Assignment 2]

1.Create a directory structure under /home/archive_project with subdirectories for docs, images, and scripts.

Using the mkdir command

The mkdir command is used to create directories in Linux. You can use it with the -p option to create nested directories, meaning it will create parent directories if they don't already exist.

Here is the command to create /home/archive_project with the subdirectories docs, images, and scripts:

sudo mkdir -p /home/archive_project/docs /home/archive_project/images /home/archive_project/scripts

- **sudo**:
- This command runs the command with superuser (administrator) privileges. You need superuser privileges to create directories at the system level (/home directory).
- You may be prompted for your password to confirm that you have permission to run this command.
- **mkdir**:
- This is the command used to create directories.
- **-p**: This option tells mkdir to create parent directories as needed. If /home/archive_project doesn't already exist, it will create the archive_project directory first, and then the docs, images, and scripts directories inside it.
- **/home/archive_project/docs**:
- This is the path where the first subdirectory (docs) will be created inside the archive_project directory.
- **/home/archive_project/images**:
- This is the path where the second subdirectory (images) will be created inside the archive_project directory.
- **/home/archive_project/scripts**:
- This is the path where the third subdirectory (scripts) will be created inside the archive_project directory.

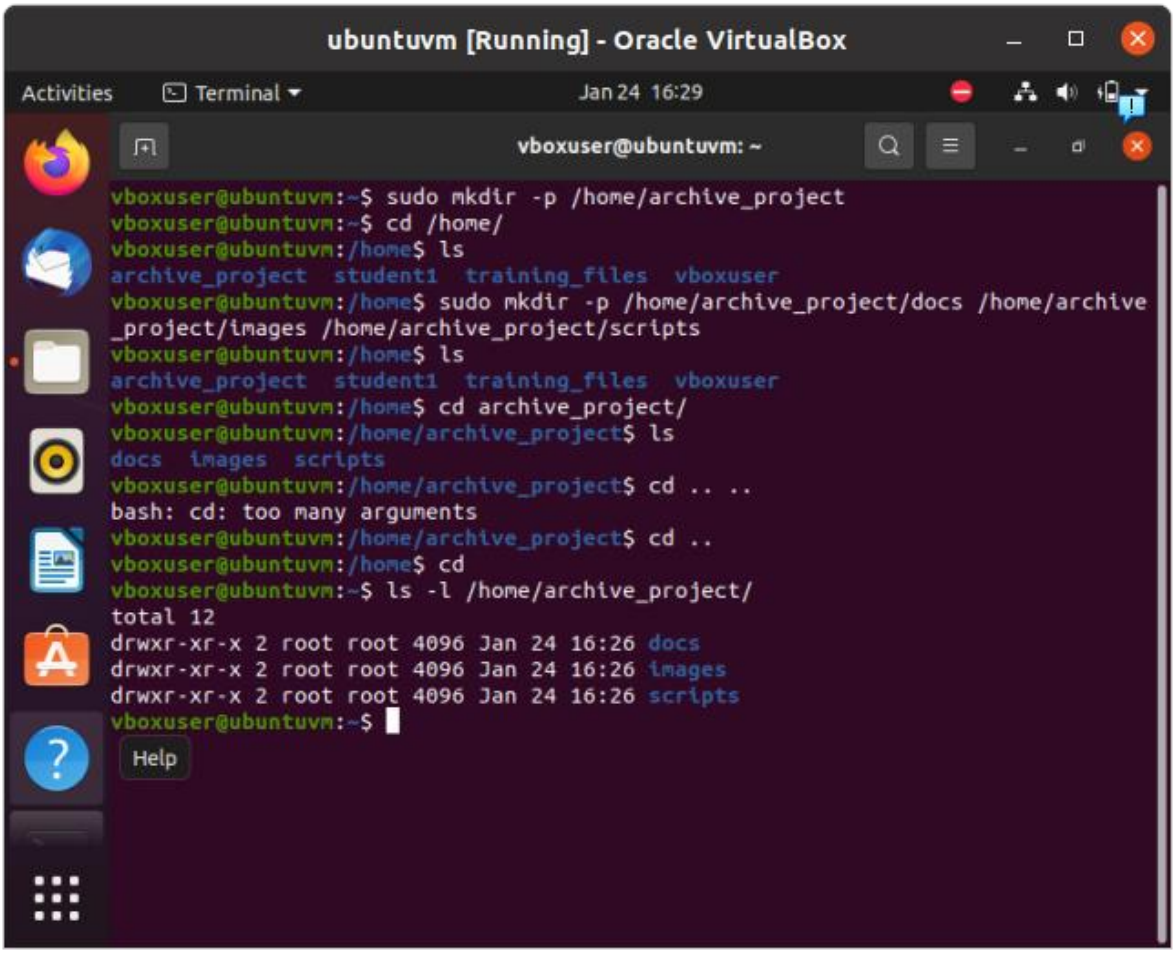
Check if /home/archive_project is created:

ls -l /home

Check the subdirectories inside /home/archive_project:

ls -l /home/archive_project

This will show you the docs, images, and scripts subdirectories inside archive_project.



2.Add some dummy text files and script files into their respective directories.

Add Dummy Text Files into the docs Directory

You'll want to create some text files in the docs directory. You can either create empty files or add some dummy content using the echo command.

Add Dummy Text Files to the docs Directory

Text files are usually just files that contain plain text data. In this case, we're adding dummy (empty) text files to the docs directory.

Navigate to the docs directory

cd /home/archive_project/docs

Create Dummy Text Files: To create empty text files, you can use the touch command. The touch command is used to create an empty file if it doesn't already exist.

Run this command to create two dummy text files inside the /home/archive_project/docs directory:

sudo touch /home/archive_project/docs/dummy_text_file1.txt /home/archive_project/docs/dummy_text_file2.txt

- **touch**: Creates an empty file (no content inside).
- **dummy_text_file1.txt** and **dummy_text_file2.txt**: These are the names of the files you're creating.

Verify the Files: After creating the files, you can list the contents of the directory to verify the files were created

ls -l /home/archive_project/docs

Add Dummy Image Files to the images Directory

We will now add dummy image files to the images directory. These will be placeholder image files

cd /home/archive_project/images

sudo touch /home/archive_project/images/image1.jpg /home/archive_project/images/image2.png

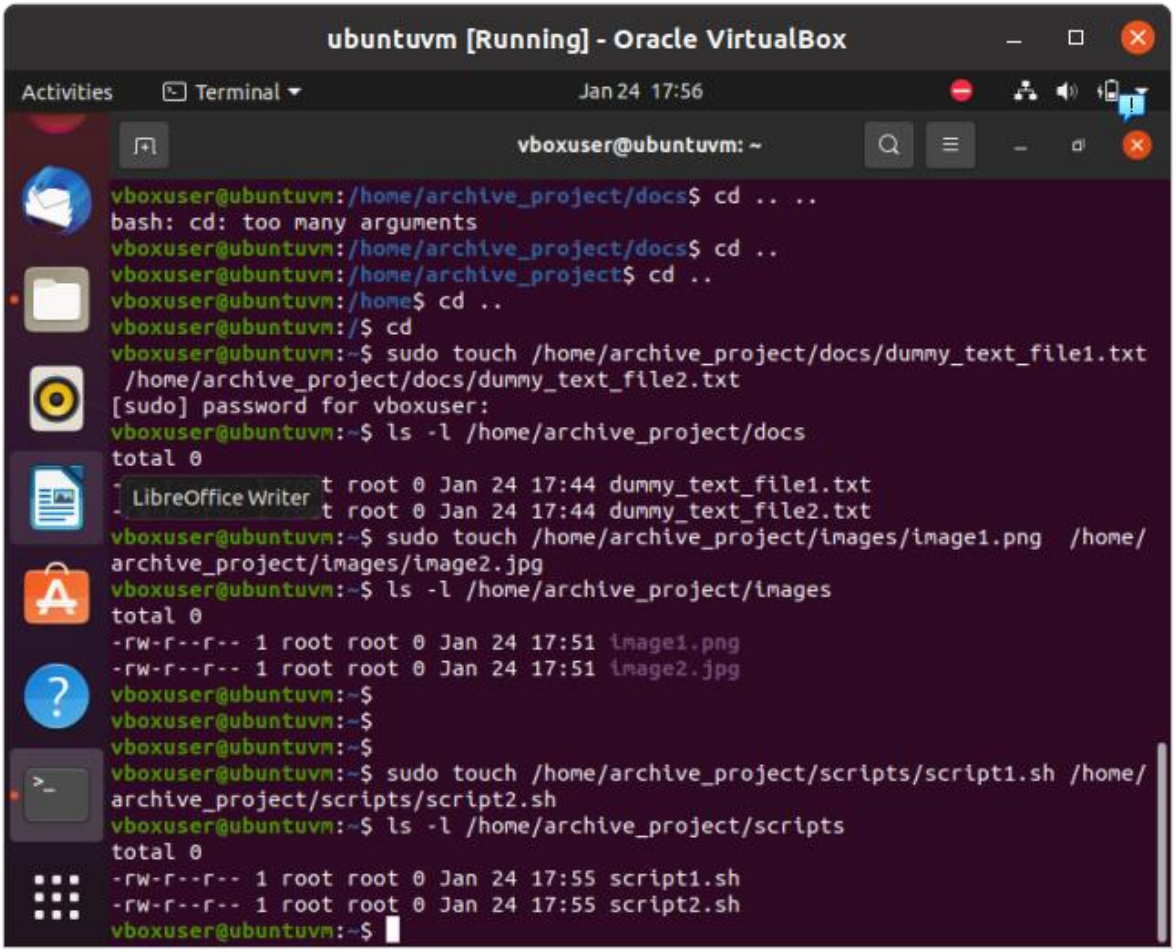
ls -l /home/archive_project/images

For the **scripts** directory, we will create empty shell script files. These files will have the .sh extension (commonly used for shell scripts)

cd /home/archive_project/scripts

sudo touch /home/archive_project/scripts/script1.sh /home/archive_project/scripts/script2.sh

ls -l /home/archive_project/scripts



3. Create a tar archive named project_backup.tar for the entire archive_project folder.

Create the project_backup.tar Archive

Use the tar command to create an archive

tar -cvf project_backup.tar /home/archive_project

- **tar**: The command used to create and manage tar archives.
- **-c**: Create a new archive.
- **-v**: Verbose mode (shows the files being added to the archive).
- **-f**: Specifies the name of the archive file to create.
- **project_backup.tar**: The name of the archive you're creating.
- **/home/archive_project**: The directory you want to archive.

sudo tar -cvf project_backup.tar /home/archive_project

Note: The sudo command is used to ensure that you have the necessary permissions to access all the files in the /home/archive_project directory, especially if some of the files or subdirectories are owned by a different user (like root).

Verify the Archive: After running the command, you can check that the tar archive was successfully created by listing the contents of the directory where you ran the command.

ls -l project_backup.tar