

backup.sh

```
#!/bin/bash
# backup.sh
# Script to back up all .txt files in current folder into backup/ with timestamp

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

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

# Find all .txt files and copy them 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

echo "☑ Backup completed!"
```

LAB4.md – File & Backup Automation

Assignment 4 – File & Backup Automation

Objective

To automate the process of **backing up all .txt files** into a **backup/** directory with a unique timestamp.

1. Script Explanation (backup.sh)

```
#!/bin/bash
```

 Tells the system to run the script with **bash**.

```
mkdir -p backup
```

↳ Creates a folder named `backup` if it doesn't exist (`-p` prevents errors).

```
timestamp=$(date +"%Y%m%d_%H%M%S")
```

↳ Stores the current date and time in `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
```

↳ Loops through all `.txt` files, checks if file exists, and copies them into `backup/` with timestamp added to filename.

```
echo "☑ Backup completed!"
```

↳ Displays a completion message.

2. Example Run

► Before

```
$ ls
notes.txt  report.txt  backup.sh
```

► Run Script

```
$ ./backup.sh
Backed up: notes.txt -> backup/notes_20250909_212300.txt
Backed up: report.txt -> backup/report_20250909_212300.txt
☑ Backup completed!
```

► After

```
$ ls backup
notes_20250909_212300.txt
report_20250909_212300.txt
```

3. Extra Questions

◊ Q1: Difference between `cp`, `mv`, and `rsync`?

- `cp` → Copies files/directories. Source remains unchanged.
- `mv` → Moves or renames files/directories. Source is removed after move.
- `rsync` → Advanced tool for syncing files/directories (local & remote), supports incremental backups.

◊ Q2: How to schedule scripts automatically?

Use **cron jobs** (Linux scheduler).

1. Open cron editor:

```
crontab -e
```

2. Add a schedule (example: run every day at 2 AM):

```
0 2 * * * /path/to/backup.sh
```

3. Save and exit. Cron will run script automatically.

Conclusion

- Created `backup.sh` to back up `.txt` files with timestamps.
- Tested successfully with example `.txt` files.
- Learned about **file management commands (`cp`, `mv`, `rsync`)** and **automation using cron**.

Deliverables:

- `backup.sh` (script)
- `LAB4.md` (report)

- PDF version for submission terminal screenshots

```
chayanika@drishti-VirtualBox:~$ touxh projects.txt
Command 'touxh' not found, did you mean:
  command 'touch' from deb coreutils (9.4-3ubuntu6.1)
Try: apt install <deb name>
chayanika@drishti-VirtualBox:~$ touch linux.txt
chayanika@drishti-VirtualBox:~$ ./backup.sh
bash: ./backup.sh: Permission denied
chayanika@drishti-VirtualBox:~$ chmod 777 backup.sh
chayanika@drishti-VirtualBox:~$ ./backup.sh
chayanika@drishti-VirtualBox:~$ nano backup.sh
chayanika@drishti-VirtualBox:~$ ./backup.sh
chayanika@drishti-VirtualBox:~$ nano backup.sh
chayanika@drishti-VirtualBox:~$ chmod 777 backup.sh
chayanika@drishti-VirtualBox:~$ ./backup.sh
chayanika@drishti-VirtualBox:~$ nano backup.sh
chayanika@drishti-VirtualBox:~$ chmod 777 backup.sh
chayanika@drishti-VirtualBox:~$ ./backup.sh
Backup script running...
./backup.sh: line 4: exho: command not found
chayanika@drishti-VirtualBox:~$ nano backup.sh
chayanika@drishti-VirtualBox:~$ ./backup.sh
Backup script running...
Done!
chayanika@drishti-VirtualBox:~$ nano backup.sh
chayanika@drishti-VirtualBox:~$ ./backup.sh
Backup script running...
Done!
Backed up backup-linux.txt -> backup/backup-linux_20251118_132014.txt
Backed up file.txt -> backup/file_20251118_132014.txt
Backed up linux.txt -> backup/linux_20251118_132014.txt
Backed up new.txt -> backup/new_20251118_132014.txt
chayanika@drishti-VirtualBox:~$ █
```