**Linux Programming: Assignment-7:**

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

Ans. Bash Shell script is a file containing series of commands that bash (Bourne Again Shell) interpreter can execute sequentially

Ex: #! /bin/bash

echo “this prints a message”

echo “HI…. How are you???”

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

Ans.#! /bin/bash

echo “HelloWorld”

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

Ans. The (#) in the shell script is used for documentation purpose i.e., to explain the written statements. The interpreter ignores the line starting with (#), and hence the line is not executed.

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

Ans. In shell scripting, it treats all the variables as strings by default. We can’t explicitly declare types like int, float, double etc.…

Ex: dice=6

Name=” James”

Pi= ”3.14”

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

Ans.#! /bin/bash

echo “Current date and time is:”

date

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

Ans.

|  |  |
| --- | --- |
| **constant** | **variable** |
| The value of constant cannot be changed | The value of variable can be changed |
| You need to use read-only built-in command to use a constant | All standard bash variables are by default variables |
| Ex: SCALE=15 | Ex: readonly PI=3.14 |

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

Ans. #! /bin/bash

echo "Enter the first number:"

read n1

echo "Enter the second number:"

read n2

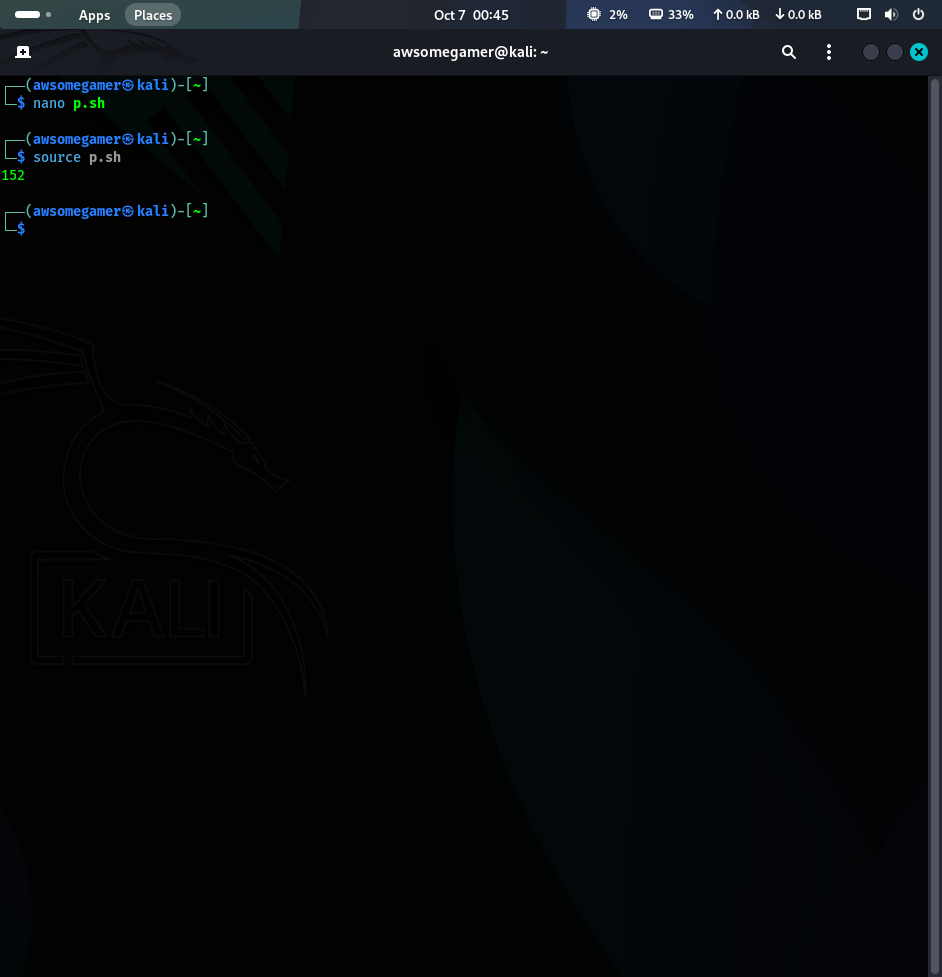
sum=$((n1 + n2))

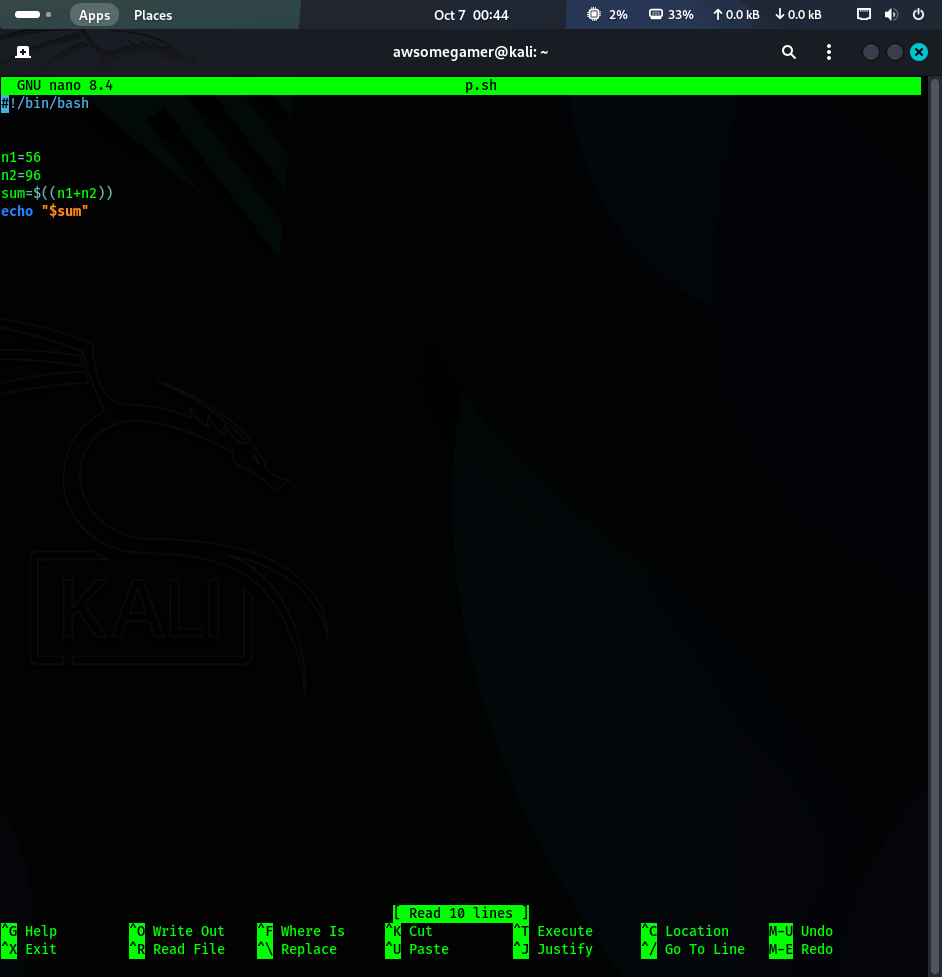
echo "The sum is: $sum"

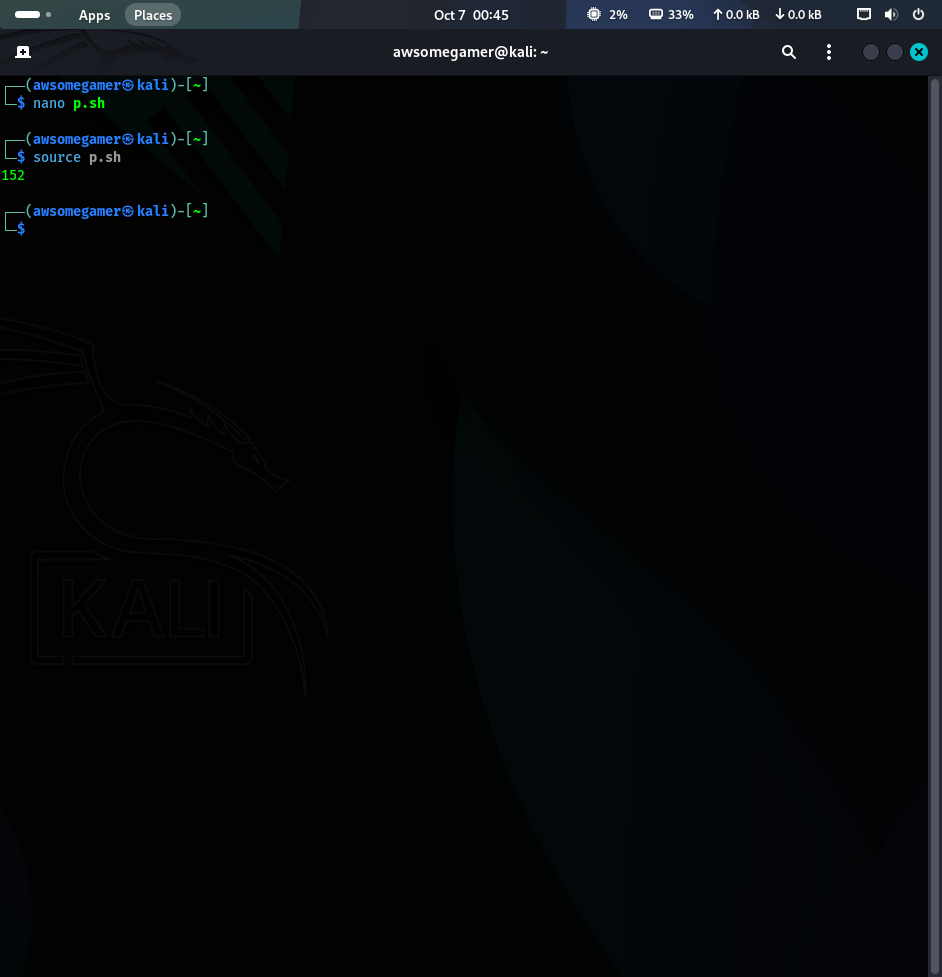
1. **What is the use of source command in shell scripting?**

Ans. Command: source

It is used to read and execute commands from file in the present shell environment. This means that variables, functions defined in [filename] become available in the shell after source command completes.







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

Ans.

* Command 1: set -x

This command is commonly used to enable tracing. prints each command and its arguments after parameter substitution and variable expansion, but before the command is executed.

* Command 2: set -n

This reads the command but doesn’t execute them. This is useful for syntax check without running script’s command.

1. **Write a bash script to create and delete a file.**

Ans. #! /bin/bash

FILE\_NAME="temp\_file.txt"

echo "Attempting to create file: $FILE\_NAME"

touch "$FILE\_NAME"

if [ -f "$FILE\_NAME”]; then

echo "File created successfully."

else

echo "Error: File creation failed."

exit 1

fi

echo "Attempting to delete file: $FILE\_NAME"

rm "$FILE\_NAME"

if [! -f "$FILE\_NAME”]; then

echo "File deleted successfully."

else

echo "Error: File deletion failed."

exit 1

fi

echo "DONE."