

Shell Scripting with Bash

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".

```
#!/bin/bash
file_name="myfile.txt"
if [ -f "$file_name" ]; then
    echo "File exists"
else
    echo "File not found"
fi
```

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.

```
#!/bin/bash

while true; do
    echo -n "Enter a number (0 to exit): "
    read number
```

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

if [ "$((number % 2))" -eq 0 ]; then
    echo "$number is even"
else
    echo "$number 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.

```
#!/bin/bash

print_lines_count() {
    local filename="$1"
    local lines_count=$(wc -l < "$filename")
    echo "Number of lines in $filename: $lines_count"
}

print_lines_count "file1.txt"
print_lines_count "file2.txt"
print_lines_count "file3.txt"
```

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

mkdir -p TestDir

# Loop to create ten files
for ((i=1; i<=10; i++)); do
    file_name="TestDir/File$i.txt"
    echo "Creating $file_name"
    echo "$file_name" > "$file_name"
done

echo "Files created successfully."
```

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.

```
#!/bin/bash

print_debug() {
    if [ "$debug" = true ]; then
```

```
        echo "Debug: $1"
    fi
}
```

```
create_files() {
    # Loop to create ten files
    for ((i=1; i<=10; i++)); do
        file_name="TestDir/File$i.txt"
        print_debug "Creating $file_name"
        echo "$file_name" > "$file_name"
        if [ $? -ne 0 ]; then
            echo "Error: Unable to create file $file_name"
            exit 1
        fi
    done
}
```

```
if [ "$1" = "debug" ]; then
    debug=true
    echo "Debug mode enabled"
else
    debug=false
fi
```

```
if [ -d "TestDir" ]; then
```

```
    echo "Error: Directory 'TestDir' already exists"
    exit 1
fi

mkdir -p TestDir
if [ $? -ne 0 ]; then
    echo "Error: Unable to create directory 'TestDir'"
    exit 1
fi

chmod 700 TestDir
if [ $? -ne 0 ]; then
    echo "Error: Unable to change permissions of directory 'TestDir'"
    exit 1
fi

create_files

echo "Files created successfully in directory '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.

```
grep "ERROR" log_file.txt | awk '{print $1, $2, $NF}'
```

2024-05-30 14:30:00 ERROR This is an error message

2024-05-30 14:31:00 INFO This is an info message

2024-05-30 14:32:00 ERROR Another error message

The script would output:

2024-05-30 14:30:00 This is an error message

2024-05-30 14:32:00 Another error message

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.

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

Set the input file, old text, new text, and output file

```
INPUT_FILE=$1
```

```
OLD_TEXT=$2
```

```
NEW_TEXT=$3
```

```
OUTPUT_FILE=$4
```

Use sed to replace old_text with new_text and output to a new file

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

```
./replace_text.sh input.txt old_text new_text output.txt
```

This is some old_text that needs to be replaced.

old_text is not good.

This is some new_text that needs to be replaced.

new_text is not good.