Assingment 01 OS

08 Ayush Chougule

Input:

```
op=0

while [ "$op" -ne 5 ]
do

echo "Choose operation:"
echo "1. Addition (+)"
echo "2. Subtraction (-)"
echo "3. Multiplication (*)"
echo "4. Division (/)"
echo "5. Exit"
read op

if [ "$op" -eq 5 ]; then
echo "Exiting..."
break

fi

echo "Enter first number:"
read a
```

```
echo "Enter second number:"
read b

if [ "$op" -eq 1 ]; then
echo "Result: $((a + b))"
elif [ "$op" -eq 2 ]; then
echo "Result: $((a - b))"
elif [ "$op" -eq 3 ]; then
echo "Result: $((a * b))"
elif [ "$op" -eq 4 ]; then
if [ "$b" -ne 0 ]; then
echo "Result: $((a / b))"
else
echo "Error: Division by zero"

fi

else
echo "Invalid option."
fi

done
```

Output:

Choose operation:

1. Addition (+)

2. Subtraction (-)

3. Multiplication (*)

4. Division (/)

5. Exit

1

Enter first number:

5

Enter second number:

6

Result: 11

Choose operation:

1. Addition (+)

2. Subtraction (-)

3. Multiplication (*)

4. Division (/)

5. Exit

2

Enter first number:

5

Enter second number:

6

Result: -1

Choose operation:

1. Addition (+)

2. Subtraction (-)

3. Multiplication (*)

4. Division (/)

5. Exit

3

Enter first number:

5

Enter second number:

6

Result: 30

Choose operation:

1. Addition (+)

2. Subtraction (-)

3. Multiplication (*)

4. Division (/)

5. Exit

4

Enter first number:

5

Enter second number:

0

Error: Division by zero

Choose operation:

1. Addition (+)

2. Subtraction (-)

3. Multiplication (*)

4. Division (/)

5. Exit

5

Exiting...

Name: Ayush. D. Chougule.
Roll no: 06 - D1 batch
TY-CSE
PRN: 10322231720.

OS. RAM 28/7/25

Assingment - 01

* Problem Statement

i) Write a shell script to perform Arithmatic operations.
ii) Write a shell script to manage files and directory.

* Objective

i) Write menu driven shell script, to perform "+, -, *, /" operations.
ii) Write a comand or script to :
- Create a directory and subdirectory.
- Create a .txt file using cat command.
- Make a copy of it, rename it.
- Set read/write/execute for owner and read-only for group/others.
- List all files in the dictectory with owner, group and others rights.

⇒ Understanding of Linux shell commands & shell programming.

* Theory:

i) Linux shells, Types of shells, shell scripting, applications, examples

- Linux shell : A command line interface that users to communicate with linux

It executes commands, manages proces., handles file operations and runs scripts.

- Types of shells:

- Bash ⇒ default in most of Linux
- Zsh ⇒ Advanced, customisable shell
- Ksh ⇒ Korn shell with scripting improvements.
- Tcsh ⇒ enhanced C shell with better scripting
- Fish ⇒ user-friendly with syntax highlighting.

- Shell scripting : Writing a series of commands a file (script)

- Application :

- Automating system maintaince tasks like file cleanups and updates.
- Running scheduled jobs using cron.
- Deploying software and managing server enviorment
- Procesing data and log files.

- Eg: #!/bin/bash
  echo "Hello word"

Scripts enhance productivity, reduces manual efforts, ensures consistency & save time.

~~Impo~~

* FAQ's

1]. What is the significance of the ~~strong~~ shebang (#) at begining of a shell script? Explain with example of #!/bin/sh.

→ The shebang (#!) at the begining of a shell script specifies the path to the interpreter that should execute the script. It tell the system to use the 'Sh shell' to run the script, ensuring compatibility

echo

and correct execution.

eg:- #! /bin/bash
echo "Running with sh shell"

2] Write a script to backup all .txt files from curr directory into a new directory named backup & compress it.

→ mkdir -p backup // create a backup directory.

cp *.txt backup // copy all .txt files backup directory

tar -czf backup.tar.gz backup // compress the backup directory into a tar.gz file.

3] Write a shell script to genrate the fibonacci sequence

→ Intialise a = 0
b = 1

echo "fibonacci sequence upto 10 terms"

for ((i=0 ; i<10 ; i++))
do
echo -n "$a"
fn = $((a+b))
a = $ b
b = $ fn
done

echo

4] Compare the shell script with any other script that automates tasks, control systems or manipulate files.

→ Shell scripts automate tasks in linux / Unix systems using command line utilities, They are ideal for file manipulation, system control and task automation.

In contrast, python scripts offer greater flexibility, better error handeling and extensive libraries, making them suitable for complex tasks like data procesing or ~~rote~~ automation. while shell scripts excel in quick system tasks, Python is preferd for cross-platform automation & more advanced logic & data handeling.

* Conclusion: The linux shell provides several commands and programming language constructs to implement various operation.

* Input:

menu:

1. Addition
2. Subtraction
3. multiplication
4. Division

Choice: 1

Enter first no.: 8

Enter second no: 2

* Output:

Result: 4