Red Hat
Enterprise
Linux 9

RHCSA-SA1 LAB-Book

Chapter 15 - Accessing Linux File Systems

Storage management concepts:

- ✓ A file system is an organized structure of data-holding files and directories residing on a storage device, such as a physical disk or partition.
- ✓ The file system hierarchy discussed earlier assembles all the file systems into one tree of directories with a single root, the “/” directory.

Mounting: The process of adding a new file system to the existing directory tree is called mounting.

Mount point:

- ✓ The directory where the new file system is mounted is referred to as a mount point.
- ✓ This is fundamentally different concept than that used on a M Microsoft Windows system, where a new file system is represented by a separate d rive letter.

Partitions:

- ✓ Hard disks and storage devices are normally divided up into smaller chunks called partitions.
- ✓ For example, one partition could contain user home directories while another could contain system data and logs.

Data Storage: Placing data in two separate file systems on two separate partitions helps in planning data storage.

Block Devices:

- ✓ Storage devices are represented by a special file type called block device.
- ✓ The block device is stored in the /dev directory.

Volume Group:

- ✓ With LVM, one or more block devices can be aggregated into a storage pool called a volume group.
- ✓ For the volume group, a directory with the same name as the volume group exists in the /dev directory.

Logical Volume:

- ✓ Disk space is made available with one or more logical volumes.
- ✓ A logical volume is the equivalent of a partition residing on a physical disk.

Note: Both the volume group and the logical volume have names assigned upon creation.

For example:

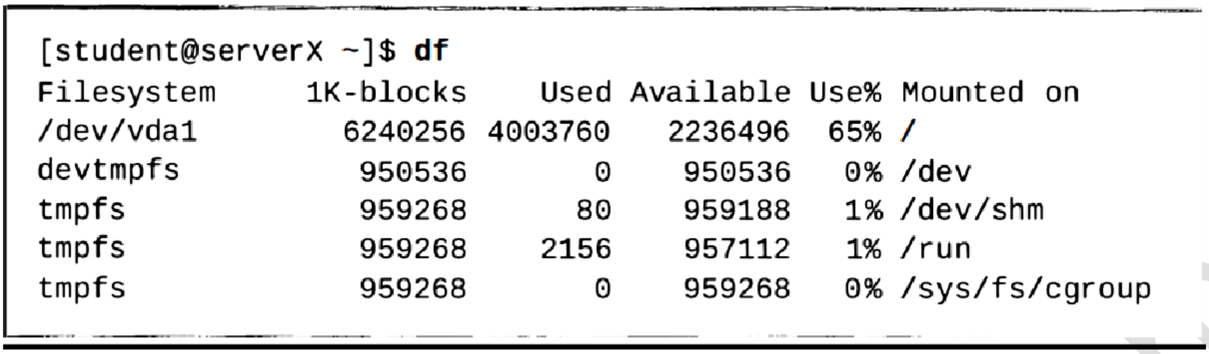
- ✓ The device file representing the mylv logical volume in the myvg volume group is /dev/myvg/mylv.

- ✓ It should be noted that LVM relies on the Device Mapper (DM) kernel driver.

- ✓ The assignment of the number is sequential beginning with zero (0).

- ✓ There is another symbolic link for every logical volume in the /dev/mapper directory with the name /dev/mapper/myvg -mylv.

- ✓ df command: df command is run without arguments, it will report total disk space, used disk space, and free disk space on all mounted regular file systems. It will report on both local and remote systems and the percentage of the total disk space that is being used.



```
[student@serverX ~]$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/vda1         6240256 4003760   2236496  65% /
devtmpfs           950536       0    950536   0% /dev
tmpfs              959268      80    959188   1% /dev/shm
tmpfs              959268    2156    957112   1% /run
tmpfs              959268       0    959268   0% /sys/fs/cgroup
```

Note:

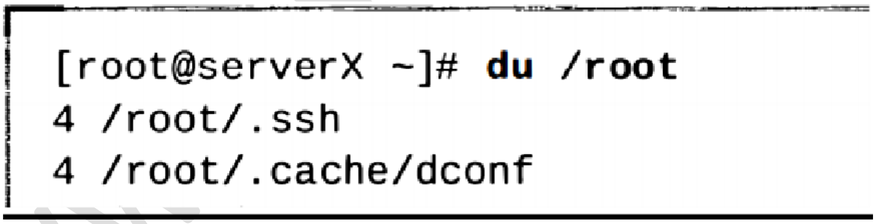
- The tmpfs and devtmpfs devices are file systems in system memory.

- All files written into tmpfs or devtmpfs disappear after system reboot.

- -h or -H command: It works like “df” command only but display the out in the human-readable format.

du command:

- Display detailed information about the space used by a certain directory tree.
- The du command shows the size of all files in the current directory tree recursively.
- For e.g



```
[root@serverX ~]# du /root
4 /root/.ssh
4 /root/.cache/dconf
```

blkid command:

The blkid command gives an overview of existing partitions with a file system on them and the UUID of the file system, as we similarly as the file system used to format the partition.

Practical on Identifying File Systems and Devices.

The device file of a SATA hard drive residing in /dev.

➔ /dev/sdc

The device file of the second partition on the first SATA hard drive in /dev.

➔/dev/sda2

The device file of a logical volume in /dev.

➔/dev/vg_install/lv_home

The device file of the second disk in a virtual machine in /dev.

➔/dev/vdb.

The device file of the third partition on the second SATA hard drive in /dev.

➔**/**dev/sdb3

The device file of the third partition on the second disk in a virtual machine in /dev.

➔/dev/vdb3

Mounting and Unmounting File Systems:

Mounting:

- ✓ The first argument of the mount command specifies the file system to mount.

- ✓ The second argument specifies the target directory where the file system is made available after mounting it.

- ✓ The target directory is referred to as a mount point.

- ✓ The mount command expects the fi le system argument in one of two different ways:
    - **The device file of the partition holding the file system, residing in /dev.**
    - **The UU/D, a universal unique identifier of the file system.**
- ✓ The /mnt directory exists by default and provides an entry point for mount points.
- ✓ It is used for manually mounting disk.
- ✓ Use the sub directory as a mount point.
- ✓ Else use main directory /mnt.

<u>Practical:</u>

Mount by device file of the partition that holds the fi le system.

➔mount /dev/vdb1 /mnt/mydata

Mount the fi le system by universal unique id, or the UUID, of the file system.

➔mount UUID="46f543fd - 7Bc9 - 4526 - a857 · 244811be2dBB" /mnt /mydata

<u>Unmounting file systems:</u>

- ✓ To unmount a file system, the un-mount command expects the mount point as an argument.

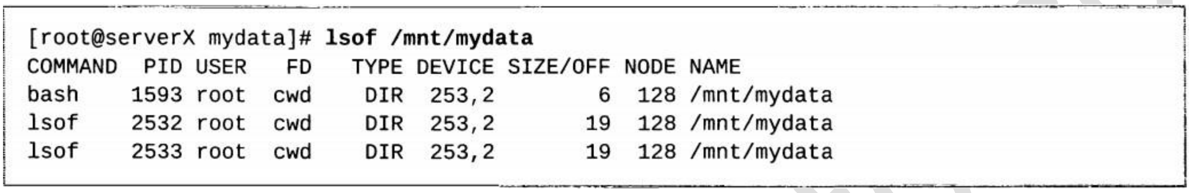
- ✓ Unmounting is not possible if the mount point is accessed by a process.
- ✓ For un-mount to be successful, the process needs to stop accessing the mount point.

- ✓ For e.g.,

lsof command:

- ✓ The lsof command lists all open files and the process accessing them in the provided directory.
- ✓ It is useful to identify which processes currently prevent the file system from successful un-mounting.

For e.g.,



```
[root@serverX mydata]# lsof /mnt/mydata
COMMAND  PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
bash    1593 root  cwd    DIR  253,2        6  128 /mnt/mydata
lsof    2532 root  cwd    DIR  253,2       19  128 /mnt/mydata
lsof    2533 root  cwd    DIR  253,2       19  128 /mnt/mydata
```

Unmounting Steps:

Step 1: umount /mnt/data

Step 2: lsblk

Accessing removable storage devices:

- ✓ Removable media, such as USB flash devices and d rives, get automatically mounted by the graphical desktop environment when plugged in.

- ✓ The amount points for the removable medium is /run/media/<user>/<label>.
- ✓ The <user> is the user logged into the graphical environment.
- ✓ The <label> is the name given to the file system when it was created.
- ✓

Practical:

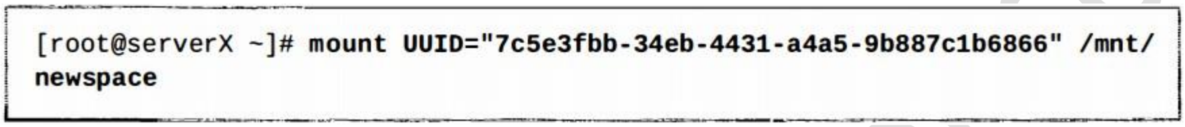
1. A new partition with a file system has been added to the second disk (vdb) on your serverX machi ne. Mount the newly available partition by UUID at the newly created mount point /mnt/newspace.

Ans:

Step:1: Use the blkid to discover the UUID of the newly added partition, vdb1, on serverX

Step:2: Create the mount point /mnt/newspace on server

Step3: Mount the file system by UUID on the /mnt/newspace directory of the serverX machi ne.

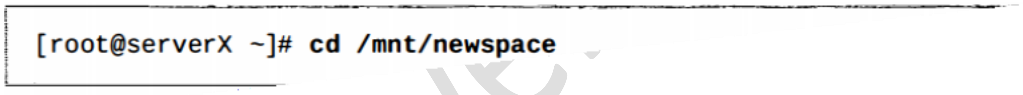


```
[root@serverX ~]# mount UUID="7c5e3fbb-34eb-4431-a4a5-9b887c1b6866" /mnt/
newspace
```

2.Change to the /mnt /newspace directory and create a new directory, /mnt/newspace/newdir, with an e mpty file, /mnt/newspace/newdir /newfile, on serverX.

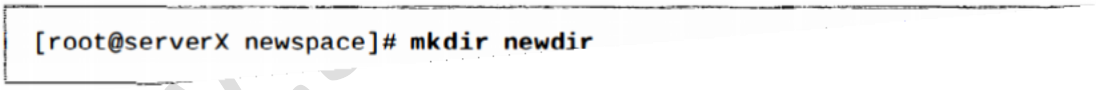
Ans:

Step:1: Change to the /mnt/newspace directory on serverX.



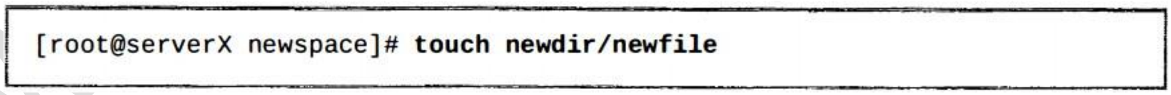
```
[root@serverX ~]# cd /mnt/newspace
```

Step:2: Create a new directory, /mnt/newspace/newdir, on serverX.



```
[root@serverX newspace]# mkdir newdir
```

Step 3: Create a new empty file, /mnt/newspace/newdir /newfile, on serverX.



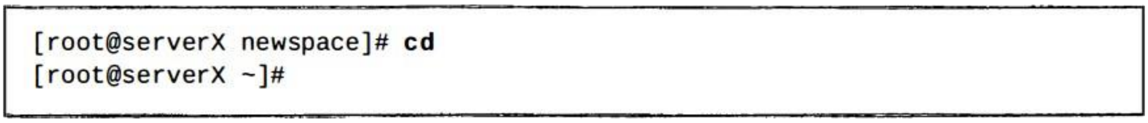
```
[root@serverX newspace]# touch newdir/newfile
```

3. Unmount the file system mou nted on the /mnt/newspace directory on serverX.

Ans:

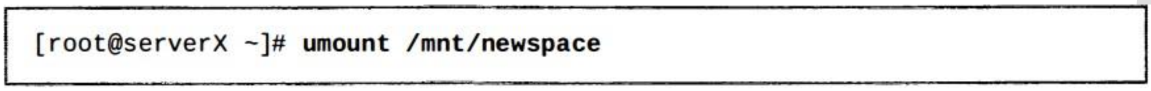
Step:1: Try to un-mount /mnt/newspace while the current directory on the shell is still /mnt/newspace on serverX.

Step:2: Change the current directory on the shel l to / root.



```
[root@serverX newspace]# cd
[root@serverX ~]#
```

Step 3: Successfully un-mount /mnt/newspace on serverX.



```
[root@serverX ~]# umount /mnt/newspace
```

Tools for finding files:

A system administrator needs tools for searching files matching certain on the file system.

This section discusses two commands that ca n search fi les in the file system.

a. **Locate:** The locate command searches a pre-generated database for file names or file paths and returns the results instantly.
b. **Find:** The find command searches the file system in rea l time by crawling through the file system.

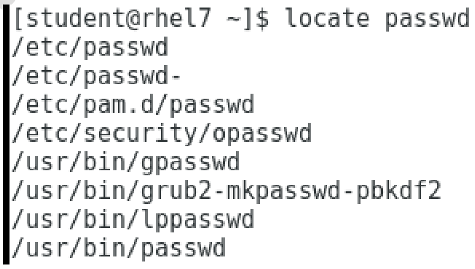
Locating Files on the System (Find command):

Practical:

1. Search for files with “passwd “in the name or path in directory trees readable by user student on serverX.

➔locate passwd

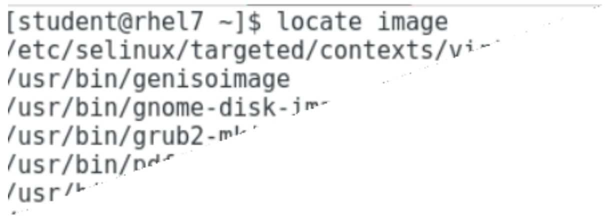
➔Output:



```
[student@rhel7 ~]$ locate passwd
/etc/passwd
/etc/passwd-
/etc/pam.d/passwd
/etc/security/opasswd
/usr/bin/gpasswd
/usr/bin/grub2-mkpasswd-pbkdf2
/usr/bin/lppasswd
/usr/bin/passwd
```

➔

2. Results are returned even when the file name or path is only a partial match to the search query.

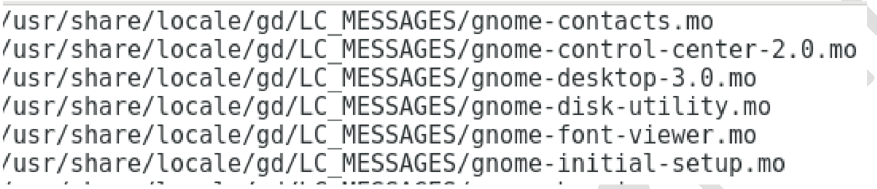


```
[student@rhel7 ~]$ locate image
/etc/selinux/targeted/contexts/vi
/usr/bin/genisoimage
/usr/bin/gnome-disk-im
/usr/bin/grub2-mk
/usr/bin/pd
/usr/
```

➔

3.-i option: performs a case-insensitive search. With this option, all possible combinations of upper - and lowercase letters match the search.

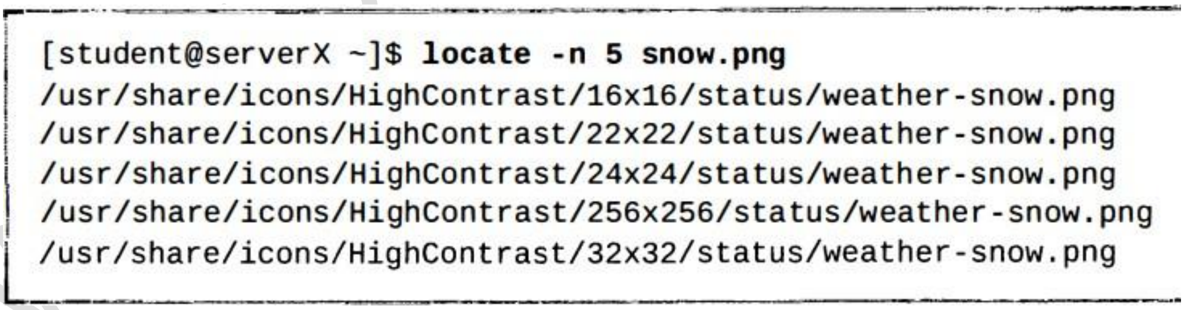
➔Output:



```
/usr/share/locale/gd/LC_MESSAGES/gnome-contacts.mo
/usr/share/locale/gd/LC_MESSAGES/gnome-control-center-2.0.mo
/usr/share/locale/gd/LC_MESSAGES/gnome-desktop-3.0.mo
/usr/share/locale/gd/LC_MESSAGES/gnome-disk-utility.mo
/usr/share/locale/gd/LC_MESSAGES/gnome-font-viewer.mo
/usr/share/locale/gd/LC_MESSAGES/gnome-initial-setup.mo
```

4.-n option: It limits the number of returned search results by locate. The following limits the search results returned by locate to the first five matches

➔locate -n 5 snow.png



```
[student@serverX ~]$ locate -n 5 snow.png
/usr/share/icons/HighContrast/16x16/status/weather-snow.png
/usr/share/icons/HighContrast/22x22/status/weather-snow.png
/usr/share/icons/HighContrast/24x24/status/weather-snow.png
/usr/share/icons/HighContrast/256x256/status/weather-snow.png
/usr/share/icons/HighContrast/32x32/status/weather-snow.png
```

Searching for files with find:

- ✓ The find command performs a real-time search in the local file systems to find files match the criteria of the command -line arguments.

- ✓ The first argument to the find command is the directory to search. If the directory argument is omitted, find will start the search in the current directory and look for matches in any of the subdirectories.
- ✓ To search the home directory of user student, give find a starting directory of /home/student and
- ✓ To search the entire system, provide a starting directory "/".

Note: find has a huge number of options that can be provided to describe exactly what kind of file should be found.

Searches ca n be based on fi le name, file size, last modified time stamp, and other file characteristics in any combi nation.

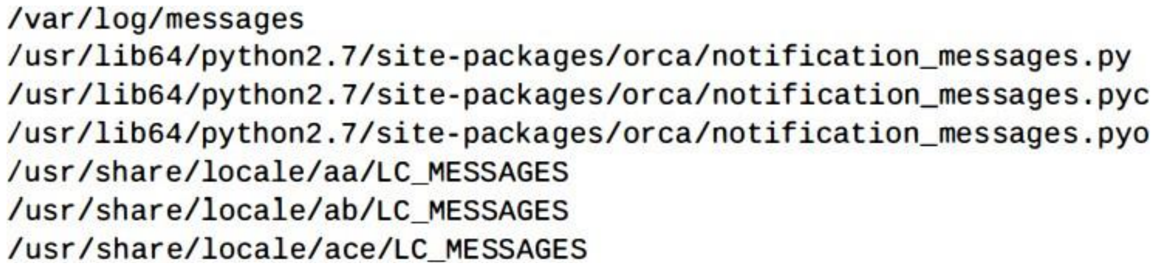
1. -name option: Followed by a file name looks up fi l es matching the given file name and returns all exact matches.
To search for files named sshd_confg in the / directory and all subdirectories on server, run:
➔ Command: find / -name sshd_config

2. Searches for files in the / directory on serverX that end in .txt:
➔Command: find / -name '*.txt'

3. To search for files in /etc/ that contain pass anywhere in their name on serverX, run
➔ Command: find /etc -name '*pass*'

<u>Queries based on ownership and permissions:</u>
1. -iname option: To perform a case -insensitive search for a given file name followed by the given name to search.
To search case – insensitive for files that have messages in their names in the / directory on serverX, run

➔find / -iname '*messages*'
➔Output:



```
/var/log/messages
/usr/lib64/python2.7/site-packages/orca/notification_messages.py
/usr/lib64/python2.7/site-packages/orca/notification_messages.pyc
/usr/lib64/python2.7/site-packages/orca/notification_messages.pyo
/usr/share/locale/aa/LC_MESSAGES
/usr/share/locale/ab/LC_MESSAGES
/usr/share/locale/ace/LC_MESSAGES
```

2. find: Can search for files based on their ownership or permissions. Useful options when searching by owner are - user and - group, which search by name, and -uid and - gid, which search by ID.

3. Search for files owned by the user student in the /home/student directory on serverX.
➔find -user student

4. Search for files owned by the group student in /home/ in the /home/student directory on serverX.
➔find -group student

5. Search for files owned by user ID 1 000 in the /home/student directory on serverX.
➔find -uid 1000

6. Search for files owned by group ID 1000 in the /home/student directory on serverX.
➔Find -gid 1000

7. Search for files owned by user root and group mail on the serverX machine.
➔find / -user root -group mail.

<u>**Queries based on particular set of permissions:**</u>

1. To use a more complex example, the following command would match any file for which the user has read, write, and execute

permissions, members of the group have read and write permissions, and others have read-only access:

```
➔find /home -prem 764
```

2. To match files for which the user has at least write and execute permissions, and group has at least written permissions, others have at least read access:

```
➔find /home -perm -324
```

3. To match files for which the suer has read permission, or the group has at least read permissions, or others have at least written access:

```
➔find /home -perm /442
```

4. To match any file in the /home/student directory for which others have at least read access on serverX run:

```
➔find -perm -004
```

5. Find all files in the /home/student directory where other has written permissions on serverX.

```
➔find -perm -002
```

Queries based on Size:

1. Search for files with a size of exactly 10 megabytes.

```
➔find -size 10M.
```

2. Find files with a size more than 10 gigabytes.

```
➔find -size +10G
```

3. List all files with a size less than 10 kilobytes.

```
➔find -size -10k
```

Queries based on Minutes:

1. To find all files that had their file content changed exactly 120 minutes ago on serverX, run:

➔find / -min 120

2. +modifier: The + modifier in front of the number of minutes looks for all files in the / that have been modified more than 200 minutes ago.

➔find / -min +200

3. -modifier: The – modifier changes the search to look for all files in the / directory which have been changed less than 150 minutes ago.

➔find / -min -150

Queries based on type options:

1. Search for all directories in the /etc folder on serverX.

➔find /etc -type d

2. Search for all soft links on the serverX system.

➔find / -type l

3. Generate a list of all block devices in the /dev directory on serverX:

➔find /dev -type b

Queries based on Links:

1. Search for all regular files with more than one hard link on the serverX machine.

➔find / -type f -links +1

Practical 1 based on locating files on the system:

Use the locate command to find various different files on the serverX machine.

1.Locate the config u ration file logrotate.cont on serverX.

➔locate logrotate.conf

2.Locate the configuration fi le networkmanager. conf, ignoring case, on serverX.

➔locate -i networkmanager.conf

Use the find command to perform rea l-time searches on the serverX machine according to the following requirements.

1. Find all files in the /var /lib directory owned by user chrony on serverX.

➔find /var/lib -user chrony

2. List all files in the /var directory owned by user root and group mail.

➔find /var/ -user root -group mail.

3. List all files in the /usr /bin directory with a file size greater than 50 kilobytes.

➔find /usr/bin -size +50k

4. Find all files in the /home/student directory that have not been changed in the last 120 minutes on serverX.

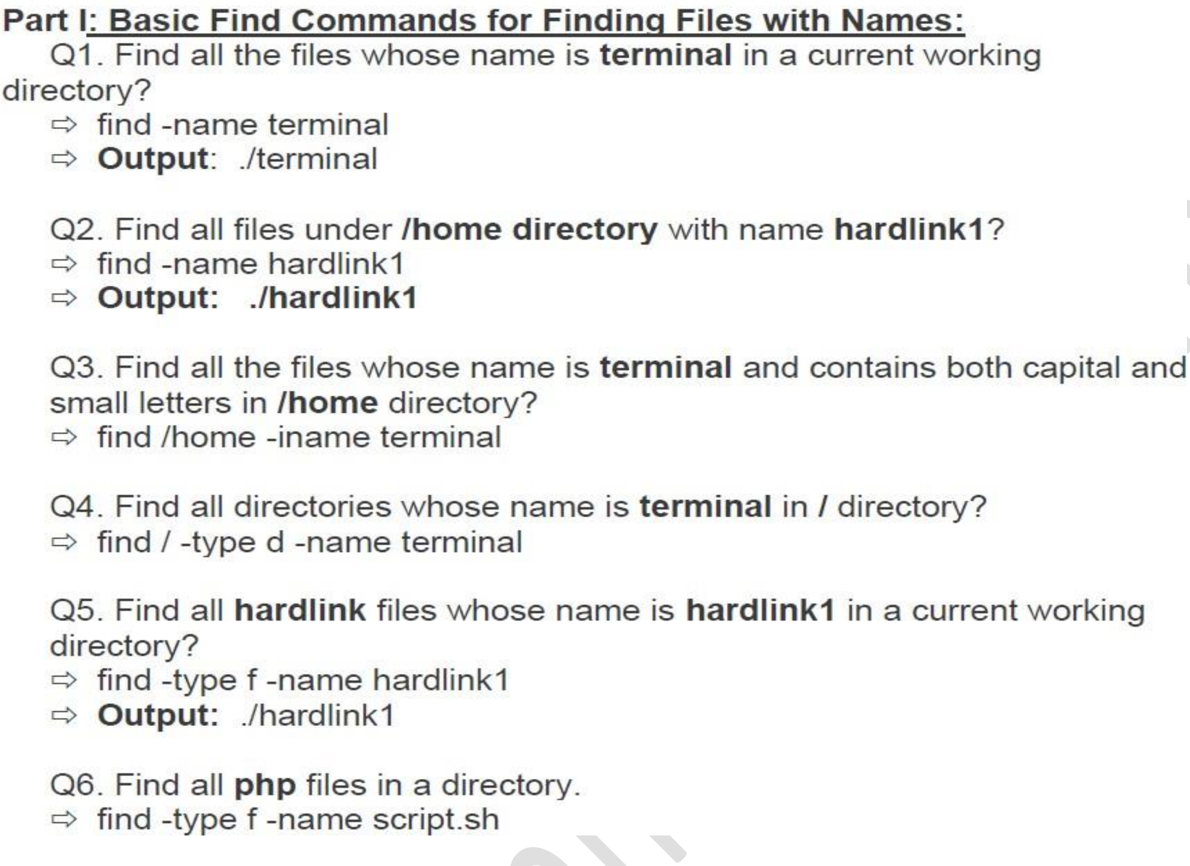
➔find /home/student -min +120

5. Find all files in the /tmp directory that have been changed in the last 240 minutes on serverX

➔find /tmp -min -240

Interview Questions

A] Based on find commands:



Part I: Basic Find Commands for Finding Files with Names:

Q1. Find all the files whose name is **terminal** in a current working directory?

⇨ find -name terminal

⇨ **Output**: ./terminal

Q2. Find all files under **/home directory** with name **hardlink1**?

⇨ find -name hardlink1

⇨ **Output: ./hardlink1**

Q3. Find all the files whose name is **terminal** and contains both capital and small letters in **/home** directory?

⇨ find /home -iname terminal

Q4. Find all directories whose name is **terminal** in **/** directory?

⇨ find / -type d -name terminal

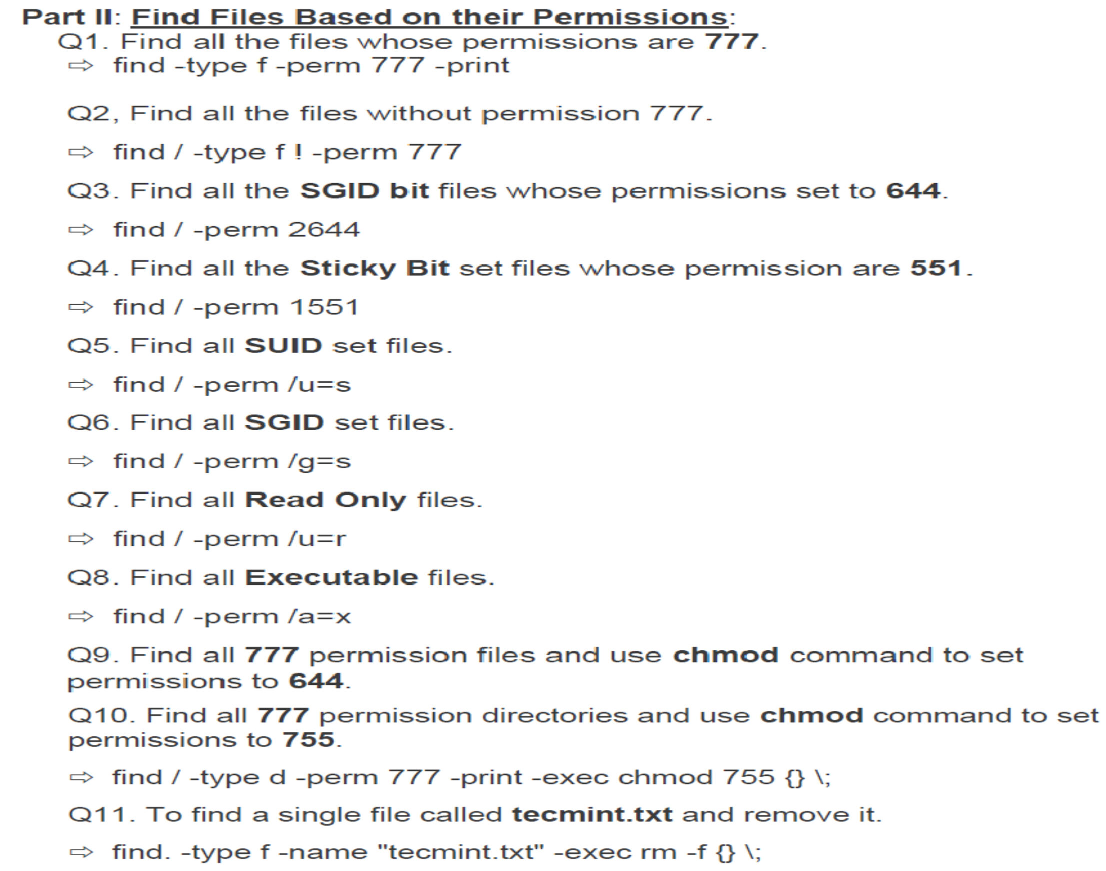
Q5. Find all **hardlink** files whose name is **hardlink1** in a current working directory?

⇨ find -type f -name hardlink1

⇨ **Output:** ./hardlink1

Q6. Find all **php** files in a directory.

⇨ find -type f -name script.sh



Part II: Find Files Based on their Permissions:

Q1. Find all the files whose permissions are **777**.

⇨ find -type f -perm 777 -print

Q2, Find all the files without permission 777.

⇨ find / -type f **!** -perm 777

Q3. Find all the **SGID bit** files whose permissions set to **644**.

⇨ find / -perm 2644

Q4. Find all the **Sticky Bit** set files whose permission are **551**.

⇨ find / -perm 1551

Q5. Find all **SUID** set files.

⇨ find / -perm /u=s

Q6. Find all **SGID** set files.

⇨ find / -perm /g=s

Q7. Find all **Read Only** files.

⇨ find / -perm /u=r

Q8. Find all **Executable** files.

⇨ find / -perm /a=x

Q9. Find all **777** permission files and use **chmod** command to set permissions to **644**.

Q10. Find all **777** permission directories and use **chmod** command to set permissions to **755**.

⇨ find / -type d -perm 777 -print -exec chmod 755 {} \;

Q11. To find a single file called **tecmint.txt** and remove it.

⇨ find. -type f -name "tecmint.txt" -exec rm -f {} \;

Q12. To find and remove multiple files such as **.mp3** or **.txt**, then use.

⇨ find . -type f -name "*.txt" -exec rm -f {} \;
⇨ OR
⇨ find . -type f -name "*.mp3" -exec rm -f {} \;

Q13. To find all empty files under a certain path.

⇨ find /tmp -type f -empty

Q14. To file all empty directories under a certain path.

⇨ find /tmp -type d -empty

Q15. To find all hidden files, use the below command.

⇨ find /tmp -type f -name “. *"

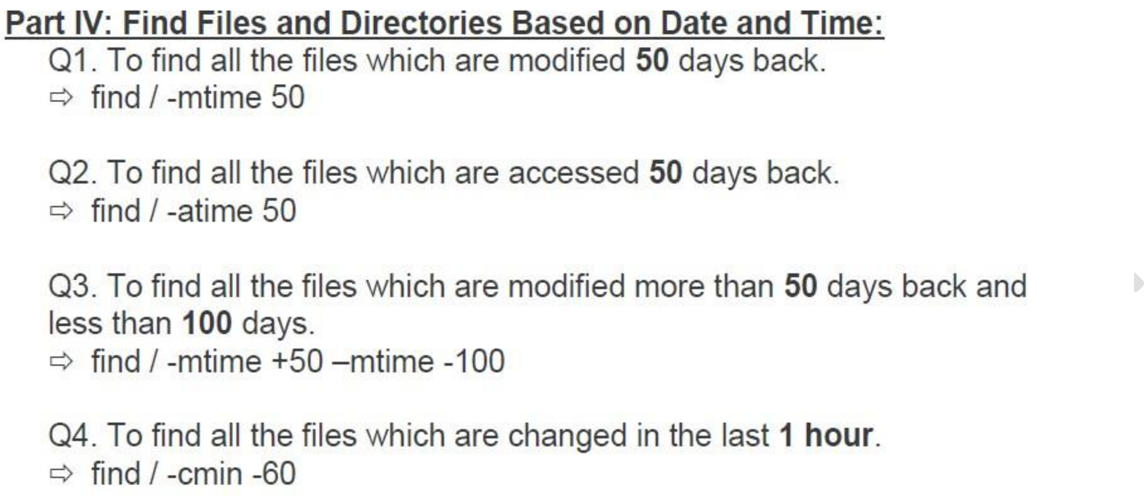
Part III: Search Files Based Owners and Groups on:

Q1. To find all or single file called **tecmint.txt** under **/** root directory of owner root.
⇨ find / -user root -name tecmint.txt

Q2. To find all files that belong to user **Tecmint** under **/home** directory.
⇨ find /home -user tecmint

Q3. To find all files that belong to the
group **Developer** under **/home** directory.
⇨ find /home -group developer

Q4. To find all **.txt** files of user **Tecmint** under **/home** directory.
⇨ find /home -user tecmint -iname "*.txt"



Part IV: Find Files and Directories Based on Date and Time:

Q1. To find all the files which are modified **50** days back.

```
⇨ find / -mtime 50
```

Q2. To find all the files which are accessed **50** days back.

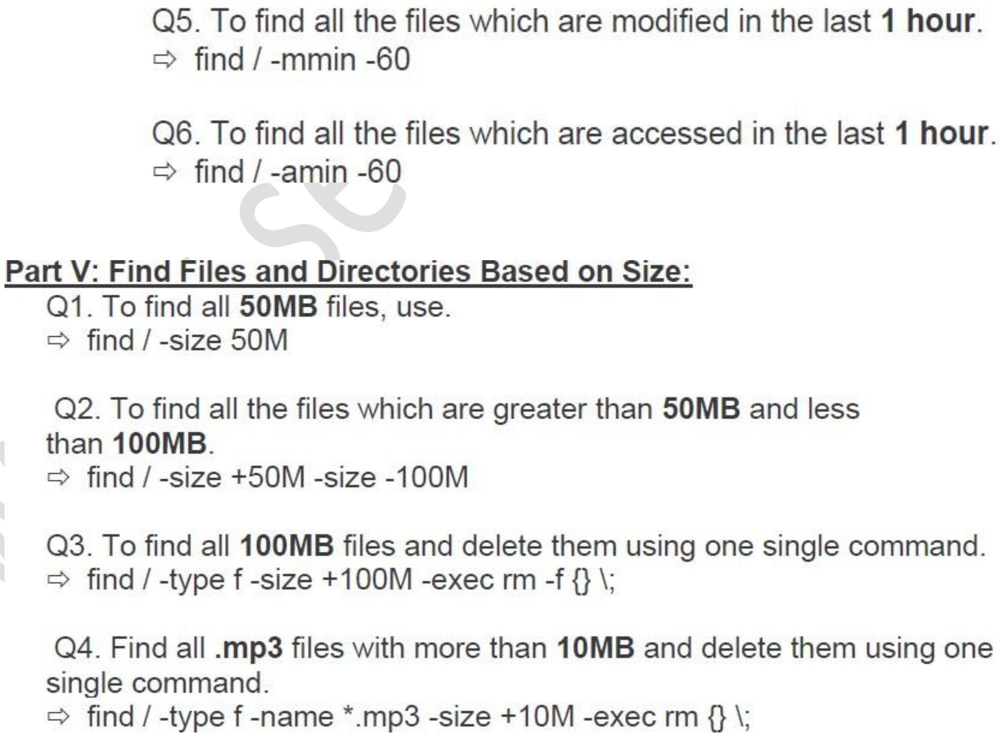
```
⇨ find / -atime 50
```

Q3. To find all the files which are modified more than **50** days back and less than **100** days.

```
⇨ find / -mtime +50 –mtime -100
```

Q4. To find all the files which are changed in the last **1 hour**.

```
⇨ find / -cmin -60
```



Q5. To find all the files which are modified in the last **1 hour**.

```
⇨ find / -mmin -60
```

Q6. To find all the files which are accessed in the last **1 hour**.

```
⇨ find / -amin -60
```

Part V: Find Files and Directories Based on Size:

Q1. To find all **50MB** files, use.

```
⇨ find / -size 50M
```

Q2. To find all the files which are greater than **50MB** and less than **100MB**.

```
⇨ find / -size +50M -size -100M
```

Q3. To find all **100MB** files and delete them using one single command.

```
⇨ find / -type f -size +100M -exec rm -f {} \;
```

Q4. Find all **.mp3** files with more than **10MB** and delete them using one single command.

```
⇨ find / -type f -name *.mp3 -size +10M -exec rm {} \;
```

END