**2 - Redirecting Errors to a Log File:**

Task: Create a shell script that generates errors, such as division by zero or file not found. Use error redirection to capture these errors in a log file. Document the error redirection process and the contents of the log file.

# Step 1: Create the Shell Script (error\_script.sh)

Begin by creating a new shell script named with the intention of

**error\_script.sh**

deliberately generating errors. This script will be used to simulate common scenarios that result in errors during execution. Open your preferred text editor, and carefully

input the following content:

#!/bin/bash

# Shell script to generate errors

# Division by zero

echo "Error 1: Attempting division by zero" result=$(echo "scale=2; 1/0" | bc)

# Accessing a non-existent file

echo "Error 2: Attempting to access a non-existent file" cat non\_existent\_file.txt

Save the script in a directory of your choosing. The script includes two intentional errors: an attempt to perform division by zero and an attempt to access a non-

existent file ( ).

**non\_existent\_file.txt**

# Step 2: Execute the Script

Navigate to the terminal and change your working directory to the location where is saved. Execute the script by using the following command:

**error\_script.sh**

bash error\_script.sh

Expected Output:

Error 1: Attempting division by zero

Error 2: Attempting to access a non-existent file cat: non\_existent\_file.txt: No such file or directory

Observe the output in the terminal, which will showcase error messages related to the intentional division by zero and the attempt to access a non-existent file. These errors are deliberately introduced for the purpose of demonstrating the error redirection process in subsequent steps.

# Step 3: Modify the Script for Error Redirection

In this step, we enhance the script ( ) to capture and redirect error

**error\_script.sh**

messages to a log file. This is accomplished by using error redirection to send standard error (stderr) to a designated log file. Open the script in a text editor and make the following modifications:

#!/bin/bash

# Shell script to generate errors with error redirection

# Log file for errors

error\_log="error\_log.txt"

# Division by zero

echo "Error 1: Attempting division by zero" result=$(echo "scale=2; 1/0" | bc) 2>> "$error\_log"

# Accessing a non-existent file

echo "Error 2: Attempting to access a non-existent file" cat non\_existent\_file.txt 2>> "$error\_log"

Explanation:

**error\_log="error\_log.txt"**

**error\_log.txt**

captured (

: Defines the name of the log file where errors will be

).

: Redirects the standard error (stderr) generated by the division by zero operation to the specified log file

**result=$(echo "scale=2; 1/0" | bc) 2>> "$error\_log"**

( ). The

**error\_log.txt**

operator is used for stderr redirection.

: Redirects the stderr generated by

**2>>**

**cat non\_existent\_file.txt 2>> "$error\_log"**

attempting to access a non-existent file to the same log file ( ).

**error\_log.txt**

Save the modified script.

# Step 4: Execute the Modified Script

Now that the script has been enhanced for error redirection, execute the modified script by navigating to the terminal and running:

bash error\_script.sh

Expected Output:

Error 1: Attempting division by zero

cat: non\_existent\_file.txt: No such file or directory

This will execute the script, and any errors produced during the process will be

captured and redirected to the specified log file ( ). Observe the terminal

**error\_log.txt**

output for the execution of the script, and we will later check the log file to confirm that error redirection is functioning as expected.

# Step 5: View the Error Log

After executing the modified script with error redirection, review the contents of the error log file to examine the captured error messages. Open a terminal and enter the following command:

cat error\_log.txt

Expected Output (if the script is executed multiple times):

cat: non\_existent\_file.txt: No such file or directory

This command displays the content of the error log ( ). Check the log file

**error\_log.txt**

to confirm that error messages generated during the script execution have been successfully redirected and recorded. The log file should provide insights into the nature of the errors, aiding in debugging and error tracking.