Name: Swarnali Ghosh

Java Pre-Skilling Training Session

Assignment -3.2

Module-3

Mail-id: swarnalighosh666@gmail.com

ASSIGNMENT-1

Ensure the script checks if a specific file (e.g., myfile.txt) exists in the current directory. If it exists, print "File exists", otherwise print "File not found".

SOLUTION:

1. Creating a dummy file named 'myfile_swar.txt' and making a shell script in vi editor named 'ss10_swar.sh': -

```
rps@rps-virtual-machine:~$ touch myfile_swar.txt
rps@rps-virtual-machine:~$ vi ss10_swar.sh
```

2. Providing permissions to access the file:

```
rps@rps-virtual-machine:~$ chmod 777 ss10_swar.sh
```

3. For showing the code where I used if, else statement:

```
rps@rps-virtual-machine:~$ vi ss10_swar.sh
```

CODE

```
#!/bin/bash

file1="myfile_swar.txt"









if [ -e "$file1" ]; then
        echo "File found"
else
        echo "File not found"
fi
~
~
```

4. Running the script to check whether the file exists or not: -

```
rps@rps-virtual-machine:~$
rps@rps-virtual-machine:~$ ./ss10_swar.sh
File found
rps@rps-virtual-machine:~$
```

ASSIGNMENT-2

Write a script that reads numbers from the user until they enter '0'. The script should also print whether each number is odd or even.

SOLUTION:

1. Creating a shell script in vi editor named ‘ss2.sh’: -

```
rps@rps-virtual-machine:~$ vi ss2.sh
```

2. Providing privilege / permission to access the file:

```
rps@rps-virtual-machine:~$ chmod 777 ss2.sh
```

3. Running the script to check whether the number is even or odd and to press '0' to quit:

```
rps@rps-virtual-machine:~$
rps@rps-virtual-machine:~$ ./ss2.sh
Enter a number(0 to quit it) :4
4 is EVEN
Enter a number(0 to quit it) :7
7 is ODD
Enter a number(0 to quit it) :0
rps@rps-virtual-machine:~$
```

Let me show the code which I have written ------

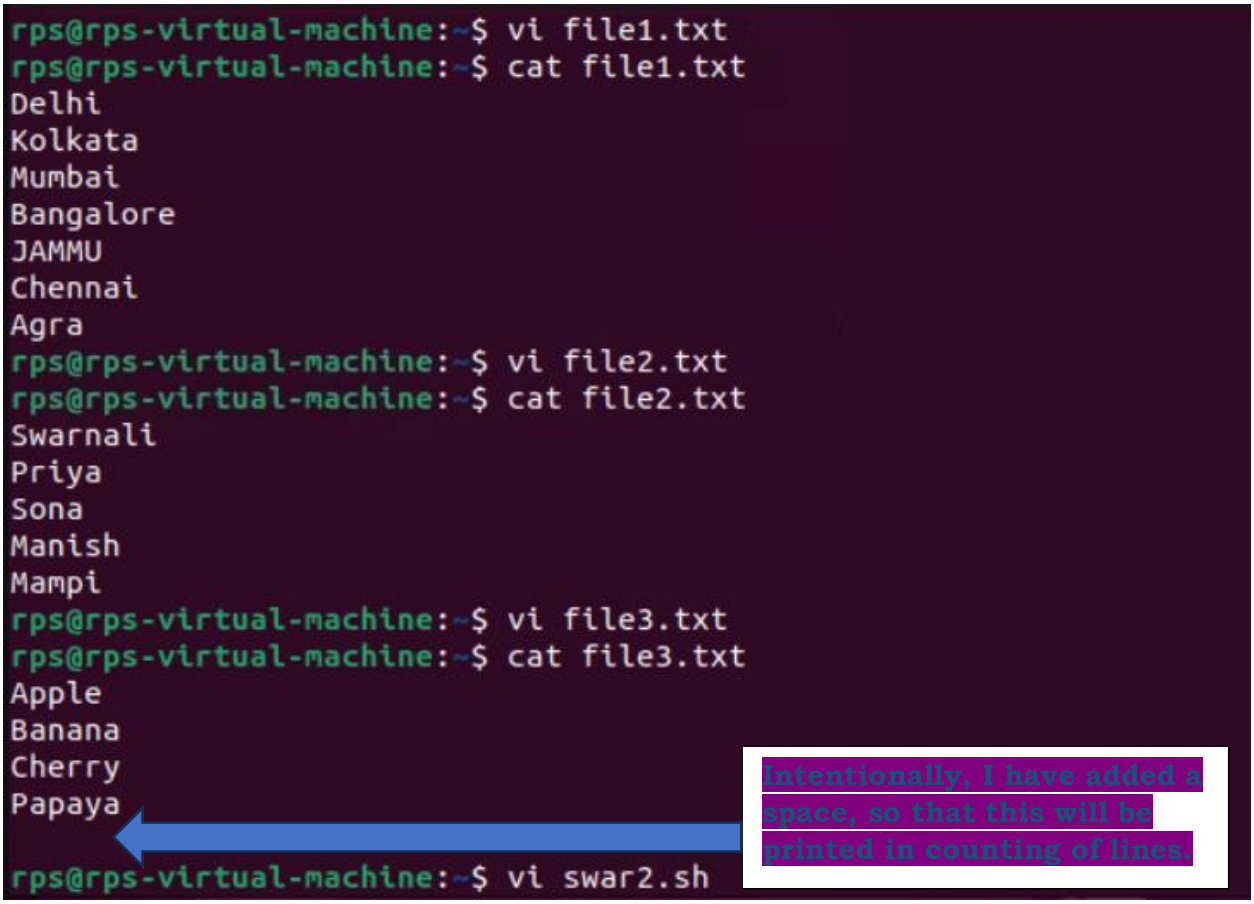
```
#!/bin/bash


while true;
do
   read -p "Enter a number(0 to quit it) :" num

   if [ "$num" -eq 0 ]; then
        break
    fi

   if [ $((num % 2)) -eq 0 ]; then
        echo "$num is EVEN"
   else
        echo "$num is ODD"
   fi

done
```

ASSIGNMENT-3

Create a function that takes a filename as an argument and prints the number of lines in the file. Call this function from your script with different filenames.

SOLUTION:

1. Firstly, I created 3files----file1.txt, file2.txt, file3.txt using vi editor & using cat we'll check the contents I inserted in the respective files: -



Thus, we could see, file1.txt---→ 7lines are present
file2.txt---→ 5lines are present
file3.txt---→ 5lines are present (including space)

2. Making a shell script in vi editor named ‘swar2.sh’:

```
rps@rps-virtual-machine:~$ vi swar2.sh
```

CODE: ----------

```
#!/bin/bash

count_lines(){

local file_name="$1"

if [[ -f "$file_name" ]]; then
        local lc=$(wc -l < "$file_name")
        echo "The file '$file_name' has $lc lines."

else

        echo "The file '$file_name' does not exists."

fi
}



count_lines "file1.txt"
count_lines "file2.txt"
count_lines "file3.txt"
~
~
```

3. Providing permission to access the file:

```
rps@rps-virtual-machine:~$ chmod 777 swar2.sh
rps@rps-virtual-machine:~$
```

4. Now, finally we need to see the output…. Counting of the lines- Exactly matches with what we have expected-→

```
rps@rps-virtual-machine:~$
rps@rps-virtual-machine:~$ ./swar2.sh
The file 'file1.txt' has 7 lines.
The file 'file2.txt' has 5 lines.
The file 'file3.txt' has 5 lines.
```

ASSIGNMENT-4

Write a script that creates a directory named TestDir and inside it, creates ten files named File1.txt, File2.txt, ... File10.txt. Each file should contain its filename as its content (e.g., File1.txt contains "File1.txt").

SOLUTION:

1. Making the shell script in vi editor named 'swarDir.sh':

```
rps@rps-virtual-machine:~$ vi swarDir.sh
```

2. Printing the file content on terminal:

```
#!/bin/bash

makeFile(){

cd TestDir/

        for((i=1;i<=10;i++))

        do
                filename="file$i.txt"
                echo "$filename" > "$filename"
        done
}

if [ -d "TestDir" ]; then
        exit 1
else
        mkdir TestDir
        makeFile
fi
~
~
```

3. Changing privileges to access the file:

```
rps@rps-virtual-machine:~$
rps@rps-virtual-machine:~$ chmod 777 swarDir.sh
```

4. Running the script to make “TestDir” directory and 10files with content in them & checking through ‘ls’ whether the directory is created or not:

```
rps@rps-virtual-machine:~$ ./swarDir.sh
rps@rps-virtual-machine:~$
rps@rps-virtual-machine:~$ ls
Desktop           myfile.txt                                  ss2.sh
Documents         node_exporter-0.18.1.linux-amd64.tar.gz     ss3.sh
Downloads         oracle-database-xe-21c-1.0-1.ol8.x86_64.rpm ss4.sh
f1.txt            Pictures                                    swar2.sh
file2.txt         Public                                      swarDir.sh
file3.txt         snap                                        swarnali_s1.sh
hello             ss10_swar.sh                                Templates
Music             ss1.h                                       TestDir
myfile_swar.txt   ss1.sh                                      Videos
rps@rps-virtual-machine:~$
```

5. Changing the directory to ‘TestDir’:

```
rps@rps-virtual-machine:~$
rps@rps-virtual-machine:~$ cd TestDir/
```

6. Checking whether all the files have been created or not:

```
rps@rps-virtual-machine:~/TestDir$ ls -l
total 40
-rw-rw-r-- 1 rps rps 11 May 19 11:48 file10.txt
-rw-rw-r-- 1 rps rps 10 May 19 11:48 file1.txt
-rw-rw-r-- 1 rps rps 10 May 19 11:48 file2.txt
-rw-rw-r-- 1 rps rps 10 May 19 11:48 file3.txt
-rw-rw-r-- 1 rps rps 10 May 19 11:48 file4.txt
-rw-rw-r-- 1 rps rps 10 May 19 11:48 file5.txt
-rw-rw-r-- 1 rps rps 10 May 19 11:48 file6.txt
-rw-rw-r-- 1 rps rps 10 May 19 11:48 file7.txt
-rw-rw-r-- 1 rps rps 10 May 19 11:48 file8.txt
-rw-rw-r-- 1 rps rps 10 May 19 11:48 file9.txt
rps@rps-virtual-machine:~/TestDir$
```

7. According to the question, each file should contain its filename as its content, so now checking this through ‘cat’: -

8. Yes, it is printing fine!!!

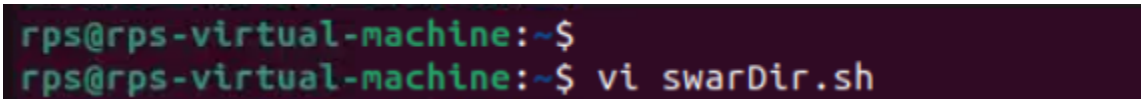
ASSIGNMENT-5

Modify the script to handle errors, such as the directory already existing or lacking permissions to create files.

Add a debugging mode that prints additional information when enabled.

SOLUTION:

1. Editing existing shell script named ‘swarDir.sh’ using vi editor:



2. Here is the code, Printing the file content on terminal:

```
#!/bin/bash




set -x
makeFile(){



mkdir TestDir
cd TestDir/

        for((i=1;i<=10;i++))

        do
                filename="file$i.txt"
                echo "$filename" > "$filename"
        done
}


handleError(){
        echo "Directory already exists"
        exit 1
}

trap 'handleError' ERR

makeFile
```

3. Removing ‘TestDir’ that was present due to earlier script:

```
rps@rps-virtual-machine:~$
rps@rps-virtual-machine:~$ rm -r TestDir/
```

9. Running the script to make “TestDir” directory and 10files with content in them:

```
rps@rps-virtual-machine:~$ ./swarDir.sh
++ trap handleError ERR
++ makeFile
++ mkdir TestDir
++ cd TestDir/
++ (( i=1 ))
++ (( i<=10 ))
++ filename=file1.txt
++ echo file1.txt
++ (( i++ ))
++ (( i<=10 ))
++ filename=file2.txt
++ echo file2.txt
++ (( i++ ))
++ (( i<=10 ))
++ filename=file3.txt
++ echo file3.txt
++ (( i++ ))
++ (( i<=10 ))
++ filename=file4.txt
++ echo file4.txt
++ (( i++ ))
++ (( i<=10 ))
++ filename=file5.txt
++ echo file5.txt
++ (( i++ ))
++ (( i<=10 ))
++ filename=file6.txt
++ echo file6.txt
++ (( i++ ))
++ (( i<=10 ))
++ filename=file7.txt
++ echo file7.txt
++ (( i++ ))
++ (( i<=10 ))
++ filename=file8.txt
++ echo file8.txt
++ (( i++ ))
++ (( i<=10 ))
++ filename=file9.txt
++ echo file9.txt
++ (( i++ ))
++ (( i<=10 ))
++ filename=file10.txt
++ echo file10.txt
++ (( i++ ))
++ (( i<=10 ))
+++ handleError
+++ echo 'Directory already exists'
Directory already exists
+++ exit 1
rps@rps-virtual-machine:~$
```

10.Changing the directory:

```
rps@rps-virtual-machine:~$ cd TestDir/
rps@rps-virtual-machine:~/TestDir$
rps@rps-virtual-machine:~/TestDir$
```

11.Checking whether all the files have been created or not: -

```
rps@rps-virtual-machine:~/TestDir$
rps@rps-virtual-machine:~/TestDir$ ls -l
total 40
-rw-rw-r-- 1 rps rps 11 May 19 13:19 file10.txt
-rw-rw-r-- 1 rps rps 10 May 19 13:19 file1.txt
-rw-rw-r-- 1 rps rps 10 May 19 13:19 file2.txt
-rw-rw-r-- 1 rps rps 10 May 19 13:19 file3.txt
-rw-rw-r-- 1 rps rps 10 May 19 13:19 file4.txt
-rw-rw-r-- 1 rps rps 10 May 19 13:19 file5.txt
-rw-rw-r-- 1 rps rps 10 May 19 13:19 file6.txt
-rw-rw-r-- 1 rps rps 10 May 19 13:19 file7.txt
-rw-rw-r-- 1 rps rps 10 May 19 13:19 file8.txt
-rw-rw-r-- 1 rps rps 10 May 19 13:19 file9.txt
rps@rps-virtual-machine:~/TestDir$
rps@rps-virtual-machine:~/TestDir$
```

ASSIGNMENT-6

Given a sample log file, write a script using grep to extract all lines containing "ERROR". Use awk to print the date, time, and error message of each extracted line.

Data Processing with sed

SOLUTION:

- Logging **Functions:**
  - `log_info`, `log_error`, and `log_warn` functions log messages with a timestamp and appropriate log level to `logfile.log`.
- Logging **Examples:**
  - We call the logging functions to create some log entries.
- Extracting **"ERROR" Lines:**
  - `grep "ERROR" $LOGFILE`: Extracts lines from `logfile.log` that contain the word "ERROR".
- Processing **with `awk`:**
  - `awk '{print $1, $2, substr($0, index($0,$4))}'`: Processes each extracted line to print:
    - `$1`: The date.
    - `$2`: The time.
    - `substr($0, index($0,$4))`: The substring starting from the fourth field, which is the error message.
- Optional **`sed` Processing:**
  - `sed 's/error/ERROR/g'`: Modifies the error messages by replacing "error" with "ERROR" (case-insensitive).

```bash
#!/bin/bash

LOGFILE="logfile.log"

log_info() {
    echo "$(date '+%Y-%m-%d %H:%M:%S') INFO $1" >> $LOGFILE
}

log_error() {
    echo "$(date '+%Y-%m-%d %H:%M:%S') ERROR $1" >> $LOGFILE
}

log_warn() {
    echo "$(date '+%Y-%m-%d %H:%M:%S') WARN $1" >> $LOGFILE
}

log_info "This is an informational message"
log_error "An error occurred"
log_warn "This is a warning message"
log_error "Another error message"


grep "ERROR" $LOGFILE | awk '{print $1, $2, substr($0, index($0,$4))}' | sed 's/error/ERROR/g'
```

1. Here, we could see ‘logfile.log’ is created: -

main.bash | logfile.log

```
1  2024-05-19 16:05:49 INFO This is an informational message
2  2024-05-19 16:05:49 ERROR An error occurred
3  2024-05-19 16:05:49 WARN This is a warning message
4  2024-05-19 16:05:49 ERROR Another error message
5
```

2. After running the script: -

```
2024-05-19 16:05:49 An ERROR occurred
2024-05-19 16:05:49 Another ERROR message


...Program finished with exit code 0
Press ENTER to exit console.
```

ASSIGNMENT-7

Create a script that takes a text file and replaces all occurrences of "old_text" with "new_text". Use sed to perform this operation and output the result to a new file.

SOLUTION:

1. Making the shell script in vi editor named 'swar5.sh':

```
rps@rps-virtual-machine:~$ vi swar5.sh
rps@rps-virtual-machine:~$
```

2. Writing the code in script:

- The script first checks if the correct number of arguments (3) is provided. If not, it prints a usage message and exits.
- The input file name, the old text to be replaced, and the new text are assigned to variables.
- Check Input File: It checks if the input file exists. If not, it prints an error message and exits.
- Perform Replacement: The sed command is used to perform the replacement.
  The s/$old_text/$new_text/g syntax tells sed to substitute old_text with new_text globally (i.e., replace all occurrences).
  The result is redirected to a new output file.
- Check Success: The script checks if the 'sed' command was successful and prints an appropriate message.

```
#!/bin/bash

if [ "$#" -ne 3 ]; then
        echo "Usage: $0 input_file old_text new_text"
        exit 1
fi

input_file=$1
old_text=$2
new_text=$3
output_file="output_$(basename "$input_file")"


if [ ! -f "$input_file" ];then
        echo "Error: Input file '$input_file' not found."
        exit 1
fi

sed "s/$old_text/$new_text/g" "$input_file" > "$output_file"

if [ $? -eq 0 ]; then
        echo "Replacement successful. Output saved to '$output_file'."
else
        echo "Error occurred"
        exit 1
fi
~
~
~
```

3. Assigning permission to access the file:

```
rps@rps-virtual-machine:~$
rps@rps-virtual-machine:~$ chmod 777 swar5.sh
```

4. Checking output: (as I was not having input.txt file thus printing not found)

```
rps@rps-virtual-machine:~$ ./swar5.sh input.txt old_text new_text
Error: Input file 'input.txt' not found.
rps@rps-virtual-machine:~$
```

5. Creating a file named input.txt and here, it is showing Replacement successful.

```
rps@rps-virtual-machine:~$
rps@rps-virtual-machine:~$ touch input.txt
rps@rps-virtual-machine:~$
rps@rps-virtual-machine:~$ ./swar5.sh input.txt old_text new_text
Replacement successful. Output saved to 'output_input.txt'.
rps@rps-virtual-machine:~$
rps@rps-virtual-machine:~$
```

Thus, replaces all occurrences of "old_text" with "new_text". And showing the result to a new file.