

TABLE OF CONTENTS:

TABLE OF CONTENTS:

- [1. Difference between the grep and sed commands?](#)
- [2. cd tilde \(~\), where does it go?](#)
- [3. Remove the execution for the user for a particular file?](#)
- [4. Why do we use passwd?](#)
- [5. How to print all the empty lines if it is present?](#)
- [6. How to delete the last line of a file?](#)
- [7. How to remove the group permission of the group?](#)
- [8. Which user logs in to the server?](#)
- [9. Copy a directory from one location to another location recursively?](#)
- [10. How to check the size of the file?](#)
- [11. Definition of shell in Linux?](#)
- [12. How to check resource utilization in Linux?](#)
- [13. How to manage disk partitions and file management in Linux?](#)
- [14. How to set a Linux server against unauthorized service?](#)
- [15. Difference between forcefully and gracefully killing in Linux?](#)
- [16. What is the lifecycle of Linux and Ubuntu?](#)
- [17. List out files by sorting them largest file first?](#)
- [18. What is the dead phase in Linux?](#)
- [19. How to check who the root user is?](#)
- [20. How to copy permissions of one file to another?](#)
- [21. How to delete the first two characters in every line using a Linux command?](#)
- [22. How to delete the lines where the pattern is not present?](#)
- [23. How to delete the unique lines from the file in Linux?](#)
- [24. How to find the total size of a directory in Linux?](#)
- [25. How to give execute permission to .ssh files in Linux?](#)
- [26. How to find .log files that are greater than 500 bytes?](#)
- [27. How to check which user had sudo access on a Linux system?](#)
- [28. How to change the owner to admin in Linux?](#)
- [29. How to display the top 5 memory-consuming processes?](#)
- [30. How to find and delete the files that are older than 1 day in the current directory?](#)
- [31. How to sort the processes based on port number?](#)
- [32. How do you format a disk in Linux?](#)
- [33. Use of htop?](#)
- [34. What is a process ID?](#)
- [35. How to search a pattern in recursive manner?](#)
- [36. What is the quickest way to check the reachability of the server?](#)
- [37. How to troubleshoot using netstat?](#)

- [38. How to check all established connections using netstat?](#)
- [39. How to list symbolic \(soft\) links only?](#)
- [40. How to find all the users belonging to a specific group?](#)
- [41. How to check which process is using port 80?](#)
- [42. What is the lsof command?](#)
- [43. How to find all executable files?](#)
- [44. How to find system uptime?](#)
- [45. How to display open ports with netstat \(TCP protocols\)?](#)
- [46. Difference between ping and telnet?](#)
- [47. How to check file permissions?](#)
- [48. How to display only the username from /etc/passwd?](#)
- [49. How to extract an archive file?](#)
- [50. What does cd - do?](#)
- [51. How do you check the system?](#)
- [52. How do you check open connections?](#)
- [53. How do you display the Linux version?](#)
- [54. Difference between kernel and Linux version?](#)
- [55. How do you identify which process is consuming more CPU?](#)
- [56. How do you kill the specific process?](#)
- [57. How to show the last command?](#)
- [58. How do you check system reboot history?](#)
- [59. How to install packages on the system?](#)
- [60. How to check the difference between files?](#)
- [61. How to check the last modified time of a file?](#)
- [62. How do you find a .txt file modified a month ago and delete it?](#)
- [63. What command is used to list all the files?](#)
- [64. How do you find files .txt files above 7 days?](#)
- [65. How do you check which process is using a specific port?](#)
- [66. How do you display the username?](#)
- [67. How do you check the name logged in the system?](#)
- [68. How do you display who and all login?](#)
- [69. Difference between uname and whoami?](#)
- [70. How do you print the second column in a closed file?](#)
- [71. How do you view the running process?](#)
- [72. How do you check which kernel version you're using?](#)
- [73. How do you display the system information?](#)
- [74. How do you display a compressed file without extracting?](#)
- [75. How do you find files owned by a specific user?](#)
- [76. How do you copy a directory recursively?](#)
- [77. How to compress the directory?](#)
- [78. Can you explain symbolic and numeric representation?](#)

79. How to view all the mounted file systems?
80. Do you know what is mounted?
81. How do you search command location?
82. Do you know what an alias is?
83. If you restart the system alias be present?
84. What is the difference between /bin and /usr/bin?
85. How do you check the current CPU load?
86. How do you check the system's default gateway?
87. How do you check open connections on port 22?
88. What is the difference between the Kernel and the Linux version?
89. How do you kill all processes with a specific name?
90. How do you check the history of system reboots?
91. How to list installed packages in the system?
92. What is apt?
93. How do you display the package name?
94. How to check the open file description?
95. How do you check network traffic on an interface?
96. What are route tables?
97. How to check the differences between two files?
98. How to display the last modified date of that file?
99. Is it possible to keep multiple root users?
100. Why does every Linux system have only 1 root user?
101. rsync and scp?
102. If the transfer is stopped in between, rsync will start from where it was stopped, whereas scp will not?
103. Prerequisites for the rsync or scp command (before running these, what steps do we need to take?)
104. Why does a hard link not have "l" and a soft link will have "l" displayed in the (ls -l command)?
105. How to freeze the terminal?
106. Ctrl + S?
107. Printing the last 10 lines of history
108. What is the difference between a private and a public repository?
109. Difference between memory usage and disk usage?
110. Linux architecture?
111. Lifecycle of Linux?
112. How to count the number of empty lines in a file?
113. How to delete a set of lines in a closed file?
114. How to compress and extract multiple files?
115. How to check if gzip is internal or external?
116. How to find files by permission?

- [117. How to search and replace text in a closed file using sed?](#)
- [118. How to move to the first or last line in the vi editor?](#)
- [119. How to append text to a file without opening it?](#)
- [120. How to compress the log directory?](#)
- [121. Time and date — how to replace time and date?](#)
- [122. Tar command flags?](#)
- [123. How to display lines that do not have a matched pattern?](#)
- [124. How can we lock the username temporarily?](#)
- [125. How to find users with a UID greater than 1000?](#)
- [126. Purpose of /etc/passwd and /etc/shadow \(encrypted passwords\)](#)
- [127. How to change group ownership for a set of files?](#)
- [128. How to change the username in Linux?](#)
- [129. How to lock and unlock a user?](#)
- [130. How to check a deleted user?](#)
- [131. Use case of awk — is it applicable?](#)
- [132. Display only empty lines using awk?](#)
- [133. How to change the group name?](#)
- [134. Where can we use the tee command?](#)
- [135. How to create a directory and a file name?](#)
- [136. Any other command to create a file other than touch?](#)
- [137. How to delete empty lines?](#)
- [138. ls command with its flags?](#)
- [139. How to search and replace in a closed file?](#)
- [140. In the vi editor, copy a range of lines to another file?](#)
- [141. Display all contents of files, excluding empty lines?](#)
- [142. How to list the tar files?](#)
- [143. grep command and its flags?](#)
- [144. Display the 1st and 2nd columns by skipping empty lines?](#)
- [145. Purpose of gzip and gunzip?](#)
- [146. Use case of wget and curl?](#)
- [147. How to show the lines with more than 10 characters?](#)
- [148. How to display the 2nd, 8th, and 11th columns?](#)
- [149. Search and replace multiple patterns in a closed file?](#)
- [150. Skipping the first column by excluding empty lines?](#)
- [151. How to display duplicate lines?](#)
- [152. How to display only blank or empty lines?](#)
- [153. How to manage disk partitions?](#)
- [154. How to display lines ending with a specific word?](#)
- [155. Finding files modified within a specific time?](#)
- [156. Find the processes running on the ports?](#)
- [157. How to find users who have root-level permissions?](#)

- [158. How to find a root user?](#)
- [159. How do you manage groups and users?](#)
- [160. What is the use of apt?](#)
- [161. Why do we use the double dots \(..\) in the cd command? Is there an alternative to switching directories?](#)
- [162. How to display line numbers, 2nd and last column, using the awk command?](#)
- [163. How to find file permissions?](#)
- [164. How to delete lines from 3 to 7 using sed?](#)
- [165. How to display the user name?](#)
- [166. I want to find the files that are more than 200 MB.](#)
- [167. How to check system memory usage?](#)
- [168. host -A?](#)
- [169. htop?](#)
- [170. What is an inode?](#)
- [171. What is a process ID?](#)
- [172. How to find and delete all files that are below 1KB?](#)
- [173. How to check all the netstat commands?](#)
- [174. How to find files 4k to 6k?](#)
- [175. How to find files that are modified in the last 24 hours?](#)
- [176. How do you search and replace 10 to 20 by ignoring cases?](#)
- [177. How to list all the running processes in Linux?](#)
- [178. How to find and delete all the files that are 7 days old?](#)
- [179. How do you check which specific port?](#)
- [180. How to display disk usage?](#)
- [181. How do you display all the users logged in?](#)
- [182. Difference between whoami and uname?](#)
- [183. How do you extract the 2nd column?](#)
- [184. What does .. signify in Linux?](#)
- [185. How do you make a script executable?](#)
- [186. How do you check the kernel version?](#)
- [187. How do you display the system information?](#)
- [188. How do you display the content of a compressed file?](#)
- [189. How do you find all the files owned by specific users?](#)
- [190. How many ways can we create an empty file?](#)
- [191. How to change the permissions of a file?](#)
- [192. How to view all the mounted views on the system?](#)
- [193. What is the difference between shutdown and poweroff?](#)
- [194. How do you create a swap file \(low RAM\)?](#)
- [195. How do you search command location?](#)
- [196. What is an alias?](#)
- [197. What are the directories in the root directory?](#)

- [198. Difference between bin and /bin?](#)
- [199. Explore the directories in the root directory?](#)
- [200. How do you check the current CPU load?](#)
- [201. How to check system memory and CPU usage?](#)
- [202. What is the use of ps -aef?](#)
- [203. How to check open ports in a new / Linux system?](#)
- [204. Purpose of nohup command?](#)
- [205. Purpose of the systemctl command in Linux?](#)
- [206. How to find all the commits executed?](#)
- [207. What is the purpose of git init?](#)
- [208. How to check a specific commit?](#)
- [209. How to get the hash for the commits?](#)
- [210. How to push one particular commit to the remote?](#)
- [211. How to schedule a shell script to run every day at 9 am?](#)
- [212. Different types of loops in shell scripting?](#)
- [213. What will = do in the string operator?](#)
- [214. After creating a remote repository, how to change that particular repository's URL?](#)
- [215. How to change the commits of a few previous commits?](#)
- [216. How to compare the two commits?](#)
- [217. Difference between git push and git push --force?](#)
- [218. Is it a good practice to use git push force?](#)
- [219. How do we undo the git add before committing?](#)
- [220. How do you see the changes made in a file across the commits?](#)
- [221. What are the limitations of crontab?](#)
- [222. What are the uses of cron jobs?](#)
- [223. How do you list cron jobs?](#)
- [224. How do you schedule a shell file to execute every minute?](#)
- [225. What do the 5 stars represent in cron jobs?](#)
- [226. How to clone a repository?](#)
- [227. What is a shell?](#)
- [228. How to run a shell script?](#)
- [229. Schedule a shell script to run every day at midnight?](#)
- [230. How to schedule a job \(shell script\) every 20 mins for multiple files?](#)
- [231. How to add paths between two files in a crontab?](#)
- [232. Difference between -d and -D while deleting the branches?](#)
- [233. What will happen internally when we clone a repository?](#)
- [234. How to debug a shell script?](#)
- [235. How do you write comments in a shell script?](#)
- [236. Will the shell script file execute if it doesn't have the execute permission?](#)
- [237. What is the purpose of reading in the shell script?](#)
- [238. Difference between = and == in shell scripting?](#)

- [239. What is the purpose of the switch statement?](#)
- [240. Uses of shell scripting?](#)
- [241. /bin/bash, which shell will it use to execute the script?](#)
- [242. What is the difference between read and the parameters?](#)
- [243. How do you secure a Linux server?](#)
- [244. How do you optimize Linux system performance?](#)
- [245. What is the difference between CVS and DVCS?](#)
- [246. How do you schedule a recurring task in Linux?](#)
- [247. Purpose of SSH?](#)
- [248. How to see which branch in our local system is tracking the branch in remote?](#)
- [249. What will git add -p do?](#)
- [250. How to limit the number of commits?](#)
- [251. How to see the commits made by the user?](#)
- [252. How do I find which commit modified a specific file?](#)
- [253. Suppose I have a process overlapping with another using the same application. How do I kill it?](#)
- [254. If I want to download content recursively, how can I achieve it?](#)
- [255. How to list CPU information?](#)
- [256. How can you access the server without using the IP or the server name?](#)
- [257. How to compare the content of the files?](#)
- [258. How to find if the package is external or internal?](#)
- [259. How to kill a process using its name?](#)
- [260. How to display the current shell?](#)
- [261. What does \\$_ display?](#)
- [262. Default port numbers \(HTTPS, SSH, Telnet, TCP, UDP\)?](#)
- [263. What is SIGKILL? Explore different types of kill signals.](#)
- [264. What is the difference between Unix and Linux?](#)
- [265. What is a GUI?](#)
- [266. What is the use of a shell script?](#)
- [267. What is a Kernel?](#)
- [268. What is an Interpreter?](#)
- [269. What is a Compiler?](#)
- [270. What is Root User?](#)
- [271. Explain the lifecycle of Linux.](#)
- [272. What are Environment Variables?](#)
- [273. Two types of variables in the shell?](#)
- [274. What are Personal Parameters?](#)
- [275. What does \\$! represent?](#)
- [276. What are Positional Parameters?](#)
- [277. What are Special Parameters?](#)
- [278. Pros and Cons of Shell Scripting](#)

- [279. What is the file system in Linux?](#)
- [280. What are the primary components of Linux?](#)
- [281. What are the three editors in Linux?](#)
- [282. Which is the most used shell script?](#)
- [283. What command is used to read a file?](#)
- [284. Alternative command for echo?](#)
- [285. How do you pass arguments in a shell script?](#)
- [286. Write a shell script to check if a file exists or not.](#)
- [287. What is the difference between '' and " "?](#)
- [288. What is Standard Error?](#)
- [289. How do you read lines from a file in a shell script?](#)
- [290. What is a Cron job?](#)
- [291. What is a Superblock?](#)
- [292. What is the Sleeping Phase?](#)
- [293. What are Special Variables in Shell Script?](#)
- [294. What is the job of a Cron job?](#)
- [295. What is CLI?](#)
- [296. What is an Interpreter?](#)
- [297. Phases of Linux Life Cycle?](#)
- [298. How to display the USB devices in a tree format?](#)
- [299. How to display disk data information?](#)
- [300. How to list all block devices in Linux?](#)
- [301. How to show all hardware information in Linux?](#)
- [302. How to list unmounted disks or partitions?](#)

1. Difference between the grep and sed commands?

Answer:

grep is used for searching and filtering text patterns in files, while sed (stream editor) is used for searching, finding, replacing, inserting, or deleting text in a stream or file.

2. cd tilde (~), where does it go?

Answer:

cd ~ takes you to the home directory of the current user (e.g., /home/username).

3. Remove the execution for the user for a particular file?

Answer:

chmod u-x filename

4. Why do we use passwd?

Answer:

The passwd command is used to set or change a user's password in Linux.

5. How to print all the empty lines if it is present?

Answer:

grep '^\$' filename

6. How to delete the last line of a file?

Answer:

sed '\$d' filename

7. How to remove the group permission of the group?

Answer:

chmod g-rwx filename

8. Which user logs in to the server?

Answer:

Who
or
last

9. Copy a directory from one location to another location recursively?

Answer:

cp -r source_dir destination_dir

10. How to check the size of the file?

Answer:

ls -lh filename

or

du -h filename

11. Definition of shell in Linux?

Answer:

A shell is a command-line interpreter that provides a user interface to access the operating system's services (e.g., bash, zsh, sh).

12. How to check resource utilization in Linux?

Answer:

top

or

htop

13. How to manage disk partitions and file management in Linux?

Answer:

Tools like fdisk, parted, and lsblk are used to create, delete, or modify disk partitions, and df, du, mount, and umount help in file system management.

14. How to set a Linux server against unauthorized service?

Answer:

Disable unwanted services using systemctl disable <service> and use firewalls (ufw/iptables), SELinux, and strong user authentication.

15. Difference between forcefully and gracefully killing in Linux?

Answer:

Graceful kill (SIGTERM - 15): Gives the process time to clean up.

Forceful kill (SIGKILL - 9): Immediately terminates without cleanup.

16. What is the lifecycle of Linux and Ubuntu?

Answer:

It includes development, testing, release, support (LTS/regular), and end-of-life (EOL). Ubuntu LTS versions get 5 years of support.

17. List out files by sorting them largest file first?

Answer:

```
ls -lhS  
or  
du -ah | sort -rh
```

18. What is the dead phase in Linux?

Answer:

A dead (zombie) process has completed execution but still has an entry in the process table waiting for the parent to read its exit status.

19. How to check who the root user is?

Answer:

The root user has UID 0.

Check using:

```
id root
```

20. How to copy permissions of one file to another?

Answer:

```
chmod --reference=file1 file2
```

21. How to delete the first two characters in every line using a Linux command?

Answer:

```
sed 's/^..//' filename
```

22. How to delete the lines where the pattern is not present?

Answer:

```
grep 'pattern' filename  
or using sed:  
sed -n '/pattern/p' filename
```

23. How to delete the unique lines from the file in Linux?

Answer:

```
sort filename | uniq -u
```

24. How to find the total size of a directory in Linux?

Answer:

```
du -sh directory_name
```

25. How to give execute permission to .ssh files in Linux?

Answer:

```
chmod 700 ~/.ssh  
chmod 600 ~/.ssh/authorized_keys
```

26. How to find .log files that are greater than 500 bytes?

Answer:

```
find . -name "*log" -size +500c
```

27. How to check which user had sudo access on a Linux system?

Answer:

```
grep '^sudo' /etc/group  
or  
getent group sudo  
  
dscl . -read /Groups/wheel GroupMembership (macOS)  
Or  
Sudo cat /etc/sudoers
```

28. How to change the owner to admin in Linux?

Answer:

```
sudo chown admin filename
```

29. How to display the top 5 memory-consuming processes?

Answer:

```
ps aux --sort=-%mem | head -n 6
```

30. How to find and delete the files that are older than 1 day in the current directory?

Answer:

```
find . -type f -mtime +1 -delete
```

31. How to sort the processes based on port number?

Answer:

```
sudo netstat -tulpn | sort -k4
```

32. How do you format a disk in Linux?

Answer:

```
sudo mkfs.ext4 /dev/sdX
```

(Replace /dev/sdX with the correct device.)

33. Use of htop?

Answer:

htop is an interactive process viewer that shows real-time CPU, memory, and process usage, allowing you to manage tasks easily.

34. What is a process ID?

Answer:

A Process ID (PID) is a unique identifier assigned by the system to every running process.

35. How to search a pattern in recursive manner?

Answer:

```
grep -r <pattern> <directory>
```

36. What is the quickest way to check the reachability of the server?

Answer:

```
ping <hostname or IP>
```

37. How to troubleshoot using netstat?

Answer:

Use netstat to check open ports, established connections, listening services, and network statistics to identify issues.

38. How to check all established connections using netstat?

Answer:

```
netstat | grep ESTABLISHED
```

39. How to list symbolic (soft) links only?

Answer:

```
find . -type l
```

40. How to find all the users belonging to a specific group?

Answer:

```
getent group groupname
```

```
groups <username>
```

41. How to check which process is using port 80?

Answer:

```
sudo lsof -i :80
```

42. What is the lsof command?

Answer:

lsof stands for List Open Files — it shows all open files and the processes that opened them.

43. How to find all executable files?

Answer:

```
find . -type f -perm /111
```

44. How to find system uptime?

Answer:

```
uptime
```

45. How to display open ports with netstat (TCP protocols)?

Answer:

```
netstat -tuln
```

46. Difference between ping and telnet?

Answer:

ping: Tests network connectivity (ICMP protocol).

telnet: Tests port availability and application-level connectivity (TCP).

47. How to check file permissions?

Answer:

```
ls -l filename
```

48. How to display only the username from /etc/passwd?

Answer:

`cut -d: -f1 /etc/passwd`

49. How to extract an archive file?

Answer:

`tar -xvf archive.tar`

50. What does cd - do?

Answer:

`cd -` switches to the previous working directory.

51. How do you check the system?

Use:

`hostnamectl`

Shows system information like OS, kernel, architecture, etc. `system_profiler`(macOS)

52. How do you check open connections?

Use:

`netstat -tuln` or `ss -tuln`

Lists all open TCP/UDP connections and listening ports.

53. How do you display the Linux version?

Use:

`cat /etc/os-release`

Shows OS name and version. Also, `uname -r` for kernel version.

54. Difference between kernel and Linux version?

Kernel version is the version of Linux core; Linux version (distribution) is the OS built around it.

55. How do you identify which process is consuming more CPU?

Use:

`top` or `ps aux --sort=-%cpu | head`

Lists top CPU-consuming processes.

56. How do you kill the specific process?

Use:

kill <PID> or kill -9 <PID> if it doesn't stop. Find PID using ps -ef.

57. How to show the last command?

Use:

!! to rerun the last command, or history to see all previous ones.

58. How do you check system reboot history?

Use:

Last reboot — shows all previous reboots with date and time.

59. How to install packages on the system?

For Ubuntu/Debian: sudo apt install <pkg>

For RHEL/CentOS: sudo yum install <pkg>

60. How to check the difference between files?

Use:

diff file1 file2 — shows line-by-line differences.

61. How to check the last modified time of a file?

Use:

ls -l filename or stat filename to show modification time.

62. How do you find a .txt file modified a month ago and delete it?

Use:

find . -name "*.txt" -mtime +30 -delete

63. What command is used to list all the files?

Use:

ls -a — lists all files, including hidden ones.

64. How do you find and files .txt files above 7 days?

Use:

```
find . -name "*.txt" -mtime +7
```

65. How do you check which process is using a specific port?

Use:

```
sudo lsof -i :<port> or netstat -tulnp | grep <port>
```

66. How do you display the username?

Use:

```
whoami
```

67. How do you check the name logged in the system?

Use:

```
who — lists all logged-in users.
```

68. How do you display who and all login?

Use:

```
last — shows user login history.
```

69. Difference between uname and whoami?

uname shows system info; whoami shows current user.

70. How do you print the second column in a closed file?

Use:

```
awk '{print $2}' filename
```

71. How do you view the running process?

Use:

```
ps -ef or top
```

72. How do you check which kernel version you're using?

Use:

```
uname -r
```

73. How do you display the system information?

Use:

```
Hostnamectl or system_profiler(macOS)
```

74. How do you display a compressed file without extracting?

Use:

`zcat file.gz or zless file.gz`

75. How do you find files owned by a specific user?

Use:

`find / -user username`

76. How do you copy a directory recursively?

Use:

`cp -r source destination`

77. How to compress the directory?

Use:

`tar -czvf archive.tar.gz directory/`

78. Can you explain symbolic and numeric representation?

Symbolic: `chmod u+x file`

Numeric: `chmod 755 file`

79. How to view all the mounted file systems?

Use:

`df -h or mount`

80. Do you know what is mounted?

Mounting means making a storage device accessible to the OS at a mount point.

81. How do you search command location?

Use:

`which command`

82. Do you know what an alias is?

Alias is a shortcut for a command (e.g., `alias ll='ls -la'`).

83. If you restart the system alias be present?

No, temporary aliases are lost unless added to `~/.bashrc`.

84. What is the difference between `/bin` and `/usr/bin`?

`/bin` has essential binaries; `/usr/bin` has user-level programs.

85. How do you check the current CPU load?

Use:

uptime or top

86. How do you check the system's default gateway?

Use:

ip route show — check line starting with 'default via'.

87. How do you check open connections on port 22?

Use:

netstat -an | grep 22 or ss -an | grep 22

88. What is the difference between the Kernel and the Linux version?

Same as Q54 — kernel = core; Linux version = OS distribution.

89. How do you kill all processes with a specific name?

Use:

pkill process or killall process

90. How do you check the history of system reboots?

Use:

last reboot

91. How to list installed packages in the system?

Ubuntu/Debian: dpkg --get-selections

RHEL/CentOS: rpm -qa

92. What is apt?

APT (Advanced Package Tool) — manages packages on Debian/Ubuntu systems.

93. How do you display the package name?

Use:

dpkg -l | grep <pkg> brew info <pkgname>(macOS)

94. How to check the open file description?

Use:

lsof

95. How do you check network traffic on an interface?

Use:

iftop or nload

96. What are route tables?

Routing tables define how network packets travel. Use ip route to check. netstat -nr(macOS)

97. How to check the differences between two files?

Use:

diff file1 file2

98. How to display the last modified date of that file?

Use:

stat filename

99. Is it possible to keep multiple root users?

No, only one root account (UID 0) exists; others can use sudo.

100. Why does every Linux system have only 1 root user?

For security and control, only one superuser can manage all resources.

101. rsync and scp?

rsync is used for efficient file transfer and synchronization between systems—it copies only the changes (differential sync).

scp is a simple file copy command using SSH—it copies entire files every time.

102. If the transfer is stopped in between, rsync will start from where it was stopped, whereas scp will not?

Yes, correct. rsync supports resuming interrupted transfers using the --partial or --append options.

scp does not resume partial transfers; it restarts from scratch.

103. Prerequisites for the rsync or scp command (before running these, what steps do we need to take?)

Ensure SSH service is running on both systems.

The remote user must have proper permissions.

Install the rsync or openssh-client packages if not installed.

Know the IP address or hostname of the remote system.

Verify network connectivity (ping test).

104. Why does a hard link not have “l” and a soft link will have “l” displayed in the (ls -l command)?

In ls -l, soft links show l at the start (e.g., lrwxrwxrwx) because they are symbolic links (special files pointing to another path).

Hard links are regular files sharing the same inode, so they appear as normal files without l.

105. How to freeze the terminal?

Press Ctrl + S — this freezes (pauses) the terminal output.

To unfreeze, press Ctrl + Q.

106. Ctrl + S?

Used to pause terminal output (software flow control).

To resume, press Ctrl + Q.

107. Printing the last 10 lines of history

Use:

history | tail -10

108. What is the difference between a private and a public repository?

Public repository: Anyone can view or clone the code.Q

Private repository: Access is restricted to authorized users only.

109. Difference between memory usage and disk usage?

Memory usage: Refers to RAM being used by running processes.

Disk usage: Refers to the amount of storage used on the disk drive.

110. Linux architecture?

Main components:

Hardware – physical devices.

Kernel – core managing hardware and resources.

System libraries – provide functions for applications.

System utilities – perform basic operations.

User space – where applications run.

111. Lifecycle of Linux?

Booting → Kernel initialization → init/systemd start → runlevel targets → user login → process execution.

112. How to count the number of empty lines in a file?

Use:

`grep -c '^$' filename`

113. How to delete a set of lines in a closed file?

Use:

`sed '5,10d' filename > newfile`

(Deletes lines 5 to 10 and saves the rest to a new file.)

114. How to compress and extract multiple files?

Compress: `tar -czvf archive.tar.gz file1 file2 dir/`

Extract: `tar -xzvf archive.tar.gz`

115. How to check if gzip is internal or external?

Use:

`type gzip`

If it shows a path like `/usr/bin/gzip`, it's an external binary.

If it says "built-in," it's internal.

116. How to find files by permission?

Use:

`find /path -type f -perm 644`

(Example: finds files with permission 644.)

117. How to search and replace text in a closed file using sed?

Use:

`sed -i 's/oldtext/newtext/g' filename`

118. How to move to the first or last line in the vi editor?

First line: press `gg`

Last line: press `G`

119. How to append text to a file without opening it?

Use:

`echo "text to append" >> filename`

120. How to compress the log directory?

Use:

`tar -czvf logs.tar.gz /var/log`

121. Time and date — how to replace time and date?

View: `date`

Set: `sudo date -s "2025-10-25 10:30:00"`

122. Tar command flags?

-c → create archive

-x → extract

-v → verbose

-f → specify filename

-z → compress with gzip

Example: tar -czvf file.tar.gz dir/

123. How to display lines that do not have a matched pattern?

Use:

grep -v "pattern" filename

124. How can we lock the username temporarily?

Use:

sudo passwd -l username

125. How to find users with a UID greater than 1000?

Use:

awk -F: '\$3 > 1000 {print \$1, \$3}' /etc/passwd

dscl . -list /Users uniqueID | awk '\$2>1000'(macOS)

126. Purpose of /etc/passwd and /etc/shadow (encrypted passwords)

/etc/passwd → stores user account info (username, UID, GID, shell).

/etc/shadow → stores encrypted passwords and password aging info (accessible only by root).

127. How to change group ownership for a set of files?

Use:

chgrp groupname file1 file2

or recursively: chgrp -R groupname directory/

128. How to change the username in Linux?

Use:

`sudo usermod -l newname oldname`

129. How to lock and unlock a user?

Lock: `sudo passwd -l username`

Unlock: `sudo passwd -u username`

130. How to check a deleted user?

Check `/etc/passwd` and `/home/` — if home directory is missing but user entry exists/deleted, you'll see traces.

Or check logs in `/var/log/secure` or `/var/log/auth.log`.

131. Use case of awk — is it applicable?

awk is used for pattern scanning, text processing, and column extraction.

Example: `awk '{print $1, $3}' file.txt`

132. Display only empty lines using awk?

Use:

`awk '/^$/ filename`

133. How to change the group name?

Use:

`sudo groupmod -n newgroup oldgroup`

134. Where can we use the tee command?

tee is used to write output to both the terminal and file simultaneously.

Example: `echo "hello" | tee file.txt`

135. How to create a directory and a file name?

Use:

`mkdir dirname && touch dirname/filename.txt`

136. Any other command to create a file other than touch?

Yes:

> file.txt or echo "" > file.txt or cat > file.txt

137. How to delete empty lines?

Use:

sed -i '/^\$/d' filename

138. ls command with its flags?

ls -l → long listing

ls -a → show hidden files

ls -lh → human-readable sizes

ls -R → recursive

ls -t → sort by modification time

139. How to search and replace in a closed file?

Use:

sed -i 's/old/new/g' filename

140. In the vi editor, copy a range of lines to another file?

Example: Copy lines 10–20 to another file:

:10,20w newfile

141. Display all contents of files, excluding empty lines?

Use:

grep -v '^\$' filename

142. How to list the tar files?

Use:

tar -tvf archive.tar or tar -tzvf archive.tar.gz

143. grep command and its flags?

-i → case-insensitive

-v → invert match

-r → recursive search

-n → show line numbers

-c → count matches

Example: grep -rin "error" /var/log/

144. Display the 1st and 2nd columns by skipping empty lines?

Use:

awk 'NF{print \$1, \$2}' filename

145. Purpose of gzip and gunzip?

gzip → compress files (gzip file.txt → file.txt.gz)

gunzip → decompress (gunzip file.txt.gz)

146. Use case of wget and curl?

wget → download files from the internet (non-interactive).

curl → transfer data via URL, supports APIs, HTTP methods, FTP, etc.

147. How to show the lines with more than 10 characters?

Use:

awk 'length(\$0) > 10' filename

148. How to display the 2nd, 8th, and 11th columns?

Use:

awk '{print \$2, \$8, \$11}' filename

149. Search and replace multiple patterns in a closed file?

Use:

```
sed -i -e 's/pattern1/replace1/g' -e 's/pattern2/replace2/g' filename
```

150. Skipping the first column by excluding empty lines?

Use:

```
awk 'NF{$1=""}; print substr($0,2)' filename
```

151. How to display duplicate lines?

Use the `uniq -d command to show duplicate lines from a sorted file.

Example: `sort file.txt | uniq -d — this sorts the file first (required by `uniq`) and then displays only lines that appear more than once.

152. How to display only blank or empty lines?

You can use `grep '^\$' file.txt` to display only empty lines.

153. How to manage disk partitions?

Use the `fdisk` or `parted` command as root to create, delete, or resize partitions.

Example: `sudo fdisk /dev/sda` → then use options like `n` (new), `d` (delete), or `p` (print).

154. How to display lines ending with a specific word?

Use `grep 'word\$' file.txt` — the `'\$` matches the end of a line.

Example: `grep 'end\$' file.txt` will show all lines ending with "end."

155. Finding files modified within a specific time?

Use the `find` command with `-mtime` (in days) or `-mmin` (in minutes).

Example: `find /path -mtime -1` finds files modified in the last 24 hours.

156. Find the processes running on the ports?

Use `sudo netstat -tulnp` or `sudo lsof -i :PORT`.

Example: `sudo lsof -i :8080` shows which process is using port 8080.

157. How to find users who have root-level permissions?

Check the `/etc/sudoers` file or the `sudo` group.

Example: `grep '^sudo' /etc/group` lists all users with sudo (root) access.

158. How to find a root user?

Simply run `grep root /etc/passwd`.

It will display the root user details, usually: `root:x:0:0:root:/root:/bin/bash`.

159. How do you manage groups and users?

Use commands like `useradd`, `usermod`, `groupadd`, and `passwd`.

Example: `sudo useradd newuser`, `sudo passwd newuser`, and `sudo usermod -aG groupname newuser`.

160. What is the use of apt?

`apt` (Advanced Package Tool) is used to install, update, upgrade, or remove software packages in Debian-based systems.

Example: `sudo apt install vim` installs Vim editor.

161. Why do we use the double dots (..) in the cd command? Is there an alternative to switching directories?

`cd ..` moves you one directory up (to the parent).

Alternative: use `cd -` to switch back to the previous directory.

162. How to display line numbers, 2nd and last column, using the awk command?

Example: `awk '{print NR, \$2, \$NF}' file.txt`

Explanation: `NR` = line number, `\\$2` = 2nd column, `\\$NF` = last column.

163. How to find file permissions?

Use the `ls -l` command.

Example: `ls -l file.txt` → output like ` -rw-r--r--` shows owner/group/others permissions.

164. How to delete lines from 3 to 7 using sed?

Use `sed '3,7d' file.txt`

This deletes lines 3 through 7 and prints the rest to standard output.

165. How to display the user name?

Use the `whoami` command.

Example: `whoami` prints the current logged-in username.

166. I want to find the files that are more than 200 MB.

Use: `find /path -type f -size +200M`

`+200M` means files greater than 200 MB.

167. How to check system memory usage?

Use the `free-h` command.

It shows total, used, and available memory in a human-readable format.

168. host -A?

The `host -A` command displays all DNS records (A, AAAA, MX, etc.) for a domain.

Example: `host -A google.com` shows all DNS record types for Google.

169. htop?

`htop` is an interactive process viewer (like `top`, but better).

You can see CPU, memory, processes, and sort/filter them in real time.

170. What is an inode?

An inode is a data structure that stores metadata about a file (permissions, owner, timestamps, etc.) except its name.

Use `ls -i` to display inodes of files.

171. What is a process ID?

Each running process has a unique PID (Process ID).

Use `ps -e` or `pidof <process>` to find it.

172. How to find and delete all files that are below 1KB?

Use: `find /path -type f -size -1k -delete`

This finds and removes files smaller than 1 kilobyte.

173. How to check all the netstat commands?

Use: `man netstat` or `netstat --help`.

This lists all available options with descriptions.

174. How to find files 4k to 6k?

Use: `find /path -type f -size +4k -size -6k`

This finds files between 4 KB and 6 KB in size.

175. How to find files that are modified in the last 24 hours?

Use: `find /path -mtime -1`

`-1` means modified within the past 1 day (24 hours).

176. How do you search and replace 10 to 20 by ignoring cases?

Use: `sed 's/10/20/Ig' file.txt`

`I` makes it case-insensitive, `g` replaces all matches per line.

177. How to list all the running processes in Linux?

Use `ps -e` or `ps -aux`

You can also use `top` or `htop` for real-time process view.

178. How to find and delete all the files that are 7 days old?

Use: `find /path -type f -mtime +7 -delete`

This deletes files older than 7 days.

179. How do you check which specific port?

Use `sudo netstat -tulnp | grep :PORT` or `sudo lsof -i :PORT`.

Example: `sudo lsof -i :22` checks which process uses port 22.

180. How to display disk usage?

Use `df -h` to show disk usage in human-readable form.

`du -sh *` shows directory-wise usage in the current path.

181. How do you display all the users logged in?

Use the `who` or `w` command.

They display currently logged-in users with login time and terminal.

182. Difference between whoami and uname?

`whoami` → shows current logged-in username.

`uname` → shows system information (kernel name, OS type, etc.).

183. How do you extract the 2nd column?

Use `awk '{print \$2}' file.txt`

This extracts and displays only the 2nd column of data.

184. What does .. signify in Linux?

`..` refers to the parent directory of your current working directory.

Example: `cd ..` moves one level up.

185. How do you make a script executable?

Use `chmod +x script.sh`.

This adds execute permission to the file so you can run it as `./script.sh`.

186. How do you check the kernel version?

Use `uname -r`.

It shows the kernel release version, e.g., `5.15.0-84-generic`.

187. How do you display the system information?

Use `uname -a` for kernel info or `lshw` / `inxi -Fxz` for hardware details.

Example: `uname -a` shows system, kernel, and architecture.

188. How do you display the content of a compressed file?

Use `zcat`, `zmore`, or `zless`.

Example: `zcat file.gz` displays its content without decompressing.

189. How do you find all the files owned by specific users?

Use `find / -user username`

Example: `find /home -user chidu` lists files owned by "chidu."

190. How many ways can we create an empty file?

touch file.txt

> file.txt

cat /dev/null > file.txt

echo -n > file.txt

191. How to change the permissions of a file?

Use the `chmod` command.

Example: `chmod 755 file.txt` gives read/write/execute for the owner, read/execute for others.

192. How to view all the mounted views on the system?

Use `mount` or `df -h`.

Example: `df -h` shows mounted filesystems and usage.

193. What is the difference between shutdown and poweroff?

`shutdown` schedules or gracefully shuts down the system.

`poweroff` immediately turns off the system without delay.

194. How do you create a swap file (low RAM)?

Example:

...

```
sudo fallocate -l 2G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile
...
```

This creates and enables a 2GB swap file.

195. How do you search command location?

Use `which` or `whereis`.

Example: `which python3` shows the executable path of Python.

196. What is an alias?

An alias is a shortcut name for a command.

Example: `alias ll='ls -l'` — next time, typing `ll` runs `ls -l`.

197. What are the directories in the root directory?

Common ones: `/bin`, `/boot`, `/dev`, `/etc`, `/home`, `/lib`, `/opt`, `/root`, `/sbin`, `/usr`, `/var`.

Each serves a specific purpose, like binaries, configs, or system files.

198. Difference between bin and /bin?

`/bin` contains essential command binaries (like `ls`, `cp`, `mv`) needed for all users.

`bin` (without `/`) is just a relative directory — usually refers to the user's local binary directory.

199. Explore the directories in the root directory?

You can list all with `ls /`.

Each directory holds a system area: `/etc` (config), `/var` (logs), `/usr` (user apps), `/tmp` (temp files), `/dev` (devices), etc.

200. How do you check the current CPU load?

Use `uptime` or `top`.

Example: `uptime` shows system load averages for 1, 5, and 15 minutes.

201. How to check system memory and CPU usage?

Use `top`, `htop`, or `free -h`.

`top` shows real-time CPU and memory stats for all processes.

202. What is the use of ps -aef?

`ps -aef` lists all running processes in a full-format listing.

It shows PID, PPID, user, command, and more.

203. How to check open ports in a new / Linux system?

Use `sudo netstat -tuln` or `sudo ss -tuln`.

It lists all open TCP/UDP ports and their listening state.

204. Purpose of nohup command?

`nohup` allows a command to keep running after logout.

Example: `nohup ./script.sh &` runs the script in the background safely.

205. Purpose of the systemctl command in Linux?

`systemctl` manages systemd services and processes.

Example: `sudo systemctl start nginx` or `sudo systemctl enable ssh`.

206. How to find all the commits executed?

Use:

git log

Displays the complete commit history of the current branch.

For a concise one-line summary:

git log --oneline

If you want commits from all branches:

git log --oneline --all

207. What is the purpose of git init?

git init initializes a new Git repository in your current directory.

It creates a hidden .git/ folder which contains all version control information.

208. How to check a specific commit?

To view details of a specific commit:

git show <commit_hash>

209. How to get the hash for the commits?

git log --oneline

210. How to push one particular commit to the remote?

Use:

```
git push origin <commit_SHA>:<branch_name>
```

Example:

```
git push origin abc123:main
```

This pushes only that specific commit to the remote branch.

211. How to schedule a shell script to run every day at 9 am?

Add the following crontab entry -e:

```
0 9 * * * /path/to/script.sh
```

212. Different types of loops in shell scripting?

- for loop
- while loop
- until loop
- select loop

213. What will = do in the string operator?

It checks if two strings are equal. Example:

```
if [ "$a" = "$b" ]; then
    echo "Strings are equal"
fi
```

214. After creating a remote repository, how to change that particular repository's URL?

Use:

```
git remote set-url origin <new_repo_url>
```

215. How to change the commits of a few previous commits?

Use interactive rebase:

```
git rebase -i HEAD~<number_of_commits>
```

Then edit, squash, or modify commits as required.

216. How to compare the two commits?

Use:

```
git diff <commit1> <commit2>
```

217. Difference between git push and git push --force?

git push: pushes changes only if the remote branch is up-to-date.

git push --force: overwrites remote history with your local commits.

218. Is it a good practice to use git push force?

No. It's risky because it rewrites history and may delete others' work. Use --force-with-lease if absolutely needed.

219. How do we undo the git add before committing?

Use:

```
git reset <file>
```

or to unstage all:

```
git reset
```

220. How do you see the changes made in a file across the commits?

Use:

```
git log -p <filename>
```

221. What are the limitations of crontab?

1. Can't handle tasks requiring a system restart.
2. Limited logging and error handling.
3. Doesn't run if the system is off.
4. Executes with minimal environment variables.

222. What are the uses of cron jobs?

To automate repetitive tasks such as backups, cleanup, monitoring scripts, report generation, and scheduled maintenance.

223. How do you list cron jobs?

Use:

crontab -l

224. How do you schedule a shell file to execute every minute?

Add to crontab:

* * * * * /path/to/script.sh

225. What do the 5 stars represent in cron jobs?

minute hour day month weekday

Example:

```
* * * * * command
# | | | |
# | | | | └ Day of week (0-6)
# | | | └ Month (1-12)
# | | └ Day of month (1-31)
# | └ Hour (0-23)
# └ Minute (0-59)
```

226. How to clone a repository?

Use:

```
git clone <repository_url>
```

227. What is a shell?

A shell is a command-line interpreter that provides an interface between the user and the operating system kernel.

228. How to run a shell script?

Use:

```
bash script.sh
```

or make it executable:

```
chmod +x script.sh  
./script.sh
```

229. Schedule a shell script to run every day at midnight?

Add to crontab:

```
0 0 * * * /path/to/script.sh
```

230. How to schedule a job (shell script) every 20 mins for multiple files?

Use:

```
*/20 * * * * /path/to/script1.sh  
*/20 * * * * /path/to/script2.sh
```

231. How to add paths between two files in a crontab?

Use absolute paths in your cron command, e.g.:

```
* * * * * /bin/bash /home/user/scripts/script.sh > /home/user/logs/output.log 2>&1
```

232. Difference between -d and -D while deleting the branches?

-d: deletes a branch safely (only if merged).

-D: force deletes a branch (even if unmerged).

233. What will happen internally when we clone a repository?

git clone creates a new directory, copies all repository data (.git folder), checks out the default branch, and sets the remote URL.

234. How to debug a shell script?

Run:

```
bash -x script.sh
```

Or include at the top of the script:

```
set -x
```

235. How do you write comments in a shell script?

Use # for single-line comments:

```
# This is a comment
```

236. Will the shell script file execute if it doesn't have the execute permission?

Not directly. You can still run it with a bash script.sh, but ./script.sh will fail.

237. What is the purpose of reading in the shell script?

It takes input from the user. Example:

```
read name
echo "Hello, $name"
```

238. Difference between = and == in shell scripting?

=: used for string comparison in POSIX shell (sh).

==: works in extended shells like bash for string comparison.

239. What is the purpose of the switch statement?

Used for multi-way branching in shell scripts. In shell, case acts as a switch statement.

240. Uses of shell scripting?

Automation, server management, backups, log processing, deployments, and monitoring tasks.

241. /bin/bash, which shell will it use to execute the script?

It will use the Bash shell regardless of the user's default shell.

242. What is the difference between read and the parameters?

read captures user input; parameters like \$1, \$2, etc., capture command-line arguments passed to the script.

243. How do you secure a Linux server?

1. Disable root login.
2. Use SSH keys.
3. Keep software updated.
4. Configure firewalls (UFW/iptables).
5. Monitor logs and use intrusion detection tools.

244. How do you optimize Linux system performance?

1. Stop unused services.
2. Use top or htop to monitor processes.
3. Manage memory and swap.
4. Optimize I/O and disk usage.
5. Use caching and load balancing.

245. What is the difference between CVS and DVCS?

CVS (Centralized Version Control System): One central repository (e.g., SVN).

DVCS (Distributed Version Control System): Every user has a full copy of the repo (e.g., Git, Mercurial).

246. How do you schedule a recurring task in Linux?

Use: cron for minute/hour/day-based scheduling or systemd timers for advanced scheduling.

247. Purpose of SSH?

Securely connects to remote systems for administration, file transfer, and tunneling.

248. How to see which branch in our local system is tracking the branch in remote?

Use:

```
git branch -vv
```

249. What will git add -p do?

It stages changes interactively, allowing you to review and stage parts (hunks) of files.

250. How to limit the number of commits?

Use: git log -n <number> to view limited commits, or during rebase use:

```
git rebase -i HEAD~<n>
```

to modify recent commits.

251. How to see the commits made by the user?

Use:

```
git log --author="username"
```

252. How do I find which commit modified a specific file?

Use:

```
git log --follow <filename>
```

to track all commits that changed that file.

253. Suppose I have a process overlapping with another using the same application. How do I kill it?

You can find and kill the overlapping process using its PID:

```
ps aux | grep <application_name>
kill -9 <PID>
```

If multiple instances exist, you can use:

```
pkill -f <application_name>
```

254. If I want to download content recursively, how can I achieve it?

Use the wget command with the -r (recursive) option:

```
wget -r <URL>
```

For example:

```
wget -r https://example.com/
```

255. How to list CPU information?

Use any of the following commands:

```
lscpu
cat /proc/cpuinfo
```

or

```
top
```

to see CPU usage in real time.

256. How can you access the server without using the IP or the server name?

You can configure an alias or entry in the /etc/hosts file, mapping a custom name to the IP:

```
sudo nano /etc/hosts  
# Add line:  
192.168.1.10 myserver
```

Then access it using ssh user@myserver.

257. How to compare the content of the files?

Use the diff command:

```
diff file1.txt file2.txt
```

Or use cmp for byte-level comparison:

```
cmp file1.txt file2.txt
```

258. How to find if the package is external or internal?

If it's installed via the OS package manager (e.g., apt, yum), it's likely internal.

To check package info:

```
dpkg -l | grep <package_name> # Debian/Ubuntu  
rpm -qa | grep <package_name> # RHEL/CentOS
```

External packages are those downloaded manually or from third-party repositories.

259. How to kill a process using its name?

Use:

```
pkill <process_name>
```

Or to ensure all matching processes are killed:

```
killall <process_name>
```

260. How to display the current shell?

Use any of these commands:

```
echo $SHELL
```

or

```
ps -p $$
```

to see which shell the current process is using.

261. What does \$_ display?

\$_ holds the last argument of the previous command.

Example:

```
echo hello
```

```
cat $_
```

Here, \$_ expands to hello.

262. Default port numbers (HTTPS, SSH, Telnet, TCP, UDP)?

20 Common Ports		
Port	Protocol	Use
20	TCP	FTP (Data Transfer)
21	TCP	FTP (Control)
22	TCP	SSH (Secure Shell)
23	TCP	Telnet (Remote Login)
55	UDP/TCP	DNS (Domain Resolution)
67	UDP	DHCP (Server to Client)
68	UDP	DHCP (Client to Server)
69	UDP	TFTP (Trivial File Transfer)
80	TCP	HTTP (Web Browsing)
110	TCP	POP3 (Email Receiving)
123	UDP	NTP (Time Sync)
138	UDP	NetBIOS (Name Service)
139	TCP	NetBIOS (Datagram Service)
143	TCP	IMAP (Email Receiving)
161	UDP	SNMP (Network Management)
443	TCP	HTTPS (Secure Web Browsing)
445	TCP	SMB (File Sharing)
3389	TCP	RDP (Remote Desktop)

263. What is SIGKILL? Explore different types of kill signals.

1. SIGKILL (signal 9) immediately terminates a process and cannot be trapped or ignored.
2. SIGHUP (1): Hang up / reload configuration
3. SIGINT (2): Interrupt (Ctrl + C)
4. SIGQUIT (3): Quit (Ctrl +)
5. SIGTERM (15): Terminate gracefully
6. SIGKILL (9): Force kill, cannot be caught or ignored
7. SIGSTOP (19): Stop process execution

You can list all signals using:

```
kill -l
```

264. What is the difference between Unix and Linux?

Unix: A Proprietary operating system developed in the 1970s; not open source.

Linux: Open-source Unix-like OS developed by Linus Torvalds in 1991.

Unix runs mainly on specific hardware; Linux runs on all platforms.

Linux is free and community-supported; Unix is commercial (e.g., AIX, HP-UX).

265. What is a GUI?

GUI (Graphical User Interface) allows users to interact with the system using graphical elements like windows, icons, and menus instead of command-line text.

266. What is the use of a shell script?

Shell scripting automates repetitive tasks such as file management, backups, system monitoring, and deployments in Linux.

267. What is a Kernel?

The kernel is the core component of the operating system that manages system resources like CPU, memory, devices, and system calls between hardware and applications.

268. What is an Interpreter?

An interpreter executes code line by line, translating it into machine instructions at runtime.

269. What is a Compiler?

A compiler translates the entire source code into executable machine code before running the program.

270. What is Root User?

The root user is the superuser in Linux with full administrative privileges, having unrestricted access to all commands and files.

271. Explain the lifecycle of Linux.

Linux lifecycle includes:

Initialization (boot process)

Kernel loading

System processes start (init/systemd)

User login

Process execution

Shutdown/reboot

272. What are Environment Variables?

Environment variables store system-wide values used by processes and shells, such as paths and user settings.

Example:

```
echo $PATH
```

273. Two types of variables in the shell?

System Variables – predefined by the OS (e.g., \$HOME, \$USER, \$PATH)

User-defined Variables – created by the user in scripts or terminals.

274. What are Personal Parameters?

These are user-specific variables that store user preferences or input data, e.g., custom variables defined within a shell session or script.

275. What does \$! represent?

\$! gives the Process ID (PID) of the last background process executed.

Example:

```
sleep 100 &
echo $!
```

276. What are Positional Parameters?

These are variables used to access arguments passed to a script.

Example:

```
$0 - script name
$1, $2, ... - command-line arguments
$# - number of arguments
```

277. What are Special Parameters?

Special variables with predefined meanings:

\$? – exit status of last command

\$# – number of arguments

\$@ – all arguments as separate words

\$* – all arguments as one string

\$\$ – PID of current shell

\$! – PID of last background process

278. Pros and Cons of Shell Scripting

Pros:

1. Automation of repetitive tasks
2. Easy to write and execute
3. Fast for system operations

Cons:

1. Hard to debug for large scripts
2. Platform dependent
3. Limited GUI capabilities

279. What is the file system in Linux?

Linux uses a hierarchical file system structure starting from / (root). Common file systems include ext4, XFS, Btrfs, and FAT32.

280. What are the primary components of Linux?

1. Kernel
2. Shell
3. File System
4. System Libraries
5. System Utilities

281. What are the three editors in Linux?

1. vi/vim
2. nano
3. gedit

282. Which is the most used shell script?

Bash (Bourne Again Shell) is the most widely used shell for scripting.

283. What command is used to read a file?

Answer: cat command

Example:

```
cat filename.txt
```

284. Alternative command for echo?

Answer: printf

Example:

```
printf "Hello World\n"
```

285. How do you pass arguments in a shell script?

Arguments are passed when executing the script:

```
./script.sh arg1 arg2
```

Access inside the script using \$1, \$2, etc.

286. Write a shell script to check if a file exists or not.

```
#!/bin/bash
read -p "Enter filename: " file
if [ -f "$file" ]; then
    echo "File exists."
```

```
else
echo "File does not exist."
fi
```

287. What is the difference between ' ' and " "?

' ' (single quotes): preserves the literal value of characters.

" " (double quotes): allows variable and command substitution.

Example:

```
name=John
echo '$name' # Output: $name
echo "$name" # Output: John
```

288. What is Standard Error?

Standard error (stderr) is the error output stream (file descriptor 2).

You can redirect it using:

```
command 2> error.log
```

289. How do you read lines from a file in a shell script?

```
while read line; do
echo $line
done < filename.txt
```

290. What is a Cron job?

A cron job is a scheduled task in Linux that runs automatically at specific intervals using the crontab utility.

291. What is a Superblock?

A superblock contains metadata about a file system, such as size, inode info, and block count. It's critical for mounting file systems.

292. What is the Sleeping Phase?

It's the state where a process is waiting for an event or resource (like I/O). The process remains inactive until the event completes.

293. What are Special Variables in Shell Script?

They are predefined variables like \$#, \$@, \$*, \$? , \$\$, and \$! that hold information about the execution state and arguments.

294. What is the job of a Cron job?

To automate repetitive system tasks such as backups, monitoring, log rotation, and system updates.

295. What is CLI?

CLI (Command Line Interface) allows users to interact with the system by typing commands in a text-based terminal.

296. What is an Interpreter?

An interpreter executes code line by line, translating it into machine code on the fly (used in Python, Bash, etc.).

297. Phases of Linux Life Cycle?

1. Bootloader (loads the kernel)
2. Kernel initialization
3. Init/systemd (starts processes)
4. Login prompt
5. User process execution
6. Shutdown

298. How to display the USB devices in a tree format?

Use the following command:

`lsusb -t`

This shows all USB devices connected to the system in a hierarchical (tree) format, displaying their bus, device, and driver information.

299. How to display disk data information?

Use:

```
lsblk -f
```

This lists all block devices (disks and partitions) along with their filesystem type, labels, UUIDs, and mount points.

300. How to list all block devices in Linux?

Use:

```
lsblk
```

This displays all block devices such as hard drives, partitions, and removable media, along with their mount points and sizes.

301. How to show all hardware information in Linux?

Use:

```
sudo lshw
```

This provides a comprehensive hardware report, including CPU, memory, storage, network, and peripheral details.

For a summarized view:

```
sudo lshw -short
```

302. How to list unmounted disks or partitions?

You can use:

```
lsblk
```

or

```
sudo fdisk -l
```

