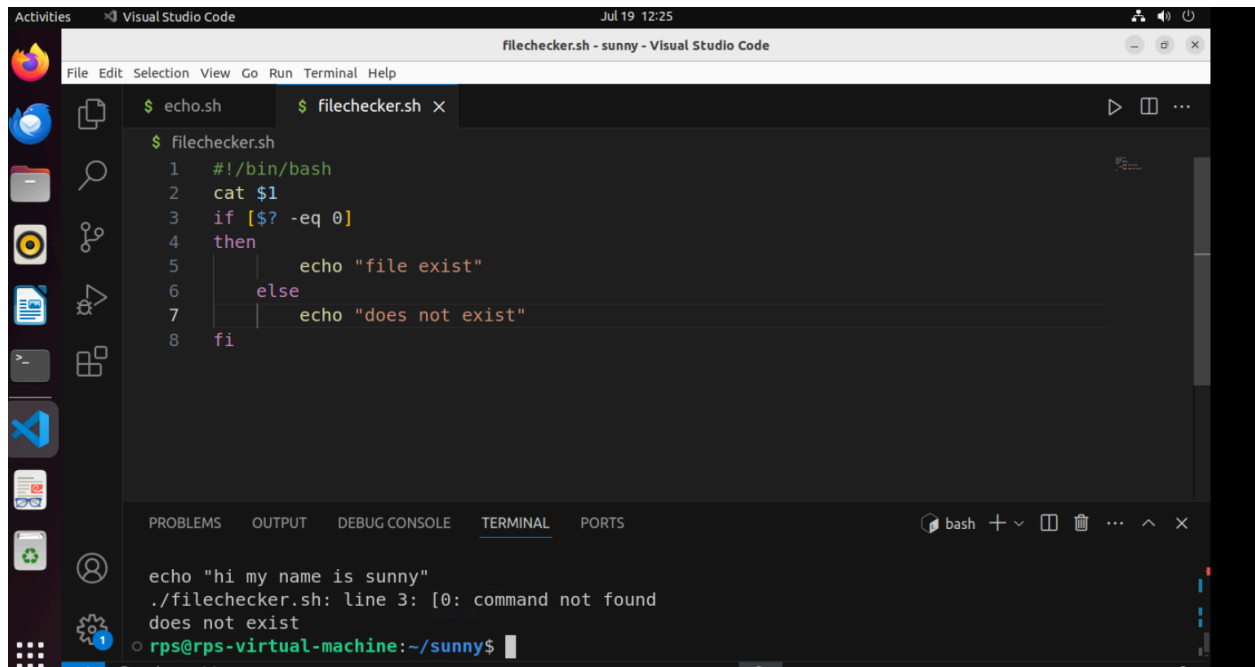


SUNNY KUMAR
19 JULY TASK

LSP(LINUX SYSTEM PROGRAMMING)

Q. writing a shell script called `file_checker.sh` that checks if a file exists or not, takes in a file name as an argument, and prints whether the file exists. The script should suppress the actual output of `cat` and only include the script's output.



The screenshot shows the Visual Studio Code editor with a file named `filechecker.sh` open. The script content is as follows:

```
1 #!/bin/bash
2 cat $1
3 if [ $? -eq 0 ]
4 then
5     echo "file exist"
6 else
7     echo "does not exist"
8 fi
```

Below the editor, the terminal window shows the execution of the script. It first runs `echo "hi my name is sunny"`, then `./filechecker.sh`, which outputs `does not exist` because the file `filechecker.sh` does not exist in the current directory. The terminal prompt is `rps@rps-virtual-machine:~/sunny$`.

`$? :-`

is a special variable that holds the exit status of the last executed command. The exit status is a numerical value returned by the command when it finishes execution. By convention, an exit status of `0` means that the command was successful, while any non-zero value indicates that an error occurred.

Example

```
#!/bin/bash
```

```
# Run a command
```

ls /some/directory

```
# Check the exit status of the command
if [ $? -eq 0 ]; then
    echo "Command succeeded."
else
    echo "Command failed."
fi
```

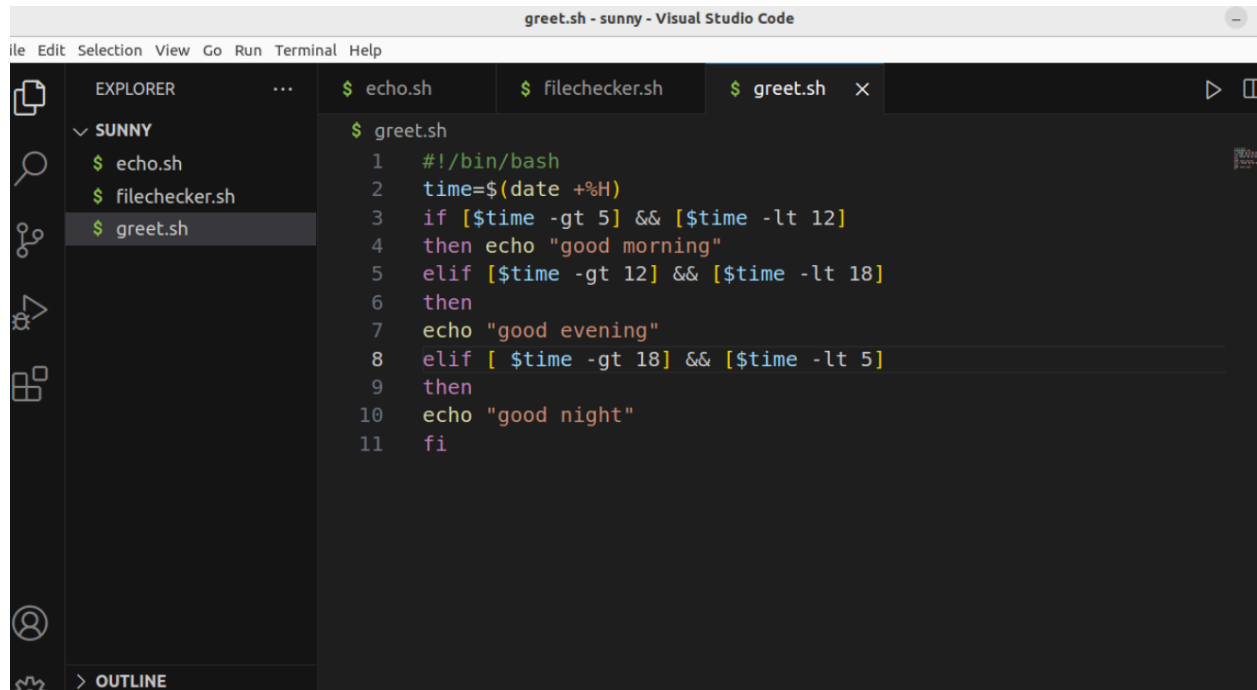
- \$\$ is the PID of the current process.
- \$? is the return code of the last executed command.
- \$# is the number of arguments in \$*
- \$* is the list of arguments passed to the current process

Q. Write a shell script called `timely_greeting` .sh that greets you based on the current time. The script should call the `date` command, extract the current hour (look into using `%H`) and then print the following greeting based on the time.

If it is between 5AM (05:00) and 12PM (12:00):
Good morning!

If it is between 12PM (12:00) and 6PM (18:00):
Good afternoon!

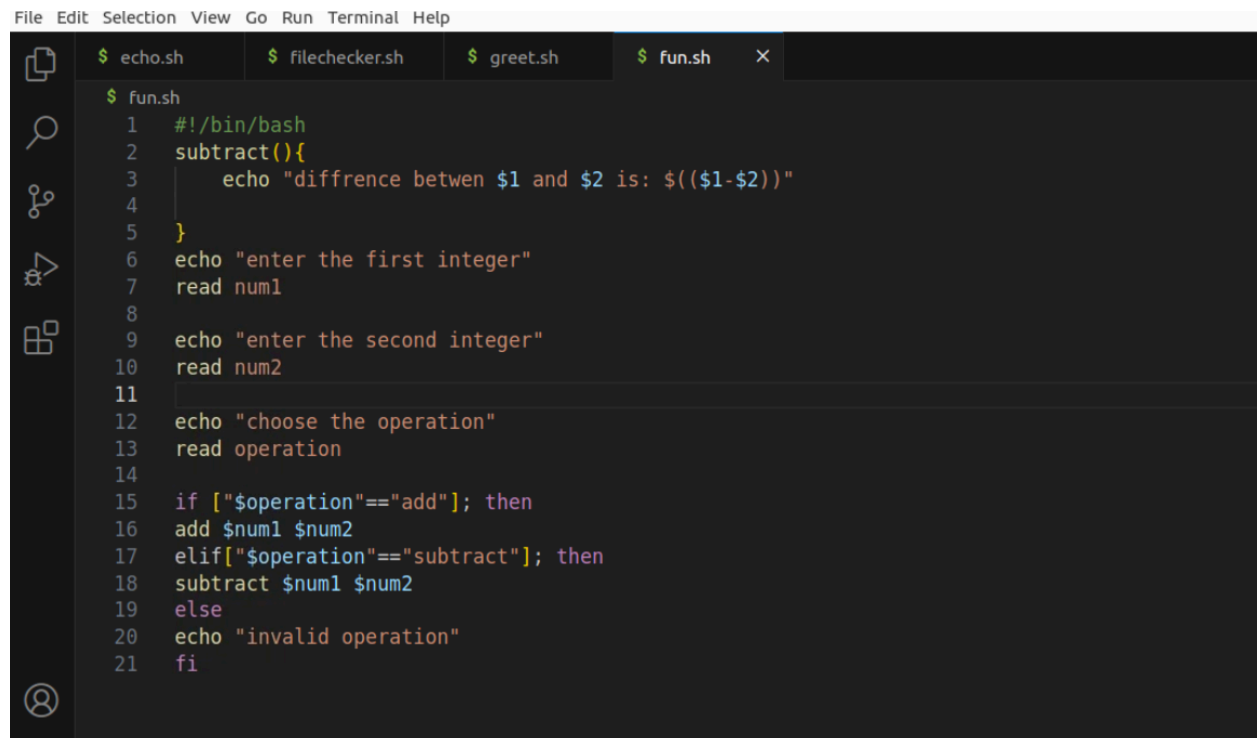
If it is between 6PM (18:00) and 5AM (5:00):
Good night!



The screenshot shows the Visual Studio Code interface with the file 'greet.sh' open. The Explorer sidebar on the left shows a folder named 'SUNNY' containing three files: 'echo.sh', 'filechecker.sh', and 'greet.sh'. The main editor area displays the content of 'greet.sh', which is a bash script that prints a greeting based on the current time of day. The script uses a series of if, elif, and fi statements to check the time and output 'good morning', 'good evening', or 'good night'.

```
greet.sh
1  #!/bin/bash
2  time=$(date +%H)
3  if [ $time -gt 5 ] && [ $time -lt 12 ]
4  then echo "good morning"
5  elif [ $time -gt 12 ] && [ $time -lt 18 ]
6  then
7  echo "good evening"
8  elif [ $time -gt 18 ] && [ $time -lt 5 ]
9  then
10 echo "good night"
11 fi
```

FUNCTION OF ADD AND SUBTRACT IN SHELL SCRIPT

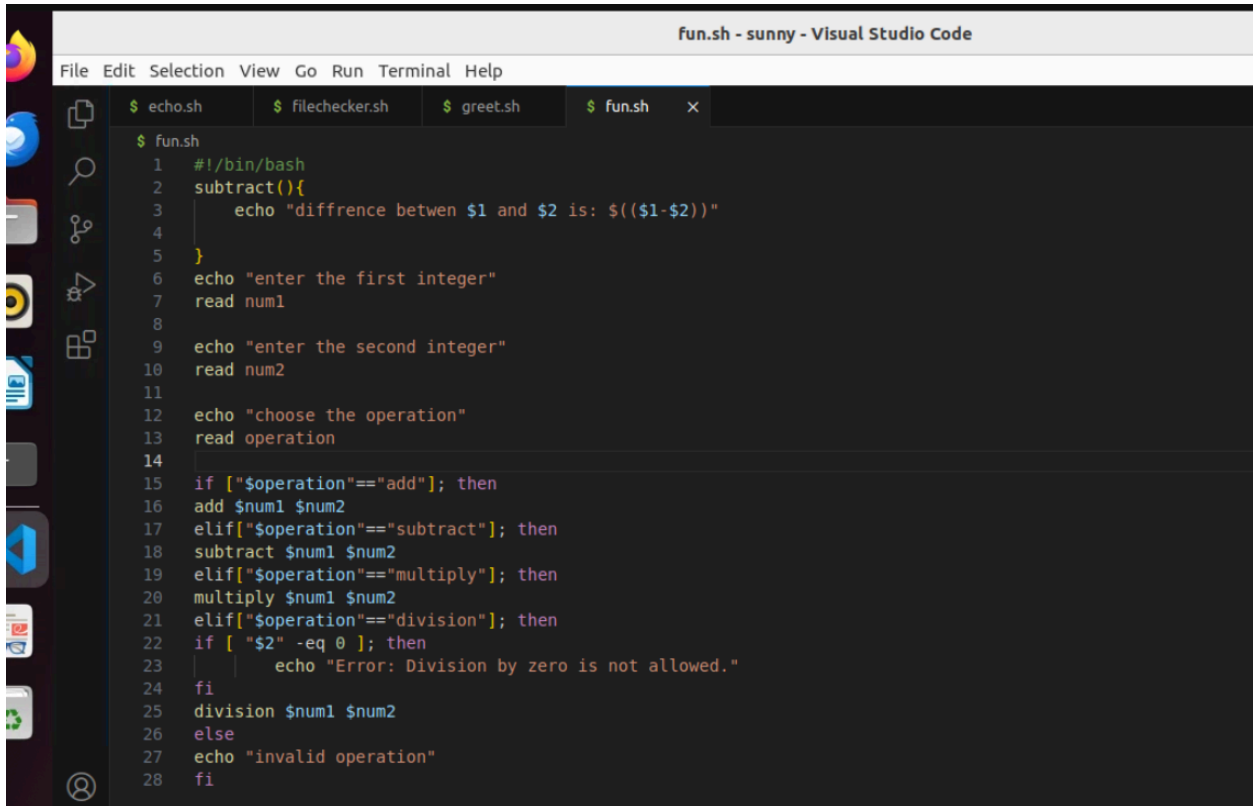


The screenshot shows the Visual Studio Code interface with the file 'fun.sh' open. The Explorer sidebar on the left shows a folder named 'SUNNY' containing four files: 'echo.sh', 'filechecker.sh', 'greet.sh', and 'fun.sh'. The main editor area displays the content of 'fun.sh', which is a bash script that performs addition and subtraction on two integers entered by the user. The script uses a function 'subtract()' and conditional statements to handle the 'add' and 'subtract' operations. It also includes an 'else' clause for invalid operations.

```
$ fun.sh
1  #!/bin/bash
2  subtract(){
3      echo "difference between $1 and $2 is: $((($1-$2))"
4  }
5
6  echo "enter the first integer"
7  read num1
8
9  echo "enter the second integer"
10 read num2
11
12 echo "choose the operation"
13 read operation
14
15 if [ "$operation"=="add" ]; then
16 add $num1 $num2
17 elif [ "$operation"=="subtract" ]; then
18 subtract $num1 $num2
19 else
20 echo "invalid operation"
21 fi
```

Q

Q. script to include add subtract multiplication and division operations, and handle division by zero appropriately. use if else instead of case

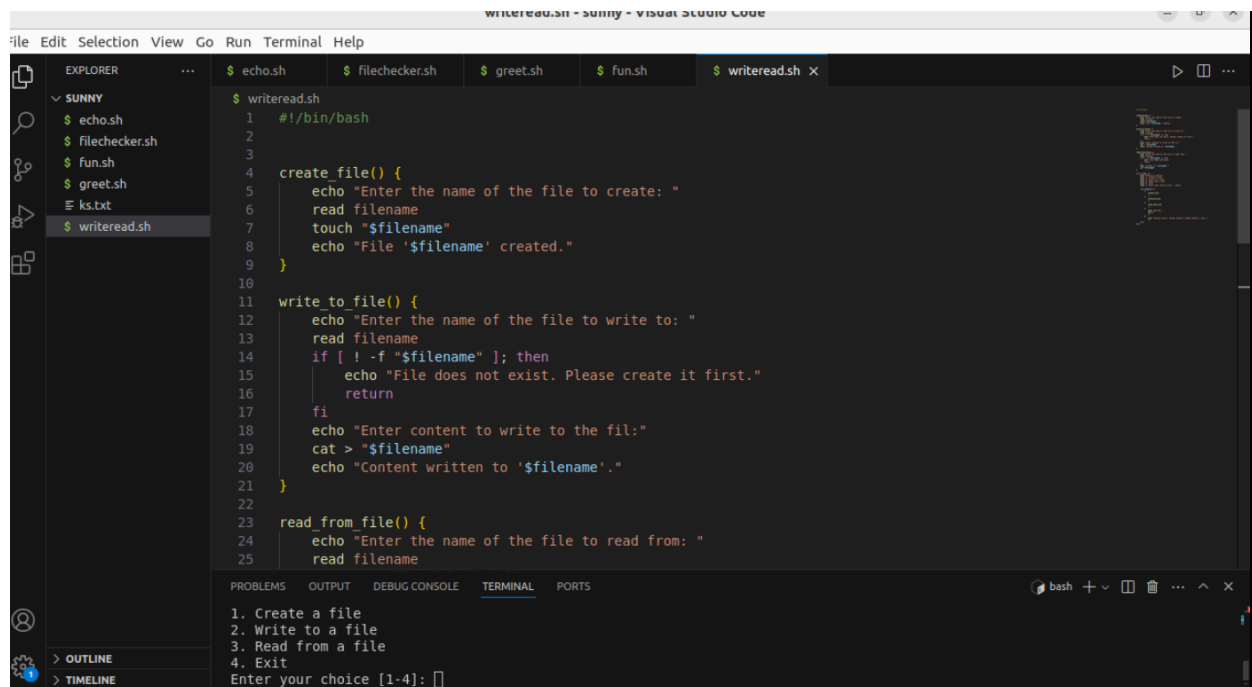


```
fun.sh - sunny - Visual Studio Code
File Edit Selection View Go Run Terminal Help
$ echo.sh $ filecheckers.sh $ greet.sh $ fun.sh x
$ fun.sh
1  #!/bin/bash
2  subtract(){
3      echo "difference between $1 and $2 is: $((($1-$2)))"
4  }
5
6  echo "enter the first integer"
7  read num1
8
9  echo "enter the second integer"
10 read num2
11
12 echo "choose the operation"
13 read operation
14
15 if ["$operation"=="add"]; then
16     add $num1 $num2
17 elif["$operation"=="subtract"]; then
18     subtract $num1 $num2
19 elif["$operation"=="multiply"]; then
20     multiply $num1 $num2
21 elif["$operation"=="division"]; then
22     if [ "$2" -eq 0 ]; then
23         echo "Error: Division by zero is not allowed."
24     fi
25     division $num1 $num2
26 else
27     echo "invalid operation"
28 fi
```

Q.Title: Create, Write, and Read Files Using Shell Script

Objective:

Develop a shell script that can create a file, write user-provided content into the file, and then read and display the content of the file.



The screenshot shows the Visual Studio Code interface with the 'writeread.sh' script open in the editor. The script contains three functions: 'create_file()', 'write_to_file()', and 'read_from_file()'. The terminal at the bottom shows the script being executed, and the user is prompted to enter a filename and content.

```
#!/bin/bash

create_file() {
    echo "Enter the name of the file to create: "
    read filename
    touch "$filename"
    echo "File '$filename' created."
}

write_to_file() {
    echo "Enter the name of the file to write to: "
    read filename
    if [ ! -f "$filename" ]; then
        echo "File does not exist. Please create it first."
        return
    fi
    echo "Enter content to write to the file:"
    cat > "$filename"
    echo "Content written to '$filename'."
}

read_from_file() {
    echo "Enter the name of the file to read from: "
    read filename
}
```

1. Create a file
2. Write to a file
3. Read from a file
4. Exit
Enter your choice [1-4]:

```
[1]+ Stopped /home/rps/sunny/writeread.sh
rps@rps-virtual-machine:~/sunny$ ./writeread.sh
Enter the filename:
sun.txt
Enter the content for the file:
hi my name is sunny kumar sharma and currently i am learning linux system programming
Display the file:
hi my name is sunny kumar sharma and currently i am learning linux system programming
rps@rps-virtual-machine:~/sunny$
```