Campus Hires 2026
LINUX Assessment Module 3

S Vimal

Scenario: Automating file backup and Reporting to the system. Create a shell script called "backup_manager.sh" that performs the following tasks incorporating the concepts suggested.

Requirements:

1. Command-line Arguments and Quoting:
The script must accept three arguments: Source directory: A directory containing files to back up. Backup directory: The destination where files will be backed up. File extension: A specific file extension to filter (e.g., .txt).
Example: ./backup_manager.sh "/home/user/source" "/backup" ".txt"

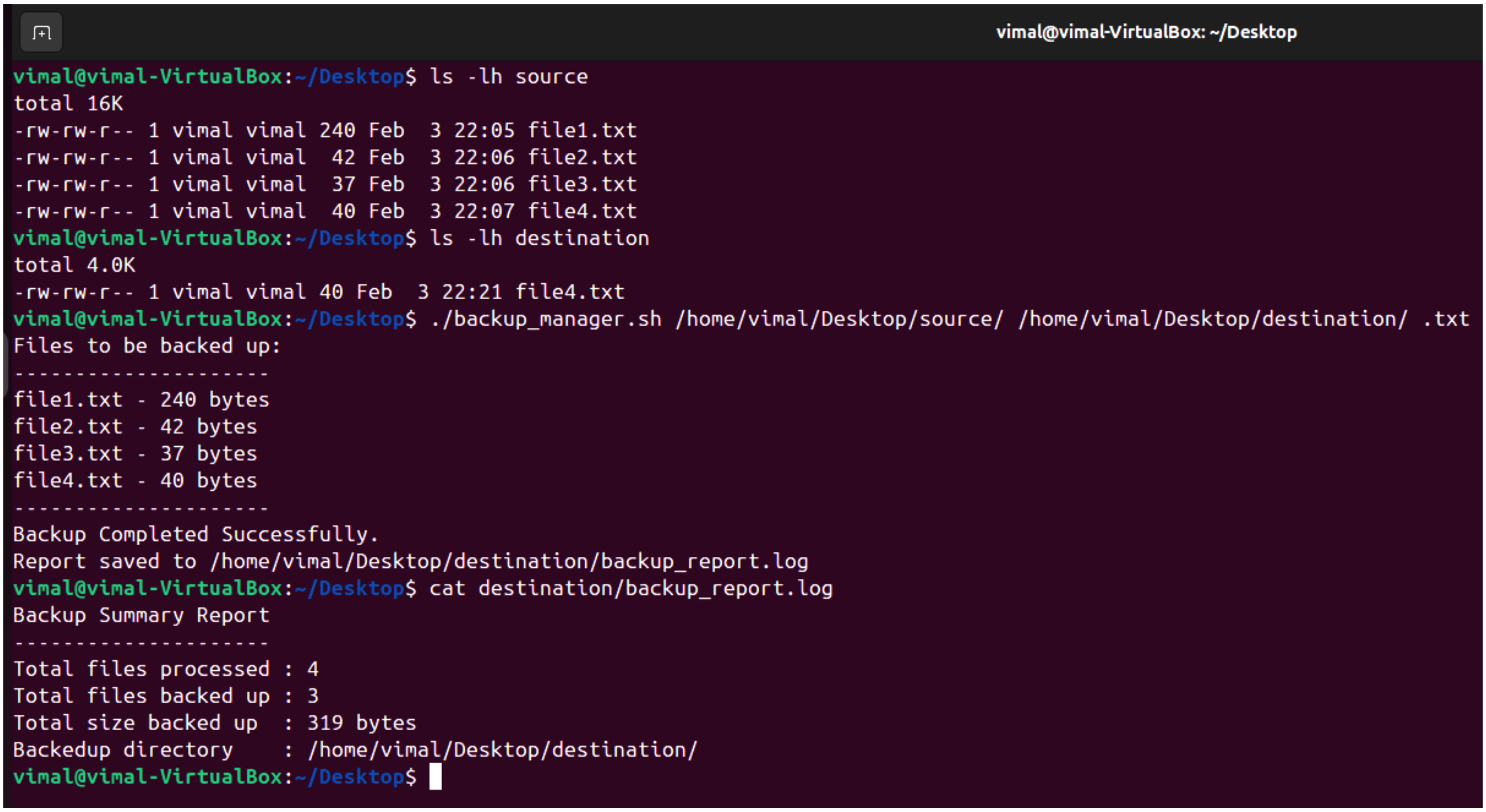
2. Globbing:
The script should use globbing to find all files in the source directory matching the provided file extension.

3. Export Statements:
Use export to set an environment variable BACKUP_COUNT, which tracks the total number of files backed up during the script execution.

4. Array Operations:
Store the list of files to be backed up in an array.
Print the names of these files along with their sizes before performing the backup.

5. Conditional Execution:
If the backup directory does not exist, create it. If creation fails, exit with an error.
If the source directory is empty or contains no files matching the extension, exit with a message.
If a file already exists in the backup directory with the same name, only overwrite it if it is older than the source file (compare timestamps).

6. Output Report:
After the backup, generate a summary report displaying:
Total files processed.
Total size of files backed up.
The path to the backup directory.
The report should be saved in the backup directory as backup_report.log.



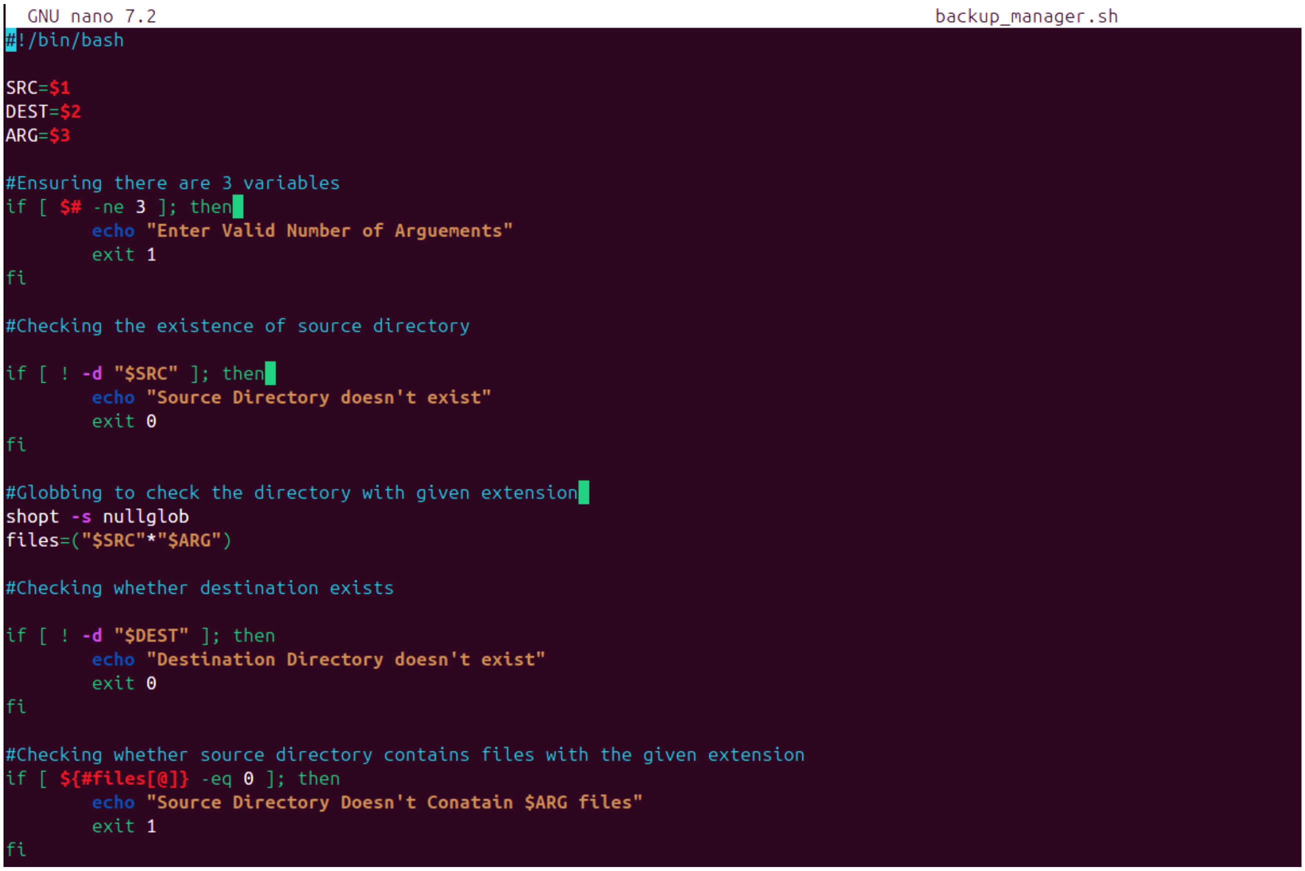
```
vimal@vimal-VirtualBox:~/Desktop$ ls -lh source
total 16K
-rw-rw-r-- 1 vimal vimal 240 Feb  3 22:05 file1.txt
-rw-rw-r-- 1 vimal vimal  42 Feb  3 22:06 file2.txt
-rw-rw-r-- 1 vimal vimal  37 Feb  3 22:06 file3.txt
-rw-rw-r-- 1 vimal vimal  40 Feb  3 22:07 file4.txt
vimal@vimal-VirtualBox:~/Desktop$ ls -lh destination
total 4.0K
-rw-rw-r-- 1 vimal vimal 40 Feb  3 22:21 file4.txt
vimal@vimal-VirtualBox:~/Desktop$ ./backup_manager.sh /home/vimal/Desktop/source/ /home/vimal/Desktop/destination/ .txt
Files to be backed up:
---------------------
file1.txt - 240 bytes
file2.txt - 42 bytes
file3.txt - 37 bytes
file4.txt - 40 bytes
---------------------
Backup Completed Successfully.
Report saved to /home/vimal/Desktop/destination/backup_report.log
vimal@vimal-VirtualBox:~/Desktop$ cat destination/backup_report.log
Backup Summary Report
---------------------
Total files processed : 4
Total files backed up : 3
Total size backed up  : 319 bytes
Backedup directory    : /home/vimal/Desktop/destination/
vimal@vimal-VirtualBox:~/Desktop$ 
```

// source directory and destination directory were created and files were added, file4 is also present in destination which is the latest among source and destination, for the test case of not copying the already existing latest file.

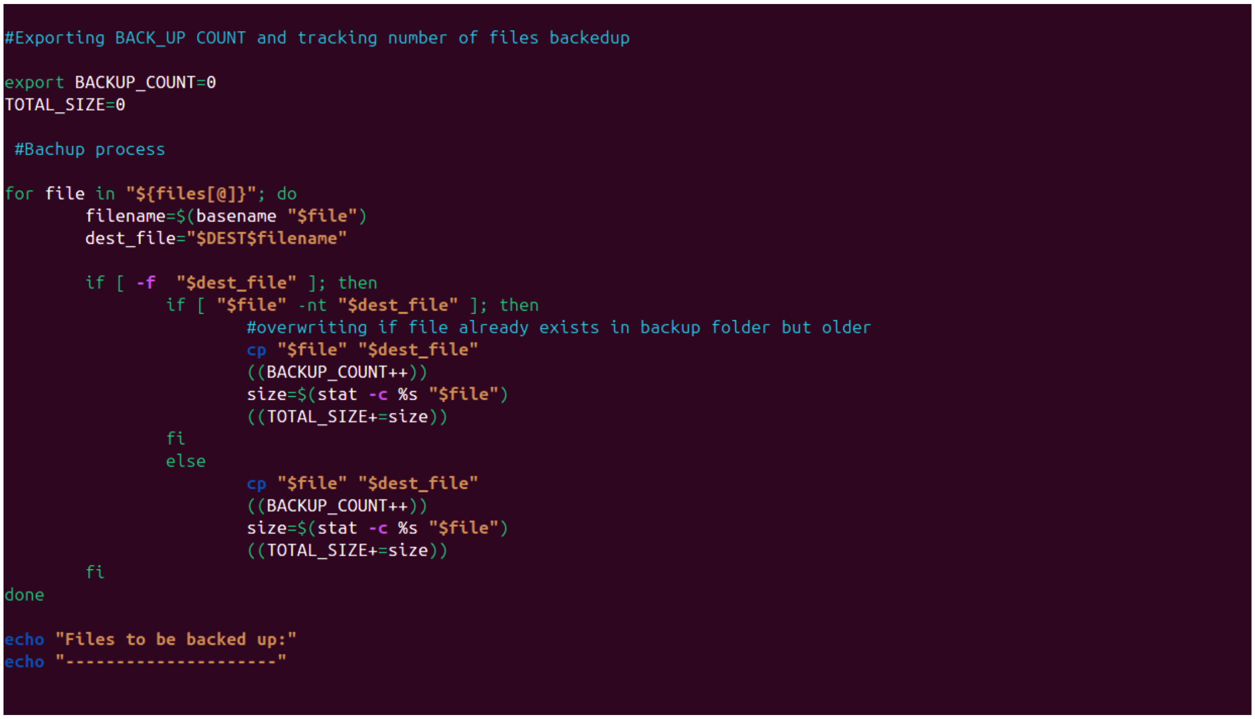
// .txt extension is used as third argument for backup process

// the script is executed and backup report is also generated and the content can be seen using the cat command

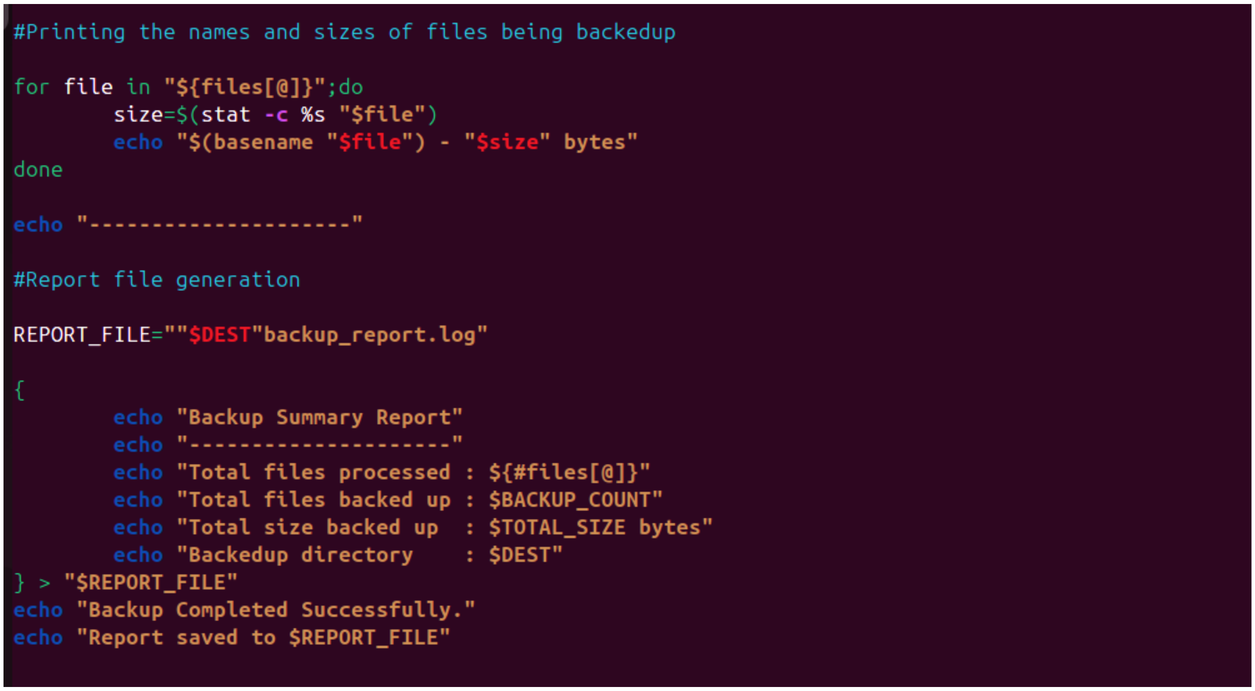
backup_manager.sh



```
  GNU nano 7.2                                                                          backup_manager.sh
#!/bin/bash

SRC=$1
DEST=$2
ARG=$3

#Ensuring there are 3 variables
if [ $# -ne 3 ]; then
        echo "Enter Valid Number of Arguements"
        exit 1
fi

#Checking the existence of source directory

if [ ! -d "$SRC" ]; then
        echo "Source Directory doesn't exist"
        exit 0
fi

#Globbing to check the directory with given extension
shopt -s nullglob
files=("$SRC"*"$ARG")

#Checking whether destination exists

if [ ! -d "$DEST" ]; then
        echo "Destination Directory doesn't exist"
        exit 0
fi

#Checking whether source directory contains files with the given extension
if [ ${#files[@]} -eq 0 ]; then
        echo "Source Directory Doesn't Conatain $ARG files"
        exit 1
fi
```



```
#Exporting BACK_UP COUNT and tracking number of files backedup

export BACKUP_COUNT=0
TOTAL_SIZE=0

 #Bachup process

for file in "${files[@]}"; do
        filename=$(basename "$file")
        dest_file="$DEST$filename"

        if [ -f  "$dest_file" ]; then
                if [ "$file" -nt "$dest_file" ]; then
                        #overwriting if file already exists in backup folder but older
                        cp "$file" "$dest_file"
                        ((BACKUP_COUNT++))
                        size=$(stat -c %s "$file")
                        ((TOTAL_SIZE+=size))
                fi
                else
                        cp "$file" "$dest_file"
                        ((BACKUP_COUNT++))
                        size=$(stat -c %s "$file")
                        ((TOTAL_SIZE+=size))
        fi
done

echo "Files to be backed up:"
echo "---------------------"
```



```
#Printing the names and sizes of files being backedup

for file in "${files[@]}";do
        size=$(stat -c %s "$file")
        echo "$(basename "$file") - "$size" bytes"
done

echo "---------------------"

#Report file generation

REPORT_FILE=""$DEST"backup_report.log"

{
        echo "Backup Summary Report"
        echo "---------------------"
        echo "Total files processed : ${#files[@]}"
        echo "Total files backed up : $BACKUP_COUNT"
        echo "Total size backed up  : $TOTAL_SIZE bytes"
        echo "Backedup directory    : $DEST"
} > "$REPORT_FILE"
echo "Backup Completed Successfully."
echo "Report saved to $REPORT_FILE"
```