

# Advanced shell script solutions:

Follow – Krishan Bhatt 

## 1. How do you check if a user is root?

```
#!/bin/bash
if [ "$EUID" -eq 0 ]; then
    echo "You are running as root."
else
    echo "You are not root."
fi
```

## 2. How do you read user input and use it in the script?

```
#!/bin/bash
read -p "Enter your name: " user_name
echo "Hello, $user_name!"
```

## 3. How do you check if a file exists?

```
#!/bin/bash
file="/etc/passwd"
if [ -f "$file" ]; then
    echo "$file exists."
else
    echo "$file does not exist."
fi
```

## 4. How do you list all files in a directory?

```
#!/bin/bash
directory="/etc"
echo "Files in $directory:"
ls "$directory"
```

## 5. How do you find the number of lines in a file?

```
#!/bin/bash
file="/etc/passwd"
lines=$(wc -l < "$file")
echo "The file $file has $lines lines."
```

## 6. How do you create a backup of a directory?

```
#!/bin/bash
src="/etc"
dest="/tmp/backup"
mkdir -p "$dest"
cp -r "$src" "$dest"
echo "Backup completed to $dest"
```

## 7. How do you check for disk space usage?

```
#!/bin/bash
df -h
```

## 8. How do you monitor a running process?

```
#!/bin/bash
process_name="sshd"
if pgrep "$process_name" > /dev/null; then
    echo "$process_name is running."
else
    echo "$process_name is not running."
fi
```

## 9. How do you calculate the sum of two numbers?

```
#!/bin/bash
read -p "Enter first number: " num1
read -p "Enter second number: " num2
sum=$((num1 + num2))
echo "The sum is: $sum"
```

## 10. How do you compress and extract files?

### Compress:

```
#!/bin/bash
tar -czf archive.tar.gz /etc
echo "Files compressed to archive.tar.gz"
```

### Extract:

```
#!/bin/bash
tar -xzf archive.tar.gz
echo "Files extracted from archive.tar.gz"
```

## 11. How do you schedule a task to run in the background?

```
#!/bin/bash
(sleep 10 && echo "Task completed") &
echo "Background task started."
```

## 12. How do you count the number of logged-in users?

```
#!/bin/bash
users=$(who | wc -l)
echo "Number of logged-in users: $users"
```

## 13. How do you create a simple menu in a shell script?

```
#!/bin/bash
echo "1. Show date"
```

```

echo "2. List files"
echo "3. Exit"
read -p "Choose an option: " choice
case $choice in
    1) date ;;
    2) ls ;;
    3) exit ;;
    *) echo "Invalid option" ;;
esac

```

## 14. How do you monitor memory usage?

```

#!/bin/bash
free -h

```

## 15. How do you send an email from a shell script?

```

#!/bin/bash
echo "Test email body" | mail -s "Test Email" user@example.com
echo "Email sent!"

```

## 16. How do you compare two strings?

```

#!/bin/bash
str1="hello"
str2="world"
if [ "$str1" = "$str2" ]; then
    echo "Strings are equal."
else
    echo "Strings are not equal."
fi

```

## 17. How do you find the largest file in a directory?

```

#!/bin/bash
directory="/etc"
largest_file=$(find "$directory" -type f -exec du -h {} + | sort -rh |
head -1)
echo "Largest file: $largest_file"

```

## 18. How do you rename multiple files at once?

```

#!/bin/bash
for file in *.txt; do
    mv "$file" "${file%.txt}.bak"
done
echo "Renamed all .txt files to .bak"

```

## 19. How do you kill a process by name?

```

#!/bin/bash
pkill -f "process_name"
echo "Process killed."

```

## 20. How do you check if a service is active?

```
#!/bin/bash
service_name="ssh"
systemctl is-active --quiet "$service_name" && echo "$service_name is active" || echo "$service_name is inactive"
```

## 21. How do you loop through files in a directory?

```
#!/bin/bash
for file in /etc/*; do
    echo "File: $file"
done
```

## 22. How do you find and delete empty files?

```
#!/bin/bash
find /path/to/directory -type f -empty -delete
echo "Empty files deleted."
```

## 23. How do you check if a command exists on the system?

```
#!/bin/bash
if command -v curl &> /dev/null; then
    echo "curl is installed."
else
    echo "curl is not installed."
fi
```

## 24. How do you create a file with the current timestamp?

```
#!/bin/bash
timestamp=$(date +%Y%m%d_%H%M%S)
touch "file_${timestamp}.txt"
echo "File created: file_${timestamp}.txt"
```

## 25. How do you perform basic arithmetic operations?

```
#!/bin/bash
read -p "Enter first number: " a
read -p "Enter second number: " b
echo "Sum: $((a + b))"
echo "Difference: $((a - b))"
echo "Product: $((a * b))"
echo "Quotient: $((a / b))"
```

## 26. How do you extract a specific column from a text file?

```
#!/bin/bash
cut -d':' -f1 /etc/passwd
```

## 27. How do you monitor real-time logs from a file?

```
#!/bin/bash
tail -f /var/log/syslog
```

## 28. How do you create a script that sends a notification?

```
#!/bin/bash
notify-send "Shell Scripting" "Your script has completed successfully."
```

## 29. How do you generate random passwords?

```
#!/bin/bash
password=$(openssl rand -base64 12)
echo "Generated password: $password"
```

## 30. How do you count words in a file?

```
#!/bin/bash
file="example.txt"
words=$(wc -w < "$file")
echo "Number of words in $file: $words"
```

## 31. How do you automate backups with timestamps?

```
#!/bin/bash
src="/path/to/source"
dest="/path/to/backup_$(date +%Y%m%d_%H%M%S).tar.gz"
tar -czf "$dest" "$src"
echo "Backup saved to $dest"
```

## 32. How do you print the last N lines of a file?

```
#!/bin/bash
tail -n 10 /var/log/syslog
```

## 33. How do you replace text in a file?

```
#!/bin/bash
sed -i 's/old_text/new_text/g' file.txt
```

## 34. How do you find files modified in the last N days?

```
#!/bin/bash
find /path/to/directory -type f -mtime -7
```

## 35. How do you ping a list of servers from a file?

```
#!/bin/bash
while read -r server; do
```

```
ping -c 1 "$server" && echo "$server is reachable" || echo "$server
is unreachable"
done < servers.txt
```

### 36. How do you schedule a cron job with a script?

```
#!/bin/bash
echo "0 2 * * * /path/to/script.sh" | crontab -
echo "Cron job added."
```

### 37. How do you filter unique lines from a file?

```
#!/bin/bash
sort file.txt | uniq > unique_lines.txt
```

### 38. How do you split a file into smaller chunks?

```
#!/bin/bash
split -l 100 large_file.txt chunk_
echo "File split into chunks."
```

### 39. How do you capture the output of a command into a variable?

```
#!/bin/bash
output=$(ls /etc)
echo "Files
```

### 40. How do you send a GET request to a URL?

```
#!/bin/bash
url="https://example.com"
response=$(curl -s "$url")
echo "Response from $url:"
echo "$response"
```

### 41. How do you calculate the factorial of a number?

```
#!/bin/bash
read -p "Enter a number: " num
factorial=1
for ((i = 1; i <= num; i++)); do
    factorial=$((factorial * i))
done
echo "Factorial of $num is $factorial"
```

### 42. How do you extract a compressed .tar.gz file to a specific directory?

```
#!/bin/bash
file="archive.tar.gz"
dest="/path/to/extract"
mkdir -p "$dest"
tar -xzf "$file" -C "$dest"
echo "Extracted to $dest"
```

### 43. How do you check the status of multiple services?

```
#!/bin/bash
services=("ssh" "cron" "nginx")
for service in "${services[@]"; do
    systemctl is-active --quiet "$service" && echo "$service is running"
|| echo "$service is stopped"
done
```

### 44. How do you find and replace text in multiple files?

```
#!/bin/bash
find /path/to/files -type f -name "*.txt" -exec sed -i
's/old_text/new_text/g' {} +
echo "Text replaced in all files."
```

### 45. How do you calculate the disk usage of a directory?

```
#!/bin/bash
dir="/path/to/directory"
usage=$(du -sh "$dir")
echo "Disk usage of $dir: $usage"
```

### 46. How do you create a progress bar in a script?

```
#!/bin/bash
for i in {1..100}; do
    echo -ne "Progress: $i%\r"
    sleep 0.05
done
echo -ne "\nDone!\n"
```

### 47. How do you find duplicate files in a directory?

```
#!/bin/bash
find /path/to/directory -type f -exec md5sum {} + | sort | uniq -w32 -d
```

### 48. How do you restart a service if it is not running?

```
#!/bin/bash
service="nginx"
if ! systemctl is-active --quiet "$service"; then
    systemctl restart "$service"
    echo "$service restarted."
else
    echo "$service is running."
fi
```

### 49. How do you check the number of open file descriptors?

```
#!/bin/bash
echo "Open file descriptors: $(lsof | wc -l)"
```

## 50. How do you create a script that executes commands in parallel?

```
#!/bin/bash
commands=("sleep 3" "ls -l" "date")
for cmd in "${commands[@]}; do
    ($cmd &)
done
wait
echo "All commands executed."
```