Module – 3

Assignment Submission – embedUR Linux Course

02.02.2026

Baalamurugan S P

baalamurugan.sp@gmail.com

Question – 1

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.

```
baala@ItsB-Laptop:~/assignments/module3$ nano backup_manager.sh
baala@ItsB-Laptop:~/assignments/module3$ ./backup_manager.sh "/home/baala/assignments/module2" "/home/baala/
assignments/module3" ".txt"
Files to be backed up:
--------------------------------------
config.txt - Size: 66 bytes
filtered_log.txt - Size: 148 bytes
log.txt - Size: 730 bytes
updated_config.txt - Size: 66 bytes
--------------------------------------

Backup completed successfully!
Total files backed up: 4
Total size: 1010 bytes
Report saved to: /home/baala/assignments/module3/backup_report.log
baala@ItsB-Laptop:~/assignments/module3$ ls
backup_manager.sh  backup_report.log  config.txt  filtered_log.txt  log.txt  updated_config.txt
baala@ItsB-Laptop:~/assignments/module3$ cat backup_report.log
Backup Report
==========================================
Date: Mon Feb  2 15:10:40 UTC 2026
Source Directory: /home/baala/assignments/module2
Backup Directory: /home/baala/assignments/module3
File Extension Filter: .txt
------------------------------------------
Total Files Processed: 4
Total Size Backed Up: 1010 bytes
Backup Location: /home/baala/assignments/module3
==========================================
baala@ItsB-Laptop:~/assignments/module3$ cat config.txt
server=localhost
db_host=localhost
api_url=http://localhost:8080

baala@ItsB-Laptop:~/assignments/module3$
```

Code:

```
#!/bin/bash

SOURCE_DIR="$1"

BACKUP_DIR="$2"

FILE_EXT="$3"

if [ $# -ne 3 ]; then

    echo "Error: Exactly 3 arguments required"

    echo "Usage: $0 <source_dir> <backup_dir> <file_extension>"

    exit 1

fi

if [ ! -d "$SOURCE_DIR" ]; then

    echo "Error: Source directory '$SOURCE_DIR' does not exist"

    exit 1

fi

if [ ! -d "$BACKUP_DIR" ]; then

    mkdir -p "$BACKUP_DIR"

    if [ $? -ne 0 ]; then

        echo "Error: Failed to create backup directory '$BACKUP_DIR'"

        exit 1

    fi
```

```
fi

shopt -s nullglob
FILES=("$SOURCE_DIR"/*"$FILE_EXT")
shopt -u nullglob

if [ ${#FILES[@]} -eq 0 ]; then
    echo "No files with extension '$FILE_EXT' found in '$SOURCE_DIR'"
    exit 0
fi

declare -a file_array
total_size=0
count=0

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

for file in "${FILES[@]}"; do
    if [ -f "$file" ]; then
        file_array+=("$file")
        filename=$(basename "$file")
        filesize=$(stat -f%z "$file" 2>/dev/null || stat -c%s "$file" 2>/dev/null)
        echo "$filename - Size: $filesize bytes"
        total_size=$((total_size + filesize))
    fi
done

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

for file in "${file_array[@]}"; do
    filename=$(basename "$file")
    dest_file="$BACKUP_DIR/$filename"

    should_copy=1

    if [ -f "$dest_file" ]; then
        if [ "$file" -nt "$dest_file" ]; then
            should_copy=1
```

```
        else
            should_copy=0
        fi
    fi

    if [ $should_copy -eq 1 ]; then
        cp "$file" "$dest_file"
        if [ $? -eq 0 ]; then
            count=$((count + 1))
        fi
    fi
done

export BACKUP_COUNT=$count

report_file="$BACKUP_DIR/backup_report.log"

{
    echo "Backup Report"
    echo "=============================================="
    echo "Date: $(date)"
    echo "Source Directory: $SOURCE_DIR"
    echo "Backup Directory: $BACKUP_DIR"
    echo "File Extension Filter: $FILE_EXT"
    echo "----------------------------------------------"
    echo "Total Files Processed: $count"
    echo "Total Size Backed Up: $total_size bytes"
    echo "Backup Location: $BACKUP_DIR"
    echo "=============================================="
} > "$report_file"

echo ""
echo "Backup completed successfully!"
echo "Total files backed up: $BACKUP_COUNT"
echo "Total size: $total_size bytes"
echo "Report saved to: $report_file"
exit 0
```