# Exercise 2

**Name    :   Dharun S**

**Reg No :   73772114128**

**Dept    :   B.E_Computer Science and Engineering**

**College:   K S Rangasamy college of Technology**

# Screenshots

1.  **Run the status command. Notice how it tells you what branch you are in.**
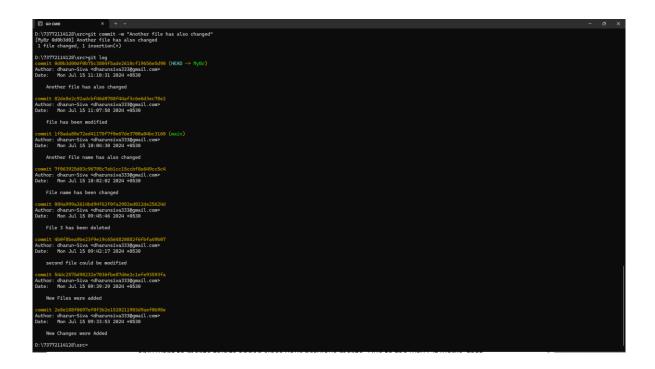


2.  **Use the branch command to create a new branch**.

**3. Use the checkout command to switch to it.**



**4. Make a couple of commits in the branch – perhaps adding a new file and/or editing existing ones.**
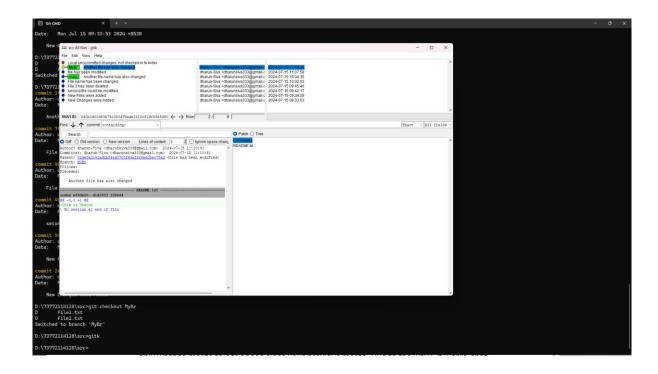
**5. Use the log command to see the latest commits. The two you just made should be at the top of the list.**
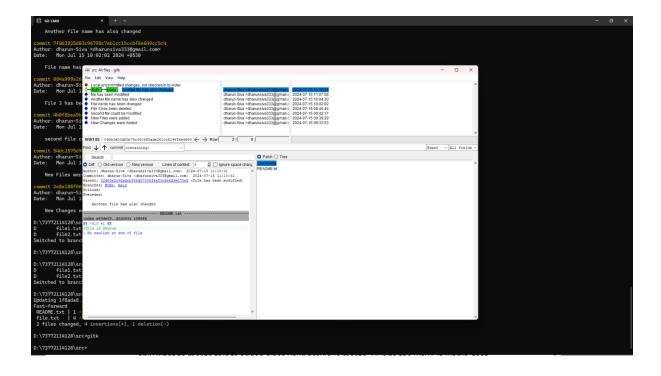


**6. Use the checkout command to switch back to the master branch. Run log again. Notice your commits don't show up now. Check the files also – they should have their original contents.**
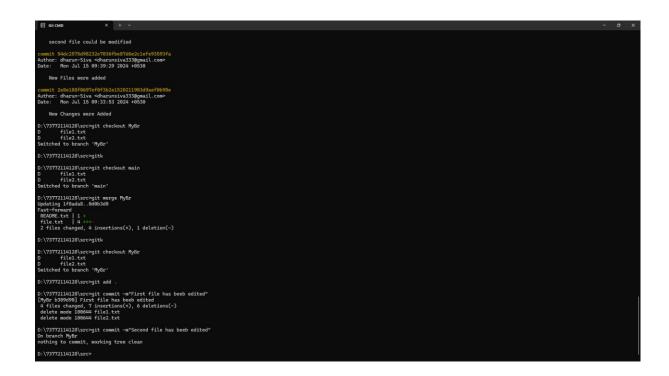
**7. Use the checkout command to switch back to your branch. Use gitk to take a look at the commit graph; notice it's linear**
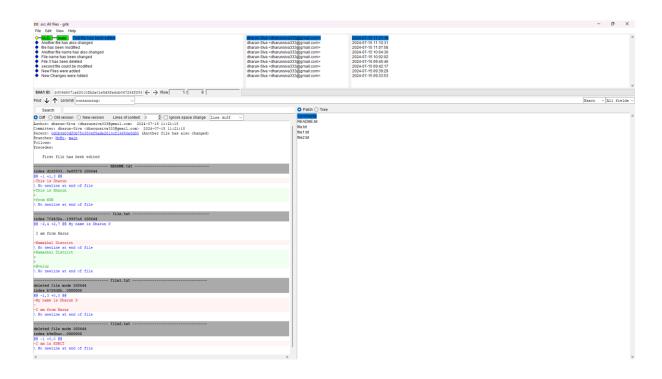


**8. Now checkout the master branch again. Use the merge command to merge your branch in to it. Look for information about it having been a fast-forward merge. Look at git log, and see that there is no merge commit.**

**9. Switch back to your branch. Make a couple more commits.**



**10. Switch back to master. Make a commit there, which should edit a different file from the ones you touched in your branch – to be sure there is no conflict.**

**11. Now merge your branch again. (Aside: you don't need to do anything to inform Git that you only want to merge things added since your previous merge. Due to the way Git works, that kind of issue simply does not come up, unlike in early versions of Subversion.)**



**12. Look at git log. Notice that there is a merge commit. Also look in gitk. Notice the DAG now shows how things forked, and then were joined up again by a merge commit.**