Feel free to use this template to structure your report. The text in this template describes what you should focus on in each section. When writing your report, replace this text with your own.

Overview

The requirements of this practical was to develop a shell script that would copy files and data from the sourcedata directory into new sub-directories within a new directory of the user's choosing within the current working directory. Specifically, the sub-directory names correspond with the titles of the trees' descriptions. Inside these sub-directories, the description of the appropriate tree, as well as the image (and copyright information of the image) was to be copied. Also, the trees data from the trees_public csv file was to be copied and stored individually in a new file called data.csv in each sub-directory.

I was able to fully meet all the requirements of this practical.

Design

My script can be roughly split into two structures: an if statement and a for loop.

The if statement was used to check if the target directory already exists, if not then it would create a new one.

The for loop iterates through all the files in the description sub-directory. It then extracts the title names from each of the trees' description files by using the **head** command to find the first line and the **gawk** command to select the 2nd element after setting the delimiter to ": ". This is then assigned to a new variable named title**.** It uses the title to name new sub-directories in the target directory.
Then, the names of the .txt files in the description sub-directory(e.g. sycamore from sycamore.txt) is used to search the image directory for their corresponding image and copyright files. These are copied into the appropriate sub-directory of the target directory.

Using **grep**, the trees_public csv file is searched for lines containing the description title. Initially, I wanted to use **gawk** compare if the 9th element (Popular Names) was equal to the description title but I was having issues implementing the comparison. I was able to use **grep** instead as I noticed that the popular names are only mentioned once per line and that all the strings in the Popular names field were surrounded in quotation marks. The latter is important in situations where the description title is also substring of a different Popular name(e.g. Sycamore and Sycamore 'cultivar'). Using regex I was use "title" as a pattern.

Testing

I tested by first creating a diagram describing the structure of the target directory.

Target Directory:

- Tree1_title
    - Tree1_name.txt
    - Tree1_name.jpg
    - Tree1_name-copyright.txt
    - data.csv
- Tree2_title

    - Tree2_name.txt
    - Tree2_name.jpg
    - Tree2_name-copyright.txt
    - data.csv
- Tree3_title
    - Tree3_name.txt
    - Tree3_name.jpg
    - Tree3_name-copyright.txt
    - data.csv

When I ran my script, I initially check to see if the tree structure of the target directory looked like the above. Making sure that the subdirectory names corresponded with the description title where TreeN_name.txt is the copied description file.

Once I perfected the structure, I made sure that the data.csv files contained the correct records. To do this I used the commands to compare the number of line between the expected output and the actual:

```
$ grep -i “\“title\”” trees_public.csv | wc -l
```

```
$ wc -1 data.csv
```

Note: The file paths have been shortened for simplicity

I also compared the number of lines with Microsoft excel’s find command. When these values were all equal, I knew that the files data was correct.

<u>Evaluation</u>

I believe I have successfully created a script that satisfies all the requirements. Most of the requirements were simple to implement, except for the data extraction part. However, I was able to find an alternative method than my initial plan, which allowed me to complete this practical.