

Digvijay Thakare

Day_7 Assignment

Que 1-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".

1. Solution-First, I opened the terminal in WSL.
2. Then, I created a new file called check_file.sh using vi by typing:

vi check_file.sh

3.Once in vi, I pressed i to enter insert mode.

4.I typed the following script to check if myfile.txt exists:

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

```
# Define the filename
```

```
FILE="My_file.txt"
```

```
# Check if the file exists
```

```
if [ -e "$FILE" ]; then
```

```
    echo "File exists"
```

```
else
```

```
    echo "File not found"
```

```
fi
```

5.After typing the script, I pressed Esc to exit insert mode.

6.To save the file and quit vi, I typed :wq and pressed Enter.

7.Next, I made the script executable by running:

```
chmod u+x check_file.sh
```

8.Finally, I ran the script with:

```
./check_file.sh
```

Que2-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

```
#!/bin/bash

# Initialize a variable to store the user input
number=1

# Loop until the user enters '0'
while [ "$number" -ne 0 ]; do

    # Prompt the user to enter a number
    echo -n "Enter a number (0 to quit): "

    read number

    # Check if the number is not 0
    if [ "$number" -ne 0 ]; then

        # Check if the number is even
        if [ $((number % 2)) -eq 0 ]; then
            echo "$number is even"
        else
            echo "$number is odd"
        fi
    fi

done

echo "Goodbye!"
```

Output-

digu@Digvijay:/mnt/c/Users/Digvija/Desktop/WiproFullstack/LinuxPractice\$ vi check_odd_even.sh

```
digu@Digvijay:/mnt/c/Users/Digvija/Desktop/WiproFullstack/LinuxPractice$  
chmod +x check_odd_even.sh
```

```
digu@Digvijay:/mnt/c/Users/Digvija/Desktop/WiproFullstack/LinuxPractice$  
./check_odd_even.sh
```

```
Enter a number (0 to quit): 56
```

```
56 is even
```

```
Enter a number (0 to quit): 75
```

```
75 is odd
```

```
Enter a number (0 to quit): 0
```

```
Goodbye!
```

Que 3 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

```
#!/bin/bash  
  
# Function to count the number of lines in a file  
  
count_lines() {  
    local FILENAME=$1  
    # Check if the file exists  
    if [ -f "$FILENAME" ]; then  
        local LINES=$(wc -l < "$FILENAME")  
        echo "The file '$FILENAME' has $LINES lines."  
    else  
        echo "The file '$FILENAME' does not exist."  
    fi  
}
```

Call the function with different filenames

count_lines "file1.txt"

count_lines "file2.txt"

count_lines "file3.txt"

Output-

digu@Digvijay:/mnt/c/Users/Digvija/Desktop/WiproFullstack/LinuxPractice\$
./count_lines.sh

The file 'file1.txt' has 1 lines.

The file 'file2.txt' has 2 lines.

The file 'file3.txt' has 3 lines.

Que4 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").

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

```
# Create a directory named TestDir
```

```
mkdir -p TestDir
```

```
# Change to the TestDir directory
```

```
cd TestDir
```

```
# Loop to create 10 files
```

```
for i in {1..10}
```

```
do
```

```
    # Create the filename
```

```
FILENAME="File${i}.txt"
# Create the file and write its name as the content
echo "$FILENAME" > "$FILENAME"
done
echo "Files created in TestDir:"
ls
```

Output-

```
digu@Digvijay:/mnt/c/Users/Digvija/Desktop/WiproFullstack/LinuxPractic
e$ ./create_files.sh
```

Files created in TestDir:

File1.txt File2.txt File4.txt File6.txt File8.txt

File10.txt File3.txt File5.txt File7.txt File9.txt

Que 5 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-

```
Solution-#!/bin/bash
```

```
# Enable debugging mode if the DEBUG variable is set to 1
```

```
DEBUG=1
```

```
debug() {
```

```
    if [ "$DEBUG" -eq 1 ]; then
```

```
        echo "DEBUG: $1"
```

```
    fi
```

```
}
```

```
# Create a directory named TestDir
```

```
DIR_NAME="TestDir"
```

```
debug "Attempting to create directory $DIR_NAME"
```

```
if mkdir -p "$DIR_NAME"; then
```

```
    debug "Directory $DIR_NAME created successfully or already exists."
```

```
else
```

```
    echo "Error: Failed to create directory $DIR_NAME"
```

```
    exit 1
```

```
fi
```

```
# Change to the TestDir directory
```

```
if cd "$DIR_NAME"; then
```

```
    debug "Changed to directory $DIR_NAME"
else
    echo "Error: Failed to change to directory $DIR_NAME"
    exit 1
fi

# Loop to create 10 files
for i in {1..10}
do
    # Create the filename
    FILENAME="File${i}.txt"
    debug "Creating file $FILENAME"

    # Create the file and write its name as the content
    if echo "$FILENAME" > "$FILENAME"; then
        debug "File $FILENAME created and written successfully."
    else
        echo "Error: Failed to create or write to file $FILENAME"
        exit 1
    fi
done

echo "Files created in $DIR_NAME:"
ls

debug "Script completed successfully."
```

Output-

```
digu@Digvijay:/mnt/c/Users/Digvija/Desktop/WiproFullstack/LinuxPractice$ vi  
create_files_with_debug.sh
```

```
digu@Digvijay:/mnt/c/Users/Digvija/Desktop/WiproFullstack/LinuxPractice$  
chmod +x create_files_with_debug.sh
```

```
digu@Digvijay:/mnt/c/Users/Digvija/Desktop/WiproFullstack/LinuxPractice$  
./create_files_with_debug.sh
```

DEBUG: Attempting to create directory TestDir

DEBUG: Directory TestDir created successfully or already exists.

DEBUG: Changed to directory TestDir

DEBUG: Creating file File1.txt

DEBUG: File File1.txt created and written successfully.

DEBUG: Creating file File2.txt

DEBUG: File File2.txt created and written successfully.

DEBUG: Creating file File3.txt

DEBUG: File File3.txt created and written successfully.

DEBUG: Creating file File4.txt

DEBUG: File File4.txt created and written successfully.

DEBUG: Creating file File5.txt

DEBUG: File File5.txt created and written successfully.

DEBUG: Creating file File6.txt

DEBUG: File File6.txt created and written successfully.

DEBUG: Creating file File7.txt

DEBUG: File File7.txt created and written successfully.

DEBUG: Creating file File8.txt

DEBUG: File File8.txt created and written successfully.

DEBUG: Creating file File9.txt

DEBUG: File File9.txt created and written successfully.

DEBUG: Creating file File10.txt

DEBUG: File File10.txt created and written successfully.

Files created in TestDir:

File1.txt File10.txt File2.txt File3.txt File4.txt File5.txt File6.txt File7.txt
File8.txt File9.txt

DEBUG: Script completed successfully.

Que 6 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.

Solution-

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

```
LOGFILE="sample.log"
```

```
if [ ! -f "$LOGFILE" ]; then
```

```
    echo "Log file $LOGFILE does not exist."
```

```
    exit 1
```

```
fi
```

```
# Extract lines containing "ERROR" and process them
```

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

Output-

```
digu@Digvijay:/mnt/c/Users/Digvija/Desktop/WiproFullstack/LinuxPractic  
e$ vi process_log.sh
```

```
digu@Digvijay:/mnt/c/Users/Digvija/Desktop/WiproFullstack/LinuxPractic  
e$ chmod +x process_log.sh
```

digu@Digvijay:/mnt/c/Users/Digvija/Desktop/WiproFullstack/LinuxPractice\$./process_log.sh

2024-05-18 10:24:01 Unable to connect to database

2024-05-18 10:25:30 Failed to load configuration file

Que 7

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-

```
vi replace_text.sh
```

```
#!/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.txt"
```

```
if [ ! -f "$INPUT_FILE" ]; then
```

```
    echo "Error: Input file '$INPUT_FILE' does not exist."
```

```
    exit 1
```

```
fi
```

```
# Use sed to replace all occurrences of old_text with new_text
```

```
sed "s/$OLD_TEXT/$NEW_TEXT/g" "$INPUT_FILE" > "$OUTPUT_FILE"
```

echo "All occurrences of '\$OLD_TEXT' have been replaced with '\$NEW_TEXT' in '\$OUTPUT_FILE'."

Output-

```
digu@Digvijay:/mnt/c/Users/Digvija/Desktop/WiproFullstack/LinuxPractice$  
cat sample.txt
```

This is the old_text that will be replaced.

We need to replace old_text with new_text.

old_text appears multiple times in this file.

```
digu@Digvijay:/mnt/c/Users/Digvija/Desktop/WiproFullstack/LinuxPractice$ vi  
replace_text.sh
```

```
digu@Digvijay:/mnt/c/Users/Digvija/Desktop/WiproFullstack/LinuxPractice$  
./replace_text.sh sample.txt old_text new_text
```

All occurrences of 'old_text' have been replaced with 'new_text' in 'output.txt'.

```
digu@Digvijay:/mnt/c/Users/Digvija/Desktop/WiproFullstack/LinuxPractice$  
cat output.txt
```

This is the new_text that will be replaced.

We need to replace new_text with new_text.

new_text appears multiple times in this file