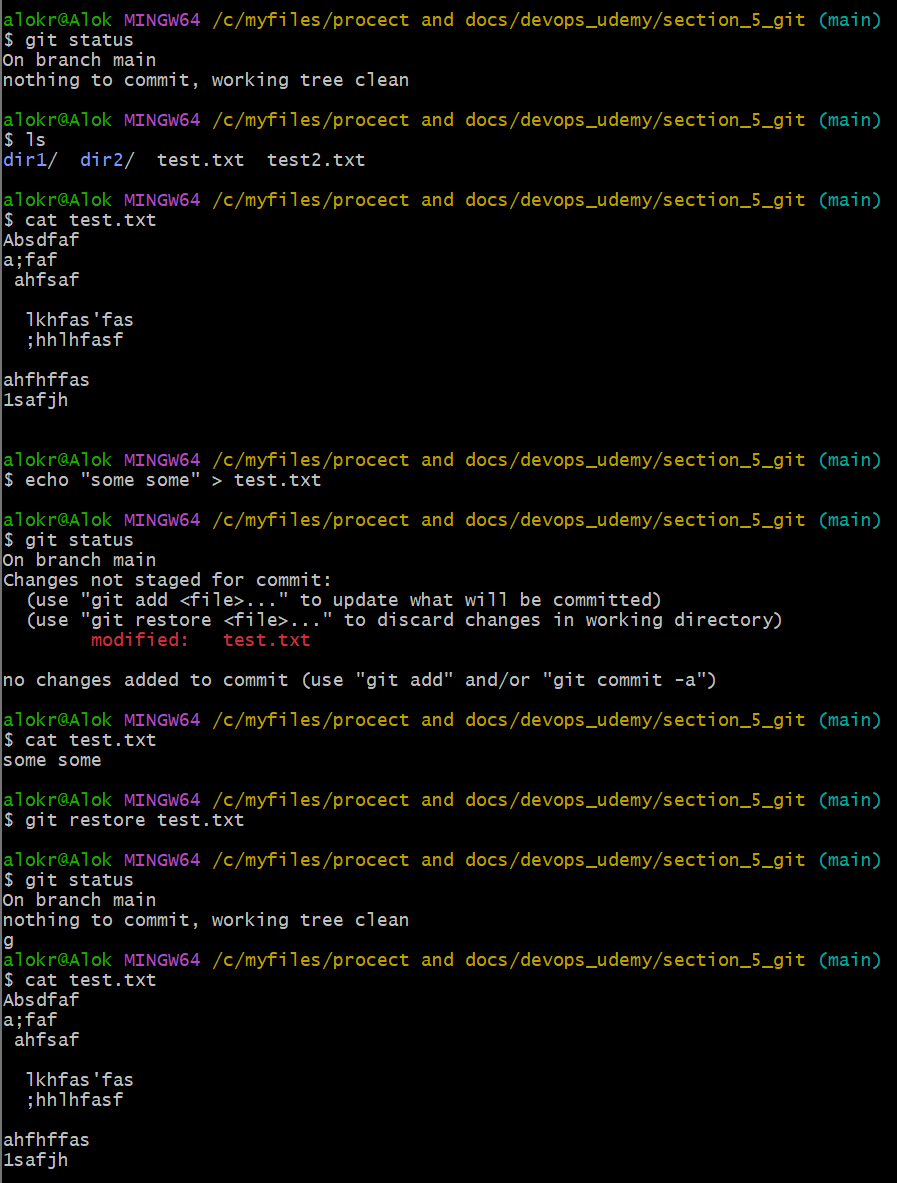
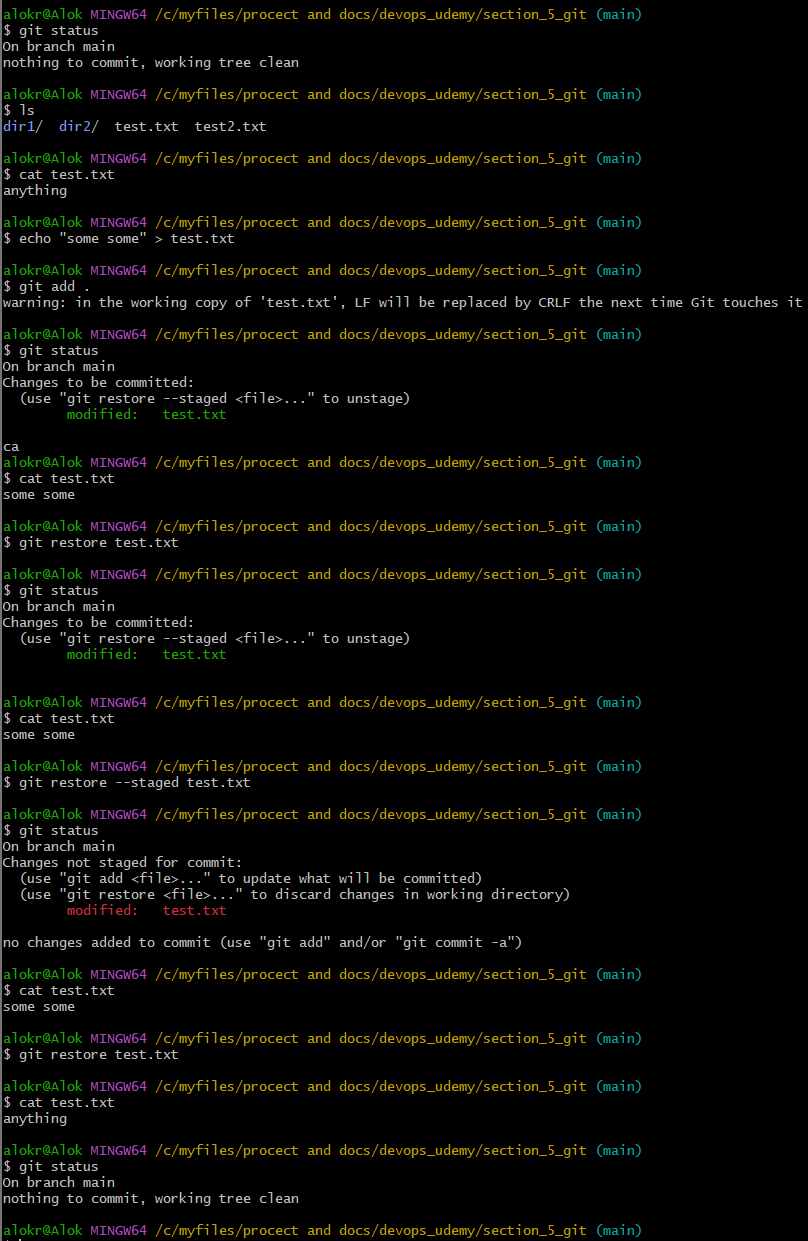
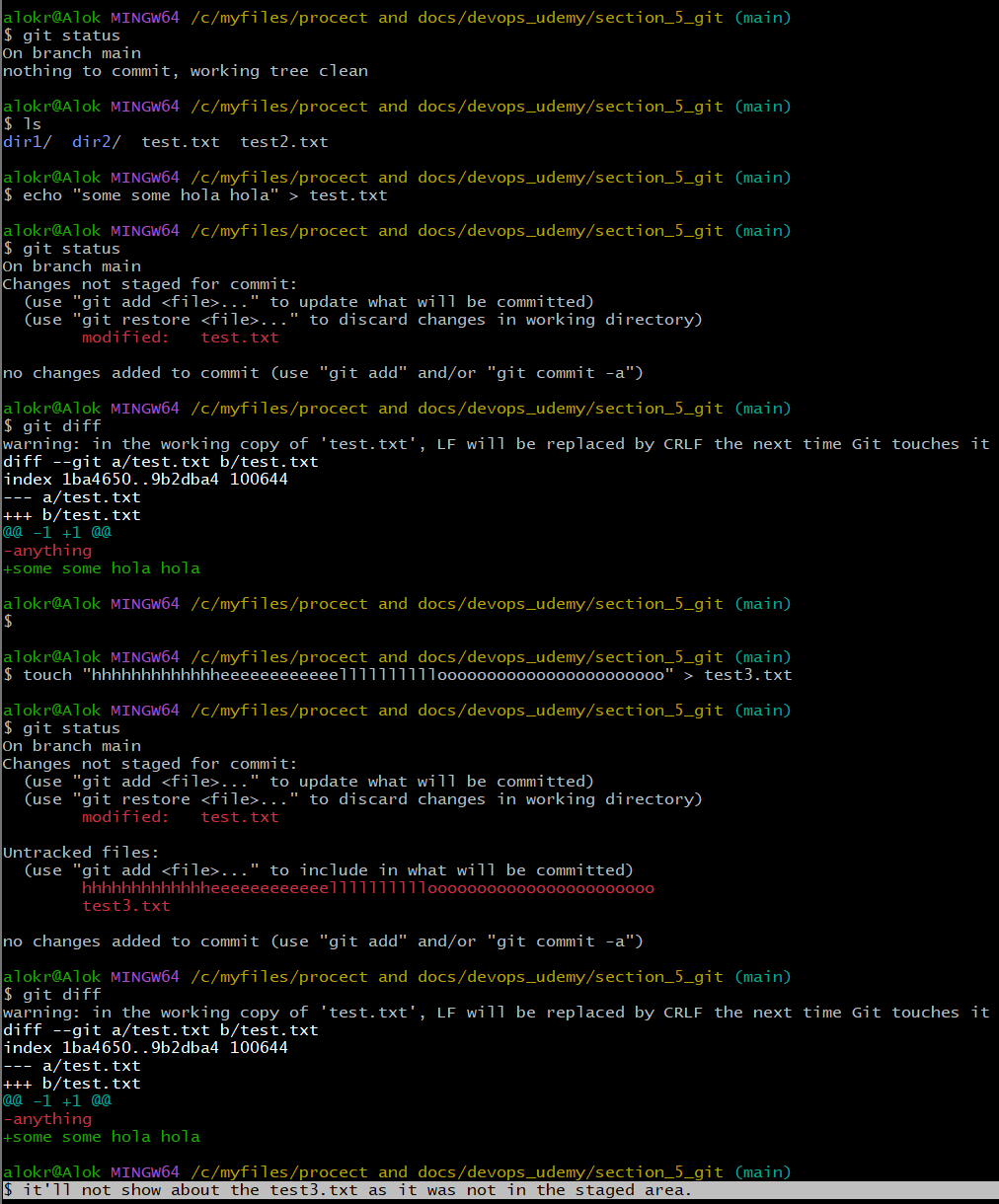
* **git push --all origin**
  + Push all the branches to the remote repository.
* **git checkout <filename>**
  + If you had changed one file and have not commited yet, then running this command will erase everything you updated and make it same as the last commited version.
  + It doesn’t effect any untracked file. (as git doesn’t know about this file)
  + If you have alreadyy staged the file (using git add) then you are executing this command, then it doesn’t effect this as the changes are already staged.
  + *Hence, if the file is not untracked & the changes are unstaged, then only the* ***git checkout <filename>*** *will reset this file to the last commit version.*
* **git restore <filename>**
  + Same as *git checkout <filename>.*
  + But it is recommended to use in stead of *git checkout <filename>.*
  + git checkout should be used for switching among the branches only.
  + It has some additional features like you can rollback to a specific commit in the past. I.e.
    - ***--source=<commit>***
  + 
  + ***git restore --staged <filename>*** : It unstaged the file from staging area.
    - Let the file is already committed, now you have changed (added or deleted) some lines in side the file, and staged it (git add command). Now you are using this command *git restore --staged <filename>*, it’ll only unstage the current updation (means the added/deleted lines) as it the file was pushed to tracking stage(git add) in the previous commit.
    - Ex:
      * The file is committed.
      * Now changed come lines. And executed *git add .*
      * Now executed *git restore --staged <filename>*
      * Now the changed in side the fill will be there, but unstaged.
      * Now execute *git restore <filename>*
      * Now the changes done in this file will be no more.
    - 
* **git diff**
  + It shows the differences between *working directory* and *staging area*.
  + 
* **git diff --cached**
  + It shows the difference between *staging area* and *last commit (HEAD).*