~
```

mousepad filename

```
(kali㉿kali)-[~/Documents]
$ mousepad file1.txt
```



The screenshot shows the Mousepad application window. The title bar reads "/Documents/file1.txt - Mousepad". The menu bar includes File, Edit, Search, View, Document, and Help. Below the menu is a toolbar with icons for new, open, save, cut, copy, paste, and others. A status bar at the bottom shows line numbers 1, 2, and 3. The main text area displays the same two lines of text as the other editors: "sdsd" and "sdsa".

If the target file doesn't exist, these editors will create one. Nano is recommended if you want to quickly edit text files. Meanwhile, use vi or jed for scripting and programming.

grep command

The global regular expression or grep command lets you find a word by searching the content of a file. This Linux command prints all lines containing the matching strings, which is useful for filtering large log files.

For example, to display lines containing blue in the colors.txt file, enter: # grep blue colors.txt

Kindly note that the content to be grepped is case sensitive

```
└─(kali㉿kali)-[~/Desktop]
$ ls
alphabets.txt
colors.txt
file1.txt
hello.txt
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx~
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx~
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx~~
level10.txt
newfile.txt
StegSeek
uganda.txt
work

└─(kali㉿kali)-[~/Desktop]
$ grep "Blue" colors.txt
Blue
Navy Blue
Sky Blue
Light Blue

4... └─(kali㉿kali)-[~/Desktop]
$ ┌──(kali㉿kali)-[~/Desktop]
└─$ su Uganda
Password:
$ whoami
Uganda
$ users
kali
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx~
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx~
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx~~
"the quieter yo
```

head command

The head command prints the first ten lines of a text file or piped data in your command-line interface. Here's the general syntax: # head [option] [file]

For instance, to view the first ten lines of alphabets.txt in the current directory, enter: # head alphabets.txt

```
k └──(kali㉿kali)-[~/Desktop]
    $ head alphabets.txt
A
B
C
D newfile.txt
E
F
G
H
I
J
```



```
k └──(kali㉿kali)-[~/Desktop]
    $
```

The head command accepts several options, such as:

-n – changes the number of lines printed. For example, head -n 5 shows the first five lines.

```
alphabets.txt
└──(kali㉿kali)-[~/Desktop]
    $ head -n 5 alphabets.txt
A
B
C file1.txt
D
E
```



```
└──(kali㉿kali)-[~/Desktop]
    $ heado.txt
```

-q – disables headers specifying the file name.

```
(kali㉿kali)-[~/Desktop]
$ head -q alphabets.txt
A
B
C
D newfile.txt
E
F
G
H
I
J

(kali㉿kali)-[~/Desktop]
$
```

tail command

The tail command displays the last ten lines of a file, which is useful for checking new data and errors. Here's the syntax: # tail [option] [file]

For example, enter the following to show the last ten lines of the alphabets.txt file: # tail -n alphabets.txt

```
(kali㉿kali)-[~/Desktop]
$ tail -n 10 alphabets.txt
```

Q
R
S file1.txt
T
U
V
W hello.txt
X
Y
Z

```
(kali㉿kali)-[~/Desktop]
```

useradd, userdel commands

Use useradd to create a new Linux user account and change its password with the passwd command. Here are the syntaxes:

```
# sudo useradd [option] username
# sudo passwd username
```

```
(kali㉿kali)-[~/Desktop]
$ sudo useradd Uganda
```



```
(kali㉿kali)-[~/Desktop]
$ sudo passwd Uganda
```

New password:

Retype new password:

passwd: password updated successfully

```
(kali㉿kali)-[~/Desktop]
```

Both the useradd and passwd commands require sudo privileges. To delete a user, use the userdel command: # sudo userdel

username

```
└─(kali㉿kali)-[~/Desktop]
└─$ sudo userdel Uganda
```

```
└─(kali㉿kali)-[~/Desktop]
```

```
└─$ █
```

tee command

The tee command writes the user's input to Terminal's output and files. Here's the basic syntax: command | tee [option] file1
For example, the following pings Localhost and prints the output in Terminal, ping_result.txt, and the localhost.txt file: # ping google.com | tee ping_result.txt localhost.txt

```
└─(kali㉿kali)-[~/Desktop] 0:16 /usr/lib
└─$ ping 127.0.0.1 | tee ping_result.txt localhost.txt
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.021 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.052 ms
^C
Ssl 06:08 0:00 /usr/lib
└─(kali㉿kali)-[~/Desktop] 0:00 /usr/lib
└─$ ls
alphabets.txt
colors.txt
file1.txt
hello.txt
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx~
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx~
jpt.ctx111847494839.ctx70859000665.ctx89993817799.ctx73357648822.ctx~~~
```

level10.txt
localhost.txt
newfile.txt
ping_result.txt
StegSeek
uganda.txt
work

```
└─(kali㉿kali)-[~/Desktop]
└─$ cat localhost.txt
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.021 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.052 ms
└─(kali㉿kali)-[~/Desktop]
└─$ cat ping_result.txt
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.021 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.052 ms
└─(kali㉿kali)-[~/Desktop]
└─$ █
```

chmod command

The chmod command modifies directory or file permissions in Linux. Here's the basic syntax: # chmod [option] [permission] [file_name]

In Linux, each file is associated with three user classes – owner, group member, and others. It also has three permissions – read, write and execute. If an owner wants to grant all permissions to every user, the command looks like this: chmod -rwxrwxrwx filename.txt

```
(kali㉿kali)-[~/Desktop]
$ ls -la uganda.txt
-rw-r--r-- 1 kali kali 0 Oct 28 15:35 uganda.txt

(kali㉿kali)-[~/Desktop]
$ chmod -rwxrwxrwx uganda.txt

(kali㉿kali)-[~/Desktop]
$ ls -la uganda.txt
----- 1 kali kali 0 Oct 28 15:35 uganda.txt

(kali㉿kali)-[~/Desktop]
```

chown command

The chown command lets you change a file, directory, or symbolic link's ownership to the specified username. Here's the syntax: chown [option] owner[:group] file(s)

For example, to make linuxuser2 the owner of filename.txt, use: chown linuxuser2 filename.txt

```
(kali㉿kali)-[~/Desktop]
$ sudo chown Uganda uganda.txt
[sudo] password for kali:
```

users Command

The 'users' command is used to display the login names of users logged in on the system. Here is the syntax: # users

```
(kali㉿kali)-[~/Desktop]
$ users
kali
```

df command

Use the df command to check a Linux system's disk space usage in percentage and kilobyte (KB). Here's the syntax: # df [options] [file]

If you don't specify the item, this command will display information about every mounted file system.

```
(kali㉿kali)-[~/Desktop]
$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            958212       0   958212   0% /dev
tmpfs           199848     1160   198688   1% /run
/dev/sda1      82083148 23569284  54298316 31% /
tmpfs           999236       0   999236   0% /dev/shm
tmpfs            5120       0     5120   0% /run/lock
tmpfs           199844      88   199756   1% /run/user/1000
hello.txt

(kali㉿kali)-[~/Desktop]
```

These are some acceptable options:

-m – displays information on the file system usage in MBs.

```
(kali㉿kali)-[~/Desktop]
$ df -m
Filesystem      1M-blocks  Used Available Use% Mounted on
udev             936       0     936   0% /dev
tmpfs            196       2     195   1% /run
/dev/sda1      80160  23017  53026 31% /
tmpfs            976       0     976   0% /dev/shm
tmpfs              5       0       5   0% /run/lock
tmpfs            196       1     196   1% /run/user/1000
file.txt

(kali㉿kali)-[~/Desktop]
$
```

-k – prints file system usage in KBs.

```

└──(kali㉿kali)-[~/Desktop]
$ df -k
Filesystem      1K-blocks      Used Available Use% Mounted on
udev            958212        0    958212   0% /dev
tmpfs           199848     1168    198680   1% /run
/dev/sda1       82083148 23569284  54298316 31% /
tmpfs           999236        0    999236   0% /dev/shm
tmpfs            5120        0      5120   0% /run/lock
tmpfs           199844       88    199756   1% /run/user/1000
hello.txt

└──(kali㉿kali)-[~/Desktop]
$ 

```

-T – shows the file system type in a new column.

```

└──(kali㉿kali)-[~/Desktop]
$ df -T
Filesystem  Type  1K-blocks      Used Available Use% Mounted on
udev        devtmpfs  958212        0    958212   0% /dev
tmpfs        tmpfs   199848     1168    198680   1% /run
/dev/sda1    ext4   82083148 23569288  54298312 31% /
tmpfs        tmpfs   999236        0    999236   0% /dev/shm
tmpfs        tmpfs    5120        0      5120   0% /run/lock
tmpfs        tmpfs   199844       88    199756   1% /run/user/1000
file.txt

└──(kali㉿kali)-[~/Desktop]
$ 

```

du command

Use du to check a file or directory's storage consumption. Remember to specify the directory path when using this command, for example: # du /home/user/Desktop

```
└─(kali㉿kali)-[~/Desktop]
$ du /home/kali/Desktop
12      /home/kali/Desktop/StegSeek/package
148     /home/kali/Desktop/StegSeek/.demo
64      /home/kali/Desktop/StegSeek/.git/hooks
8       /home/kali/Desktop/StegSeek/.git/logs/refs/heads
8       /home/kali/Desktop/StegSeek/.git/logs/refs/remotes/origin
12      /home/kali/Desktop/StegSeek/.git/logs/refs/remotes
24      /home/kali/Desktop/StegSeek/.git/logs/refs
32      /home/kali/Desktop/StegSeek/.git/logs
4       /home/kali/Desktop/StegSeek/.git/refs/tags
8       /home/kali/Desktop/StegSeek/.git/refs/heads
8       /home/kali/Desktop/StegSeek/.git/refs/remotes/origin
12      /home/kali/Desktop/StegSeek/.git/refs/remotes
28      /home/kali/Desktop/StegSeek/.git/refs
1692    /home/kali/Desktop/StegSeek/.git/objects/pack
4       /home/kali/Desktop/StegSeek/.git/objects/info
1700    /home/kali/Desktop/StegSeek/.git/objects
4       /home/kali/Desktop/StegSeek/.git/branches
8       /home/kali/Desktop/StegSeek/.git/info
1880    /home/kali/Desktop/StegSeek/.git
772     /home/kali/Desktop/StegSeek/src
40      /home/kali/Desktop/StegSeek/tests/stegseek/data
56      /home/kali/Desktop/StegSeek/tests/stegseek
84      /home/kali/Desktop/StegSeek/tests/steghide/data
444     /home/kali/Desktop/StegSeek/tests/steghide
504     /home/kali/Desktop/StegSeek/tests
16      /home/kali/Desktop/StegSeek/.github/workflows
20      /home/kali/Desktop/StegSeek/.github
3404    /home/kali/Desktop/StegSeek
16      /home/kali/Desktop/work
761740   /home/kali/Desktop
```

The du command has several options, such as:

-s – shows the specified folder's total size.

```
└─(kali㉿kali)-[~/Desktop]
$ du -s /home/kali/Desktop
761740  /home/kali/Desktop

└─(kali㉿kali)-[~/Desktop]
$
```

-m – provides folder and file information in MB.

```
(kali㉿kali)-[~/Desktop]
$ du -m /home/kali/Desktop
1      /home/kali/Desktop/StegSeek/package
1      /home/kali/Desktop/StegSeek/.demo
1      file1.txt
1      /home/kali/Desktop/StegSeek/.git/hooks
1      /home/kali/Desktop/StegSeek/.git/logs/refs/heads
1      /home/kali/Desktop/StegSeek/.git/logs/refs/remotes/origin
1      /home/kali/Desktop/StegSeek/.git/logs/refs/remotes
1      /home/kali/Desktop/StegSeek/.git/logs/refs
1      /home/kali/Desktop/StegSeek/.git/logs
1      hello.txt
1      level10.txt
1      newfile.txt
1      colors.txt
1      /home/kali/Desktop/StegSeek/package
1      /home/kali/Desktop/StegSeek/.demo
1      /home/kali/Desktop/StegSeek/.git/hooks
1      /home/kali/Desktop/StegSeek/.git/logs/refs/heads
1      /home/kali/Desktop/StegSeek/.git/logs/refs/remotes/origin
1      /home/kali/Desktop/StegSeek/.git/logs/refs/remotes
1      /home/kali/Desktop/StegSeek/.git/logs
1      /home/kali/Desktop/StegSeek/.git/refs
2      /home/kali/Desktop/StegSeek/.git/objects/pack
1      /home/kali/Desktop/StegSeek/.git/objects/info
2      /home/kali/Desktop/StegSeek/.git/objects
1      /home/kali/Desktop/StegSeek/.git/branches
1      /home/kali/Desktop/StegSeek/.git/info
2      /home/kali/Desktop/StegSeek/.git
1      /home/kali/Desktop/StegSeek/src
1      /home/kali/Desktop/StegSeek/tests/stegseek/data
1      /home/kali/Desktop/StegSeek/tests/stegseek
1      /home/kali/Desktop/StegSeek/tests/steghide/data
1      /home/kali/Desktop/StegSeek/tests/steghide
1      /home/kali/Desktop/StegSeek/tests
1      /home/kali/Desktop/StegSeek/.github/workflows
1      /home/kali/Desktop/StegSeek/.github
4      /home/kali/Desktop/StegSeek
1      /home/kali/Desktop/work
744     /home/kali/Desktop
```

-k – displays information in KB.

```
(kali㉿kali)-[~/Desktop]
$ du -k /home/kali/Desktop
12      /home/kali/Desktop/StegSeek/package
148     /home/kali/Desktop/StegSeek/.demo
64      file1.txt /home/kali/Desktop/StegSeek/.git/hooks
8       /home/kali/Desktop/StegSeek/.git/logs/refs/heads
8       /home/kali/Desktop/StegSeek/.git/logs/refs/remotes/origin
12      /home/kali/Desktop/StegSeek/.git/logs/refs/remotes
24      /home/kali/Desktop/StegSeek/.git/logs/refs
32      /home/kali/Desktop/StegSeek/.git/logs
4       hello.txt /home/kali/Desktop/StegSeek/.git/refs/tags
8       /home/kali/Desktop/StegSeek/.git/refs/heads
8       /home/kali/Desktop/StegSeek/.git/refs/remotes/origin
12      /home/kali/Desktop/StegSeek/.git/refs/remotes
28      /home/kali/Desktop/StegSeek/.git/refs
1692    /home/kali/Desktop/StegSeek/.git/objects/pack
4       /home/kali/Desktop/StegSeek/.git/objects/info
1700    /home/kali/Desktop/StegSeek/.git/objects
4       /home/kali/Desktop/StegSeek/.git/branches
8       /home/kali/Desktop/StegSeek/.git/info
1880    /home/kali/Desktop/StegSeek/.git
772     /home/kali/Desktop/StegSeek/src
40      /home/kali/Desktop/StegSeek/tests/stegseek/data
56      /home/kali/Desktop/StegSeek/tests/stegseek
84      /home/kali/Desktop/StegSeek/tests/steghide/data
444     /home/kali/Desktop/StegSeek/tests/steghide
504     /home/kali/Desktop/StegSeek/tests
16      /home/kali/Desktop/StegSeek/.github/workflows
20      /home/kali/Desktop/StegSeek/.github
3404    /home/kali/Desktop/StegSeek
16      /home/kali/Desktop/work
761740   /home/kali/Desktop
```

top command

The top command displays running processes and the system's real-time condition, including resource utilization. It helps identify resource-intensive processes, enabling you to disable them easily.

To run the command, enter top into your command-line interface. # top

```
(kali㉿kali)-[~/Desktop]
$ top
```

And these are the results displayed


```

alphabets.txt          5.3%] Tasks: 76, 177 thr, 123 kthr; 1 run
0[##*                4.5%] Load average: 0.41 0.41 0.37
1[##*                1.9%] Uptime: 02:12:40
2[*                 1.3%]
3[##*                Mem[|||||]#*662M/1.91G
file[                 Swap[ 0K/1024M]

```

[Main] [I/O]

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
891	root	20	0	386M	134M	60668	S	5.2	6.9	3:27.86	/usr/lib
67299	kali	20	0	8028	4352	3452	R	3.9	0.2	0:00.38	htop -C
1406	kali	20	0	357M	44900	30600	S	3.2	2.2	0:44.87	/usr/bin
1225	kali	20	0	1278M	102M	77396	R	2.6	5.3	1:47.45	xfwm4

-h – displays the help message and exits.

```

└─(kali㉿kali)-[~/Desktop]
$ htop -h
htop 3.2.2
(c) 2004-2019 Hisham Muhammad. (c) 2020-2023 htop dev team.
Released under the GNU GPLv2+.

-C --no-color           Use a monochrome color scheme
-d --delay=DELAY        Set the delay between updates, in tenths of
seconds
-F --filter=FILTER       Show only the commands matching the given f
ilter
-h --help                Print this help screen
-H --highlight-changes [=DELAY] Highlight new and old processes
-M --no-mouse             Disable the mouse

```

ps command

The ps command creates a snapshot of all running processes in your system. Executing it without an option or argument will list the running processes in the shell with the following information:

- Unique process ID (PID).
- Type of the terminal (TTY).
- Running time (TIME).
- Command that launches the process (CMD).

Syntax: # ps

```
└─(kali㉿kali)-[~/Desktop]
```

```
$ ps
```

PID	TTY	TIME	CMD
41710	pts/0	00:00:23	zsh
69376	pts/0	00:00:00	ps

```
└─(kali㉿kali)-[~/Desktop]
```

```
$ elop.txt
```

The ps command accepts several options, including:

-T – displays all processes associated with the current shell session.

-u username – lists processes associated with a specific user.

```
└─(kali㉿kali)-[~/Desktop]
```

```
$ ps -T
```

PID	SPID	TTY	TIME	CMD
41710	41710	pts/0	00:00:23	zsh
69924	69924	pts/0	00:00:00	ps

```
level10.txt
```

```
└─(kali㉿kali)-[~/Desktop]
```

```
$ ps -u
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
kali	41710	0.6	0.4	15076	9080	pts/0	Ss	05:08	0:23	/usr/bin
kali	43806	0.0	0.3	13876	7452	pts/1	Ss	05:12	0:00	/usr/bin
kali	70240	100	0.2	11212	4840	pts/0	R+	06:06	0:00	ps -u

```
└─(kali㉿kali)-[~/Desktop]
```

```
$
```

-A – shows all the running processes.

```
(kali㉿kali)-[~/Desktop]
```

```
$ ps -A
```

PID	TTY	TIME	CMD
1	?	00:00:02	systemd
2	?	00:00:00	kthreadd
3	?	00:00:00	rcu_gp
4	?	00:00:00	rcu_par_gp
5	?	00:00:00	slub_flushwq
6	?	00:00:00	netns
10	?	00:00:00	mm_percpu_wq
11	?	00:00:00	rcu_tasks_kthread
12	?	00:00:00	rcu_tasks_rude_kthread
13	?	00:00:00	rcu_tasks_trace_kthread

Scroll till the end

37312	?	00:00:01	kworker/3:2-events
63877	?	00:00:00	kworker/2:1-ata_sff
65349	?	00:00:00	kworker/u64:1-events_unbound
66384	?	00:00:00	kworker/2:0-ata_sff
68032	?	00:00:00	kworker/u64:0-writeback
68909	?	00:00:00	kworker/2:2-events
70508	pts/0	00:00:00	ps

```
(kali㉿kali)-[~/Desktop]
```

```
$
```

kill command

Use the kill command to terminate an unresponsive program using its identification number (PID). To check the PID, run the following command: # ps ux

To stop the program, enter the syntax below: kill [signal_option] pid

```
kali      72561  0.0  0.2  11212  47
```

```
└─(kali㉿kali)-[~/Desktop]  
└─$
```

```
└─(kali㉿kali)-[~/Desktop]  
└─$ kill 72561
```

There are 64 signals for terminating a program, but SIGTERM and SIGKILL are the most common. SIGTERM is the default signal that lets the program save its progress before stopping. Meanwhile, SIGKILL forces programs to stop and discard unsaved progress.

The advanced commands are more and not limited to the ones demonstrated here. Please go ahead and explore by making further research.