## 16-05 ASSIGNMENTS (1-7)

**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:**

```
$ vi condition.sh

 #!/bin/bash
filename="myfile.txt"
if [ -e "$filename" ]
then
        echo "File exists."
else
        echo "File not found."
fi

$ chmod +x condition.sh
$ ./condition.sh
```

**OUTPUT:**

```
 File not found.
```

```
$ vi condition.sh

 #!/bin/bash
filename="condition.sh"
if [ -e "$filename" ]
then
        echo "File exists."
else
        echo "File not found."
fi

$ chmod +x condition.sh
$ ./condition.sh
```

**OUTPUT:**

```
 File exists.
```

**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:**

$ vi oddoreven.sh

```
#!/bin/bash
while true
do
	read num
	if [ “$num” -eq 0 ]
	then
		echo “exit”
		break
	fi
	if [ $((num % 2)) -eq 0 ]
	then
		echo “$num is even”
	else
		echo “$num is even”
	fi
done
```

```
$ chmod +x oddoreven.sh
$ ./oddoreven.sh
```

**OUTPUT:**

```
3
3 is odd
4
4 is even
9
9 is odd
0
exit
```

**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:**

$ vi assign3.sh

```
#!/bin/bash
function countlines() {
      filename="$1"
      numlines =$(wc -l < "$filename")
      echo "$filename file has $numlines lines"
      }
countlines condition.sh
countlines oddoreven.sh
```

```
$ chmod +x assign3.sh
$ ./assign3.sh
```

**OUTPUT:**

```
condition.sh file has 8 lines
oddoreven.sh file has 15 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:**

$ vi assign4.sh

```
#!/bin/bash
mkdir TestDir
cd TestDir
for ((i=1; i<=10; i++))
do
      filename=”File${i}.txt”
      echo “$filename” > “$filename”
done
```

```
$ chmod +x assign4.sh
$ ./assign4.sh
```

**OUTPUT:**

File1.txt, File2.txt,....File10.txt will be created in TestDir directory

**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.**

$ vi assign5.sh

```
#!/bin/bash
debug_msg() {
   if [ "$DEBUG" = "true" ]
 then
       echo "[DEBUG] $1"
   fi
}

create_directory() {
   if [ -d "$1" ]
 then
       echo "Directory $1 already exists."
       return 1
   fi

   mkdir -p "$1"
   if [ $? -ne 0 ]
 then
       echo "Error: Failed to create directory $1"
       return 1
   fi

   debug_msg "Directory $1 created successfully."
}

DEBUG="false"

while getopts ":d" opt
 do
   case $opt in
       d)
           DEBUG="true"
             ;;
      \?)
           echo "Invalid option: -$OPTARG" >&2
           exit 1
           ;;
   esac
done

shift $((OPTIND - 1))

if [ $# -eq 0 ]
 then
   echo "Usage: $0 [-d] directory_name"
   exit 1
fi

directory_name="$1"
```

```
create_directory "$directory_name"
if [ $? -ne 0 ]; then
    exit 1
fi

echo "Directory created successfully."
```

```
$ chmod +x assign5.sh
$ ./assign5.sh -d test_dir
```

**OUTPUT:**

```
[DEBUG] Directory test_dir created successfully.
Directory created successfully.

[DEBUG] Directory test_dir already exists.
Directory test_dir already exists.

Error: Failed to create directory /root/test_dir
```

**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:**

```
$ vi assign6.sh

filename="textfile.txt"

if [ ! -f “$filename" ]; then
   echo "Error: File 'filename' not found."
   exit 1
fi
grep "ERROR" "$filename" | awk '{print $1, $2, $0}

$ chmod +x assign6.sh
$ ./assign6.sh
```

**OUTPUT:**

```
Date-Time: 2024-05-16 14:23:45 Error: ERROR: ERROR massage 1
Date-Time: 2024-05-16 14:25:10 Error: ERROR: ERROR massage 2
```

**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 assign7.sh

```
#!/bin/bash
if [ "$#" -ne 3 ]
then
      echo "Modify: $0 <file_name> <old_text> <new_text>"
      exit 1
fi
file_name="$1"
old_text="$2"
new_text="$3"
new_file="${file_name}.new"
sed "s/$old_text/$new_text/g" "$file_name" > "$new_file"
echo “Modify completed successfully”
```

```
$ chmod +x assign7.sh
$ ./assihn7.sh  sampletext.txt  Evening  Night
```

**OUTPUT:**

Modify completed successfully

Old file is sampletext.txt “This is Thursday Evening”

New file is sampletext.txt.new “This is Thursday Night”