

# Unix Bash Assignment

**\*Note:** All the Assignments in the PDF are executed using **WSL2** using **vi editor**, and all the file used to complete the Assignment is Given Below

```
harish@HARISH: ~/Wipro/As: X + v
harish@HARISH:~/Wipro/Assignment$ ls
Assignment1.sh Assignment3.sh Assignment5.sh Assignment7.sh logFile.log newFile.txt
Assignment2.sh Assignment4.sh Assignment6.sh TestDir      myfile.txt sample.txt
harish@HARISH:~/Wipro/Assignment$ |
```

**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"

## Step 1:

Create a Shell Script file for of any name starting with **vi** with the extension of ".sh"

**Syntax: vi filename.sh**

```
harish@HARISH: ~/Wipro/As: X + v
harish@HARISH:~/Wipro/Assignment$ vi Assignment1|.sh
```

## Step 2:

The above code will open the **vi editor** in the editor write the code in Bash shell scripting language.

```
harish@HARISH: ~/Wipro/As: X + v
if [ -f "myfile.txt" ];
then
    echo "file exists"
else
    echo "file not found"
fi
~
```

The above code is a simple if loop to check if the file exists in the current directory or not. "-f" will return true if file exists.

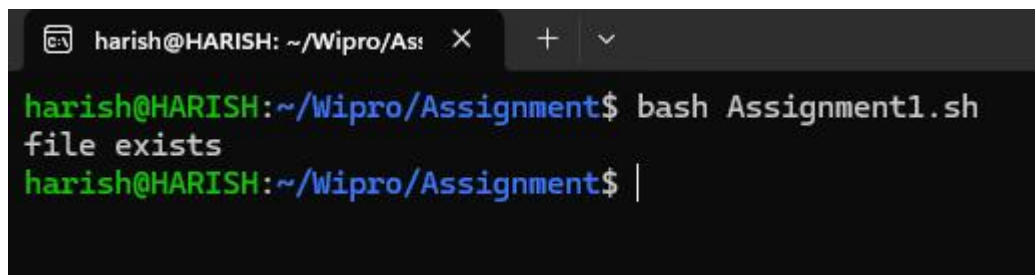
### Step 3:

After writing code close the vi editor by press esc, then ":wq:" to save and quit to the cmd

### Step 4:

To Execute the code use bash filename. And the output will be shown below

**Syntax:** bash filename.sh

A terminal window with a dark background. The prompt is 'harish@HARISH: ~/Wipro/As!'. The user enters 'bash Assignment1.sh'. The output is 'file exists'. The prompt returns to 'harish@HARISH: ~/Wipro/Assignment\$' with a cursor at the end.

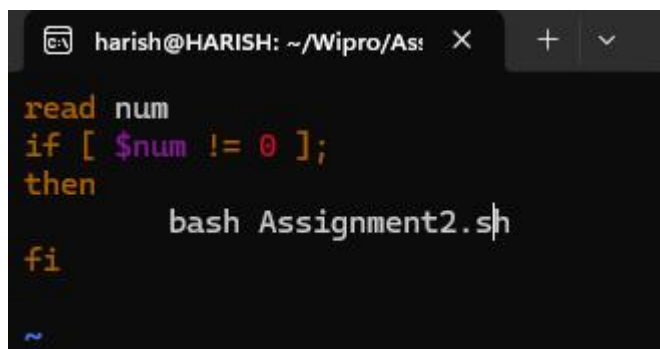
```
harish@HARISH: ~/Wipro/Assignment$ bash Assignment1.sh
file exists
harish@HARISH: ~/Wipro/Assignment$ |
```

**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.

### Answer:

To take input from the user "read" cmd is used

### Program:

A terminal window with a dark background. The prompt is 'harish@HARISH: ~/Wipro/As!'. The user enters a script to read a number and call another script based on whether it's zero or not.

```
read num
if [ $num != 0 ];
then
    bash Assignment2.sh
fi
~
```

Here we use recursion to call the same function until the user gives input as "0".

## Output:

```
harish@HARISH: ~/Wipro/As! X + v
harish@HARISH:~/Wipro/Assignment$ vi Assignment2.sh
harish@HARISH:~/Wipro/Assignment$ bash Assignment2.sh
1
2
3
4
5
0
harish@HARISH:~/Wipro/Assignment$ |
```

**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.

## Answer:

To return the number of lines in a code we use "wc" and the file name of which we want to know the number of lines

**Syntax: wc -l filename**

This will return number of lines and file name

If we don't want the file name then we append < file name at the end.

**Syntax: wc -l < filename**

## Program:

```
harish@HARISH: ~/Wipro/As! X + v
numberOfLines(){
wc -l < $1
}
numberOfLines $1
~
```

## Output:

```
harish@HARISH: ~/Wipro/As: X + v
harish@HARISH:~/Wipro/Assignment$ bash Assignment3.sh logFile.log
20
harish@HARISH:~/Wipro/Assignment$ |
```

Here logFile.log is the file name passing as command line arguments, as it returned 20 which is 20 lines in it.

**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").

## Answer:

To create a new Directory we use "**mkdir**" followed by directory name  
And to create new file with default text we use the following syntax

**Syntax:** "Text" > filename.txt

## Program:

```
harish@HARISH: ~/Wipro/As: X + v
mkdir TestDir
cd TestDir
for((i = 1 ; i<11;i++))
do
echo "File$i" > "File$i".txt
done
ls
cd ..|
~
~
```

## Output:

```
harish@HARISH: ~/Wipro/As: X + v
harish@HARISH:~/Wipro/Assignment$ bash Assignment4.sh
File1.txt File10.txt File2.txt File3.txt File4.txt File5.txt File6.txt File7.txt File8.txt File9.txt
harish@HARISH:~/Wipro/Assignment$ |
```

The above program creates a directory and in the directory we use for loop to traverse 10 times and creates files 10 times. And "ls" cmd to print all the file in the directory. And comes back using "cd ..".

**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.

### Answer:

The previous assignment executes only once because a directory can't be created again with the same name.

So to rectify that we use -d to check if the file exists or not

### Program:

```
harish@HARISH: ~/Wipro/As: X + v
if [ ! -d "$TestDir" ];
then
    echo "The Directory TestDir Already Created."
    echo "The TestDir contains the following files in it:"
    cd TestDir
    | ls
    cd ..
else
    mkdir TestDir
    cd TestDir
    for((i = 1 ; i<11;i++))
    do
        echo "File$i" > "File$i".txt
    done
    ls
    cd ..
fi
~
~
~
```

### Output:

```
harish@HARISH: ~/Wipro/Assignment$ bash Assignment5.sh
The Directory TestDir Already Created.
The TestDir contains the following files in it:
File1.txt File10.txt File2.txt File3.txt File4.txt File5.txt File6.txt File7.txt File8.txt File9.txt
harish@HARISH: ~/Wipro/Assignment$
```

**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.

**Answer:**

**Grep** is a command used for searching and manipulating text in a file  
**Awk** is a programming language used for manipulating text in a file.

**logFile:**

To perform grep and awk operation we create a sample log file with some content in it.

```
harish@HARISH: ~/Wipro/Asr
harish@HARISH:~/Wipro/Assignment$ vi Assignment6.sh
harish@HARISH:~/Wipro/Assignment$ cat logFile.log
[ERROR] 2023-05-15 14:30:45 - Connection timed out: Unable to establish a connection to the database server.
[ERROR] 2023-05-15 14:35:20 - File not found: The requested file "report.pdf" does not exist in the specified directory.
[ERROR] 2023-05-15 14:40:12 - Authentication failed: Invalid credentials. Please check your username and password.
[ERROR] 2023-05-15 14:45:55 - Out of memory: The application exceeded the allocated memory limit. Consider optimizing your code.
[ERROR] 2023-05-15 14:50:30 - Null pointer exception: Attempted to access a null object reference.
[ERROR] 2023-05-15 14:55:18 - Database query failed: SQL syntax error near line 42.
[ERROR] 2023-05-15 15:00:05 - Disk space full: The system drive is running out of space. Delete unnecessary files.
[ERROR] 2023-05-15 15:05:40 - Network error: Unable to connect to the remote server. Check your internet connection.
[ERROR] 2023-05-15 15:10:22 - Division by zero: Attempted to divide a number by zero.
[ERROR] 2023-05-15 15:15:08 - Critical error: Application crashed unexpectedly. Investigate the root cause.
[INFO] 2023-05-15 14:30:45 - Application started successfully.
[INFO] 2023-05-15 14:35:20 - User logged in: Welcome, John Doe!
[INFO] 2023-05-15 14:40:12 - Data synchronization complete: 100 records updated.
[INFO] 2023-05-15 14:45:55 - Task completed: Backup process finished without errors.
[INFO] 2023-05-15 14:50:30 - Notification sent: New product added to the catalog.
[INFO] 2023-05-15 14:55:18 - Report generated: Sales summary for Q1 2023.
[INFO] 2023-05-15 15:00:05 - System update: Security patches applied successfully.
[INFO] 2023-05-15 15:05:40 - User activity: Jane Smith edited her profile.
[INFO] 2023-05-15 15:10:22 - Task scheduled: Daily data backup initiated.
[INFO] 2023-05-15 15:15:08 - Server status: All services running normally.
harish@HARISH:~/Wipro/Assignment$
```

**Program:**

```
harish@HARISH: ~/Wipro/Asr
echo "using Grep"
echo ""
grep -h 'ERROR' logFile.log
echo ""
echo "Using Awk"
echo ""
awk '/ERROR/ {$1=""; print $0}' logFile.log
~
```

**Syntax for Grep:** `grep -h 'WORD' filename`

**Syntax for Awk:** `awk '/word/ {print $column number`

Here \$1 is to make 1<sup>st</sup> column null and \$0 is to print whole line

## Output:

```
harish@HARISH: ~/Wipro/Asn X + v
harish@HARISH:~/Wipro/Assignment$ bash Assignment6.sh
using Grep

[ERROR] 2023-05-15 14:30:45 - Connection timed out: Unable to establish a connection to the database server.
[ERROR] 2023-05-15 14:35:20 - File not found: The requested file "report.pdf" does not exist in the specified directory.
[ERROR] 2023-05-15 14:40:12 - Authentication failed: Invalid credentials. Please check your username and password.
[ERROR] 2023-05-15 14:45:55 - Out of memory: The application exceeded the allocated memory limit. Consider optimizing your code.
[ERROR] 2023-05-15 14:50:30 - Null pointer exception: Attempted to access a null object reference.
[ERROR] 2023-05-15 14:55:18 - Database query failed: SQL syntax error near line 42.
[ERROR] 2023-05-15 15:00:05 - Disk space full: The system drive is running out of space. Delete unnecessary files.
[ERROR] 2023-05-15 15:05:40 - Network error: Unable to connect to the remote server. Check your internet connection.
[ERROR] 2023-05-15 15:10:22 - Division by zero: Attempted to divide a number by zero.
[ERROR] 2023-05-15 15:15:08 - Critical error: Application crashed unexpectedly. Investigate the root cause.

Using Awk

2023-05-15 14:30:45 - Connection timed out: Unable to establish a connection to the database server.
2023-05-15 14:35:20 - File not found: The requested file "report.pdf" does not exist in the specified directory.
2023-05-15 14:40:12 - Authentication failed: Invalid credentials. Please check your username and password.
2023-05-15 14:45:55 - Out of memory: The application exceeded the allocated memory limit. Consider optimizing your code.
2023-05-15 14:50:30 - Null pointer exception: Attempted to access a null object reference.
2023-05-15 14:55:18 - Database query failed: SQL syntax error near line 42.
2023-05-15 15:00:05 - Disk space full: The system drive is running out of space. Delete unnecessary files.
2023-05-15 15:05:40 - Network error: Unable to connect to the remote server. Check your internet connection.
2023-05-15 15:10:22 - Division by zero: Attempted to divide a number by zero.
2023-05-15 15:15:08 - Critical error: Application crashed unexpectedly. Investigate the root cause.
harish@HARISH:~/Wipro/Assignment$ |
```

In the above output using grep we displayed the whole line in which ERROR is present  
And using Awk we displayed the date and time and the error message without the word Error in it.



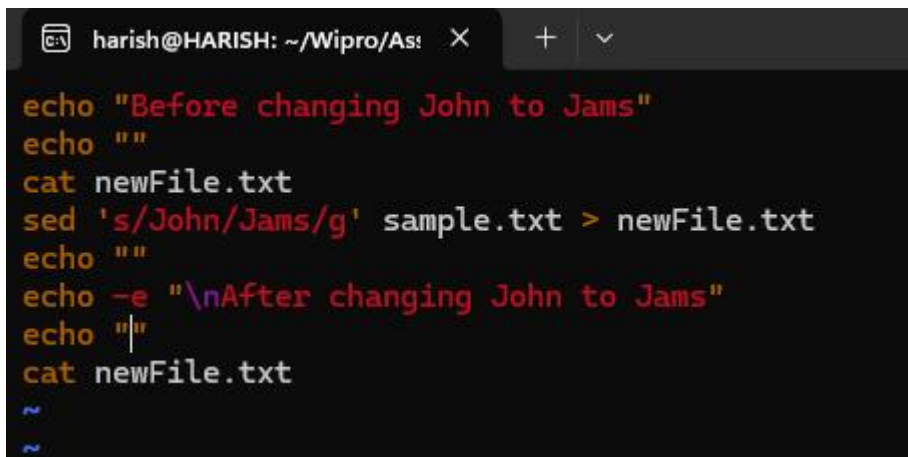
**Assignment 7:** Create a script that takes a text file and replaces all occurrences of "old\_text" with "new\_text". Use sed to perform this operation and output the result to a new file.

**Answer:**

**Sed** is a command stands for **Streaming Editor** and it can perform many functions like searching, find and replace, insertion and deletion. In this program we will replace old text with new one.

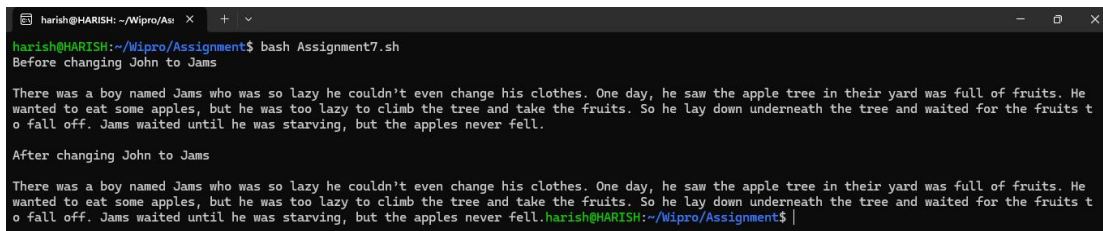
**Syntax:** sed 's/old\_text/new\_text' filename

**Program:**

A terminal window with a dark background. The prompt is 'harish@HARISH: ~/Wipro/As:'. The script contains the following commands: 'echo "Before changing John to Jams"', 'echo ""', 'cat newFile.txt', 'sed 's/John/Jams/g' sample.txt > newFile.txt', 'echo ""', 'echo -e "\nAfter changing John to Jams"', 'echo ""', and 'cat newFile.txt'. There are two tilde '~' characters at the bottom of the terminal.

```
harish@HARISH: ~/Wipro/As: X + v
echo "Before changing John to Jams"
echo ""
cat newFile.txt
sed 's/John/Jams/g' sample.txt > newFile.txt
echo ""
echo -e "\nAfter changing John to Jams"
echo ""
cat newFile.txt
~
~
```

**Output:**

A terminal window showing the execution of a script. The prompt is 'harish@HARISH: ~/Wipro/Assignment\$'. The script output is: 'Before changing John to Jams', followed by a paragraph about a boy named John, then 'After changing John to Jams', followed by the same paragraph with 'John' replaced by 'Jams'. The prompt returns to 'harish@HARISH: ~/Wipro/Assignment\$'.

```
harish@HARISH: ~/Wipro/Assignment$ bash Assignment7.sh
Before changing John to Jams

There was a boy named John who was so lazy he couldn't even change his clothes. One day, he saw the apple tree in their yard was full of fruits. He wanted to eat some apples, but he was too lazy to climb the tree and take the fruits. So he lay down underneath the tree and waited for the fruits to fall off. John waited until he was starving, but the apples never fell.

After changing John to Jams

There was a boy named Jams who was so lazy he couldn't even change his clothes. One day, he saw the apple tree in their yard was full of fruits. He wanted to eat some apples, but he was too lazy to climb the tree and take the fruits. So he lay down underneath the tree and waited for the fruits to fall off. Jams waited until he was starving, but the apples never fell.harish@HARISH:~/Wipro/Assignment$
```

In the above output the name John is replaced with Jams.