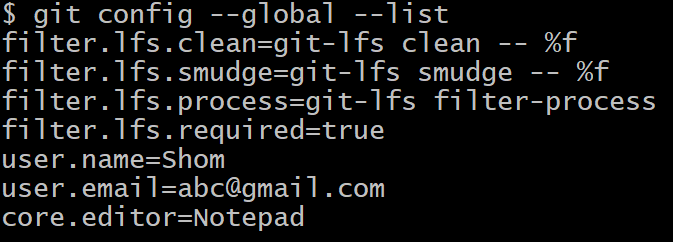
**Git**

**Git HOL-1**

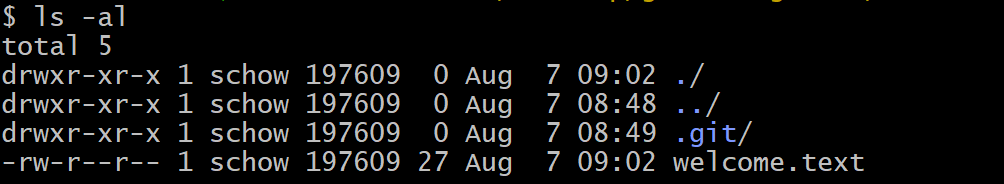
* **Setup your machine with Git Configuration**
* **Integrate notepad++.exe to Git and make it a default editor**
* **Add a file to source code repository**

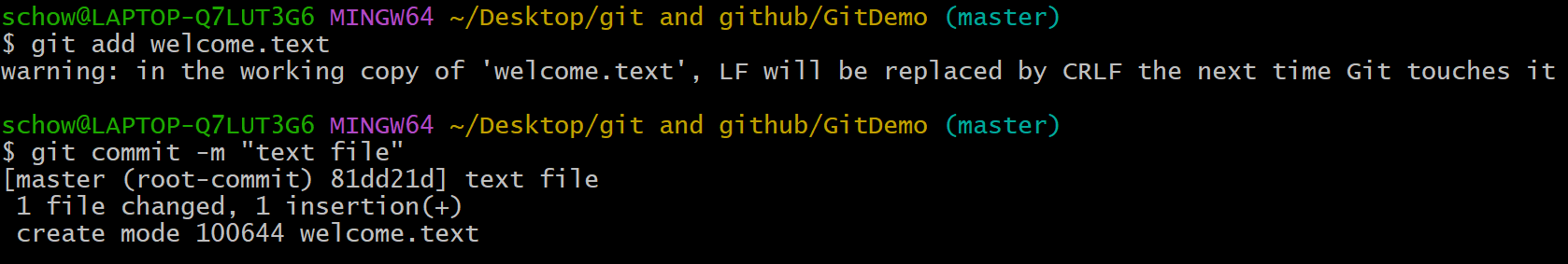
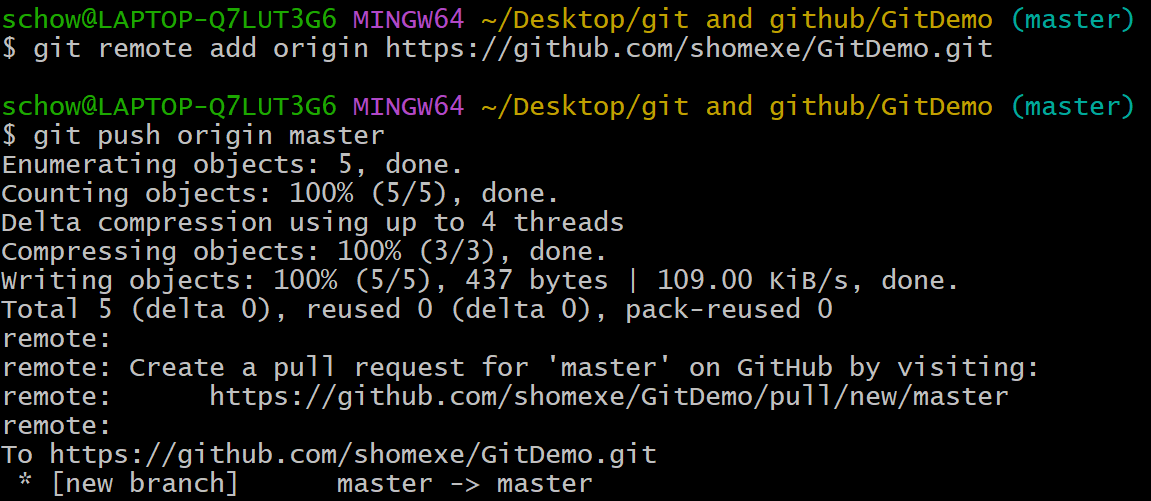
🡺solution:

Setting global configuration



Committing a file

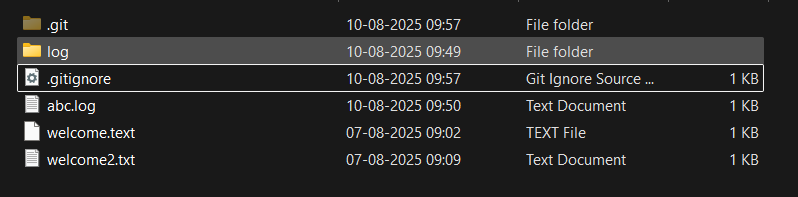
**$ echo "welcome to version control" >> welcome.text**



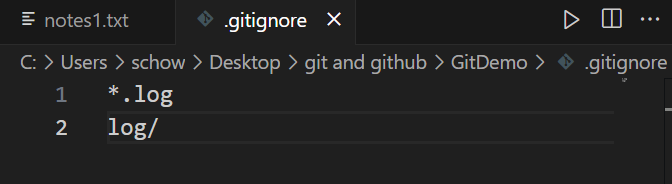
**Git HOL-2**

* **Implement git ignore command to ignore unwanted files and folders**

🡺solution:

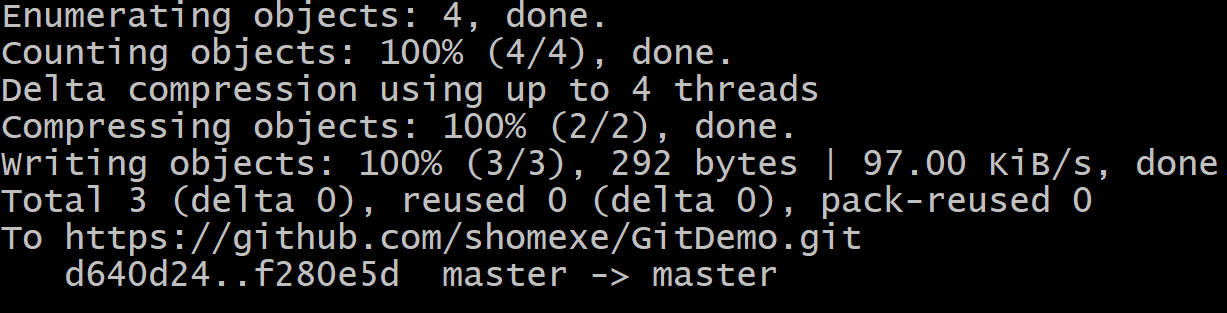
created **log folder** and a text file inside it then created a file with **.log** extension

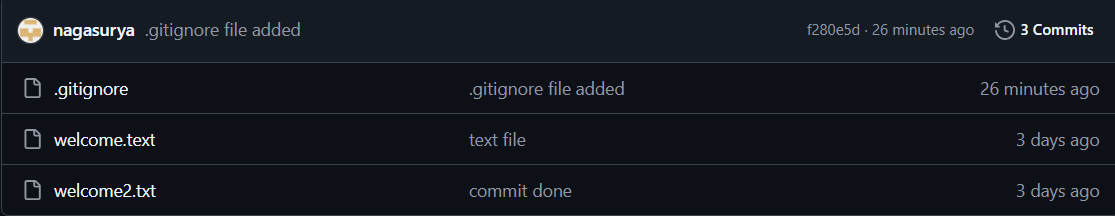
added .log file and log folder inside **.gitignore** to be ignored by git when new commit is done



**$ git add .**

**$ git commit -m “.gitignore file added”**

****



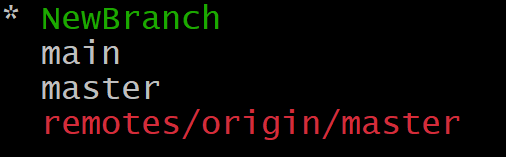
**Git HOL-3**

* **Construct a branch, do some changes in the branch, and merge it with master (or trunk)**

🡺 solution:

**$ git checkout -b NewBranch**

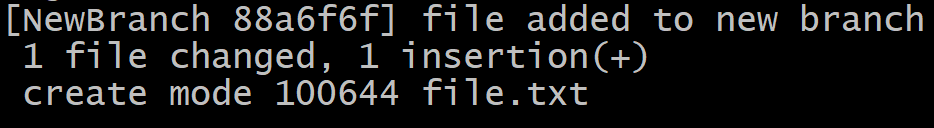
**$ git branch -a**

****

**$ echo "...file content" >> file.txt**

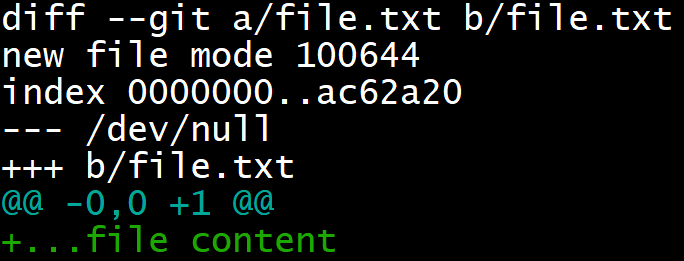
**$ git add .**

**$ git commit -m “file added to new branch”**

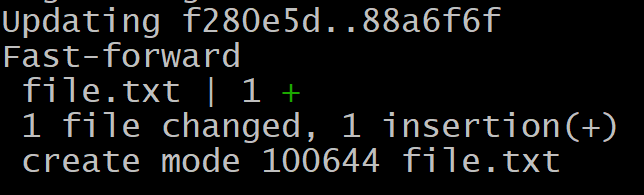


**$ git switch master**

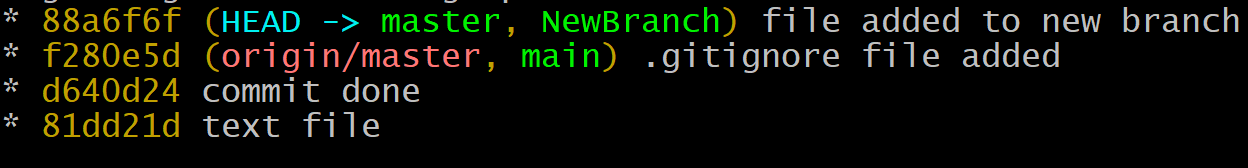
**$ git diff master..NewBranch**



**$ git merge NewBranch**



**$ git log –oneline –graph –decorate**



**$ git branch -d NewBranch**

**-----------------------------------------------------------------------------------------------------------------------------------**

**Git HOL-4**

* **Implement conflict resolution when multiple users are updating the trunk (or master) in such a way that it results into a conflict with the branch’s modification.**

🡺solution:

**$ git checkout -b GitWork**

**$ echo "hello world" > hello.xml**

**$ git add hello.xml**

**$ git commit -m "in branch GitWork"**

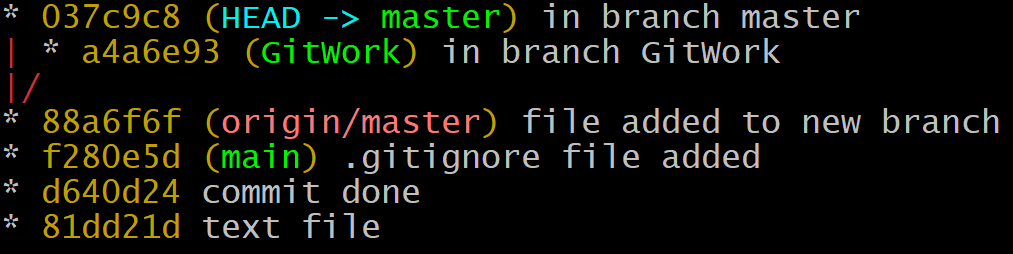
**$ git switch master**

**$ echo "hello world 2" > hello.xml**

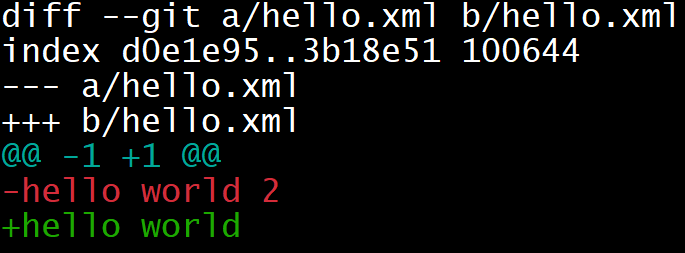
**$ git add .**

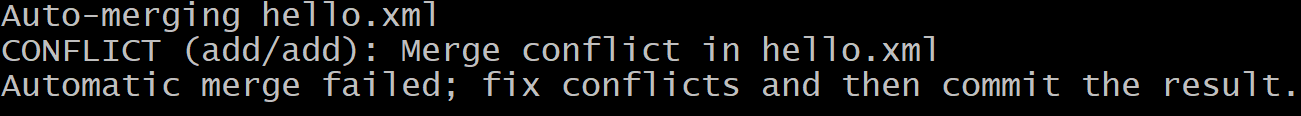
**$ git commit -m "in branch master"**

**$ git log --oneline --graph --decorate --all**

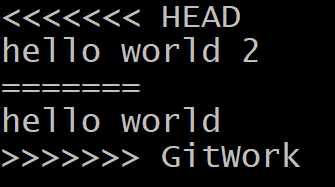
****

**$ git diff master GitWork**

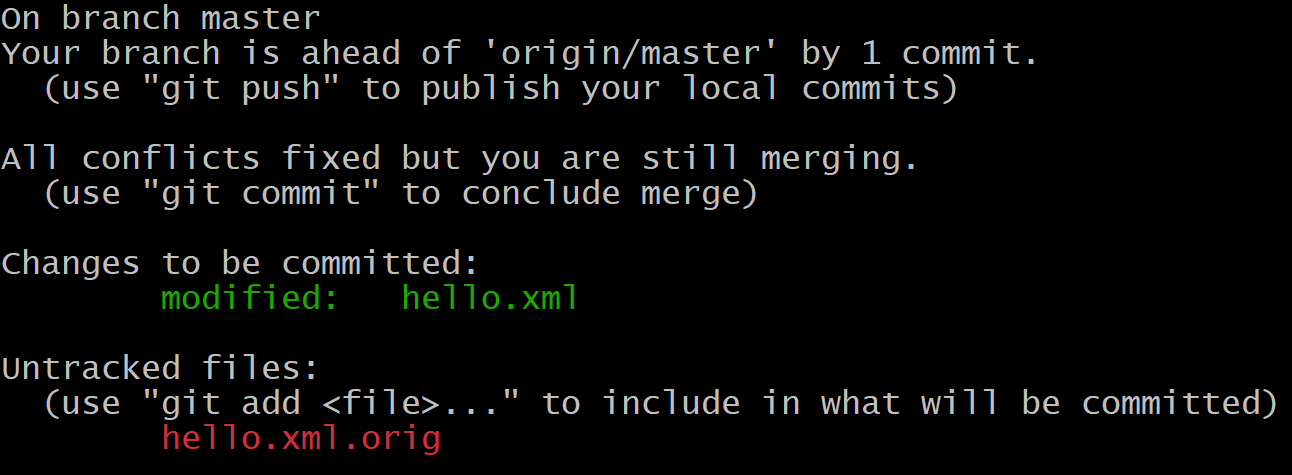


**$ git merge GitWork**

**$ cat hello.xml**



**$ git mergetool (used to resolve conflict by editing file and saving it)**

**$ git status**

**$ git add hello.xml**

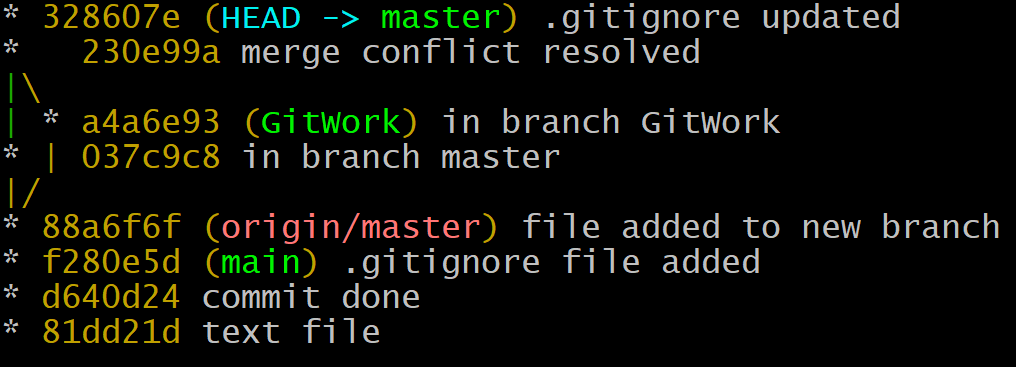
**$ git commit -m "merge conflict resolved"**

**$ echo "\*.orig" >> .gitignore**

**$ git add .gitignore**

**$ git commit -m ".gitignore updated"**

**$ git log --oneline --graph –decorate**



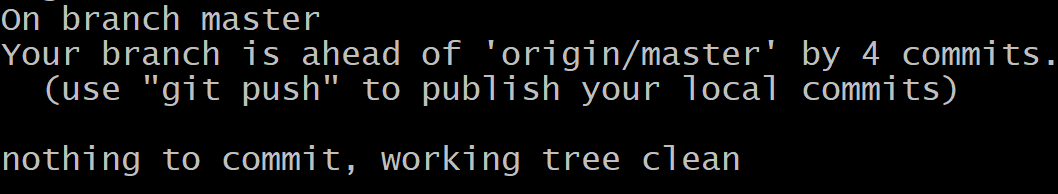
**$ git branch -d GitWork**

**Git HOL-5**

* **Execute steps involving clean up and push back to remote Git.**

🡺solution:

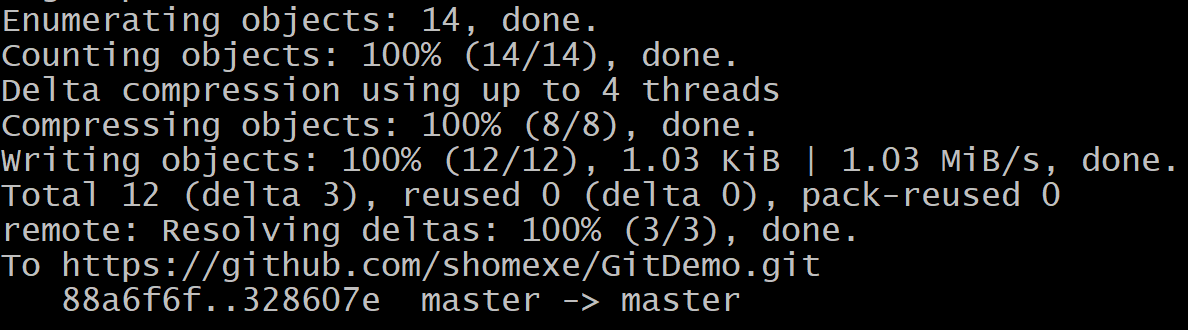
**$ git status**



**$ git branch -a**



**$ git push**



**Changes reflected on remote repository**

