Linux Programming - Assignment 7

Name – Suraj P Das

Batch - 3C
Roll no. - 66
USN - ENG25CY1006

1. Q1. What is a bash shell script? Give one example.

A bash shell script is a plain text file containing a sequence of commands and control structures written for the Bourne Again SHell (bash). It automates tasks by running commands in order. Scripts usually start with a shebang line that tells the system which interpreter to use.

```
# Example: hello.sh
#!/bin/bash
echo "This is a simple bash script."
```

2. Q2. Write a simple shell script to print “Hello World”.

A minimal bash script to print Hello World.

```
# hello_world.sh
#!/bin/bash
echo "Hello World"
```

3. Q3. What is the purpose of comments (#) in a shell script?

Comments (lines starting with #) document the script's intent, explain tricky parts, and can temporarily disable commands. They are ignored by the shell and help maintain readability and future edits.

4. Q4. How do you declare variables (int, float, double, string, Boolean, and char in a shell script?

Bash does not enforce static types like C. All variables are strings, but you can treat them as numbers in arithmetic contexts. Examples:

```
# integer (used in arithmetic)
count=5
# float (bash doesn't have native float arithmetic without external
tools)
price="12.34"  # treat as string, use bc for calculations
# string
name="Mukesh"
```

```
# boolean (convention)
is_valid=true  # use strings 'true'/'false'
# char (single-character string)
letter="A"
```

5. Q5. Write a shell script to display the current date and time of the system.

Use the date command to print a readable timestamp.

```
#!/bin/bash
# show current date & time
echo "Current date and time: $(date '+%Y-%m-%d %H:%M:%S')"
```

6. Q6. Explain the difference between a constant and a variable in bash script.

A variable stores a value that can change during script execution. A constant is a value that should not change; map this in bash by convention or by using readonly to prevent reassignment. Example: readonly PI=3.14 prevents PI from being changed.

7. Q7. Write a shell script to read two integer number from the user and compute the sum of both the number.

Read input with read and use arithmetic expansion $((...)).

```
#!/bin/bash
read -p "Enter first integer: " a
read -p "Enter second integer: " b
sum=$((a + b))
echo "Sum = $sum"
```

8. Q8. What is the use of source command in shell scripting?

The source command (or dot: .) reads and executes commands from a file in the current shell environment. It's used to load functions, variables, or configuration into the current shell without starting a subshell. Example: source ./env.sh

9. Q9. How can you debug a shell script? Give two methods.

Two common debugging methods:
1) Run with -x: bash -x script.sh (prints each command after expansion).
2) Insert set -u (treat unset variables as errors) and set -e (exit on error) or use echo/debugging prints and trap commands to inspect flow.

10. Q10. Write a bash script to create and delete a file.

Use touch to create and rm to delete. Check for existence to be safe.

```
#!/bin/bash
file="tempfile.txt"
# create
if [ ! -e "$file" ]; then
  touch "$file"
```

```
  echo "File created: $file"
else
  echo "File already exists: $file"
fi
# delete
read -p "Delete file? (y/n): " ans
if [ "$ans" = "y" ]; then
  rm -f "$file"
  echo "File deleted"
fi
```