🤖 AUTOMATION USING SHELL SCRIPT

📝 Assignment 4 – File & Backup Automation

Objective: Automate file management using Bash scripting.

📌 Task 1: Writing the Script `backup.sh`

🎯 Purpose

The script will:

1. Find all `.txt` files in the current folder.
2. Copy them into a `backup/` directory.
3. Add a **timestamp** to the filename for versioning.

📜 Script Code

```bash
#!/bin/bash
# backup.sh

# Create backup directory if it doesn’t exist
mkdir -p backup

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

# Find and copy all .txt files into backup folder with timestamp
for file in *.txt
do
  if [ -f "$file" ]; then
    cp "$file" "backup/${file%.txt}_$timestamp.txt"
    echo "Backed up: $file -> backup/${file%.txt}_$timestamp.txt"
  fi
done
```

📌 Task 2: Testing the Script

1. Create some .txt files:

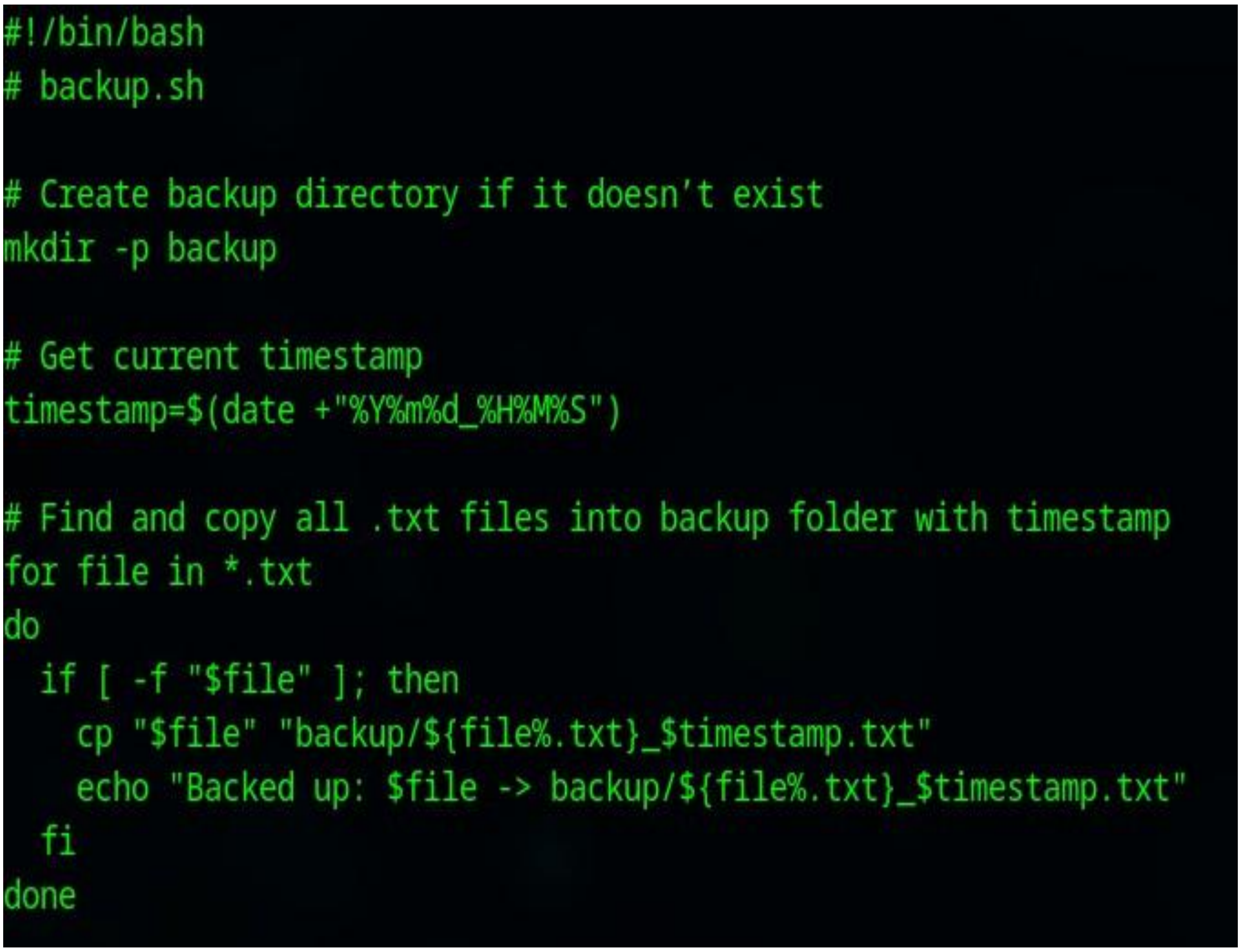
```bash
echo "Hello World" > file1.txt
echo "Backup test" > file2.txt
```

2. Run the script:

```
./backup.sh
```

Check the backup/ folder:

```
backup/file1_20250909_120530.txt
backup/file2_20250909_120530.txt
```



```
#!/bin/bash
# backup.sh

# Create backup directory if it doesn't exist
mkdir -p backup

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

# Find and copy all .txt files into backup folder with timestamp
for file in *.txt
do
  if [ -f "$file" ]; then
    cp "$file" "backup/${file%.txt}_$timestamp.txt"
    echo "Backed up: $file -> backup/${file%.txt}_$timestamp.txt"
  fi
done
```

📌 Task 3: Documentation – How the Script Works

1. `mkdir -p backup` → ensures the `backup/` folder exists.
2. `timestamp=$(date +"%Y%m%d_%H%M%S")` → creates a unique timestamp.
3. `for file in *.txt` → loops through all `.txt` files in the current directory.
4. `cp "$file" "backup/${file%.txt}_$timestamp.txt"` → copies each file to the `backup/` folder with timestamp in the name.
5. `echo` → prints a confirmation message for each file backed up.

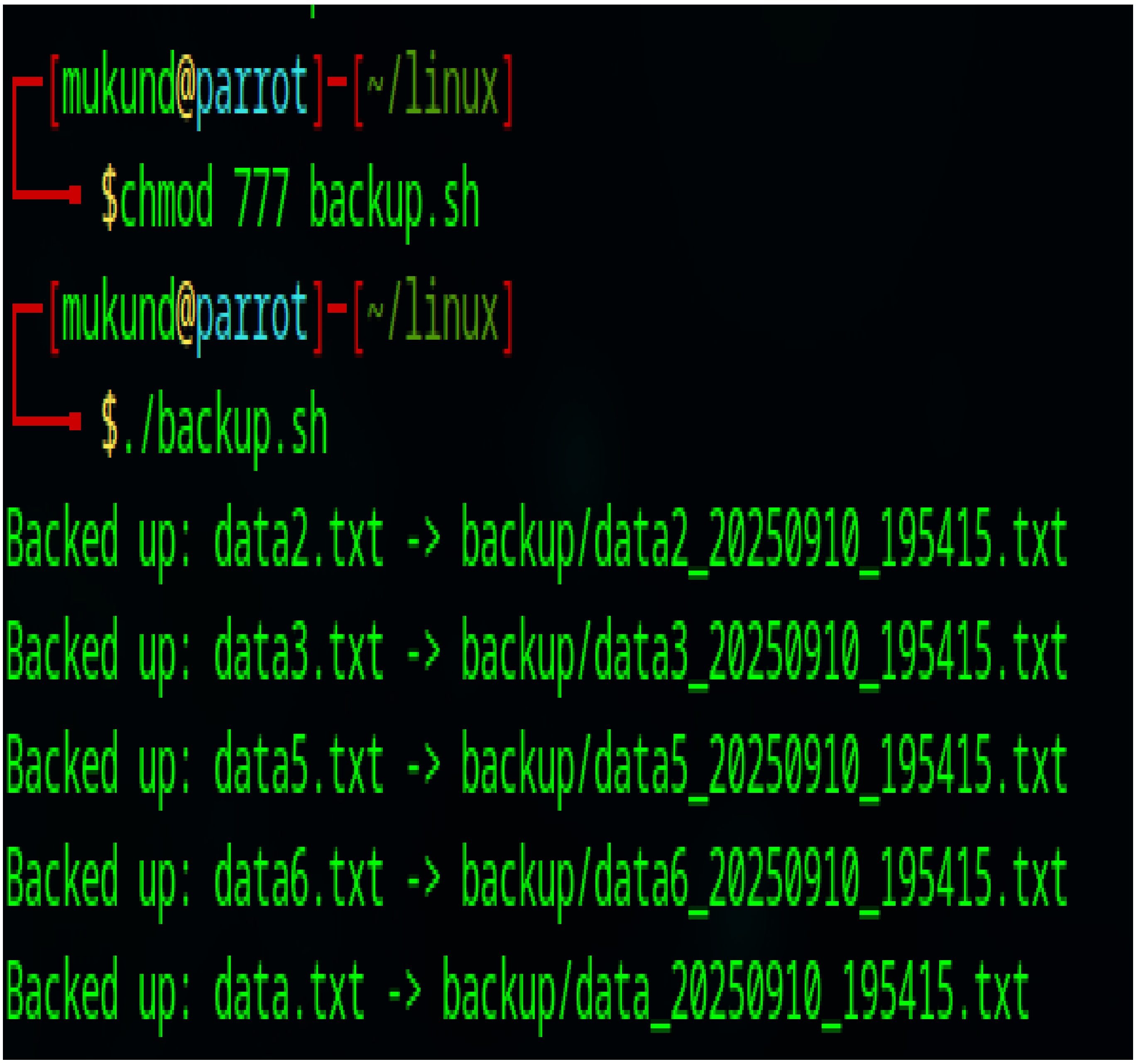
▶️ Example Run

Command:

```
./backup.sh
```

output

```
Backed up: file1.txt -> backup/file1_20250909_120530.txt
Backed up: file2.txt -> backup/file2_20250909_120530.txt
```



```
┌─[mukund@parrot]─[~/linux]
└──╼ $chmod 777 backup.sh
┌─[mukund@parrot]─[~/linux]
└──╼ $./backup.sh
Backed up: data2.txt -> backup/data2_20250910_195415.txt
Backed up: data3.txt -> backup/data3_20250910_195415.txt
Backed up: data5.txt -> backup/data5_20250910_195415.txt
Backed up: data6.txt -> backup/data6_20250910_195415.txt
Backed up: data.txt -> backup/data_20250910_195415.txt
```

❓ Extra Questions

Q1: What is the difference between `cp`, `mv`, and `rsync`?

- `cp` → copies files or directories.
- `mv` → moves or renames files/directories (the original is removed).
- `rsync` → advanced copy tool that syncs files/directories efficiently (supports incremental backup, remote copy, and compression).

Q2: How can you schedule scripts to run automatically?

✅ **Using cron jobs in Linux:**

1. Open crontab:

```
crontab -e
```

2. Add a job (example: run backup every day at midnight):

```
0 0 * * * /path/to/backup.sh
```

✅ **Using systemd timers:**

For more complex scheduling, `systemd` timers can be used instead of cron.