LAB5.md

🧪 Lab 4 – File & Backup Automation

🎯 Objective

The goal of this assignment is to automate file management using a Bash script. Specifically, the script will:

- Search for all `.txt` files in the current directory.
- Copy them to a `backup/` folder.
- Append a **timestamp** to each backup file's name to avoid overwriting.
- Be tested with sample `.txt` files.

⚙ How the Script Works

Script name: `backup.sh`

🔧 Functionality:

1. **Creates a `backup/` folder** if it does not exist.
2. **Gets the current timestamp** in the format `YYYYMMDD_HHMMSS`.
3. **Finds all `.txt` files** in the current directory.
4. **Copies each `.txt` file** to the `backup/` folder with a new name that includes the timestamp.
    - Example: `notes.txt` → `backup/notes_20250910_152201.txt`
5. **Prints a success message** for each file copied, or a warning if no `.txt` files are found.

💻 The Script: `backup.sh`

```
#!/bin/bash
# backup.sh - Backup all .txt files with timestamp

# === Create backup directory if it doesn't exist ===
BACKUP_DIR="backup"
mkdir -p "$BACKUP_DIR"

# === Get current timestamp ===
TIMESTAMP=$(date +"%Y%m%d_%H%M%S")

# === Find and copy all .txt files ===
echo "📁 Backing up .txt files..."

count=0
for file in *.txt; do
    if [ -f "$file" ]; then
        cp "$file" "$BACKUP_DIR/${file%.txt}_$TIMESTAMP.txt"
        echo "✅ $file → $BACKUP_DIR/${file%.txt}_$TIMESTAMP.txt"
```

```
        ((count++))
    fi
done

# === Final message ===
if [ "$count" -eq 0 ]; then
    echo "⚠ No .txt files found to back up."
else
    echo "🎉 Backup complete! $count files copied to '$BACKUP_DIR'."
fi
```

🧪 Example Test & Output

📝 Step 1: Create Test Files

```
echo "Hello" > notes.txt
echo "Backup test" > report.txt
```

📝 Step 2: Run the script

```
$ ./backup.sh
📁 Backing up .txt files...
✅ notes.txt → backup/notes_20250910_152201.txt
✅ report.txt → backup/report_20250910_152201.txt
🎉 Backup complete! 2 files copied to 'backup'.
```

❓ Extra Questions

---

1. What is the difference between `cp`, `mv`, and `rsync`?

Command	Description	Example
`cp`	Copies files or directories	`cp file.txt backup/`
`mv`	Moves or renames files or directories	`mv old.txt new.txt`
`rsync`	Synchronizes files and directories efficiently	`rsync -av source/ dest/`

Key Differences:

- `cp` creates a duplicate file in a new location.
- `mv` removes the file from the original location and places it in a new one.
- `rsync` is used for syncing files efficiently, especially in backup operations. It can transfer only the differences between files and works over networks.

---

2. How can you schedule scripts to run automatically?

✅ Using `cron` (Linux/macOS)

`cron` is a built-in Linux tool used to schedule tasks (cron jobs) at specific times or intervals.

Steps to Schedule a Script:

1. Open the crontab file:

```
crontab -e
```

2. Add a line to schedule your script. Example: Run backup.sh every day at 5:00 PM:

0 17 * * * /full/path/to/backup.sh