Here is the **change\_permissions.sh** script that changes the permissions of all objects (files and directories) inside **"my\_folder"** to -rw-r--r-- using symbolic notation (chmod u=rw,go=r), avoiding octal notation (644).

**Bash Script (change\_permissions.sh)**

bash

CopyEdit

#!/bin/bash

# Define the target directory

TARGET\_DIR="my\_folder"

# Check if the directory exists

if [ ! -d "$TARGET\_DIR" ]; then

echo "Error: Directory '$TARGET\_DIR' does not exist."

exit 1

fi

# Change permissions of all files and directories using symbolic mode

find "$TARGET\_DIR" -exec chmod u=rw,go=r {} \;

echo "Permissions changed to -rw-r--r-- for all objects in '$TARGET\_DIR'."

**Explanation:**

1. **Defines** "my\_folder" as the target directory.
2. **Checks if "my\_folder" exists**, exits with an error if not.
3. **Uses find with chmod u=rw,go=r** to change:
   * **User (owner)** → Read (r), Write (w)
   * **Group & Others** → Read (r)
4. **Prints a confirmation message**.

**Usage:**

1. **Make the script executable**:

bash

CopyEdit

chmod +x change\_permissions.sh

1. **Run the script**:

bash

CopyEdit

./change\_permissions.sh

**Example Output (ls -l my\_folder)**

**Before Running the Script**

bash

CopyEdit

$ ls -l my\_folder

total 0

-rwxrwxrwx 1 user user 0 Feb 16 12:00 file1.txt

-rwxr-xr-x 1 user user 0 Feb 16 12:00 script.sh

drwxrwxrwx 2 user user 4096 Feb 16 12:00 subdir

**After Running the Script**

bash

CopyEdit

$ ls -l my\_folder

total 0

-rw-r--r-- 1 user user 0 Feb 16 12:01 file1.txt

-rw-r--r-- 1 user user 0 Feb 16 12:01 script.sh

-rw-r--r-- 1 user user 0 Feb 16 12:01 subdir

**⚠️ Important Note**

* This script removes **execute (x) permissions** from **directories**, making them inaccessible (cd subdir will fail).
* If you need to **preserve directory access**, modify it like this:

bash

CopyEdit

find "$TARGET\_DIR" -type f -exec chmod u=rw,go=r {} \;

find "$TARGET\_DIR" -type d -exec chmod u=rwx,go=rx {} \;

* + This keeps directories accessible (drwxr-xr-x).

Let me know if you need any changes! 🚀

Here’s a Bash script called **manage\_apt.sh** that does the following with apt:

✅ Removes all **unused dependencies** using apt autoremove  
✅ Updates the **software package database** using apt update  
✅ Upgrades the **entire system** using apt upgrade  
✅ **Checks if the script is run as root** (to avoid permission errors)

**Bash Script (manage\_apt.sh)**

bash

CopyEdit

#!/bin/bash

# Check if the script is running as root

if [ "$EUID" -ne 0 ]; then

echo "Error: This script must be run as root (use sudo)."

exit 1

fi

echo "Starting APT package management tasks..."

# Uninstall all unused dependencies

echo "Removing unused dependencies..."

apt autoremove -y

# Update the software package database

echo "Updating package database..."

apt update -y

# Upgrade the entire system

echo "Upgrading installed packages..."

apt upgrade -y

echo "APT package management tasks completed successfully."

**Explanation of the Script**

1. **Checks if the script is run as root**:
   * If not (EUID != 0), it prints an error message and exits to avoid permission issues.
2. **Performs APT package management tasks**:
   * Runs apt autoremove -y → **Removes unused dependencies**
   * Runs apt update -y → **Updates the package database**
   * Runs apt upgrade -y → **Upgrades installed packages**
3. **Uses -y flag** to **automatically confirm prompts**.

**How to Use the Script**

1. **Make the script executable**:

bash

CopyEdit

chmod +x manage\_apt.sh

1. **Run the script with sudo (to avoid permission issues)**:

bash

CopyEdit

sudo ./manage\_apt.sh

**Example Output**

bash

CopyEdit

$ sudo ./manage\_apt.sh

Starting APT package management tasks...

Removing unused dependencies...

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

0 upgraded, 0 newly installed, 0 to remove, and 0 not upgraded.

Updating package database...

Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease

Hit:2 http://archive.ubuntu.com/ubuntu focal InRelease

...

Upgrading installed packages...

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

Calculating upgrade... Done

0 upgraded, 0 newly installed, 0 to remove, and 0 not upgraded.

APT package management tasks completed successfully.

**✅ Why This Script is Useful**

* **Prevents permission errors** by ensuring it runs as root.
* **Automates system maintenance** in one command.
* **Saves time** by handling **autoremove, update, and upgrade** in one go.

Would you like additional features like apt full-upgrade or apt clean? 🚀