LINUX PROGRAMING: ASSINMENT 5

1. What is a shell in Linux OS? How many categories of shell is currently exists in Linux? Why bash shell is very popular in Linux distribution?

A shell in Linux is a command-line interpreter that allows users to interact with the operating system by executing commands, running scripts, and managing files and processes. It acts as a bridge between the user and the kernel.

Categories of shells in Linux:

- Bourne Again Shell (bash)
- C Shell (csh)
- Korn Shell (ksh)
- Z Shell (zsh)
- TENEX C Shell (tcsh)
- Bourne Shell (sh)

Why bash is popular:

- It's the default shell in most Linux distributions.
- Offers powerful scripting capabilities.
- Supports command history, auto-completion, and job control.
- Compatible with Bourne shell scripts.

2.What does the ls -Z command display?

The ls -Z command shows SELinux security context labels for files and directories. It's useful for managing access control in SELinux-enabled systems.

3.Write a command to list all hidden files in the current directory.

ls -d

4.Explain the difference between hard links and soft links (symbolic links) in Linux.

Feature	Hard Link	Soft link
Points to	Actual data on disk	File name/ path
Works across FS	No	Yes
Broken if file deleted	No	Yes
Command	In file linkname	In -s file linkname

5.A file has permissions -rwxr-x--x. Explain who can read, write, and execute it

Owner: read (r), write (w), execute (x)

- Group: read (r), execute (x)
- Others: execute (x)

so

- Only the owner can modify the file.
- Group and others can execute it.
- Group can also read it; others cannot.

6. Write the command to change the group ownership of a file data.txt to group

staff.

Chgrp staff data.txt

7. Why is it dangerous to give 777 permissions to a file? Explain with an example.

Giving 777 means read, write, and execute permissions to everyone:

- Owner, group, and others can modify or delete the file.
- Example:

    chmod 777 script.sh

Anyone can edit or run this script, potentially injecting malicious code.

8. What is the difference between apropos (i.e., man -k) and whatis (i.e., man -f)?

- Apropos / man -k: Searches manual page descriptions for a keyword.

  Example: apropos copy
- whatis / man -f: Displays a brief description of a command.

    Example: whatis cp

9. Write a command to redirect the error output of a command to a file named error.log.

command 2> error.log

Replace command with the actual command you're running.

10. How can you use the tee command to append output to a file instead of overwriting it?

command | tee -a output.txt

This appends the output of the command to output.txt instead of replacing its content

LINUX PROGRAMING: ASSIGNMENT 6

1.Which command is used to list the contents of a directory? Justify with proper example.

Command: ls

- Justification: The ls command lists files and directories in the current or specified directory.
- Example:

ls /home/shweta/Documents

2. Write the command to create a new directory named 123test_dir.

Command:

mkdir 123test_dir

3. What is the purpose of the sed command? Justify with proper example.

Purpose: sed is a stream editor used to perform basic text transformations on an input stream.

- Example:

sed 's/Linux/Unix/' file.txt

This replaces the first occurrence of "Linux" with "Unix" in each line of file.txt.

4. Which distinct command is used to display one-line descriptions of any commands?

Command: whatis

Example:

whatis ls

Output: ls (1) - list directory contents

5. Write the command to create an empty file named “notes.txt”.

Command:

touch notes.txt

6. Differentiate between grep and awk commands with an example.

Feature	grep	awk
Purpose	Searches for patterns in text	Pattern scanning and processing language
Usage	Line-based filtering	Field-based processing
Example	grep “error”logfile.txt	awk’{print $2}’ logfile.txt

7. Write the command to give read, write, and execute permission to the owner of a file script.sh.

Command:

chmod u+rwx script.sh

8. How is chown different from chgrp? Give one example for each.

. chown: changes the file owner

sudo chown shweta script.sh

. chgrp: changes group ownership

sudo chgrp developers script.sh

9. A user complains that they cannot execute a file even though it exists in their directory. How would you troubleshoot this using ls -l, chmod, and whoami?

. step 1: check permissions

ls -l filename

. step 2: add execute permission is missing

chmod +x filename

. step 3: conform user identity

Whoami

10. Design a command pipeline to: find all .log files modified in the last 2 days in /var/log, display them on screen, and save the results into a file recent_logs.txt using tee command.

Command pipeline to find recent .log files

find /var/log -name "*.log" -mtime -2 | tee recent_logs.txt

Explanation:

. find: searches for .log files modified in last 2 days

. tee: displays output and saves to recent_logs.txt

LINUX PROGRAMMING: ASSIGNMENT 7

1. What is a bash shell script? Give one example.

A Bash shell script is a text file containing a sequence of commands written for the Bash shell to automate tasks.

Example:

```
#! / bin/bash
echo "System Info:"
una me -a
```

2. Write a simple shell script to print “Hello World”.

```
#! / bin/bash
echo "Hello World"
```

3. What is the purpose of comments (#) in a shell script?

Purpose of comments (#) in shell scripts

- Comments are used to explain code, improve readability, and prevent execution of specific lines.
- Example:

```
# This script prints Hello World
echo "Hello World"
```

4.How do you declare variables (int, float, double, string, Boolean, and char in a shell script?

Bash treats all variables as strings unless used in arithmetic context.

Type	declaration example
Int	num=10
float	num=3.14(use bc for operations)
double	same as float
string	name=" Shweta"
Boolean	flag=true or flag=flase
Char	char='A'

5.Write a shell script to display the current date and time of the system.

Shell script to display current date and time

```
#! / bin/bash
echo "Current Date and Time: $(date)"
```

6.Explain the difference between a constant and a variable in bash script.

Difference between constant and variable

Feature	constant	variable
Value	Fixed value, cannot be Changed	Can change during execution
Example	readonly pi=3.14	name="Shweta"

7. Write a shell script to read two integer number from the user and compute the sum of both the number.

Shell script to read two integers and compute sum

```
#! / bin/bash

read -p "Enter first number: " a

read -p "Enter second number: " b

sum=$((a + b))

 echo "Sum: $sum"
```

8. What is the use of source command in shell scripting?

Use of source command

- Loads and executes a script in the current shell environment.
- Useful for applying environment changes without starting a new shell.

  Example:

```
source ~/. bashrc
```

9. How can you debug a shell script? Give two methods.

Debugging a shell script

Two methods:

1. Using -x option:

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

2. Insert debug statements:

```
echo "Debug: Variable x = $x"
```

10. Write a bash script to create and delete a file.

```
#!/bin/bash
touch myfile.txt
echo "File created: myfile.txt"
rm myfile.txt
echo "File deleted: myfile.txt"
```

LINUX PROGRAMMING: ASSIGNMENT 8

1.What is a user-defined function in shell scripting? Explain with an example.

A user-defined function in shell scripting is a reusable block of code that performs a specific task. Functions help organize code, reduce repetition, and improve readability.

Syntax:

```
function_name() {
  # commands
}
```

Example:

```
Greet_user() {
  echo "Hello, $1! Welcome to Linux scripting."
}
Greet_user "Shweta"
```

2. Write a bash script with a function that multiply two integer numbers.

```
#! / bin/bash

multiply () {
  result=$(( $1 * $2 ))
  echo "Multiplication of $1 and $2 is: $result"
}
multiply 5 7
```

3. Explain how arrays (1D, 2D, and 3D) are declared in bash scripting.

declaring arrays in Bash (1D, 2D, 3D)

- 1D Array:

  fruits=("apple" "banana" "cherry")

- 2D Array (Simulated using nested indexing):

  matrix [0,0] =1

  matrix [0,1] =2

  matrix [1,0] =3

  matrix [1,1] =4

- 3D Array (Simulated with associative arrays or naming convention):

  declare -A cube

  cube ["0,0,0"] =10

  cube ["0,0,1"]=20

Bash doesn't support true multi-dimensional arrays, but you can simulate them using associative arrays or indexed keys.

4. Write a shell script to display elements of an array.

Shell script to display elements of an array

```
#! / bin/bash
colors= ("red" "green" "blue" "yellow")
echo "Array elements:"
for color in "${colors [@]}"
do
    echo "$color"
done
```

5. What is the purpose of cron in Linux?

purpose of cron in Linux

Cron is a time-based job scheduler in Linux used to automate tasks such as

backups, updates, and monitoring. It runs commands or scripts at specified times and intervals.

6. Write a cron job to run a backup script every day at midnight.

cron job to run a backup script daily at midnight

To schedule a cron job:

0 0 * * * /home/user/backup.sh

Add this line to your crontab using crontab -e.

7. How do you schedule a one-time job using at command?

schedule a one-time job using at command

at 10:00 AM tomorrow

Then type the command to run:

/home/user/backup.sh

Press Ctrl+D to save and exit.

8. Write a script to display disk usage using df and du.

script to display disk usage using df and du

```
#! / bin/bash
echo "Disk usage summary:"
 df -h
 echo -e "\nDirectory usage:"
 du -sh /home/user/*
```

9. How can you log the output of a script using the tee command?

logging output using tee comman:

```
#! / bin/bash
echo "Starting backup..." | tee -a backup.log
```

```
tar -czf backup.tar.gz /home/user | tee -a backup.log
echo "Backup completed." | tee -a backup.log
```

tee logs output to both the terminal and a file.

10. Explain with an example how shell scripting can automate system administration tasks.

Automating system administration tasks with shell scripting:

Shell scripts can automate repetitive tasks like user management, backups, updates, and monitoring.

```
#! / bin/bash
add_user() {
  username=$1
  useradd "$username"
  echo "User $username created successfully."
}
add_user "newuser"
```

This script simplifies user creation and can be extended to include password setup, group assignment, etc.

LINUX PROGRAMMING: ASSIGNMENT 9

1.Write a shell script using if…else to check if a number is even or odd.

shell script to check if a number is even or odd using if…else

```
#! / bin/bash
 read -p "Enter a number: " num
 if (( num % 2 == 0 ))
then
echo "$num is even."
Else
 echo "$num is odd."
fi
```

2. Explain the difference between if and case statements in bash.

difference between if and case statements in Bash

Feature	if statement	case statement
Use case	conditional logic with Expressions.	pattern matching for multiple values
Syntax	uses if, elif, else blocks	uses case, pattern),;;
Best For	complex conditions	multiple discrete options

Example:

- if: Check if a number is positive
- case: Menu selection like 1) Start, 2) Stop

3. Write a script to find the largest of three numbers entered by the user.

script to find the largest of three numbers

```
# ! /bin/bash
read -p "Enter first number: " a
read -p "Enter second number: " b
read -p "Enter third number: " c
if (( a >= b && a >= c ))
then
  echo "$a is the largest."
elif (( b >= a && b >= c ))
then
  echo "$b is the largest."
else
  echo "$c is the largest."
fi
```

4. How do you use a for loop to traverse an array in bash? Give an example. The array is defined as arr= (123, “Abs”, -2.3, ‘A’, 23.56, 0).

for loop to traverse an array

```
#! / bin/bash
arr=
```

```
(123 "Abs" -2.3 'A' 23.56 0)
echo "Array elements:"
for item in "${arr[@]}"
do
  echo "$item"
done
```

5. Write a shell script to loop through all files in the current directory and display their names.

```
#! / bin/bash
echo "Files in current directory:"
for file in *
do
  if [ -f "$file" ]; then
    echo "$file"
  fi
done
```

6. What is the difference between while and until loops in bash?

difference between while and until loops

Loop Type	condition check	Loop Runs When
while	while [condition]	condition is true
until	until [condition]	condition is flase

Example:

```
# while loop
count=1
while [ $count -le 5 ]
do
  echo "Count is $count"
  ((count++))
done

# until loop
count=1
until [ $count -gt 5 ]
do
  echo "Count is $count"
  ((count++))
Done
```

7. Write a countdown timer script using a while loop.

countdown timer using while loop

```
#! / bin/bash
count=10
while [ $count -gt 0 ]
do
  echo "$count"
```

```
  sleep 1
  ((count--))
done
echo "Time's up!"
```

8. How do you use break and continue statements in loops? Give examples.

using break and continue in loops

```
#! / bin/bash
for i in {1..5}
do
 if [ $i -eq 3 ]; then
  echo "Skipping 3"
  continue
 fı
 if [ $i -eq 4 ]; then
  echo "Breaking at 4"
  break
 fı
 echo "Number: $i"
done
```

9. Write a script to check if a file exists or not using the if and else loop.

```
      script to check if a file exist
#! / bin/bash
read -p "Enter filename: " filename
if [ -e "$filename" ]
then
 echo "File '$filename' exists."
else
 echo "File '$filename' does not exist.
fi
```

10. Write a script to calculate factorial of a number using for loop.

```
   script to calculate factorial using for loop
```

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