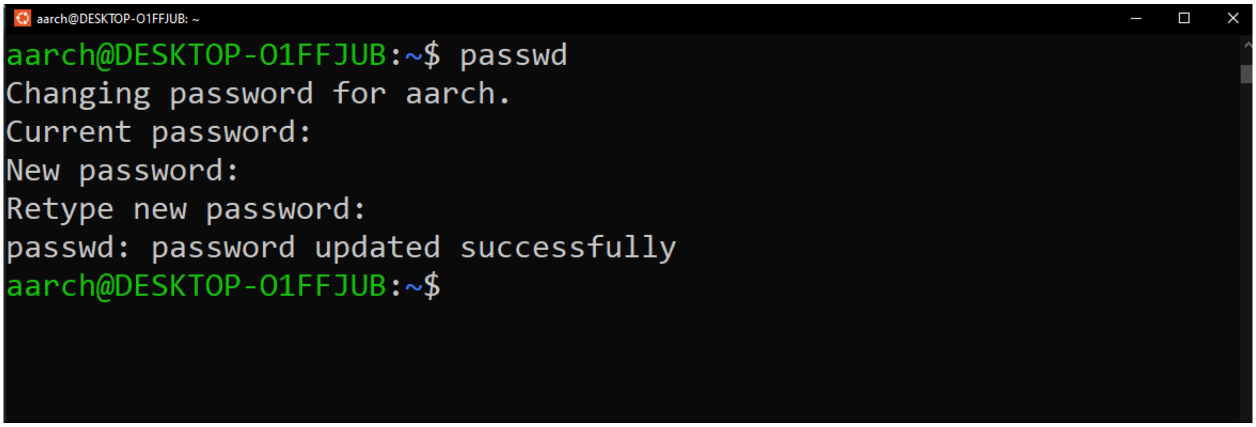
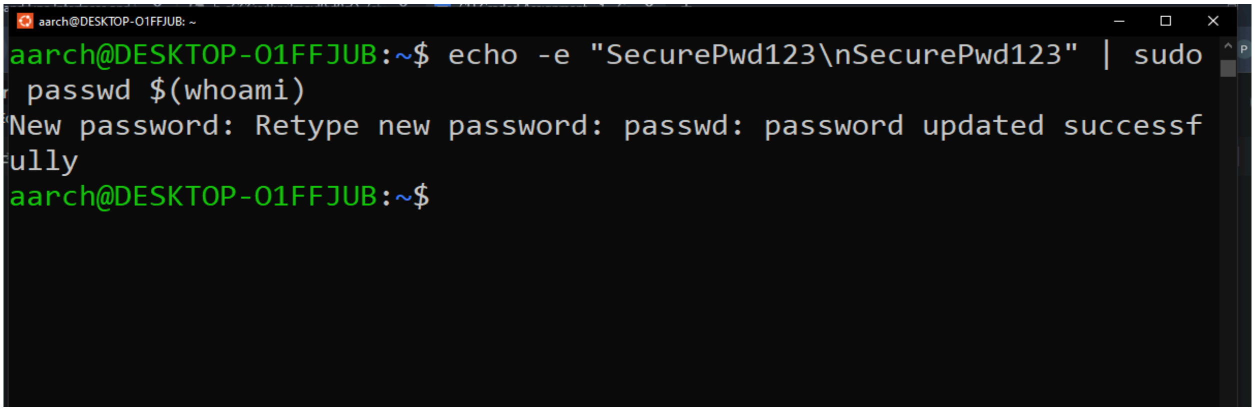
Q1

A



```
aarch@DESKTOP-O1FFJUB:~$ passwd
Changing password for aarch.
Current password:
New password:
Retype new password:
passwd: password updated successfully
aarch@DESKTOP-O1FFJUB:~$
```

To change password to user account, passwd command is used. But this command does not show if we have actually changed the password to “SecurePwd123”, as the password is hidden for security purposes.

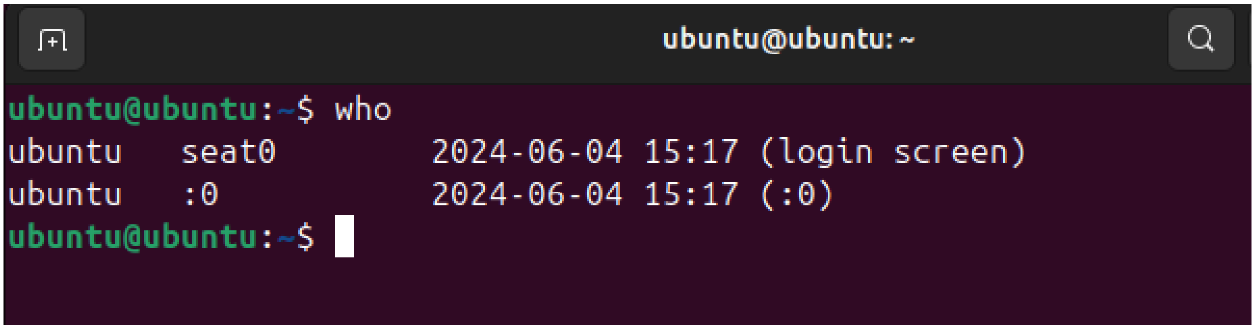


```
aarch@DESKTOP-O1FFJUB:~$ echo -e "SecurePwd123\nSecurePwd123" | sudo
 passwd $(whoami)
New password: Retype new password: passwd: password updated successf
ully
aarch@DESKTOP-O1FFJUB:~$
```

Alternatively, we can use the above method to show what password we are setting and to which user.

Here, as “sudo passwd” is used instead of “passwd”, it doesn’t prompt for the old password and only the new password is echoed and redirected as an input for the prompts from passwd.

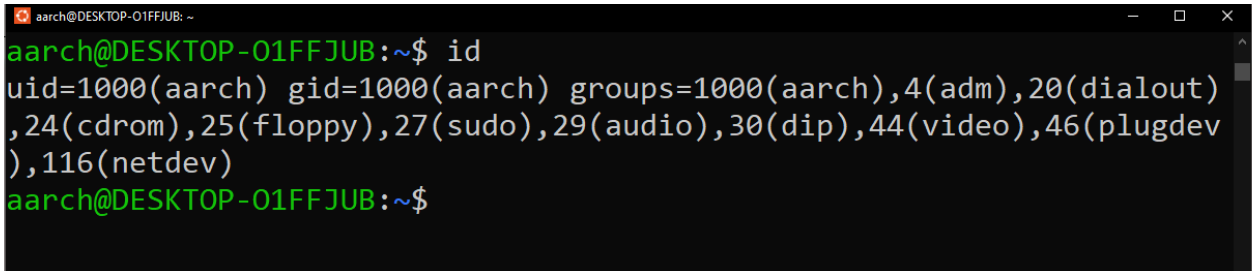
B



```
ubuntu@ubuntu:~$ who
ubuntu   seat0        2024-06-04 15:17 (login screen)
ubuntu   :0           2024-06-04 15:17 (:0)
ubuntu@ubuntu:~$ 
```

“who” command shows currently logged in users.

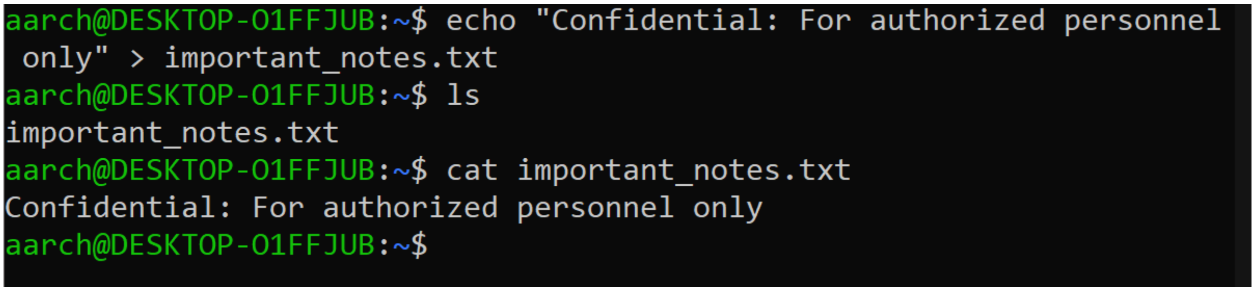
C



```
aarch@DESKTOP-O1FFJUB:~$ id
uid=1000(aarch) gid=1000(aarch) groups=1000(aarch),4(adm),20(dialout)
,24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev
),116(netdev)
aarch@DESKTOP-O1FFJUB:~$
```

Here we have used the “id” command to check user id against a username.

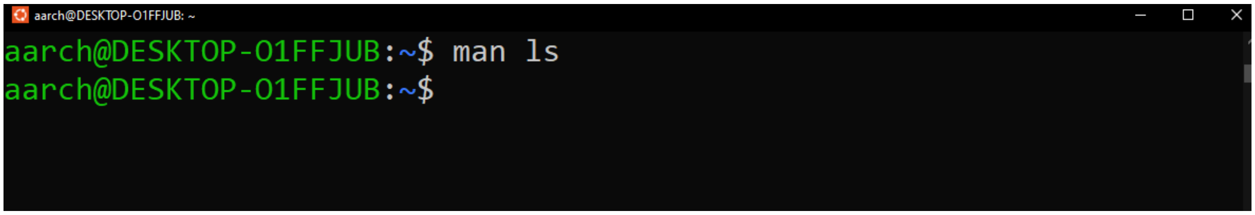
D



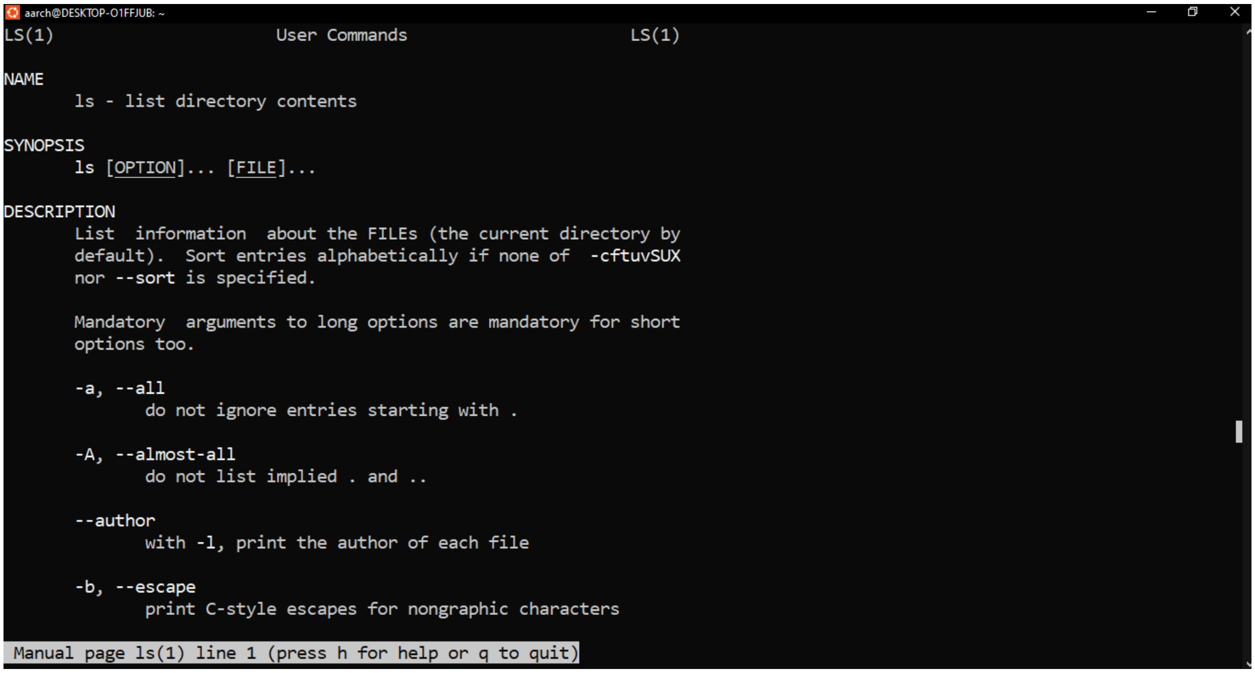
```
aarch@DESKTOP-O1FFJUB:~$ echo "Confidential: For authorized personnel
 only" > important_notes.txt
aarch@DESKTOP-O1FFJUB:~$ ls
important_notes.txt
aarch@DESKTOP-O1FFJUB:~$ cat important_notes.txt
Confidential: For authorized personnel only
aarch@DESKTOP-O1FFJUB:~$
```

“echo” command displays any text passed to it as an argument but here we are **redirecting** the text to the “important_notes.txt” file using the ">” operator.
“ls” command is used to show that an “important_notes.txt” file is created.
“cat important_notes.txt” command reads and displays the content of the “important_notes.txt” file

E



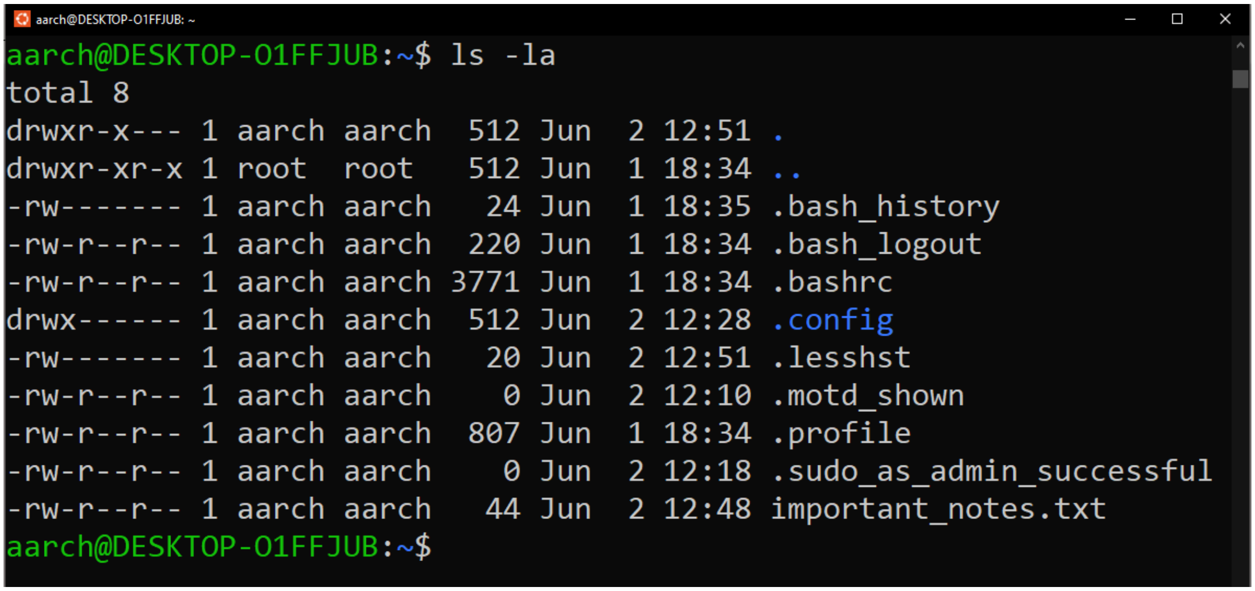
```
aarch@DESKTOP-O1FFJUB:~$ man ls
aarch@DESKTOP-O1FFJUB:~$
```



```
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
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

 Manual page ls(1) line 1 (press h for help or q to quit)
```

“man ls” command is used to display the user manual of the “ls” command. It provides a detailed view of commands which includes NAME, SYNOPSIS, DESCRIPTION, OPTIONS, etc.

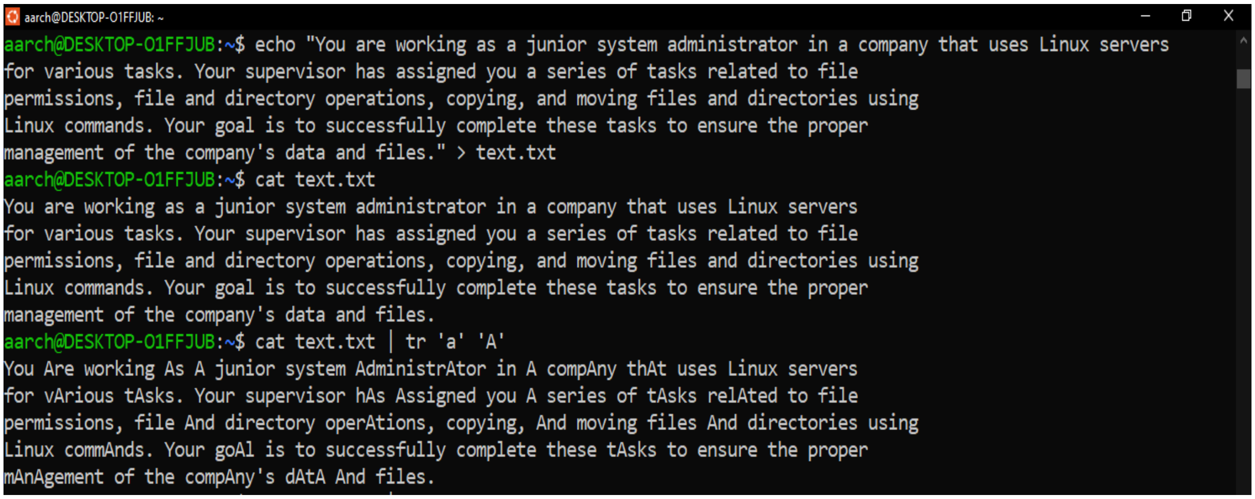
F



```
aarch@DESKTOP-O1FFJUB:~$ ls -la
total 8
drwxr-x--- 1 aarch aarch  512 Jun  2 12:51 .
drwxr-xr-x 1 root  root   512 Jun  1 18:34 ..
-rw------- 1 aarch aarch   24 Jun  1 18:35 .bash_history
-rw-r--r-- 1 aarch aarch  220 Jun  1 18:34 .bash_logout
-rw-r--r-- 1 aarch aarch 3771 Jun  1 18:34 .bashrc
drwx------ 1 aarch aarch  512 Jun  2 12:28 .config
-rw------- 1 aarch aarch   20 Jun  2 12:51 .lesshst
-rw-r--r-- 1 aarch aarch    0 Jun  2 12:10 .motd_shown
-rw-r--r-- 1 aarch aarch  807 Jun  1 18:34 .profile
-rw-r--r-- 1 aarch aarch    0 Jun  2 12:18 .sudo_as_admin_successful
-rw-r--r-- 1 aarch aarch   44 Jun  2 12:48 important_notes.txt
aarch@DESKTOP-O1FFJUB:~$
```

“ls” command is used to list the contents of a directory
“-l” option is used to list contents of a directory in long format
“-a” option is used to display all files including hidden files and directory

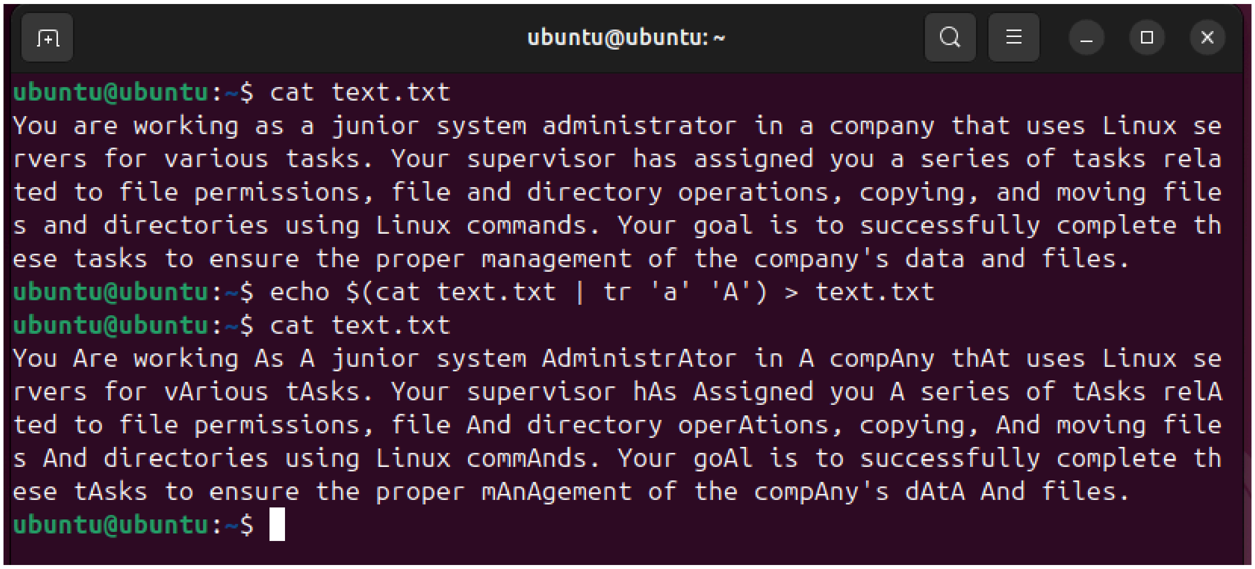
G



```
aarch@DESKTOP-O1FFJUB:~$ echo "You are working as a junior system administrator in a company that uses Linux servers
for various tasks. Your supervisor has assigned you a series of tasks related to file
permissions, file and directory operations, copying, and moving files and directories using
Linux commands. Your goal is to successfully complete these tasks to ensure the proper
management of the company's data and files." > text.txt
aarch@DESKTOP-O1FFJUB:~$ cat text.txt
You are working as a junior system administrator in a company that uses Linux servers
for various tasks. Your supervisor has assigned you a series of tasks related to file
permissions, file and directory operations, copying, and moving files and directories using
Linux commands. Your goal is to successfully complete these tasks to ensure the proper
management of the company's data and files.
aarch@DESKTOP-O1FFJUB:~$ cat text.txt | tr 'a' 'A'
You Are working As A junior system AdministrAtor in A compAny thAt uses Linux servers
for vArious tAsks. Your supervisor hAs Assigned you A series of tAsks relAted to file
permissions, file And directory operAtions, copying, And moving files And directories using
Linux commAnds. Your goAl is to successfully complete these tAsks to ensure the proper
mAnAgement of the compAny's dAtA And files.
```

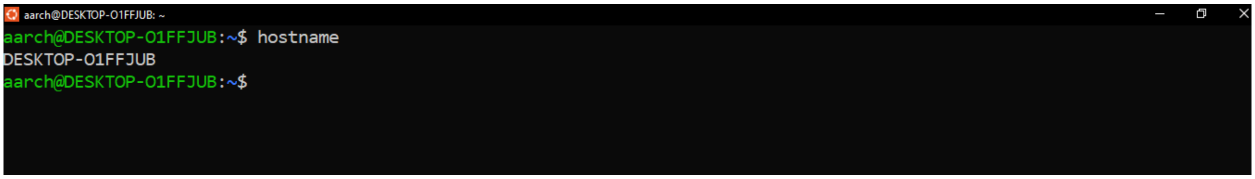
Here, we have used the “echo” command which will take the text as an argument and then we have used “>” to redirect that text to the “text.txt” file.
To show that the text is properly added to the file, we have used “cat” command.
Next we have used “cat” command which will display the text in file as output but using “|” (pipe command) we have connected the output of cat command into input of “tr” command. “Tr” command performs text transformation, here it will convert all “a” to “A”. The output is as displayed in the above screenshot.

Using the above commands we have successfully converted all a’s in the text to uppercase and displayed it but the changes won’t be reflected in the text.txt file. To do that we can take a few extra steps and do as shown in the below screenshot.



```
ubuntu@ubuntu:~$ cat text.txt
You are working as a junior system administrator in a company that uses Linux se
rvers for various tasks. Your supervisor has assigned you a series of tasks rela
ted to file permissions, file and directory operations, copying, and moving file
s and directories using Linux commands. Your goal is to successfully complete th
ese tasks to ensure the proper management of the company's data and files.
ubuntu@ubuntu:~$ echo $(cat text.txt | tr 'a' 'A') > text.txt
ubuntu@ubuntu:~$ cat text.txt
You Are working As A junior system AdministrAtor in A compAny thAt uses Linux se
rvers for vArious tAsks. Your supervisor hAs Assigned you A series of tAsks relA
ted to file permissions, file And directory operAtions, copying, And moving file
s And directories using Linux commAnds. Your goAl is to successfully complete th
ese tAsks to ensure the proper mAnAgement of the compAny's dAtA And files.
ubuntu@ubuntu:~$ 
```

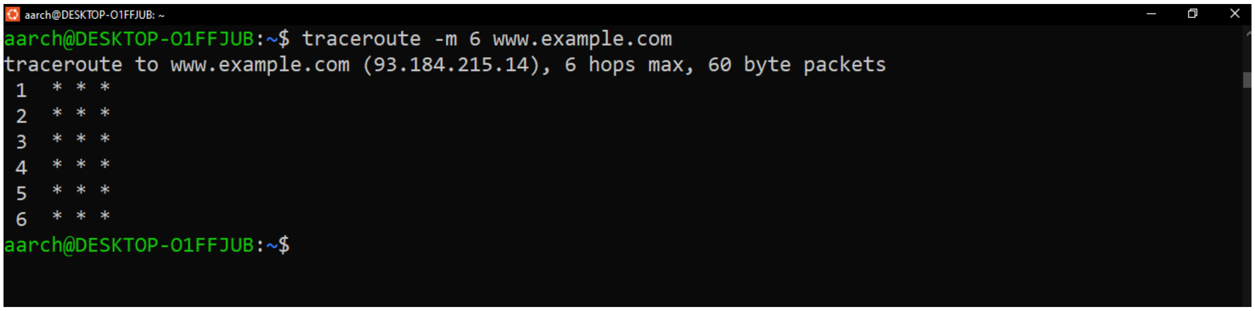
H



```
aarch@DESKTOP-O1FFJUB:~$ hostname
DESKTOP-O1FFJUB
aarch@DESKTOP-O1FFJUB:~$
```

“hostname” command prints your computer/server name.

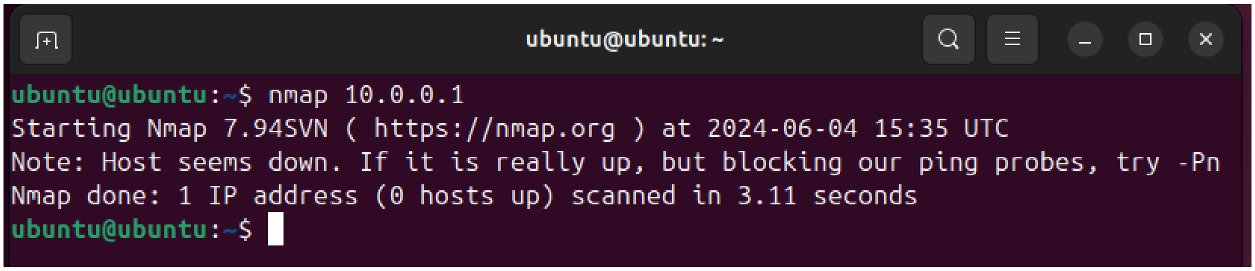
I



```
aarch@DESKTOP-O1FFJUB:~$ traceroute -m 6 www.example.com
traceroute to www.example.com (93.184.215.14), 6 hops max, 60 byte packets
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
aarch@DESKTOP-O1FFJUB:~$
```

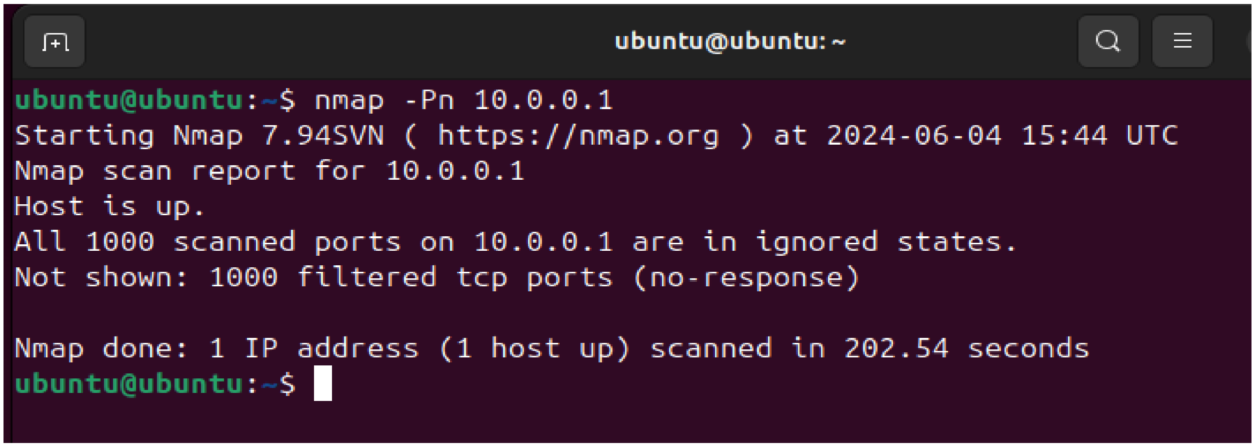
The “traceroute” command is a network diagnostic tool used to trace the route taken by packets from a source to a destination over an IP network.

J



```
ubuntu@ubuntu:~$ nmap 10.0.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-04 15:35 UTC
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.11 seconds
ubuntu@ubuntu:~$
```

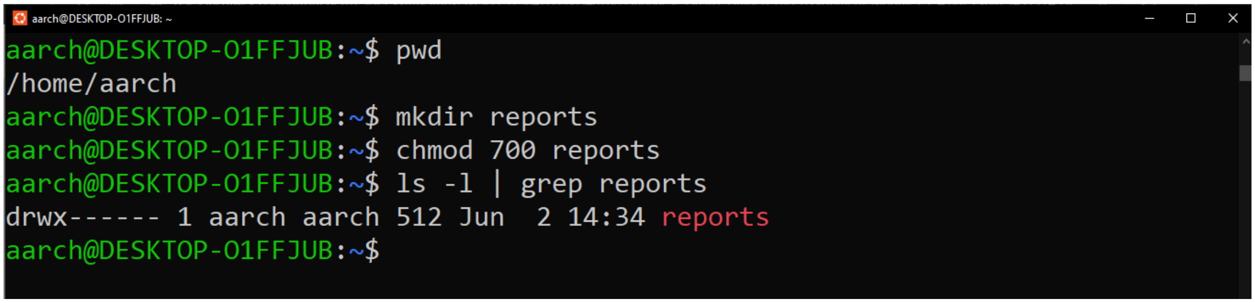
“nmap” tool is used to perform network and port scanning.
We can do basic port scan using the above shown command.
As the basic command was not able to ping the host because by default nmap only probes active machines, there is an alternative way to bypass the host discovery.
Using the “-Pn” option, nmap will scan every IP provided as if it is active, as shown in below screenshot.



```
ubuntu@ubuntu:~$ nmap -Pn 10.0.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-04 15:44 UTC
Nmap scan report for 10.0.0.1
Host is up.
All 1000 scanned ports on 10.0.0.1 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 202.54 seconds
ubuntu@ubuntu:~$
```

Q2

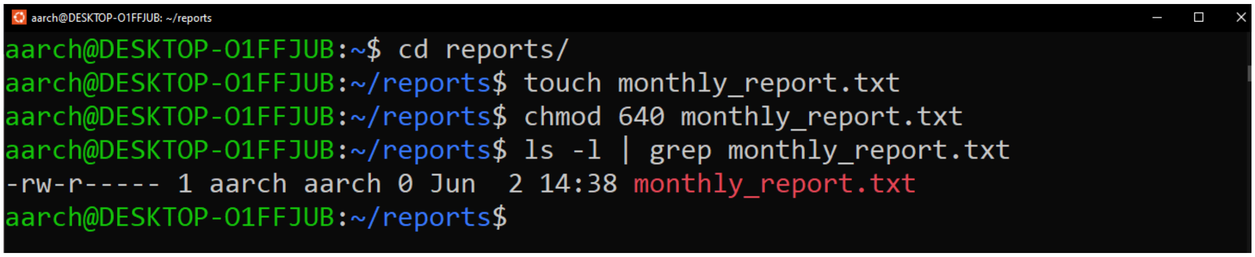
A



```
aarch@DESKTOP-O1FFJUB:~$ pwd
/home/aarch
aarch@DESKTOP-O1FFJUB:~$ mkdir reports
aarch@DESKTOP-O1FFJUB:~$ chmod 700 reports
aarch@DESKTOP-O1FFJUB:~$ ls -l | grep reports
drwx------ 1 aarch aarch 512 Jun  2 14:34 reports
aarch@DESKTOP-O1FFJUB:~$
```

Here we have used the “chmod” command to set the permission to “reports” directory. Numeric “700” indicates that only the owner can read, write, and execute in this directory, while others have no access. This can be seen in the output of “ls -l”.

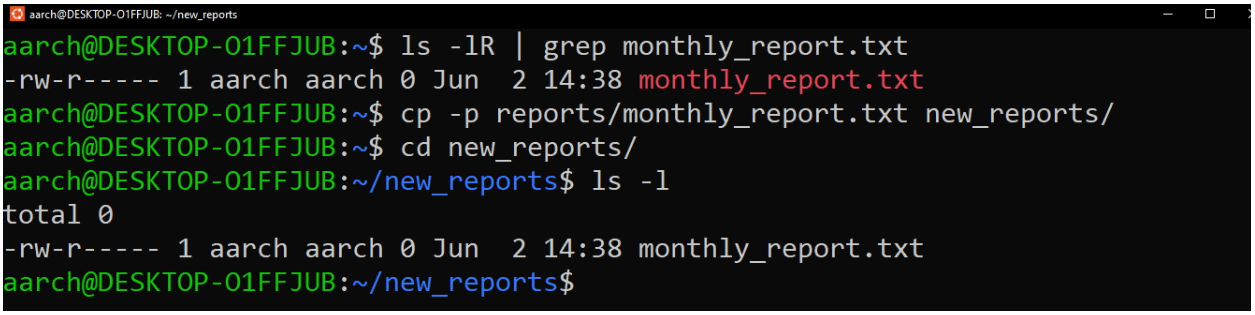
B



```
aarch@DESKTOP-O1FFJUB:~$ cd reports/
aarch@DESKTOP-O1FFJUB:~/reports$ touch monthly_report.txt
aarch@DESKTOP-O1FFJUB:~/reports$ chmod 640 monthly_report.txt
aarch@DESKTOP-O1FFJUB:~/reports$ ls -l | grep monthly_report.txt
-rw-r----- 1 aarch aarch 0 Jun  2 14:38 monthly_report.txt
aarch@DESKTOP-O1FFJUB:~/reports$
```

Here we have used the “chmod” command to set the permission to the “monthly_report.txt” file. Numeric “640” indicates that the owner can read and write, the group can read, and others have no access. This can be seen in the output of “ls -l”.

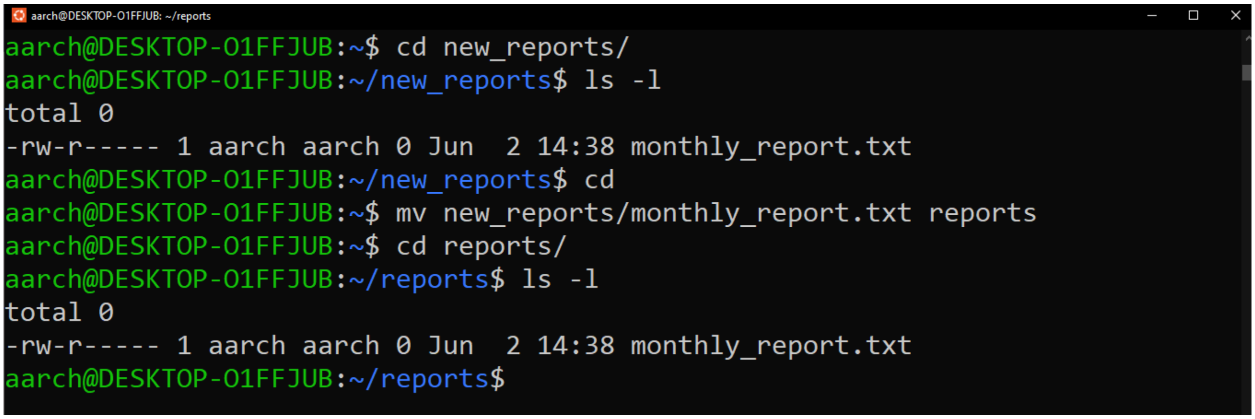
C



```
aarch@DESKTOP-O1FFJUB:~$ ls -lR | grep monthly_report.txt
-rw-r----- 1 aarch aarch 0 Jun  2 14:38 monthly_report.txt
aarch@DESKTOP-O1FFJUB:~$ cp -p reports/monthly_report.txt new_reports/
aarch@DESKTOP-O1FFJUB:~$ cd new_reports/
aarch@DESKTOP-O1FFJUB:~/new_reports$ ls -l
total 0
-rw-r----- 1 aarch aarch 0 Jun  2 14:38 monthly_report.txt
aarch@DESKTOP-O1FFJUB:~/new_reports$
```

“-p” option of the “cp” command preserves the permissions of the copied file or directory. This can be seen in the output of “ls -l”.

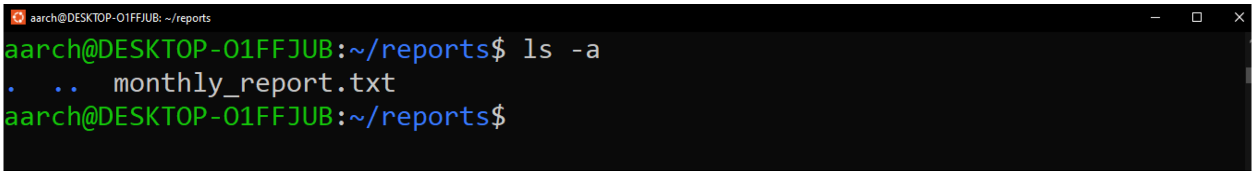
D



```
aarch@DESKTOP-O1FFJUB:~$ cd new_reports/
aarch@DESKTOP-O1FFJUB:~/new_reports$ ls -l
total 0
-rw-r----- 1 aarch aarch 0 Jun  2 14:38 monthly_report.txt
aarch@DESKTOP-O1FFJUB:~/new_reports$ cd
aarch@DESKTOP-O1FFJUB:~$ mv new_reports/monthly_report.txt reports
aarch@DESKTOP-O1FFJUB:~$ cd reports/
aarch@DESKTOP-O1FFJUB:~/reports$ ls -l
total 0
-rw-r----- 1 aarch aarch 0 Jun  2 14:38 monthly_report.txt
aarch@DESKTOP-O1FFJUB:~/reports$
```

“mv” command preserves the permissions of files and directories by default while moving them. This can be seen in the output of “ls -l”.

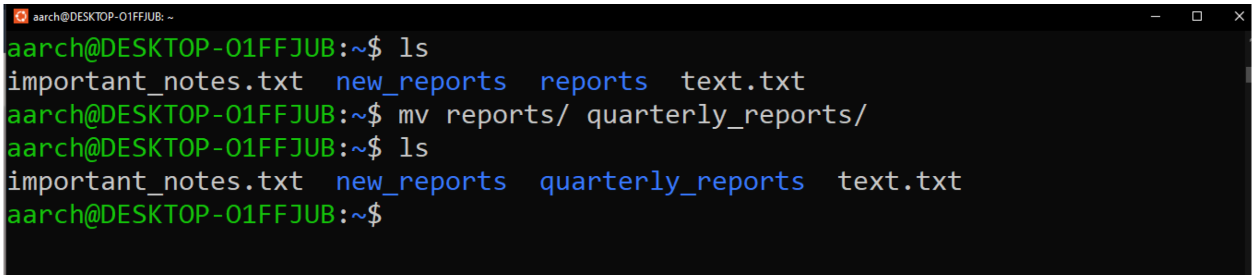
E



```
aarch@DESKTOP-O1FFJUB:~/reports$ ls -a
.  ..  monthly_report.txt
aarch@DESKTOP-O1FFJUB:~/reports$
```

“-a” option of the “ls” command prints all files and directories including hidden ones.

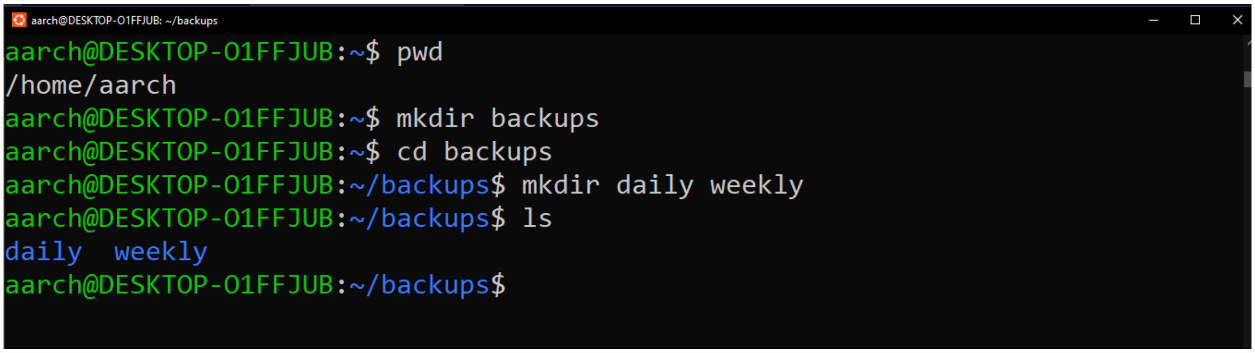
F



```
aarch@DESKTOP-O1FFJUB:~$ ls
important_notes.txt  new_reports  reports  text.txt
aarch@DESKTOP-O1FFJUB:~$ mv reports/ quarterly_reports/
aarch@DESKTOP-O1FFJUB:~$ ls
important_notes.txt  new_reports  quarterly_reports  text.txt
aarch@DESKTOP-O1FFJUB:~$
```

“mv” command is also used to rename a file or directory.

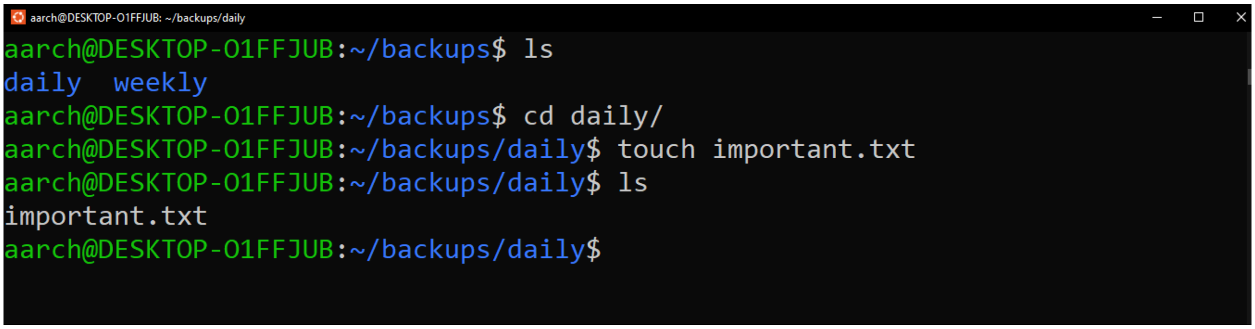
G



```
aarch@DESKTOP-O1FFJUB:~$ pwd
/home/aarch
aarch@DESKTOP-O1FFJUB:~$ mkdir backups
aarch@DESKTOP-O1FFJUB:~$ cd backups
aarch@DESKTOP-O1FFJUB:~/backups$ mkdir daily weekly
aarch@DESKTOP-O1FFJUB:~/backups$ ls
daily  weekly
aarch@DESKTOP-O1FFJUB:~/backups$
```

“mkdir” command is used to create a new directory and “cd” command is used to change or navigate to a specific directory.

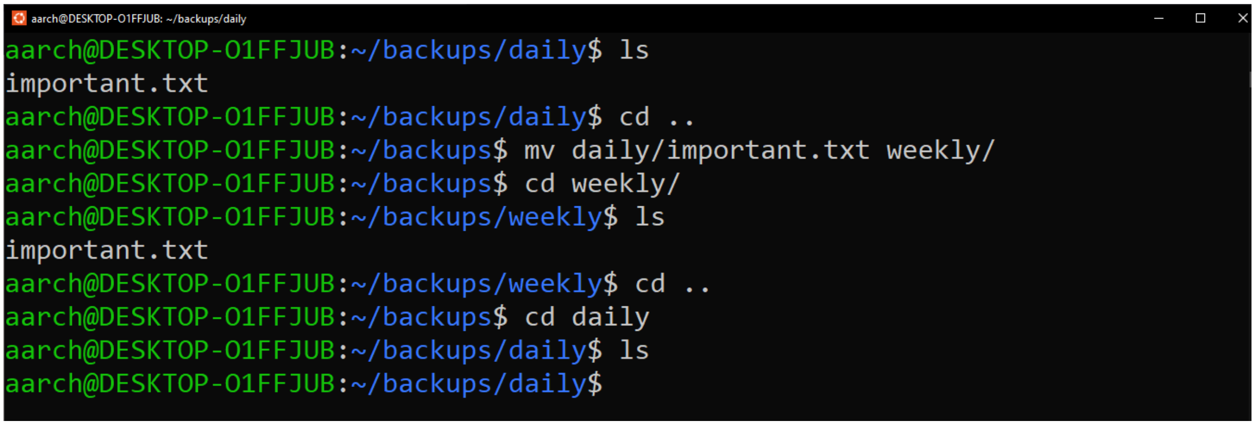
H



```
aarch@DESKTOP-O1FFJUB:~/backups$ ls
daily  weekly
aarch@DESKTOP-O1FFJUB:~/backups$ cd daily/
aarch@DESKTOP-O1FFJUB:~/backups/daily$ touch important.txt
aarch@DESKTOP-O1FFJUB:~/backups/daily$ ls
important.txt
aarch@DESKTOP-O1FFJUB:~/backups/daily$
```

“touch” command can be used to create a new empty file, if it does not already exists.

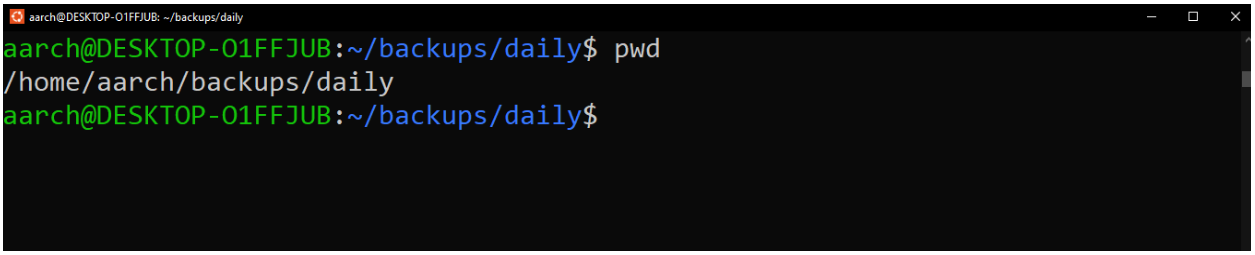
I



```
aarch@DESKTOP-O1FFJUB:~/backups/daily$ ls
important.txt
aarch@DESKTOP-O1FFJUB:~/backups/daily$ cd ..
aarch@DESKTOP-O1FFJUB:~/backups$ mv daily/important.txt weekly/
aarch@DESKTOP-O1FFJUB:~/backups$ cd weekly/
aarch@DESKTOP-O1FFJUB:~/backups/weekly$ ls
important.txt
aarch@DESKTOP-O1FFJUB:~/backups/weekly$ cd ..
aarch@DESKTOP-O1FFJUB:~/backups$ cd daily
aarch@DESKTOP-O1FFJUB:~/backups/daily$ ls
aarch@DESKTOP-O1FFJUB:~/backups/daily$
```

If the final argument of “mv” command is an existing directory, the source files/directories are moved to the target directory with their names being preserved.

J

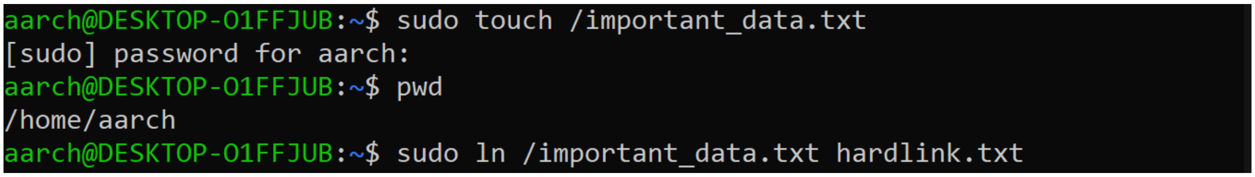


```
aarch@DESKTOP-O1FFJUB:~/backups/daily$ pwd
/home/aarch/backups/daily
aarch@DESKTOP-O1FFJUB:~/backups/daily$
```

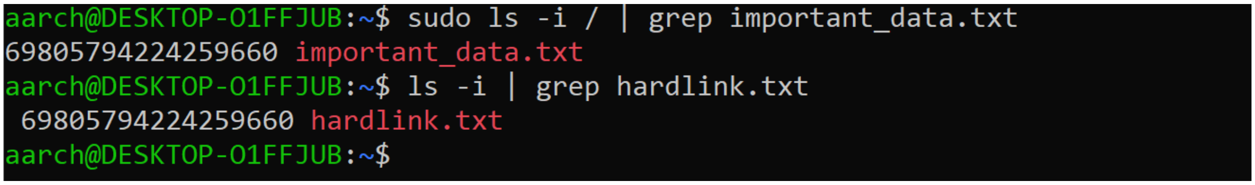
“pwd” command will display (print) the full path of the current working directory.

Q3

A



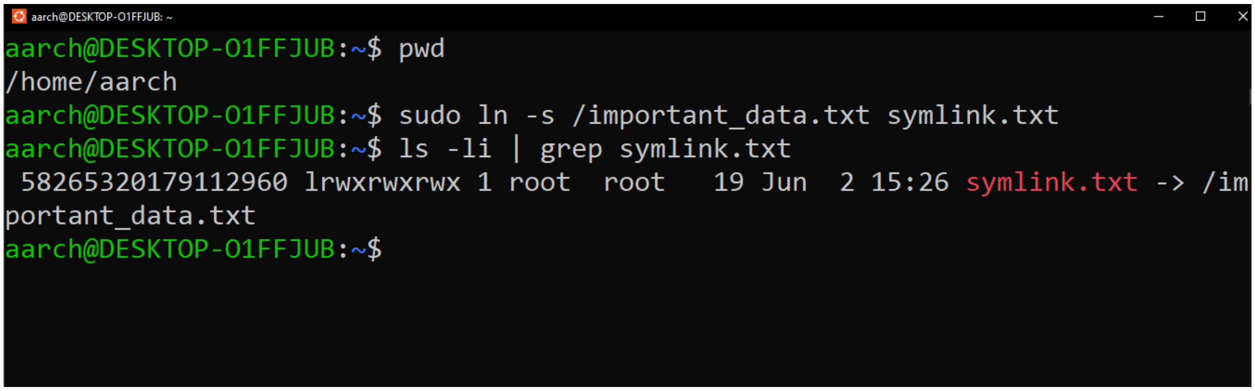
```
aarch@DESKTOP-O1FFJUB:~$ sudo touch /important_data.txt
[sudo] password for aarch:
aarch@DESKTOP-O1FFJUB:~$ pwd
/home/aarch
aarch@DESKTOP-O1FFJUB:~$ sudo ln /important_data.txt hardlink.txt
```



```
aarch@DESKTOP-O1FFJUB:~$ sudo ls -i / | grep important_data.txt
69805794224259660 important_data.txt
aarch@DESKTOP-O1FFJUB:~$ ls -i | grep hardlink.txt
 69805794224259660 hardlink.txt
aarch@DESKTOP-O1FFJUB:~$
```

“ln” command is used to create links. If no options/flags are provided, it creates a hard link. ‘ls -i’ is used to find the inode number of a file. As we can see in the output, the inode number for both “important_data.txt” and “hardlink.txt” is the same.

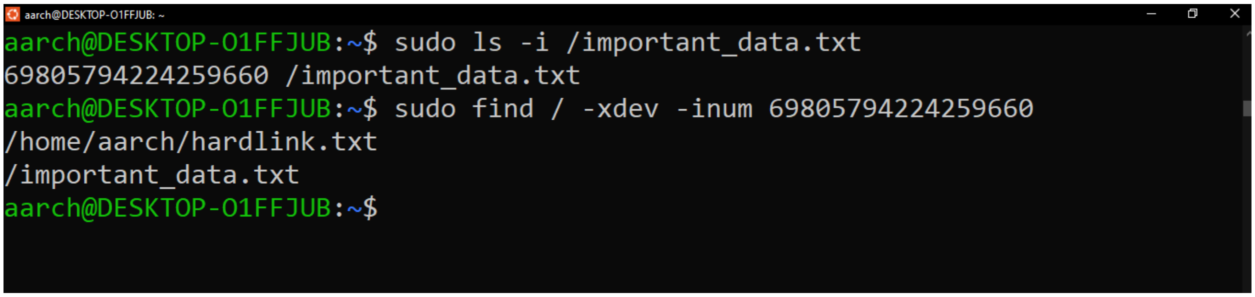
B



```
aarch@DESKTOP-O1FFJUB:~$ pwd
/home/aarch
aarch@DESKTOP-O1FFJUB:~$ sudo ln -s /important_data.txt symlink.txt
aarch@DESKTOP-O1FFJUB:~$ ls -li | grep symlink.txt
 58265320179112960 lrwxrwxrwx 1 root  root   19 Jun  2 15:26 symlink.txt -> /im
portant_data.txt
aarch@DESKTOP-O1FFJUB:~$
```

When the “-s” option is used with “ln“, it creates a symbolic link. As is visible in the output of “ls -li”, the inode number for the symlink is different from that of the source file (in previous question). The permissions are also printed in the output of “ls -li.

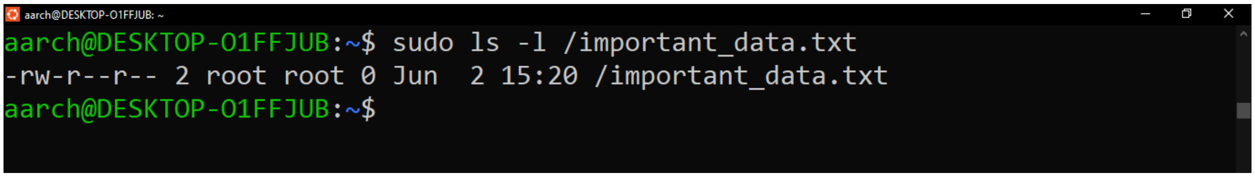
C



```
aarch@DESKTOP-O1FFJUB:~$ sudo ls -i /important_data.txt
69805794224259660 /important_data.txt
aarch@DESKTOP-O1FFJUB:~$ sudo find / -xdev -inum 69805794224259660
/home/aarch/hardlink.txt
/important_data.txt
aarch@DESKTOP-O1FFJUB:~$
```

The inode number of “important_data.txt” is found using “ls -i”. A query is made on the root dir using the “find” command. Here, the “-inum” option indicates that the query is for the inode number provided as an argument. The “-xdev” option is used to make sure that other filesystems are not queried.

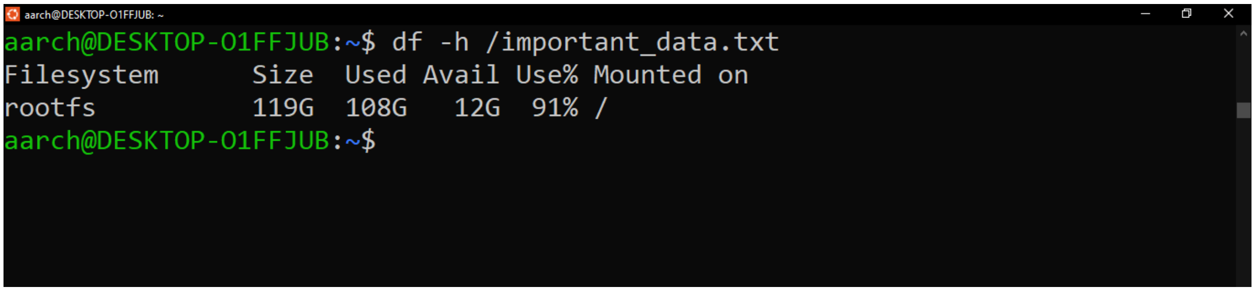
D



```
aarch@DESKTOP-O1FFJUB:~$ sudo ls -l /important_data.txt
-rw-r--r-- 2 root root 0 Jun  2 15:20 /important_data.txt
aarch@DESKTOP-O1FFJUB:~$
```

The relevant details can be found using the “ls -l” command.

E



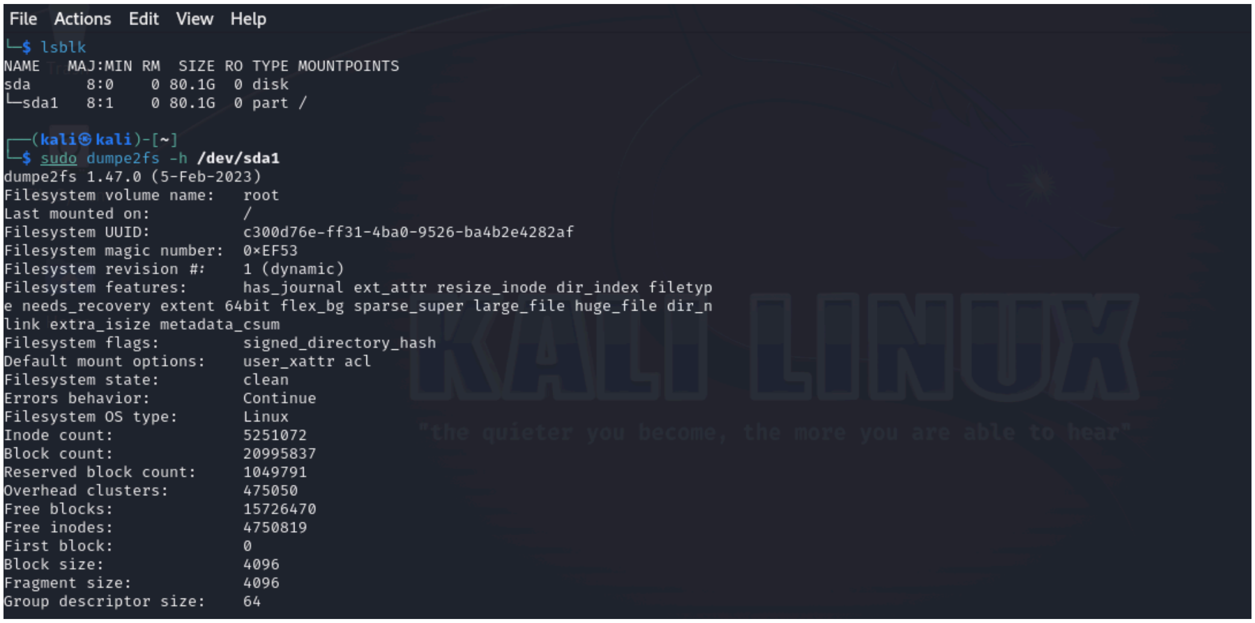
```
aarch@DESKTOP-O1FFJUB:~$ df -h /important_data.txt
Filesystem      Size  Used Avail Use% Mounted on
rootfs          119G  108G   12G  91% /
aarch@DESKTOP-O1FFJUB:~$
```

“df -h” prints the size, storage used and free space of the filesystem that contains the file provided as an argument.

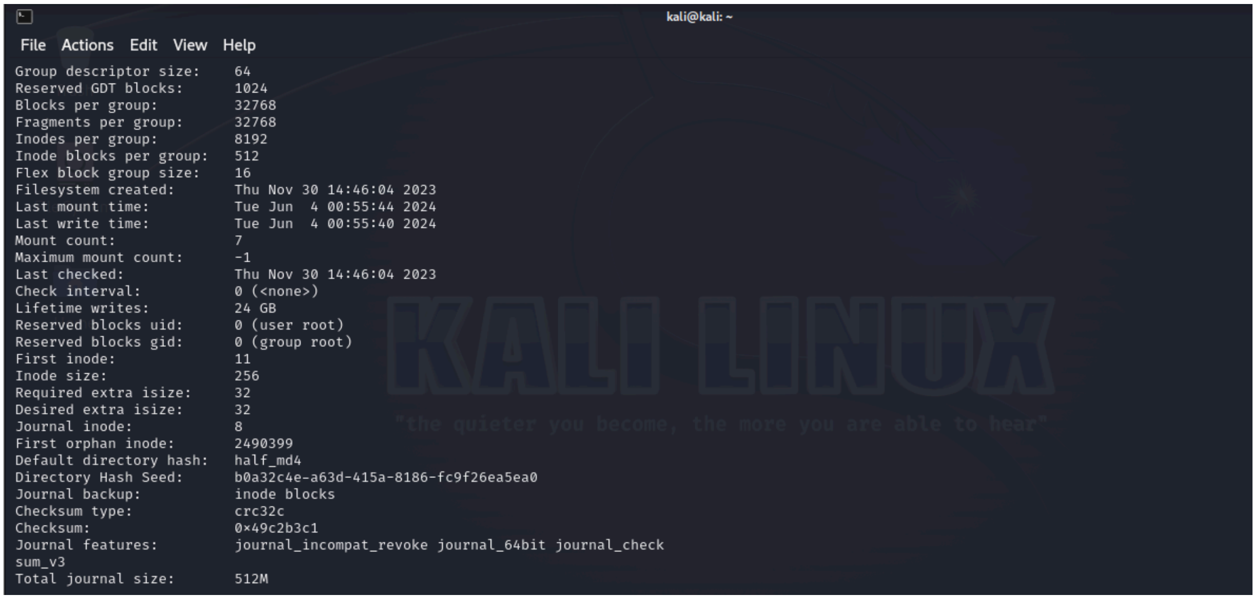
F

(This commands were not running properly on Ubuntu WSL, coursera labs & Ubuntu VM, so i have used Kali Linux VM, hope it will be considered)

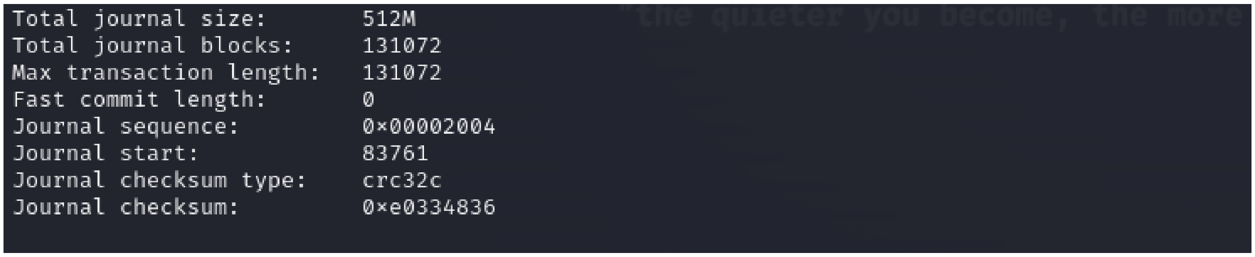
The available filesystems can be identified from the output of the “lsblk” command.



```
└─$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda      8:0    0 80.1G  0 disk
└─sda1   8:1    0 80.1G  0 part /

┌──(kali㉿kali)-[~]
└─$ sudo dumpe2fs -h /dev/sda1
dumpe2fs 1.47.0 (5-Feb-2023)
Filesystem volume name:   root
Last mounted on:          /
Filesystem UUID:          c300d76e-ff31-4ba0-9526-ba4b2e4282af
Filesystem magic number:  0×EF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetyp
e needs_recovery extent 64bit flex_bg sparse_super large_file huge_file dir_n
link extra_isize metadata_csum
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              5251072
Block count:              20995837
Reserved block count:     1049791
Overhead clusters:        475050
Free blocks:              15726470
Free inodes:              4750819
First block:              0
Block size:               4096
Fragment size:            4096
Group descriptor size:    64
```



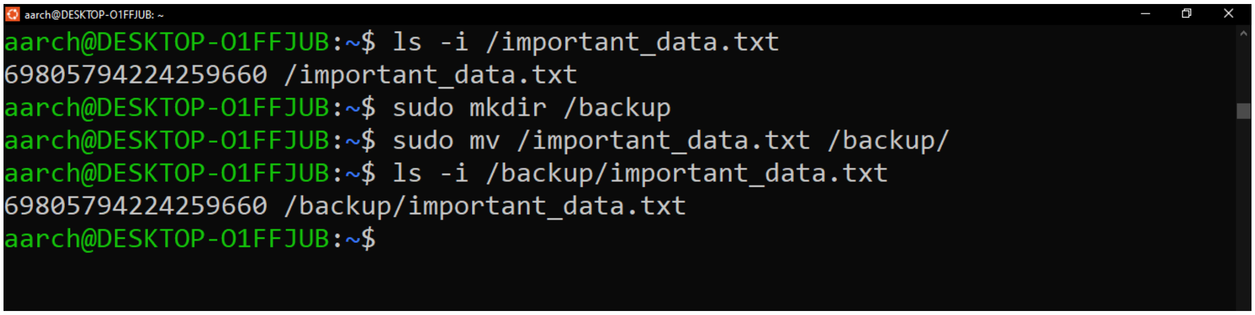
```
Group descriptor size:    64
Reserved GDT blocks:      1024
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         8192
Inode blocks per group:   512
Flex block group size:    16
Filesystem created:       Thu Nov 30 14:46:04 2023
Last mount time:          Tue Jun  4 00:55:44 2024
Last write time:          Tue Jun  4 00:55:40 2024
Mount count:              7
Maximum mount count:      -1
Last checked:             Thu Nov 30 14:46:04 2023
Check interval:           0 (<none>)
Lifetime writes:          24 GB
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               256
Required extra isize:     32
Desired extra isize:      32
Journal inode:            8
First orphan inode:       2490399
Default directory hash:   half_md4
Directory Hash Seed:      b0a32c4e-a63d-415a-8186-fc9f26ea5ea0
Journal backup:           inode blocks
Checksum type:            crc32c
Checksum:                 0×49c2b3c1
Journal features:         journal_incompat_revoke journal_64bit journal_check
sum_v3
Total journal size:       512M
```



```
Total journal size:       512M
Total journal blocks:     131072
Max transaction length:   131072
Fast commit length:       0
Journal sequence:         0×00002004
Journal start:            83761
Journal checksum type:    crc32c
Journal checksum:         0×e0334836
```

“dumpe2fs” command is used to print the superblock data of the filesystem provided as an argument.

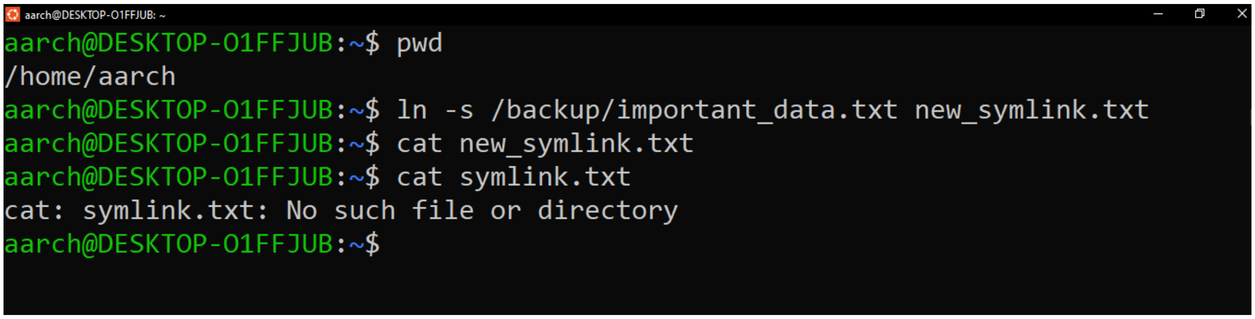
G



```
aarch@DESKTOP-O1FFJUB:~$ ls -i /important_data.txt
69805794224259660 /important_data.txt
aarch@DESKTOP-O1FFJUB:~$ sudo mkdir /backup
aarch@DESKTOP-O1FFJUB:~$ sudo mv /important_data.txt /backup/
aarch@DESKTOP-O1FFJUB:~$ ls -i /backup/important_data.txt
69805794224259660 /backup/important_data.txt
aarch@DESKTOP-O1FFJUB:~$
```

As visible in the output of both “ls -i” commands used here, relocating a file does not change the inode number associated with it.

H



```
aarch@DESKTOP-O1FFJUB:~$ pwd
/home/aarch
aarch@DESKTOP-O1FFJUB:~$ ln -s /backup/important_data.txt new_symlink.txt
aarch@DESKTOP-O1FFJUB:~$ cat new_symlink.txt
aarch@DESKTOP-O1FFJUB:~$ cat symlink.txt
cat: symlink.txt: No such file or directory
aarch@DESKTOP-O1FFJUB:~$
```

Here, as “important_data.txt” is an empty file, using “cat” on a symlink pointing to it, there is supposed to be no output.
However, an error may indicate that the file does not exist (here, meaning that the link is invalid).
As we can see in the output of “cat”, as the location of “important_data.txt” is changed, the new link is valid and the old symlink is now invalid.

I

The information related to inodes are specific to a particular superblock and since superblocks are unique to each filesystem, the same inode number in different filesystems may contain different data. So if a hard link is made across different filesystems, the link would point to the corresponding inode within the filesystem containing the link, so the link would not actually point to the source file.

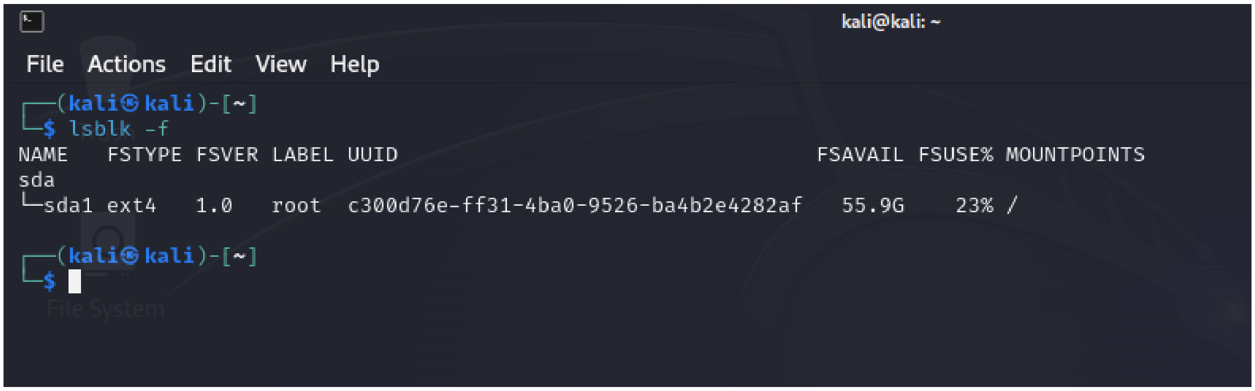
J

If a hard link is made to a file/directory, deleting the source file won’t affect the data blocks as the inode is still associated with the hard link, so to create a backup without duplicating files, a hard link can be made to the source file, which would preserve the contents even if the source file would accidentally be deleted.

Q4

(A few of the commands were not running properly on Ubuntu WSL, coursera labs & Ubuntu VM, so i have used Kali Linux VM, hope it will be considered)

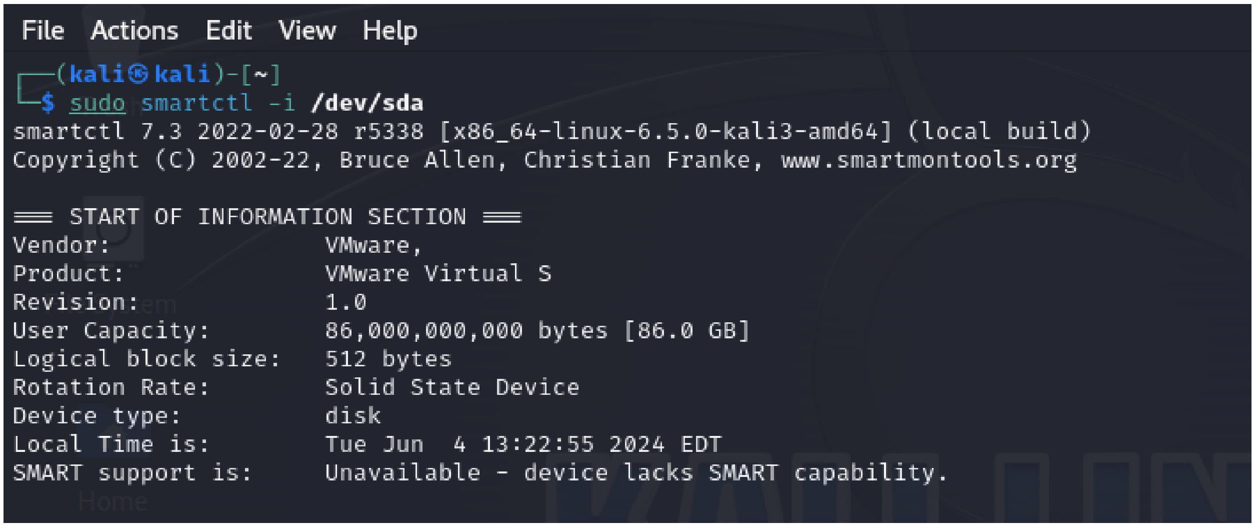
A



```
┌──(kali㉿kali)-[~]
└─$ lsblk -f
NAME   FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINTS
sda
└─sda1 ext4   1.0   root  c300d76e-ff31-4ba0-9526-ba4b2e4282af   55.9G    23% /

┌──(kali㉿kali)-[~]
└─$ 
```

“lsblk” lists the block devices available on a system along with the device name, size and mountpoint.

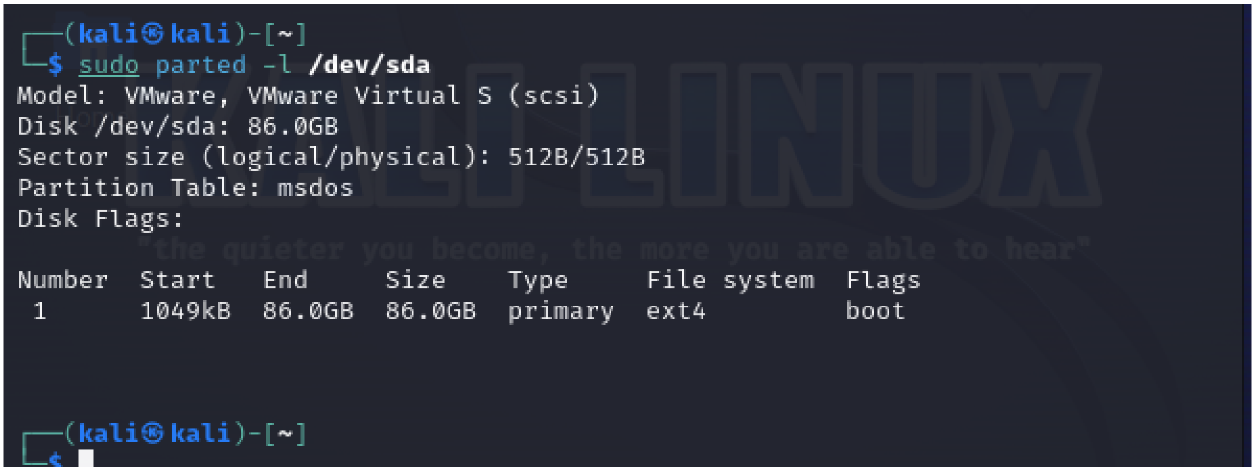
B



```
┌──(kali㉿kali)-[~]
└─$ sudo smartctl -i /dev/sda
smartctl 7.3 2022-02-28 r5338 [x86_64-linux-6.5.0-kali3-amd64] (local build)
Copyright (C) 2002-22, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===
Vendor:               VMware,
Product:              VMware Virtual S
Revision:             1.0
User Capacity:        86,000,000,000 bytes [86.0 GB]
Logical block size:   512 bytes
Rotation Rate:        Solid State Device
Device type:          disk
Local Time is:        Tue Jun  4 13:22:55 2024 EDT
SMART support is:     Unavailable - device lacks SMART capability.
```

Since S.M.A.R.T monitoring is not enabled on virtual hard drives, the “smartctl” command does not work on virtual machines.

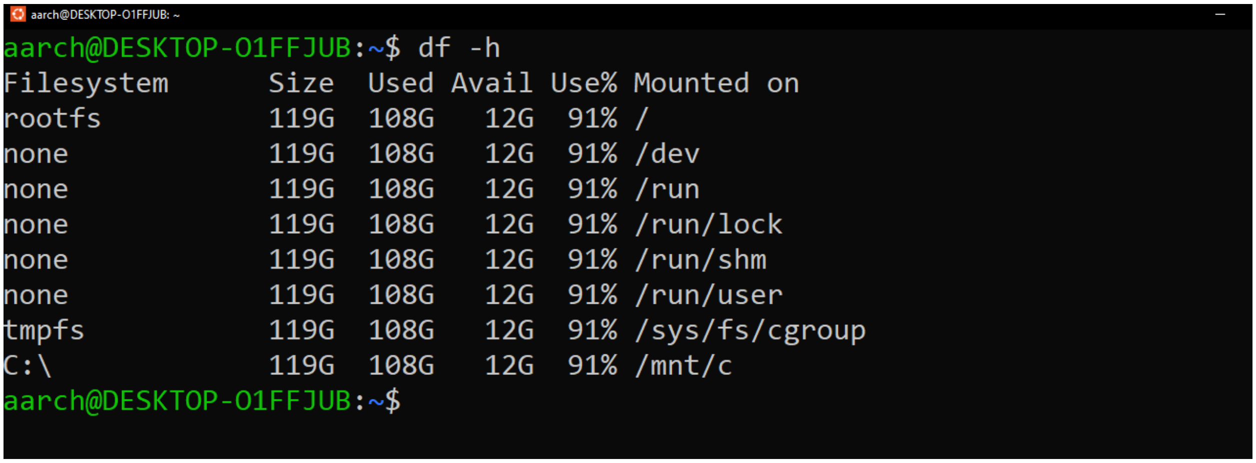
C



```
┌──(kali㉿kali)-[~]
└─$ sudo parted -l /dev/sda
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sda: 86.0GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
 1      1049kB  86.0GB  86.0GB  primary  ext4         boot


┌──(kali㉿kali)-[~]
└─$
```

“parted” command with the “-l” option lists all partitions on a disk.

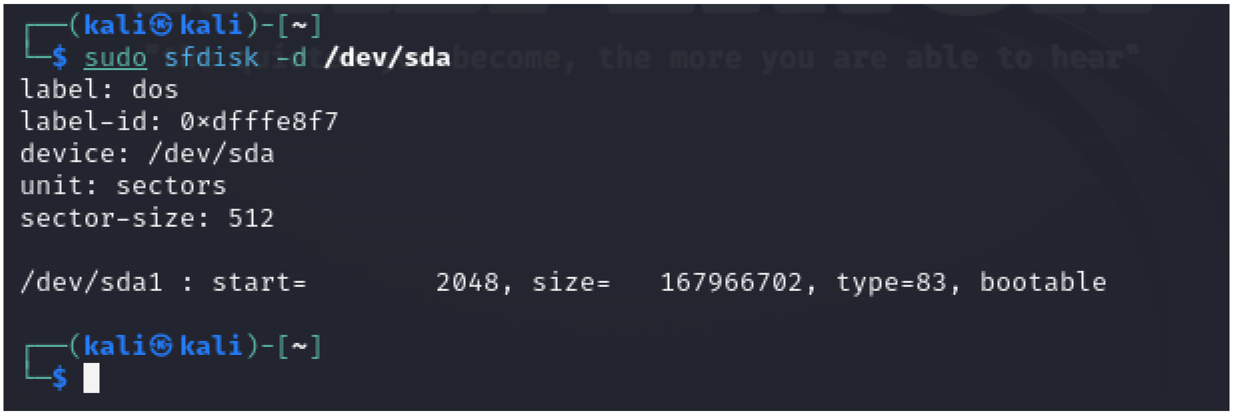
D



```
aarch@DESKTOP-O1FFJUB:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
rootfs          119G  108G   12G  91% /
none            119G  108G   12G  91% /dev
none            119G  108G   12G  91% /run
none            119G  108G   12G  91% /run/lock
none            119G  108G   12G  91% /run/shm
none            119G  108G   12G  91% /run/user
tmpfs           119G  108G   12G  91% /sys/fs/cgroup
C:\             119G  108G   12G  91% /mnt/c
aarch@DESKTOP-O1FFJUB:~$
```

Using “df -h” without providing any arguments prints the relevant details of all filesystems instead of a specific filesystem as seen in Question 3.e

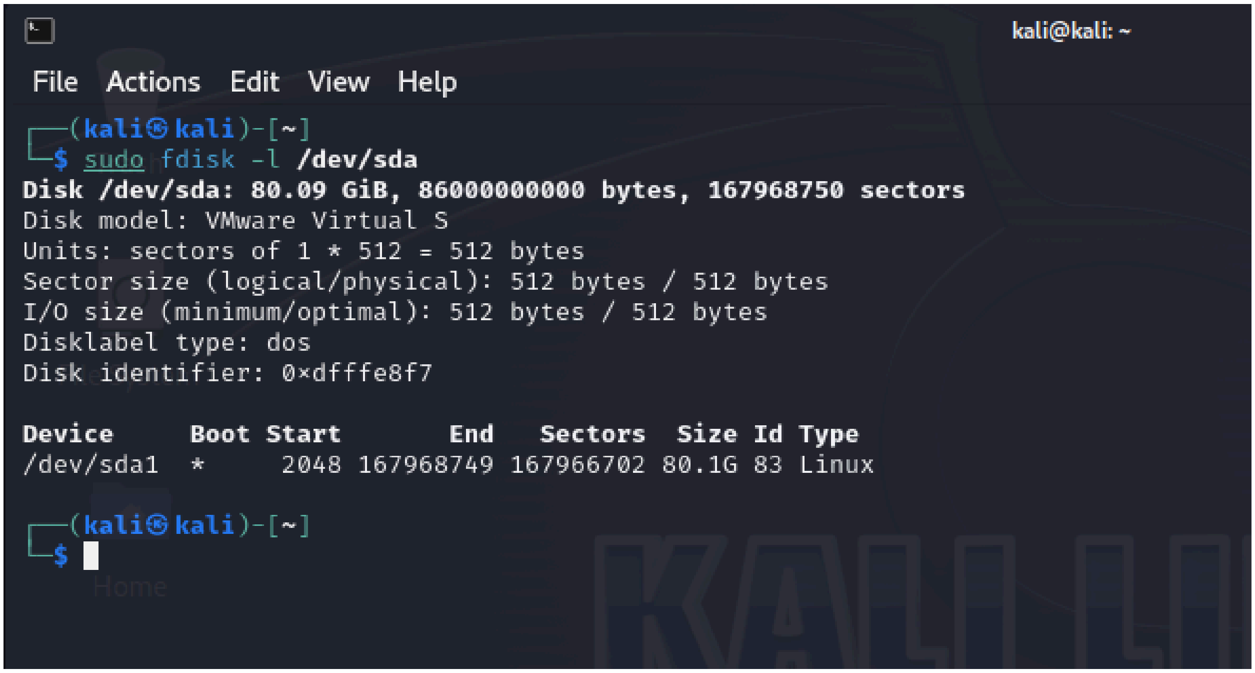
E



```
┌──(kali㉿kali)-[~]
└─$ sudo sfdisk -d /dev/sda
label: dos
label-id: 0×dfffe8f7
device: /dev/sda
unit: sectors
sector-size: 512

/dev/sda1 : start=        2048, size=   167966702, type=83, bootable

┌──(kali㉿kali)-[~]
└─$ 
```

“sfdisk -d” prints partition table of a disk

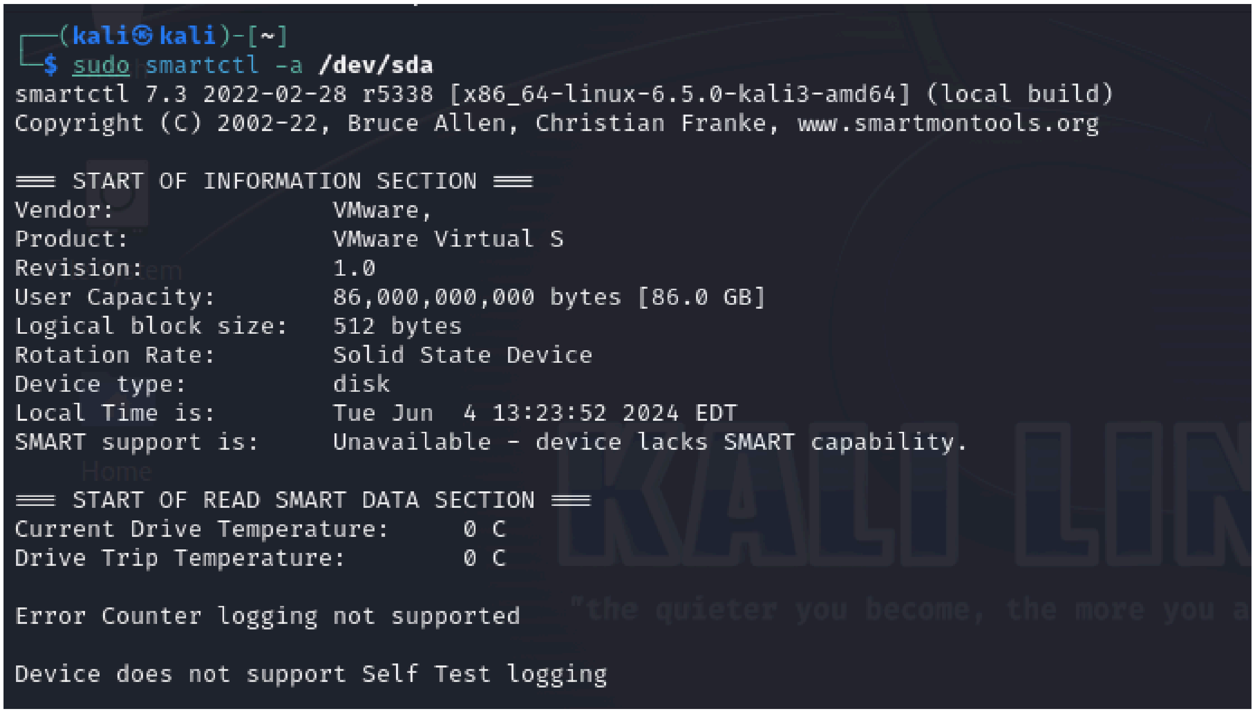
F



```
┌──(kali㉿kali)-[~]
└─$ sudo fdisk -l /dev/sda
Disk /dev/sda: 80.09 GiB, 86000000000 bytes, 167968750 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0×dfffe8f7

Device     Boot Start       End   Sectors  Size Id Type
/dev/sda1  *     2048 167968749 167966702 80.1G 83 Linux

┌──(kali㉿kali)-[~]
└─$ 
```

“fdisk -l” prints sector level information of a disk

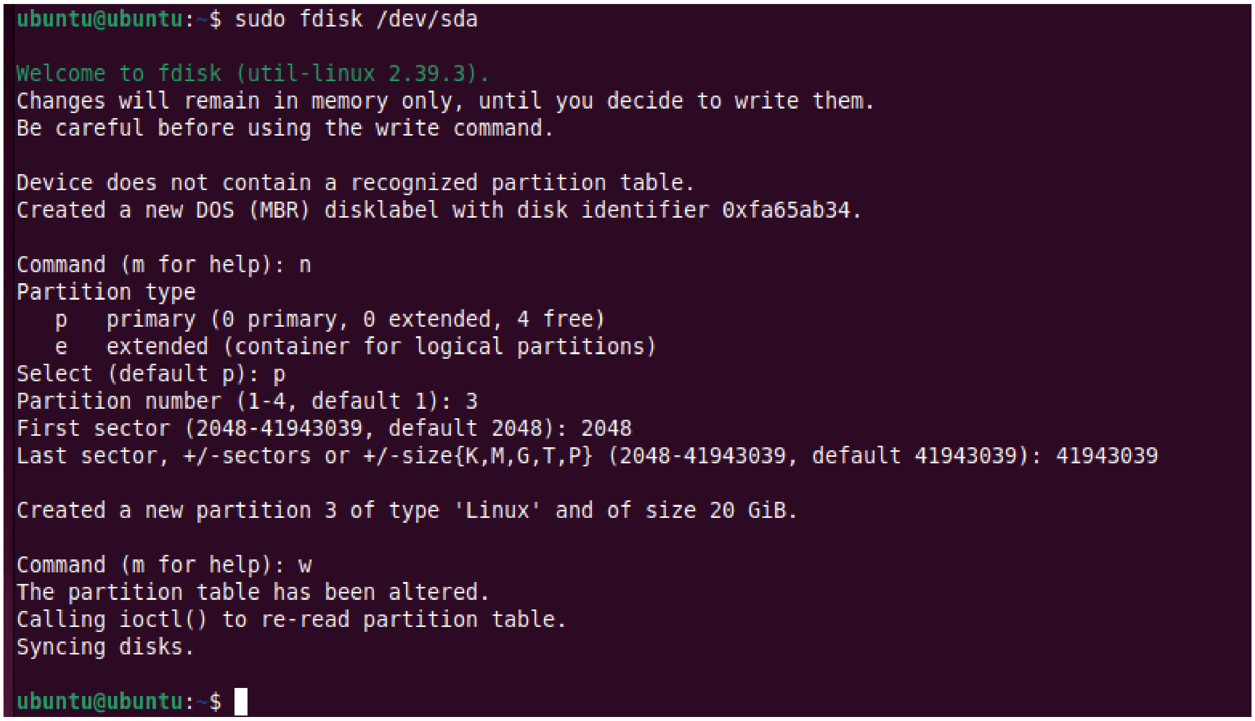
G



```
┌──(kali㉿kali)-[~]
└─$ sudo smartctl -a /dev/sda
smartctl 7.3 2022-02-28 r5338 [x86_64-linux-6.5.0-kali3-amd64] (local build)
Copyright (C) 2002-22, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===
Vendor:               VMware,
Product:              VMware Virtual S
Revision:             1.0
User Capacity:        86,000,000,000 bytes [86.0 GB]
Logical block size:   512 bytes
Rotation Rate:        Solid State Device
Device type:          disk
Local Time is:        Tue Jun  4 13:23:52 2024 EDT
SMART support is:     Unavailable - device lacks SMART capability.

=== START OF READ SMART DATA SECTION ===
Current Drive Temperature:     0 C
Drive Trip Temperature:        0 C

Error Counter logging not supported

Device does not support Self Test logging
```

Since S.M.A.R.T monitoring is not enabled on virtual hard drives, the “smartctl” command does not work on virtual machines.

H



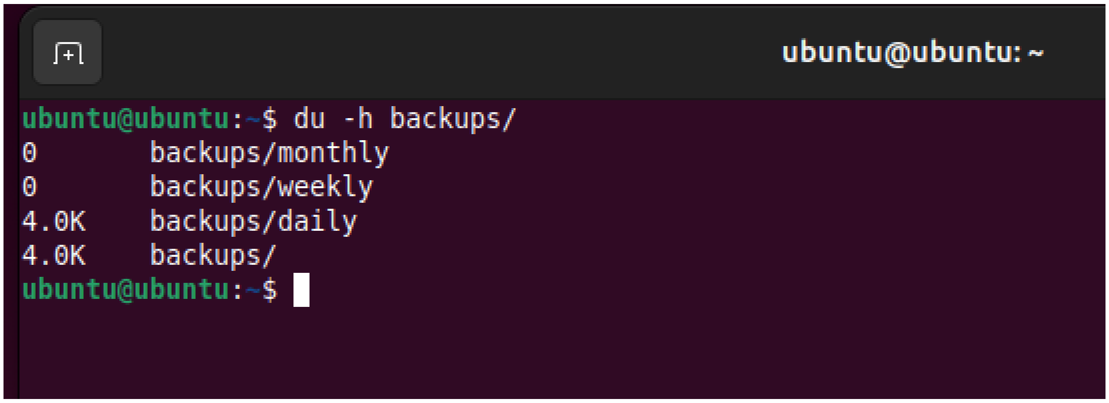
```
ubuntu@ubuntu:~$ sudo fdisk /dev/sda

Welcome to fdisk (util-linux 2.39.3).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS (MBR) disklabel with disk identifier 0xfa65ab34.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 3
First sector (2048-41943039, default 2048): 2048
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-41943039, default 41943039): 41943039

Created a new partition 3 of type 'Linux' and of size 20 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

ubuntu@ubuntu:~$ 
```

Using the “n” command in the interface provided by “fdisk”, a new partition can be made by entering the relevant details. “mkfs” can be used to make a filesystem on a partition.

I

Using “fsck -y” automatically finds and repairs damaged sectors in the target filesystem. This can’t be shown in a screenshot here as there are no bad blocks to fix.

J



```
ubuntu@ubuntu:~$ du -h backups/
0       backups/monthly
0       backups/weekly
4.0K    backups/daily
4.0K    backups/
ubuntu@ubuntu:~$ 
```

The “du -h” recursively prints the disk usage of the directory provided as an argument along with all the subdirectories