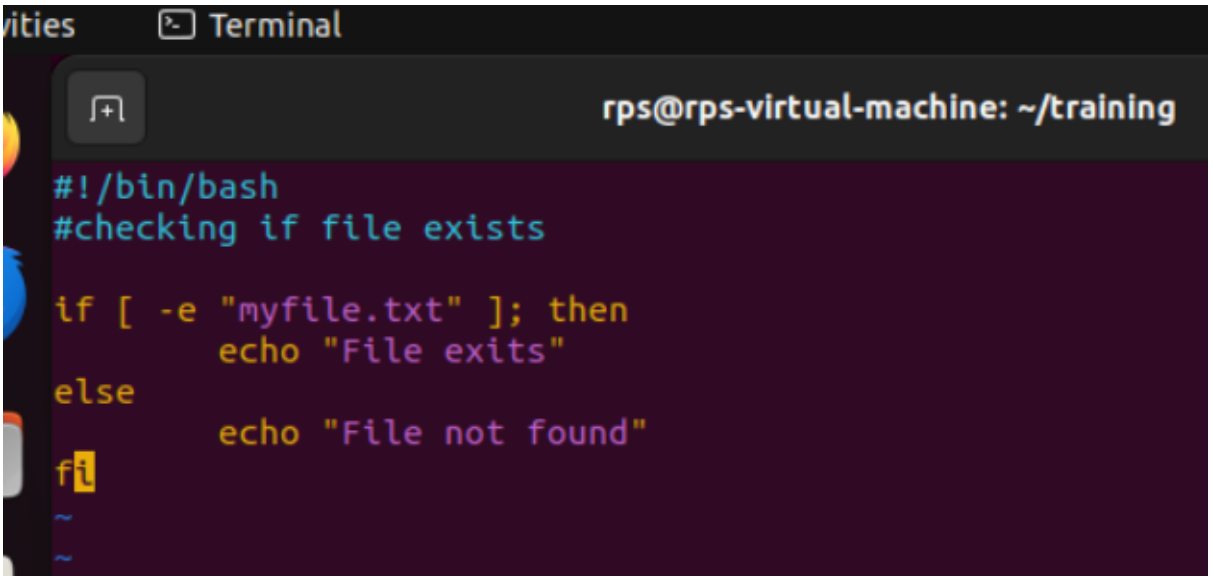
ASSIGNMENT 1

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".

check_file.sh shell script code

```
#!/bin/bash

# Check if the file exists
if [ -e "myfile.txt" ]; then
    echo "File exists"
else
    echo "File not found"
fi
```

Script file in Vim editor



Output showing script is working

```
rps@rps-virtual-machine:~/training$ ls
intro.txt
rps@rps-virtual-machine:~/training$ vim check_file.sh
rps@rps-virtual-machine:~/training$ chmod +x check_file.sh
rps@rps-virtual-machine:~/training$ ./check_file.sh
File not found
rps@rps-virtual-machine:~/training$
```

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.

Odd_even.sh shell script code

```
#!/bin/bash

while true; do
        echo "Enter a number (enter 0 to stop):"
        read num

        #check if number is 0 if it is than exit loop
        if [ "$num" -eq 0 ]; then
                echo "Exiting ... "
                break
        fi

        #check if number is even or odd
        if [ $((num%2) ) -eq 0 ]; then
                echo "$num is even"
        else
                echo "$num is odd”
        fi

done
```

Script file in Vim editor

```
#!/bin/bash

while true; do
        echo "Enter a number (enter 0 to stop):"
        read num

        #check if number is 0 if it is than exit loop
        if [ "$num" -eq 0 ]; then
                echo "Exiting..."
                break
        fi

        #check if number is even or odd
        if [ $((num%2)) -eq 0 ]; then
                echo "$num is even"
        else
                echo "$num is odd"
        fi
done
~
```

Output showing script is working

```
rps@rps-virtual-machine:~/training$ vim odd_even.sh
rps@rps-virtual-machine:~/training$ chmod +x odd_even.sh
rps@rps-virtual-machine:~/training$ ./odd_even.sh
Enter a number (enter 0 to stop):
2
2 is even
Enter a number (enter 0 to stop):
9
9 is odd
Enter a number (enter 0 to stop):
0
Exiting...
rps@rps-virtual-machine:~/training$
```

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.

line_counter.sh shell script code

```
#!/bin/bash

# Function to count the number of lines in a file
count_lines() {
    local filename="$1"
    local num_lines=$(wc -l < "$filename")
    echo "Number of lines in $filename: $num_lines"
}

# Main script
while true; do
    echo "Enter a filename (enter 'exit' to quit): "
    read filename

    # Check if the user wants to exit
    if [ "$filename" == "exit" ]; then
        echo "Exiting..."
        break
    fi

    # Check if the file exists
    if [ ! -f "$filename" ]; then
        echo "File '$filename' not found."
        continue
    fi

    # Call the function to count lines in the file
    count_lines "$filename"
done
```

Script file in Vim editor

rps@rps-virtual-machine: ~/training

```
#!/bin/bash

#Function to count number of lines in a file
count_lines(){
        local filename="$1"
        local num_lines=$(cat "$filename" | wc -l)
        echo "Number of lines in $filename: $num_lines"
}

#Main script
while true; do
        echo "Enter a filename (enter 'exit' to quit)"
        read filename

        #check if user wants to exit
        if [ "$filename" == "exit" ]; then
                echo "Exiting..."
                break;
        fi

        #check if file exits
        if [ ! -f "$filename" ]; then
                echo "File '$filename' not found."
                continue
        fi

        #call the function to count lines in the file
        count_lines "$filename"
done
```

Output showing script is working

```
rps@rps-virtual-machine:~/training$ vim line_counter.sh
rps@rps-virtual-machine:~/training$ chmod +x line_counter.sh
rps@rps-virtual-machine:~/training$ ./line_counter.sh
Enter a filename (enter 'exit' to quit)
intro.txt
Number of lines in intro.txt: 2
Enter a filename (enter 'exit' to quit)
wipro.txt
File 'wipro.txt' not found.
Enter a filename (enter 'exit' to quit)
test.txt
File 'test.txt' not found.
Enter a filename (enter 'exit' to quit)
exit
Exiting...
rps@rps-virtual-machine:~/training$ ls
check_file.sh  intro.txt  line_counter.sh  odd_even.sh  welcome.txt
rps@rps-virtual-machine:~/training$ cat intro.txt
introduction to java
welcome to wipro
rps@rps-virtual-machine:~/training$ cat welcome.txt
welcome to java programming.
rps@rps-virtual-machine:~/training$
```

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").

create_directoryfiles.sh shell script code

```
#!/bin/bash

# Create a directory named TestDir
mkdir -p TestDir

# Change to the TestDir directory
cd TestDir

# Create ten files named File1.txt, File2.txt, ..., File10.txt
for ((i=1; i<=10; i++)); do
    filename="File${i}.txt"
    echo "$filename" > "$filename"
    echo "Created $filename"
done

# Go back to the previous directory
cd ..
```

Script file in Vim editor

```
rps@rps-virtual
#!/bin/bash

#create a directory named TestDir
mkdir -p TestDir

#change to the TestDir directory
cd TestDir

#create ten files named File1.txt, File2.txt, ..., File10.txt
for ((i=1; i<=10; i++)); do
        filename="File${i}.txt"
        echo "$filename" > "$filename"
        echo "Created $filename"
done

#go back to the previos directory
cd ..
```

Output showing script is working

```
rps@rps-virtual-machine: ~/training/TestDir
rps@rps-virtual-machine:~/training$ vim create_directoryfiles.sh
rps@rps-virtual-machine:~/training$ chmod +x create_directoryfiles.sh
rps@rps-virtual-machine:~/training$ ./create_directoryfiles.sh
Created File1.txt
Created File2.txt
Created File3.txt
Created File4.txt
Created File5.txt
Created File6.txt
Created File7.txt
Created File8.txt
Created File9.txt
Created File10.txt
rps@rps-virtual-machine:~/training$ ls
check_file.sh  create_directoryfiles.sh  intro.txt  line_counter.sh  odd_even.sh  TestDir  welcome.txt
rps@rps-virtual-machine:~/training$ cd ..
rps@rps-virtual-machine:~$ ls
Desktop  Documents  Downloads  eclipse-workspace  Music  Pictures  Public  snap  Templates  training  Videos  wiprotraining
rps@rps-virtual-machine:~$ cd training
rps@rps-virtual-machine:~/training$ cd TestDir
rps@rps-virtual-machine:~/training/TestDir$ ls
File10.txt  File1.txt  File2.txt  File3.txt  File4.txt  File5.txt  File6.txt  File7.txt  File8.txt  File9.txt
rps@rps-virtual-machine:~/training/TestDir$
```

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.

modify_createdirectoryfiles.sh shell script code

```
#!/bin/bash

# Function to print debug messages
```

```
debug() {
    if [ "$DEBUG" == "true" ]; then
        echo "DEBUG: $1"
    fi
}

# Function to create files
create_files() {
    debug "Creating files..."
    for ((i=1; i<=10; i++)); do
        filename="File${i}.txt"
        echo "$filename" > "$filename"
        debug "Created $filename"
    done
}

# Main script
# Check for debug mode
if [ "$1" == "--debug" ]; then
    DEBUG="true"
else
    DEBUG="false"
fi

# Check if the TestDir directory already exists
if [ -d "TestDir" ]; then
    echo "Directory 'TestDir' already exists."
    debug "Exiting due to existing directory."
    exit 1
fi

# Attempt to create the directory
mkdir -p TestDir

# Check if the directory was created successfully
if [ $? -ne 0 ]; then
    echo "Failed to create directory 'TestDir'."
    debug "Exiting due to failed directory creation."
    exit 1
else
    echo "Directory 'TestDir' created successfully."
fi
```

```
# Change to the TestDir directory
cd TestDir || {
    echo "Failed to change directory to 'TestDir'."
    debug "Exiting due to failed directory change."
    exit 1
}

# Create files
create_files

# Go back to the previous directory
cd ..

echo "Script execution completed successfully."
```

Script file in Vim editor

```bash
#!/bin/bash

# Function to print debug messages
debug() {
    if [ "$DEBUG" == "true" ]; then
        echo "DEBUG: $1"
    fi
}

# Function to create files
create_files() {
    debug "Creating files..."
    for ((i=1; i<=10; i++)); do
        filename="File${i}.txt"
        echo "$filename" > "$filename"
        debug "Created $filename"
    done
}

# Main script
# Check for debug mode
if [ "$1" == "--debug" ]; then
    DEBUG="true"
else
    DEBUG="false"
fi

# Check if the TestDir directory already exists
if [ -d "TestDir" ]; then
    echo "Directory 'TestDir' already exists."
    debug "Exiting due to existing directory."
    exit 1
fi

# Attempt to create the directory
mkdir -p TestDir
```

```bash
# Check if the directory was created successfully
if [ $? -ne 0 ]; then
    echo "Failed to create directory 'TestDir'."
    debug "Exiting due to failed directory creation."
    exit 1
else
    echo "Directory 'TestDir' created successfully."
fi

# Change to the TestDir directory
cd TestDir || {
    echo "Failed to change directory to 'TestDir'."
    debug "Exiting due to failed directory change."
    exit 1
}

# Create files
create_files

# Go back to the previous directory
cd ..
```

Output showing script is working

```
rps@rps-virtual-machine:~/training$ vim modify_createdirectoryfiles.sh
rps@rps-virtual-machine:~/training$ chmod +x modify_createdirectoryfiles.sh
rps@rps-virtual-machine:~/training$ ./modify_createdirectoryfiles.sh --debug
Directory 'TestDir' already exists.
DEBUG: Exiting due to existing directory.
rps@rps-virtual-machine:~/training$
```

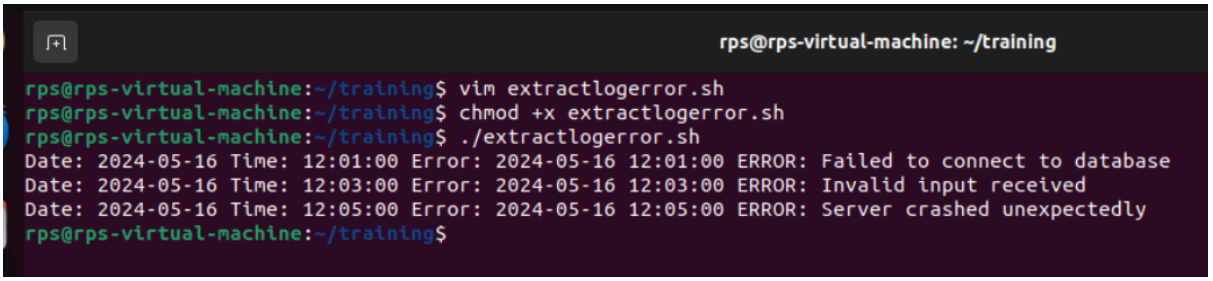
ASSIGNMENT 6

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

extractlogerror.sh shell script code

```
#!/bin/bash

# File path of the sample log file
log_file="sample.log"

# Use grep to extract all lines containing "ERROR" from the log file
error_lines=$(grep "ERROR" "$log_file")

# Use awk to parse each extracted line and print the date, time, and error message
echo "$error_lines" | awk '{print "Date:", $1, "Time:", $2, "Error:", $0}'
```

Script file in Vim editor



Log file

```
rps@rps-virtual-machine:~/training$ cat sample.log
2024-05-16 12:00:00 INFO: Application started
2024-05-16 12:01:00 ERROR: Failed to connect to database
2024-05-16 12:02:00 DEBUG: Processing request 1
2024-05-16 12:03:00 ERROR: Invalid input received
2024-05-16 12:04:00 WARNING: Resource low
2024-05-16 12:05:00 ERROR: Server crashed unexpectedly
2024-05-16 12:06:00 INFO: Application stopped
rps@rps-virtual-machine:~/training$
```

Output showing script is working

rps@rps-virtual-machine: ~/training

```
#!/bin/bash

# File path of the sample log file
log_file="sample.log"

# Use grep to extract all lines containing "ERROR" from the log file
error_lines=$(grep "ERROR" "$log_file")

# Use awk to parse each extracted line and print the date, time, and error message
echo "$error_lines" | awk '{print "Date:", $1, "Time:", $2, "Error:", $0}'
```

ASSIGNMENT 7

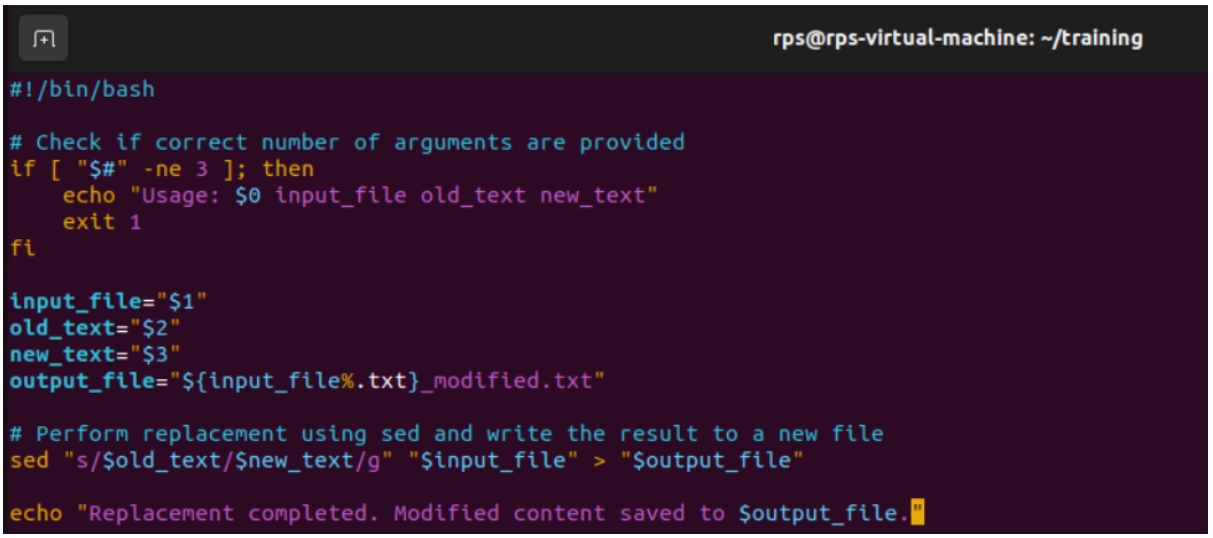
Create a script that takes a text file and replaces all occurrences of "old_text" with "new_text". Use sed to perform this operation and output the result to a new file.

replaceoldtext.sh shell script code

```
#!/bin/bash

# Check if correct number of arguments are provided
if [ "$#" -ne 3 ]; then
    echo "Usage: $0 input_file old_text new_text"
    exit 1
fi

input_file="$1"
old_text="$2"
new_text="$3"
output_file="${input_file%.txt}_modified.txt"

# Perform replacement using sed and write the result to a new file
sed "s/$old_text/$new_text/g" "$input_file" > "$output_file"
```

```
echo "Replacement completed. Modified content saved to
$output_file."
```

Script file in Vim editor



Output showing script is working