**Q1: How do you find all `.log` files in the `/var/log` directory that were modified more than 15 days ago, but exclude those files that are larger than 500MB?**

* find /var/log -name “\*.log” -mtime +15 -size -500M

OR

Find /var/log -type f -name “\*.log” -mtime +15 -size -500M

/var/log : path

-type f: finds in files( not directories)

-name “\*.log”: search for files having extension “.log”

-mtime: files modified more than 15 days ago

-size -500M : excludes files larger than 500MB ( i.e include less than 500MB)

**Q2: How can you search for all symbolic links in a directory and delete them using a single `find` command?**

* find /path/to/directory -type l -exec rm {} +
* find: The command used to search for files and directories.
* /path/to/directory: Replace this with the path to the directory you want to search in.
* -type l: This option specifies that you want to find symbolic links.
* -exec rm {} +: This part executes the rm command on the found symbolic links. The {} is a placeholder for the found items, and + allows find to pass multiple arguments to rm at once for efficiency.Bottom of Form

**Q3: Using `find`, how do you locate files in the `/home/user` directory that are owned by a user other than the current owner?**

* find /home/user ! -user $(whoami)
* find: The command used to search for files.
* /home/user: The directory where you want to search for files.
* ! -user $(whoami): This part specifies that you want to find files that are **not** owned by the current user. $(whoami) retrieves the current user's username.

**Q4: How can you find all files in the `/etc` directory that contain the word "config" in their name and have write permission for others?**

* find /etc -type f -name '\*config\*' -perm -002
* find: The command used to search for files.
* /etc: The directory where you want to search.
* -type f: This option specifies that you want to find files (not directories).
* -name '\*config\*': This matches files that contain "config" in their name. The asterisks (\*) are wildcards that match any characters before or after "config."
* -perm -002: This checks for write permission for others (the last digit, 2, represents write permissions for others).

**Q5: How can you search for all lines in a file `/etc/passwd` that do not contain the word "bash"?**

* grep -v 'bash' /etc/passwd
* grep: The command used for searching text.
* -v: This option tells grep to invert the match, meaning it will show lines that do **not** match the specified pattern.
* 'bash': The pattern you are searching for (in this case, lines that do not contain the word "bash").
* /etc/passwd: The file you are searching through.

**Q6: How do you search recursively through all files in a directory for a specific string, showing only the matching filenames and excluding binary files?**

* grep -rl --exclude-binary '\*' 'specific\_string' /path/to/directory
* grep: The command used for searching text.
* -r: This option tells grep to search recursively through the directory.
* -l: This option makes grep output only the names of files with matching lines, rather than the lines themselves.
* --exclude-binary '\*': This option excludes binary files from the search.
* 'specific\_string': Replace this with the actual string you want to search for.
* /path/to/directory: Replace this with the path to the directory you want to search in.

**Q7: How can you search for the word "error" in a file, displaying the surrounding 3 lines before and after each match?**

* rep -C 3 'error' filename
* grep: The command used for searching text.
* -C 3: This option tells grep to show 3 lines of context before and after each matching line.
* 'error': The word you are searching for.
* filename: Replace this with the name of the file you want to search in.

**Q8: Using `grep`, how would you find all occurrences of the word "root" in `/etc/passwd` that start at the beginning of a line?**

* grep '^root' /etc/passwd
* grep: The command used for searching text.
* '^root': The caret (^) signifies the start of a line, so this pattern matches lines that begin with the word "root."
* /etc/passwd: The file you are searching through.

**Q9: How can you create a compressed archive of the `/home/user` directory, excluding all `.mp4` files?**

* tar --exclude='\*.mp4' -czf user\_archive.tar.gz -C /home user
* tar: The command used to create and manipulate archive files.
* --exclude='\*.mp4': This option specifies that all .mp4 files should be excluded from the archive.
* -czf: This option tells tar to create (c), compress with gzip (z), and specify the filename (f) for the archive.
* user\_archive.tar.gz: The name of the output archive file.
* -C /home: This changes to the /home directory before starting the archive process.
* user .: This specifies that the user directory (and its contents) should be included in the archive.

**Q10: How would you extract only a specific file `file.txt` from a `.tar.gz` archive without extracting the entire archive?**

* tar -xzf archive.tar.gz file.txt
* tar: The command used for creating and extracting archive files.
* -xzf: This option tells tar to extract (x), use gzip compression (z), and specify the filename of the archive (f).
* archive.tar.gz: Replace this with the name of your .tar.gz archive.
* file.txt: The specific file you want to extract from the archive.

**Q11: How can you append a file called `newfile.txt` to an existing `archive.tar.gz` without decompressing the archive?**

* gunzip -c archive.tar.gz > archive.tar
* tar -rf archive.tar newfile.txt
* gzip archive.tar

**Q12: How would you list the contents of a `.tar.gz` archive without extracting it?**

* tar -tzf archive.tar.gz
* tar: The command used for creating and manipulating archive files.
* -t: This option tells tar to list the contents of the archive.
* -z: This option specifies that the archive is compressed with gzip.
* -f: This option specifies the filename of the archive.
* archive.tar.gz: Replace this with the name of your archive file.

**Q13: How can you list all currently running processes and then filter for processes that include the word "java", showing only the process ID and name?**

* ps aux | grep '[j]ava' | awk '{print $2, $11}'
* ps aux: This command lists all currently running processes with detailed information.
* grep '[j]ava': This filters the output to show only lines containing "java". Using brackets around j prevents the grep process itself from appearing in the results.
* awk '{print $2, $11}': This extracts and prints the second column (Process ID) and the eleventh column (the command name) from the filtered results.

**Q14: How can you display a list of all files in `/var/log`, then filter for files containing the word "error" using both `ls` and `grep` in a pipeline?**

* ls /var/log | grep 'error'

**Q15: How do you count the number of lines in a file that contain the word "failed" using `grep` and `wc`?**

* grep 'failed' filename | wc -l

**Q16: How can you display the last 20 lines of a log file and filter the lines that contain the word "critical" using a combination of `tail` and `grep`?**

* tail -n 20 logfile | grep 'critical'

**Q17: How do you display the disk usage of all files and directories in `/home/user` in humanreadable format, sorted by size in descending order?**

* du -h /home/user | sort -hr
* du -h /home/user: This command calculates the disk usage of files and directories in /home/user, with the -h option making the output human-readable (e.g., displaying sizes in KB, MB, etc.).
* |: The pipe sends the output of the du command to the next command.
* sort -hr: This sorts the output in human-readable format (-h) and in reverse order (-r), so the largest sizes are displayed first.

**Q18: How can you display the total disk usage of the `/var` directory but exclude the subdirectory `/var/log`?**

* du -sh --exclude=/var/log /var/\*
* du: The command used to estimate file space usage.
* -s: This option summarizes the total size for each argument.
* -h: This option makes the output human-readable (e.g., KB, MB).
* --exclude=/var/log: This option tells du to exclude the specified directory from the calculations.
* /var/\*: This includes all items in the /var directory.

**Q19: How can you use `du` to find the disk usage of files larger than 100MB in a directory?**

* find /path/to/directory -type f -size +100M -exec du -h {} +
* find /path/to/directory: This command starts searching in the specified directory. Replace /path/to/directory with the actual path.
* -type f: This option specifies that you want to find files (not directories).
* -size +100M: This option finds files larger than 100 megabytes. The + indicates "greater than."
* -exec du -h {} +: This executes the du command on each file found. The {} is a placeholder for the filenames found by find, and + allows passing multiple filenames at once for efficiency. The -h option makes the output human-readable.

**Q20: How do you display the disk usage of the 10 largest directories in the `/usr` directory using `du`?**

* du -h --max-depth=1 /usr | sort -hr | head -n 10
* du -h --max-depth=1 /usr: This command calculates the disk usage of all directories in /usr, showing the sizes in a human-readable format (-h) and limiting the depth to 1 to only show top-level directories.
* |: The pipe sends the output of the du command to the next command.
* sort -hr: This sorts the output in human-readable format (-h) and in reverse order (-r), so the largest sizes appear first.
* head -n 10: This takes the first 10 lines of the sorted output, which correspond to the 10 largest directories.