LINUX STARTER KIT

🛠️ Lab 5 – Starter Kit & Automation

Objective: Build a starter project environment automatically using a Bash script.

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

🎯 Purpose

The script will:

1. Create a structured project environment with folders:
   project/ scripts/ docs/ data/
2. Add placeholder `README.md` files in each folder.
3. Print a success message: **"Starter Kit Ready!"**

📜 Script Code

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

# Create project folders
mkdir -p project/scripts project/docs project/data

# Add placeholder README.md files
echo "# Scripts Folder" > project/scripts/README.md
echo "# Documentation Folder" > project/docs/README.md
echo "# Data Folder" > project/data/README.md
echo "# Main Project Folder" > project/README.md

# Success message
echo "Starter Kit Ready!"
```

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

# Create project folders
mkdir -p project/scripts project/docs project/data

# Add placeholder README.md files
echo "# Scripts Folder" > project/scripts/README.md
echo "# Documentation Folder" > project/docs/README.md
echo "# Data Folder" > project/data/README.md
echo "# Main Project Folder" > project/README.md

# Success message
echo "Starter Kit Ready!"
```

📌 Task 2: Example Run

Command:

```
./starter_kit.sh
```

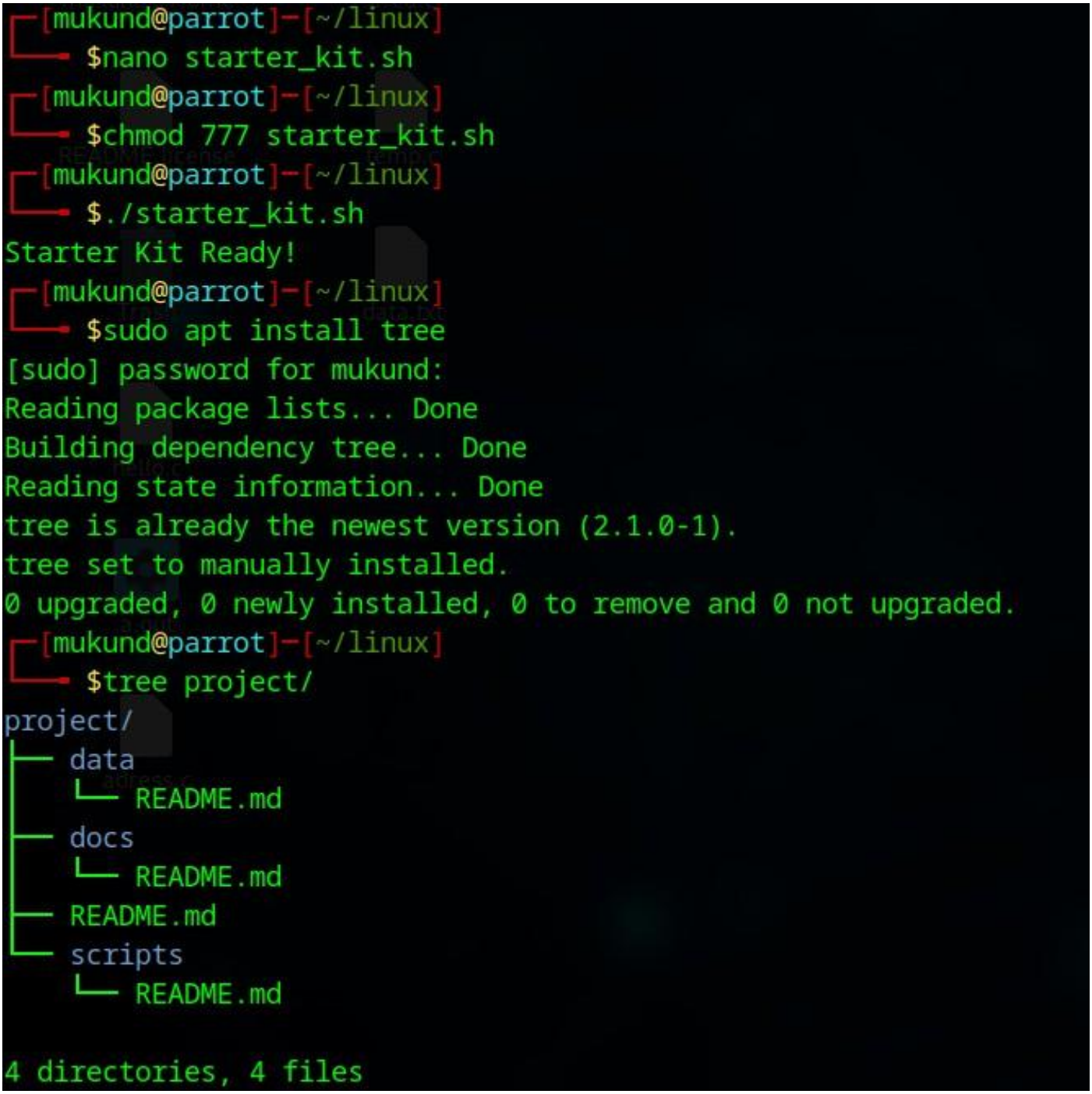
Output:

```
Starter Kit Ready!
```

Resulting Folder Structure:

```
project/
├── README.md
├── scripts/
│   └── README.md
├── docs/
│   └── README.md
└── data/
    └── README.md
```

📝 NOTE - FIRST INSTALL TREE USING `sudo apt install tree` TO ACCESS THE RESULTING FOLDER STRUCTURE AND THEN RUN THE COMMAND `tree project/` TO VIEW IT


[mukund@parrot]-[~/linux]
$nano starter_kit.sh
[mukund@parrot]-[~/linux]
$chmod 777 starter_kit.sh
[mukund@parrot]-[~/linux]
$./starter_kit.sh
Starter Kit Ready!
[mukund@parrot]-[~/linux]
$sudo apt install tree
[sudo] password for mukund:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
tree is already the newest version (2.1.0-1).
tree set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
[mukund@parrot]-[~/linux]
$tree project/
project/
├── data
│ └── README.md
├── docs
│ └── README.md
├── README.md
└── scripts
└── README.md
4 directories, 4 files

.

❓ Extra Questions

Q1: What does `mkdir -p` do?

- `mkdir -p` creates directories **recursively**.
- It ensures that parent folders are created if they don't exist, and it does **not throw an error** if the folder already exists.

Q2: Why is automation useful in DevOps?

- ✅ **Consistency:** Reduces human error by ensuring the same setup every time.
- ⚡ **Speed:** Saves time by automating repetitive setup tasks.
- 📦 **Scalability:** Makes it easy to replicate environments across multiple systems
- 🔁 **Integration:** Automation is a key principle of DevOps, enabling CI/CD pipelines and faster deployments.