LAB 3

GIT AND GITHUB

Fullname: Trương Đặng Trúc Lâm

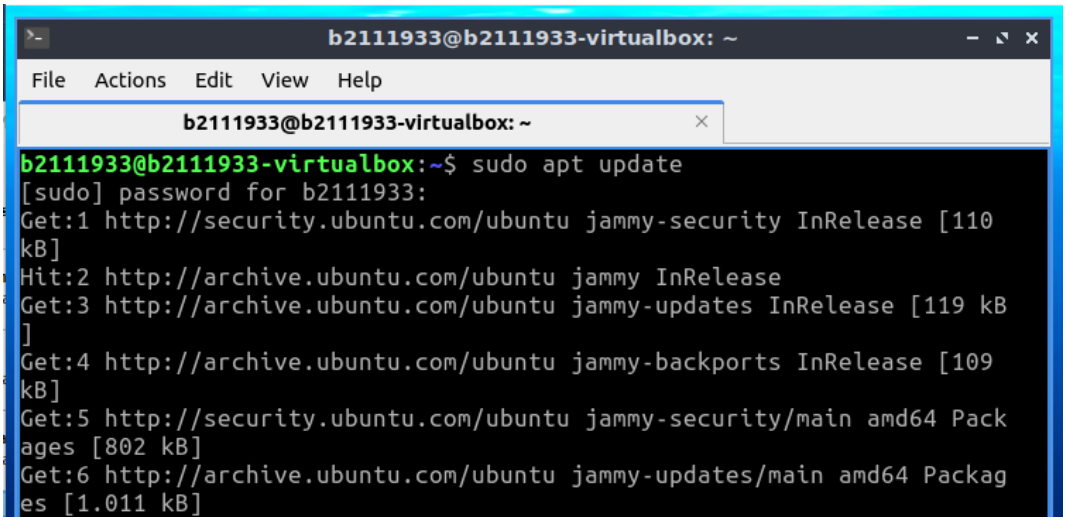
Student ID: B2111933

- Note: screenshots need to be clear and good-looking; submissions must be in PDF format.

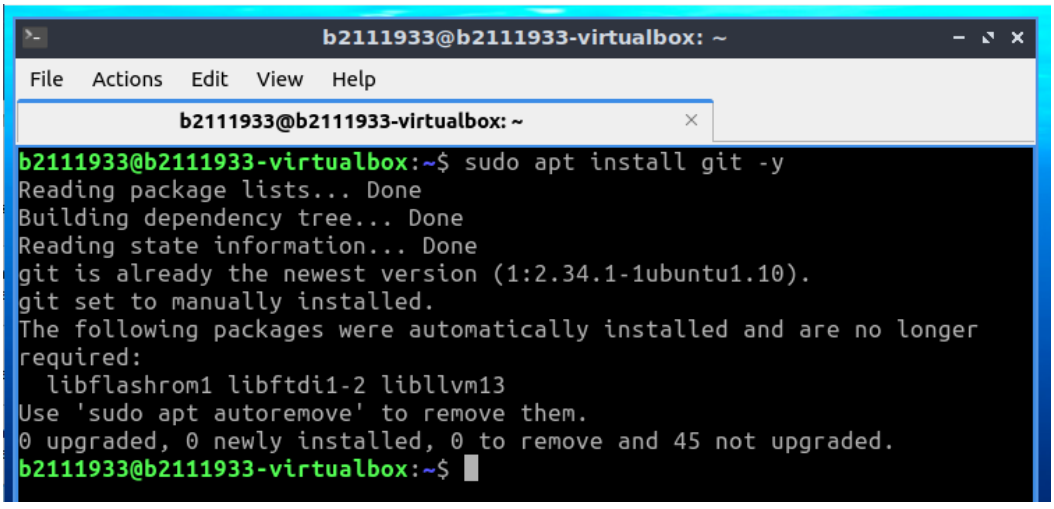
1. **Setup and making commits**

- Install and setup git on your computer (remember to set your name/email)

```
$sudo apt update ; sudo apt install git -y
```

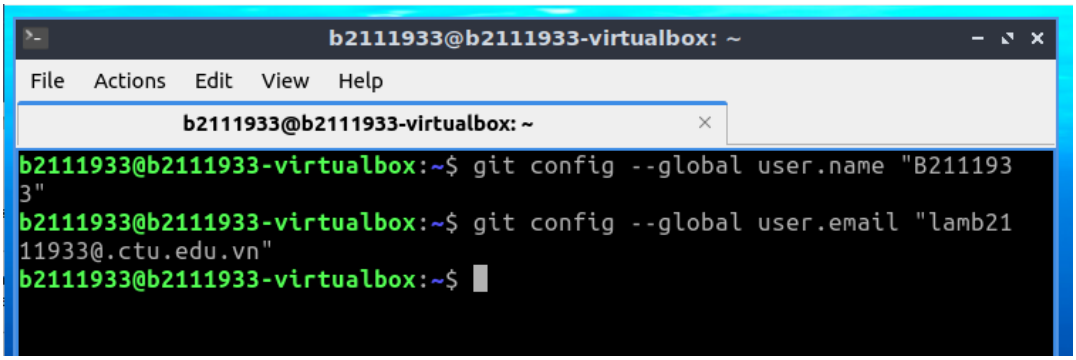


Updated successfully



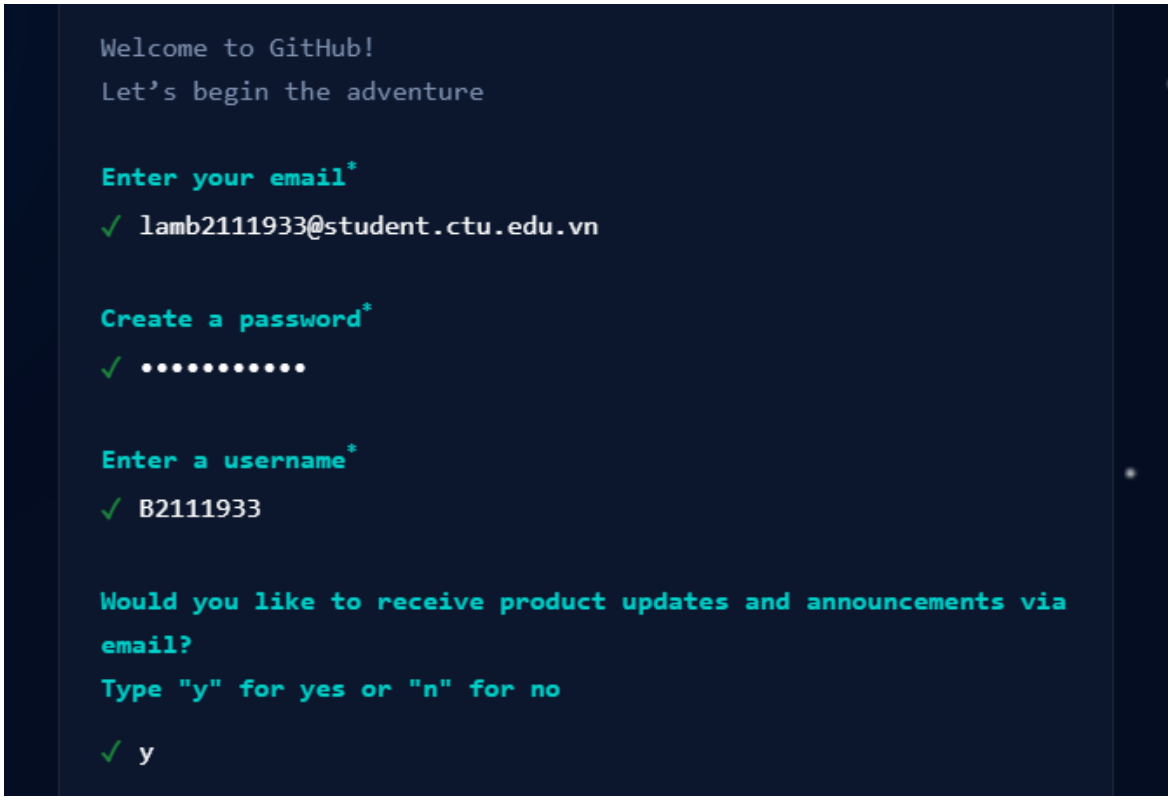
Installed **git** successfully

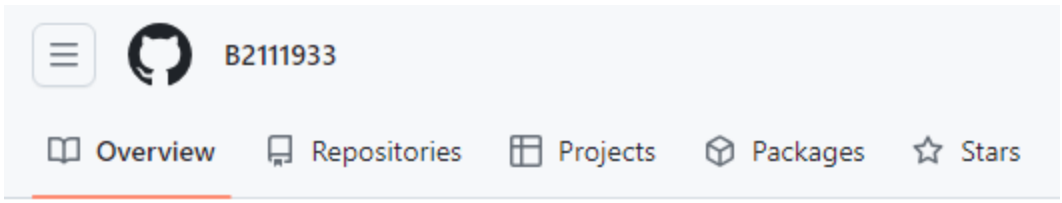
```
$git config --global user.name "Firstname Lastname"
$git config --global user.email "example@.ctu.edu.vn"
```



Finished configuration

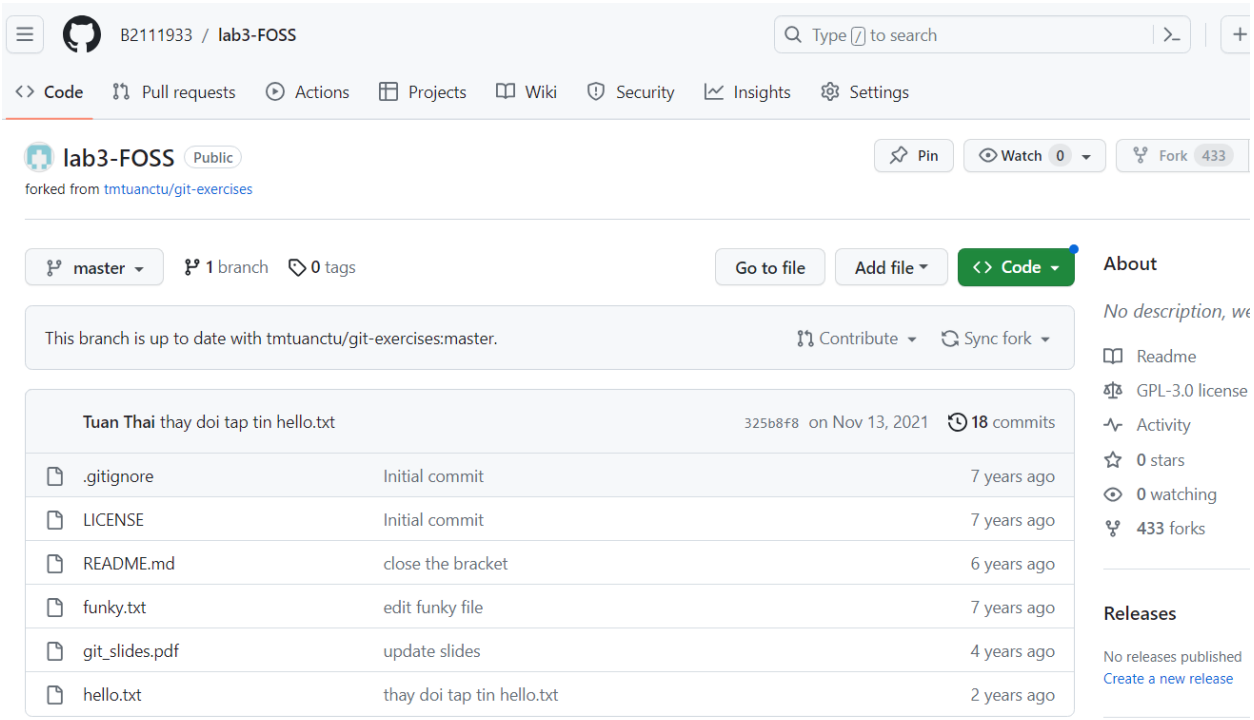
- Create an account (or login) to GitHub at https://github.com





GitHub registered

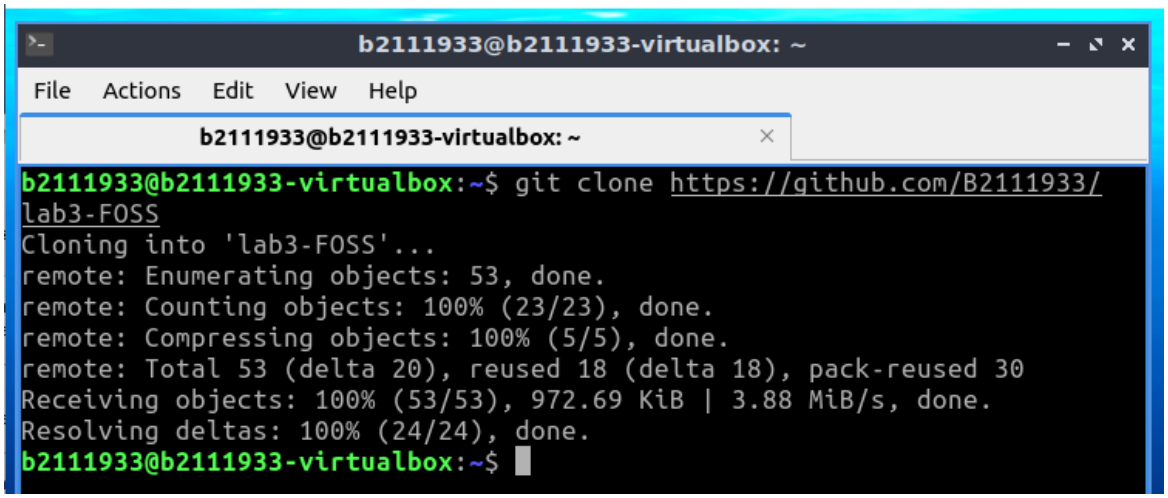
- Fork (Fork)this repository https://github.com/TuanThai/git-exercises.git to your Github account.



Forked repository **https://github.com/TuanThai/git-exercises.git** to account **B2111933**

- Clone the forked repository to your computer

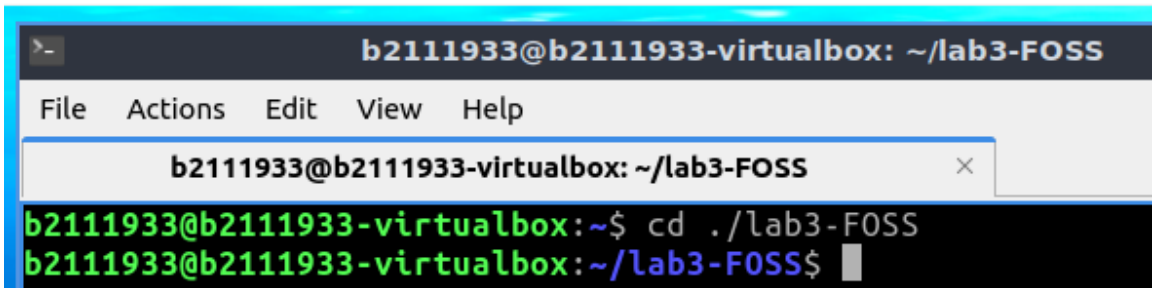
```
$git clone <url of your forked repository>
```



Cloned the forked repository to my computer successfully

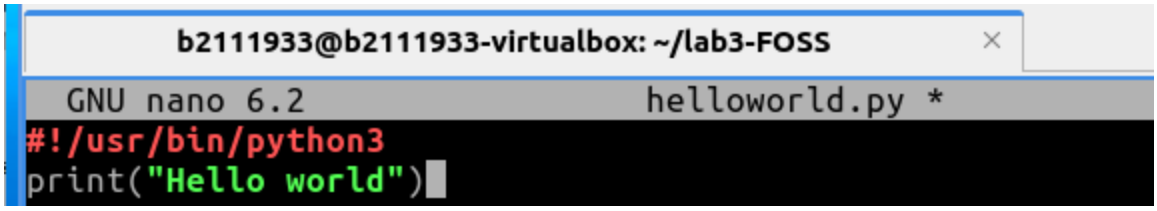
- Create and add a new file `helloword.py`

```
$cd ./git-exercises
```



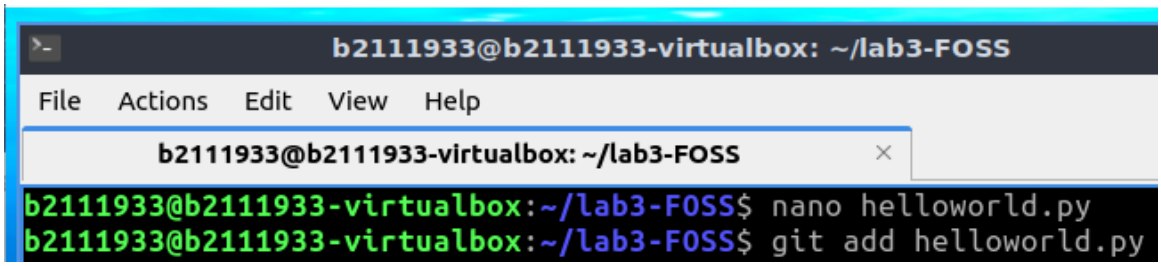
Changed the **working directory** to the **forked repository**

```
$nano helloword.py
     #!/usr/bin/python3
     print("Hello world")
```



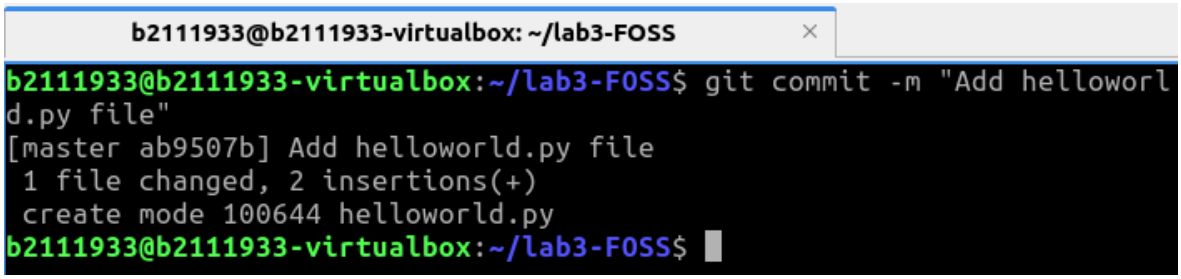
Created file **helloword.py**

```
$git add helloworld.py
```



Added file **helloword.py**

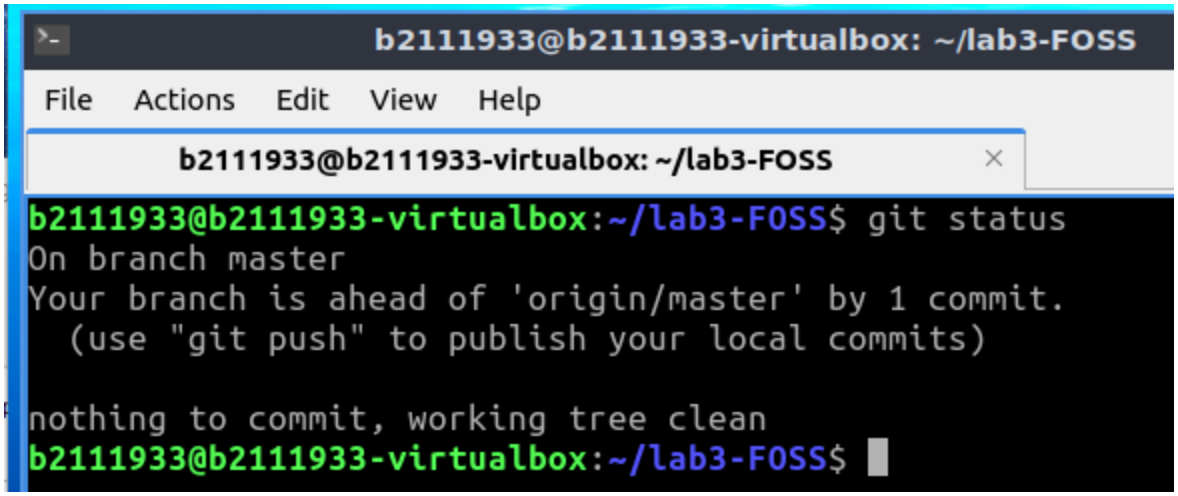
```
$git commit -m "Add hellworld.py file"
```



Committed file **helloword.py**

- Examine the state of your repo

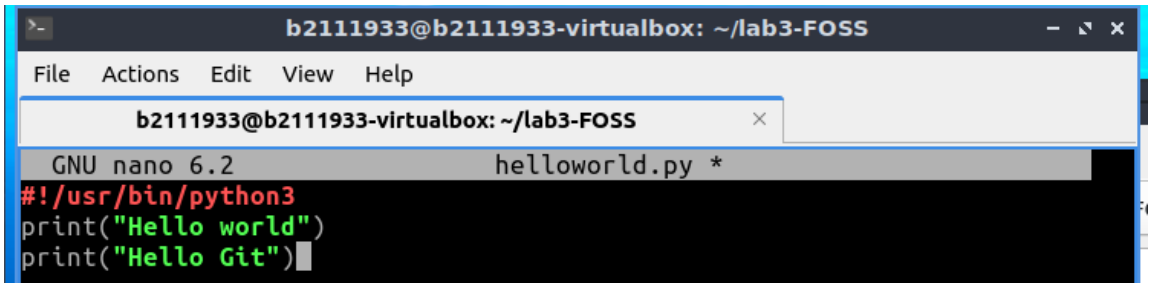
```
$git status
```



State of the repo

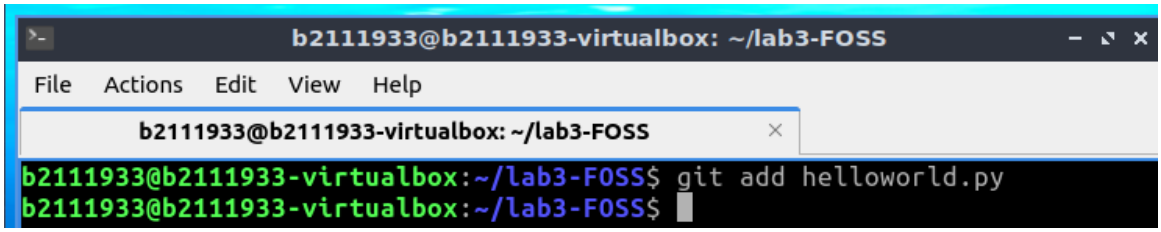
- Edit and save your new file, then add it to the staging area. Finally make a new commit with the edited file. At all stages use git status to see how your repository changes

```
$nano helloworld.py
    #!/usr/bin/python3
    print(“Hello world”)
    print(“Hello Git”)
```



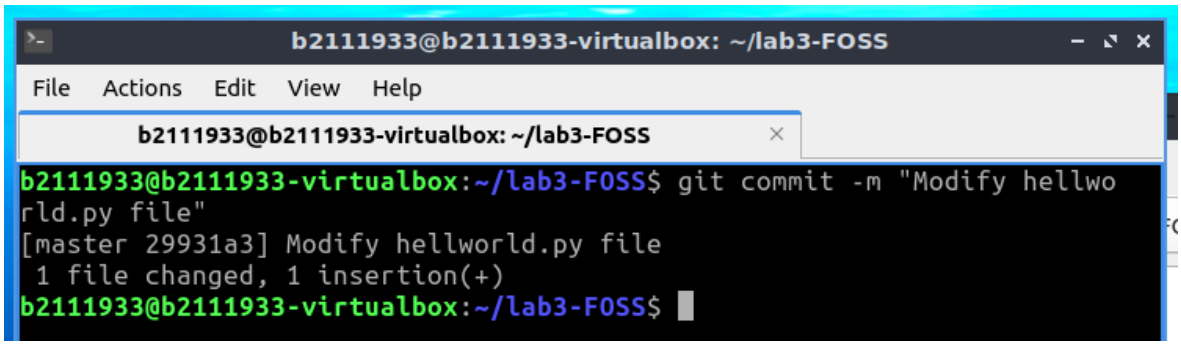
Edited file **helloworld.py**

```
$git add helloworld.py
```



Added the file to the staging area

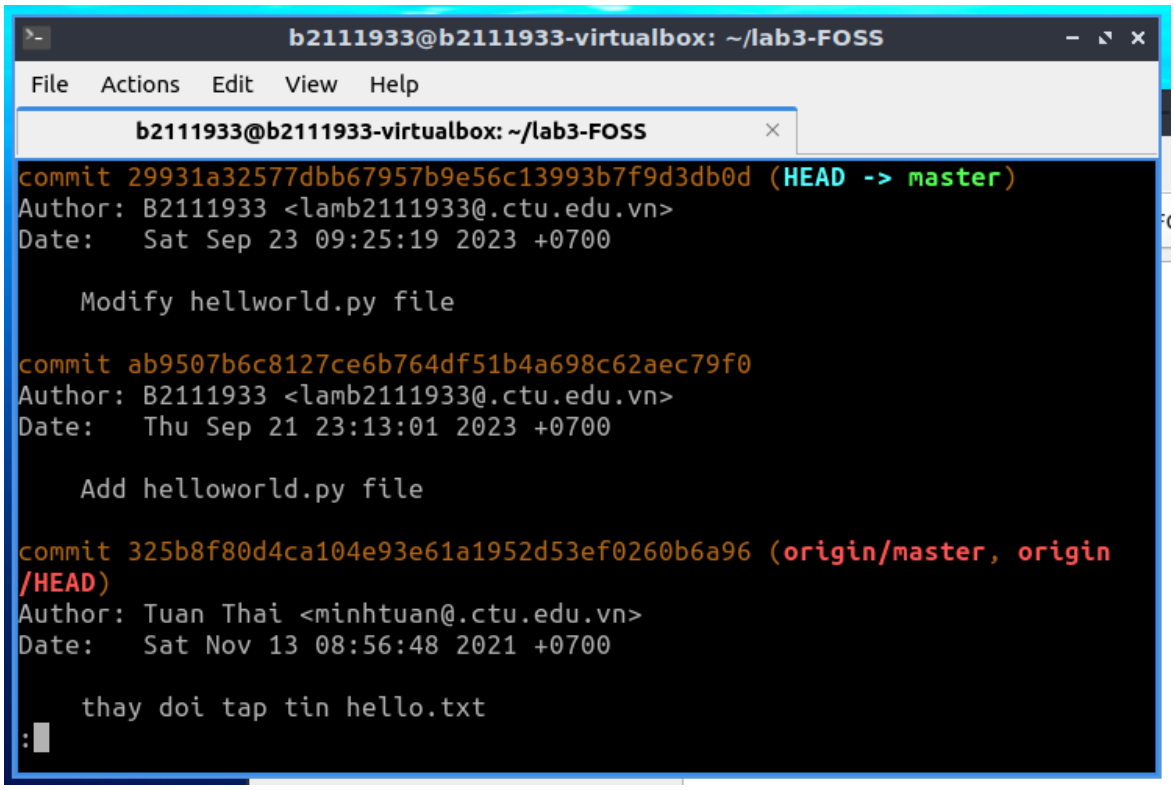
```
$git commit -m “Modify hellworld.py file”
```



Committed successfully

- View the log

```
$git log
```

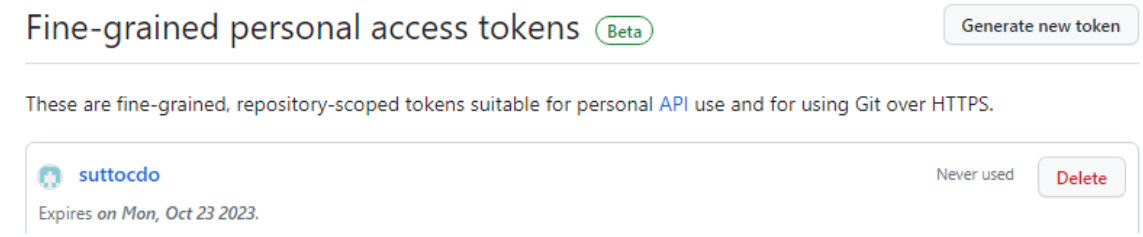


This is the log

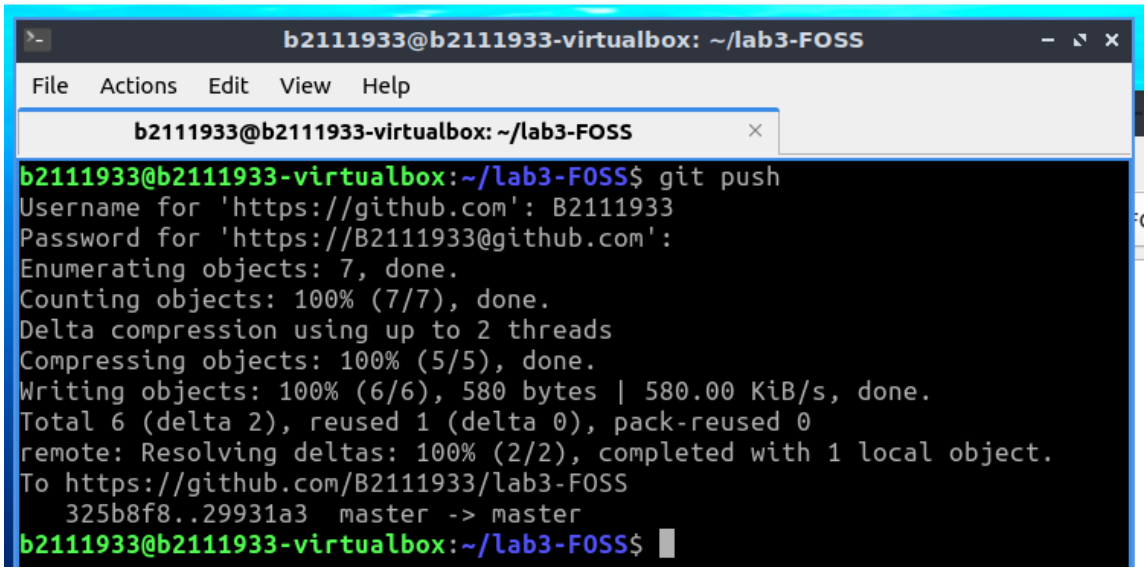
- Push the commits to the server

```
$git push
```

(Note: You need to provide a personal access token as a password, please follow this article to create one)

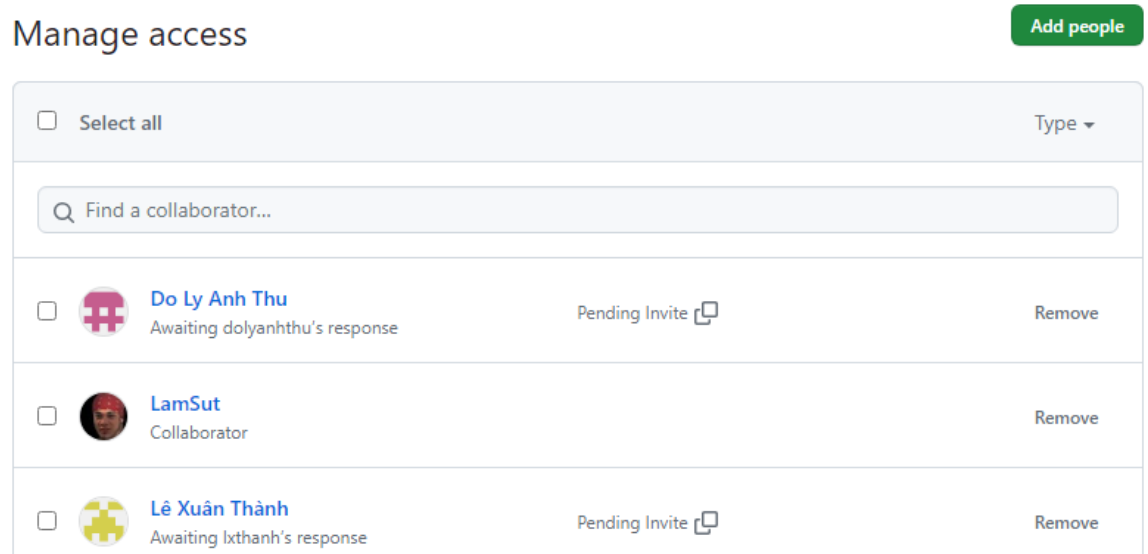


Generated a token



Pushed successfully

- Invite someone to be a collaborator of Github repository



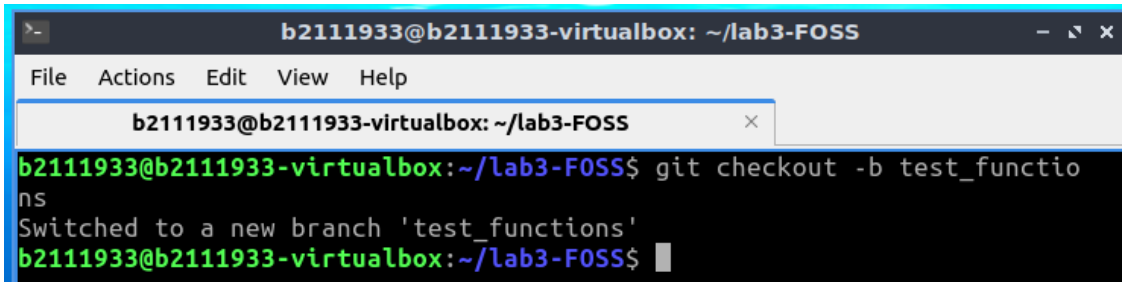
I have invited my friends to be collaborators of my Github repository

(take screenshots to show that you finish this exercise)

2. Branching and Merging

- Create a new branch “test_functions”

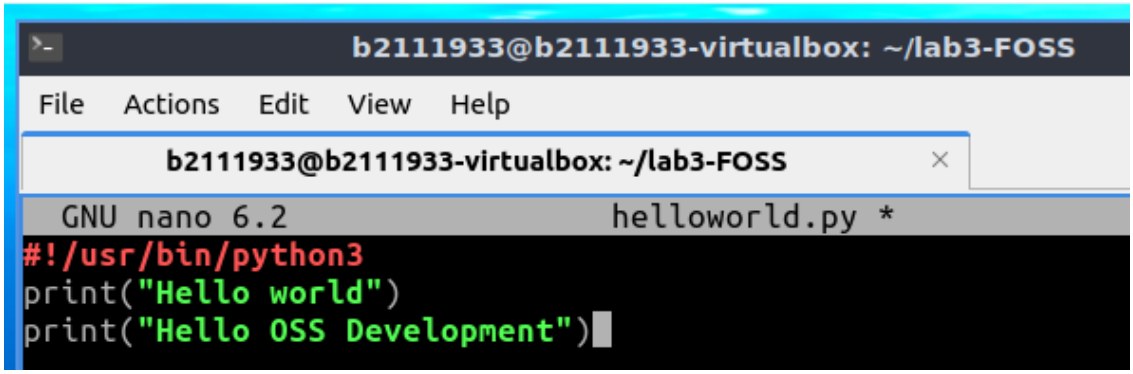
```
$git checkout -b test_functions
```



Created a new branch named **test_functions**

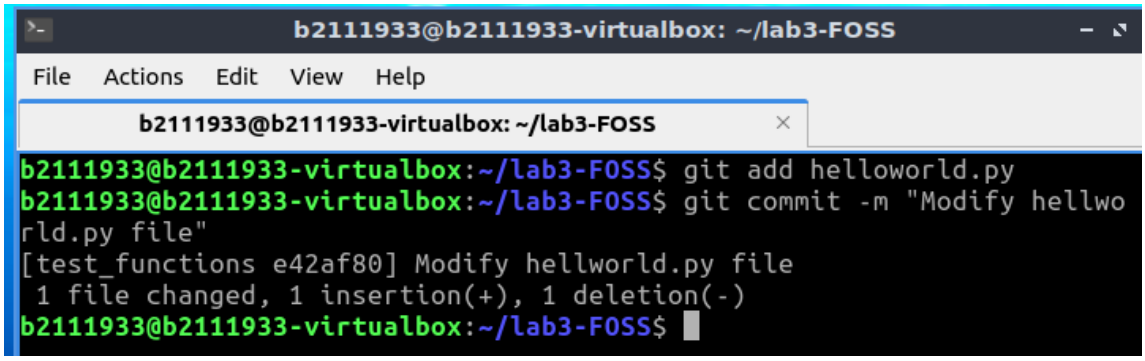
- Edit `helloworld.py` file and commit the result

```
$nano helloworld.py
    #!/usr/bin/python3
    print(“Hello world”)
    print(“Hello OSS Development”)
```



Edited file **helloworld.py**

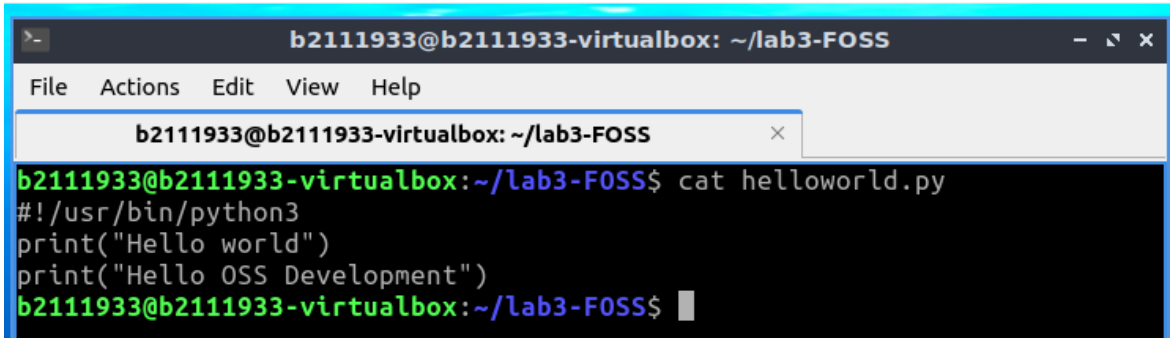
```
$git add helloworld.py
$git commit -m “Modify hellworld.py file”
```



Committed the result

- Display `helloworld.py` content

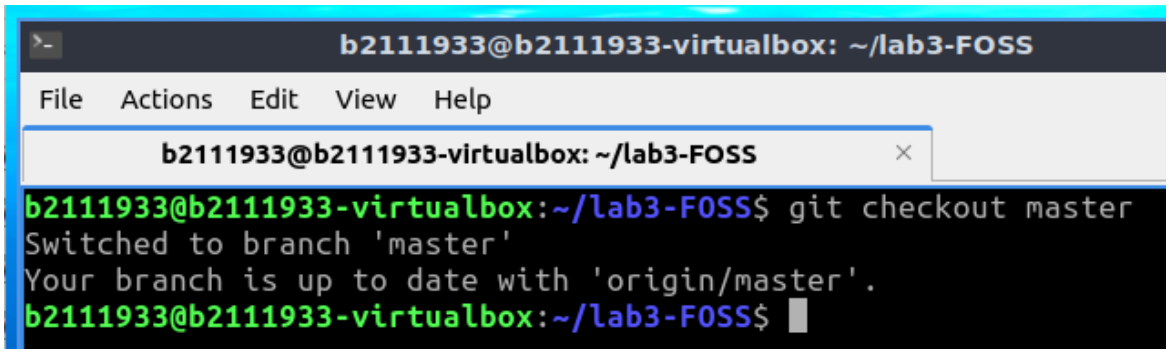
```
$cat helloworld.py
```



Displayed **helloworld.py** content (in **test_functions** branch)

- Swap back to the “master” branch

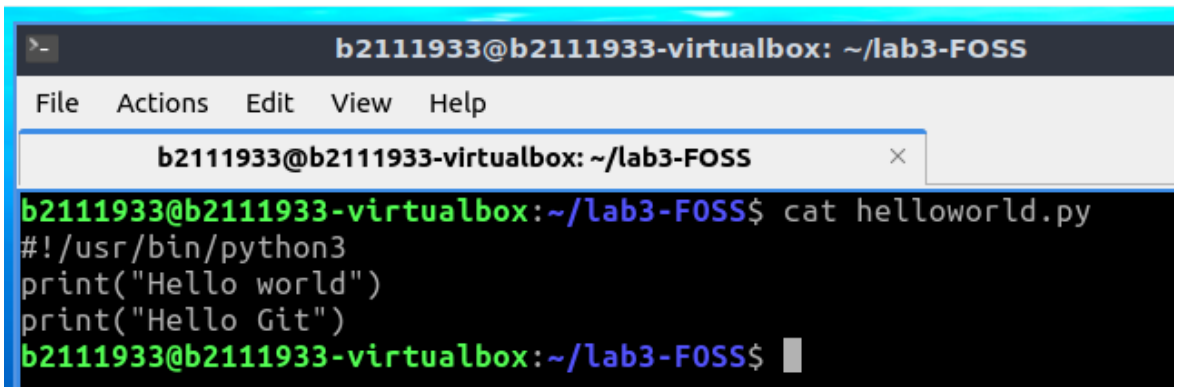
```
$git checkout master
```



Got back to the **master** branch

- Display `helloworld.py` content

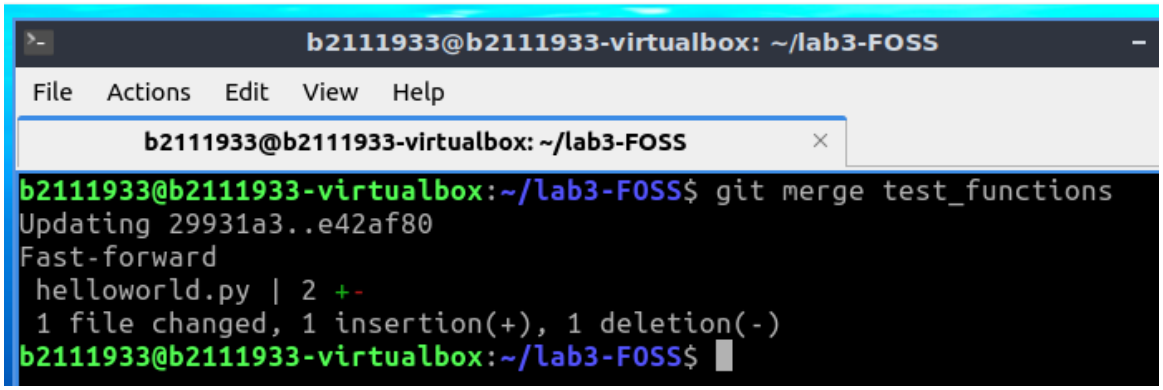
```
$cat helloworld.py
```



Displayed **helloworld.py** content (in **master** branch)

- Merge “test_functions” branch to “master”

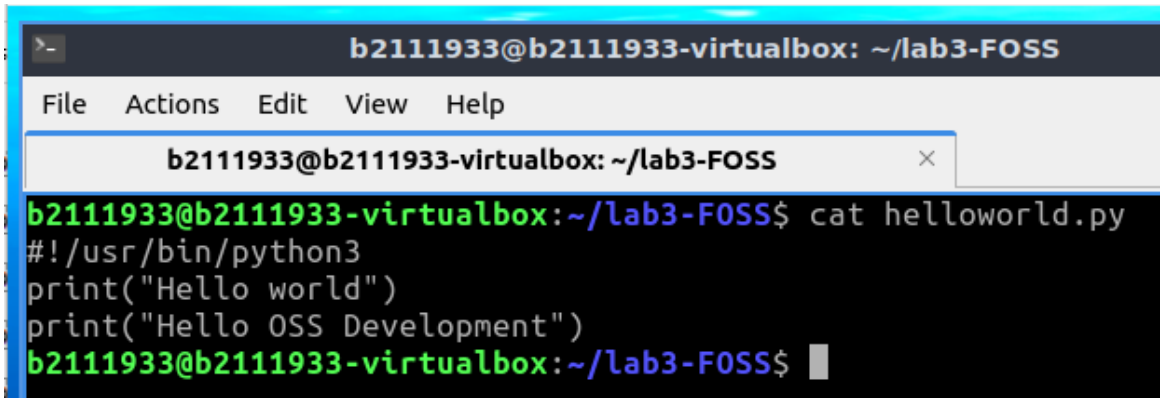
```
$git merge test_functions
```



Merged **test_functions** branch to **master**

- Display `helloworld.py` content. Are there any conflicts?

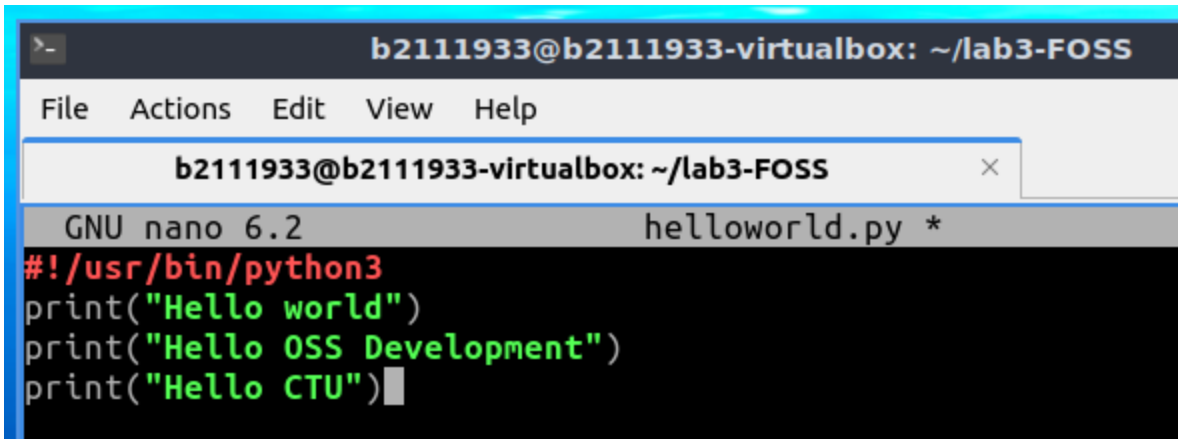
```
$cat helloworld.py
```



Displayed **helloworld.py** content, we can see that file **helloworld.py** in **master** branch was replaced by the one in **test_functions** branch. There was no conflict.

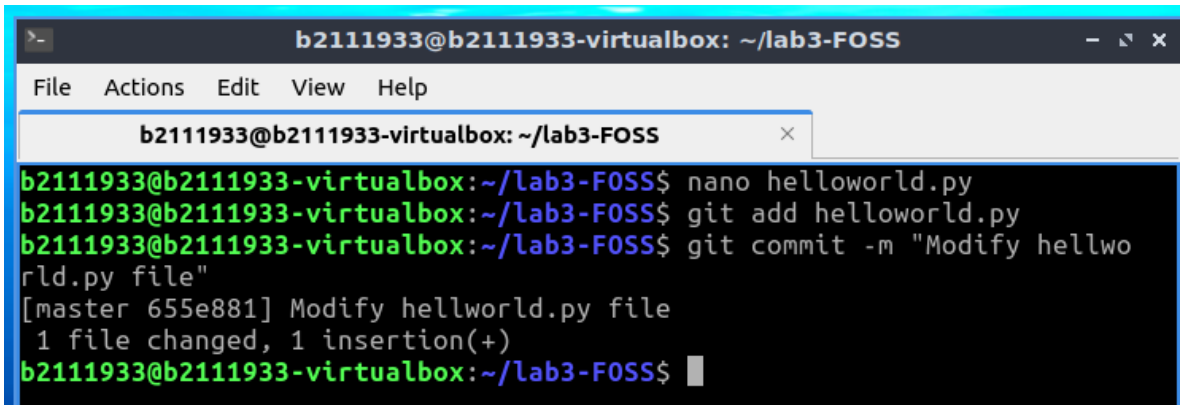
- Edit `helloworld.py` file and commit the result

```
$nano helloworld.py
    #!/usr/bin/python3
    print(“Hello world”)
    print(“Hello OSS Development”)
    print(“Hello CTU”)
```



Edited file **helloworld.py**

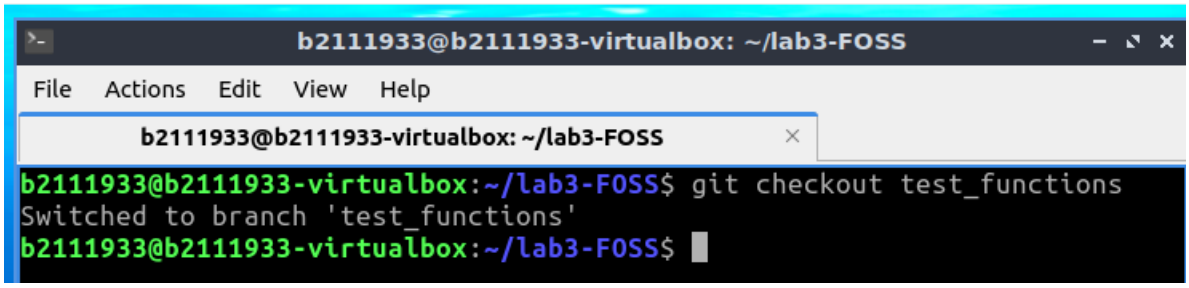
```
$git add helloworld.py
$git commit -m "Modify hellworld.py file"
```



Committed the result

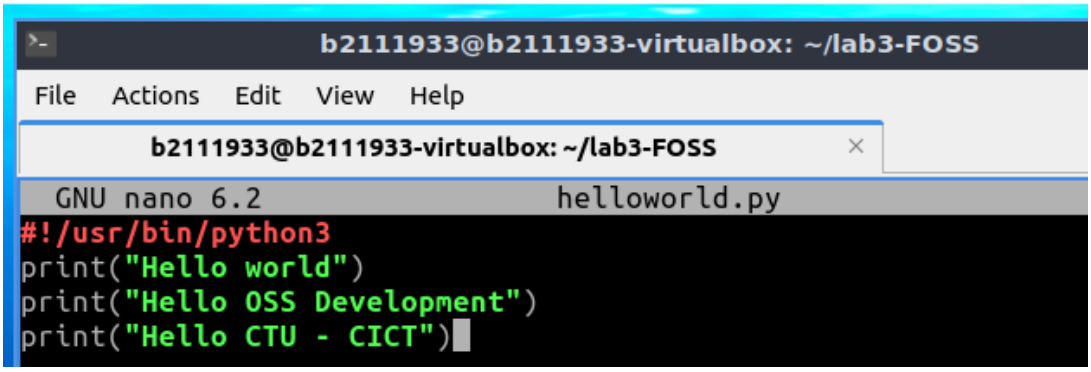
- Checkout to `test_functions`, edit `helloworld.py` file and commit the result

```
$git checkout
```



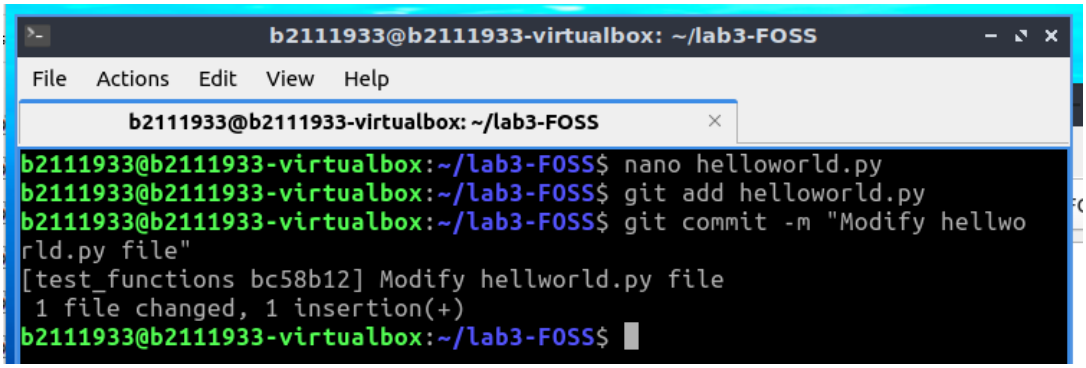
Switched to branch **test_functions**

```
$nano helloworld.py
     #!/usr/bin/python3
     print("Hello world")
     print("Hello OSS Development")
     print("Hello CTU - CICT")
```



Edited file **helloworld.py**

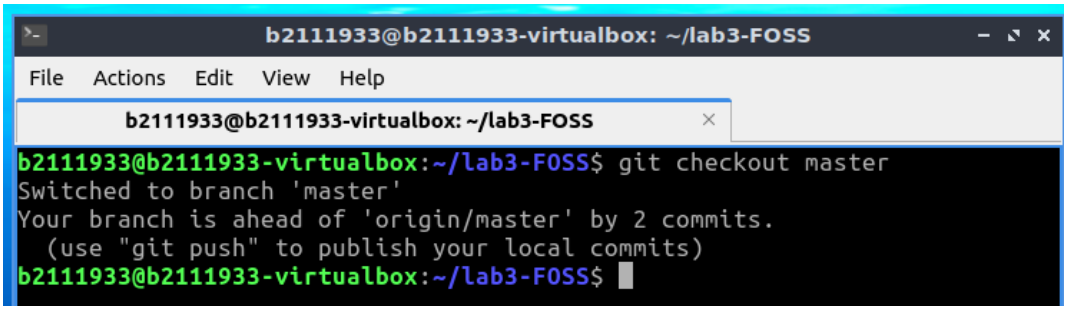
```
$git add helloworld.py
$git commit -m "Modify hellworld.py file"
```



Committed the result

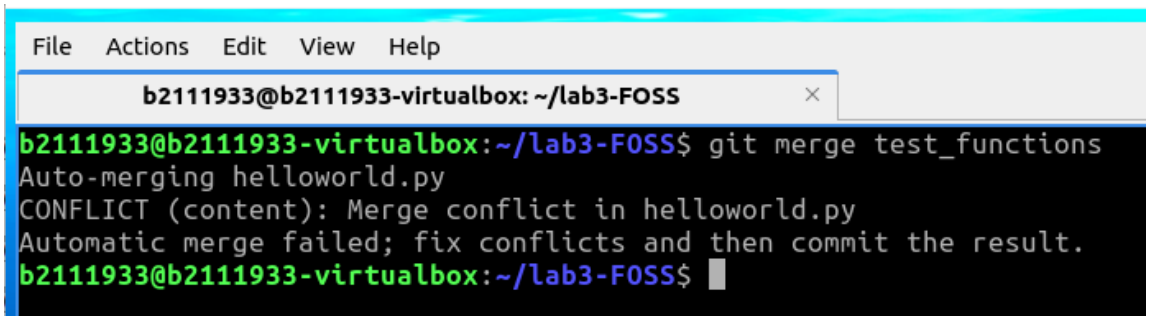
- Swap back to the "master" branch, merge "test_functions" branch to "master"

```
$git checkout master
```



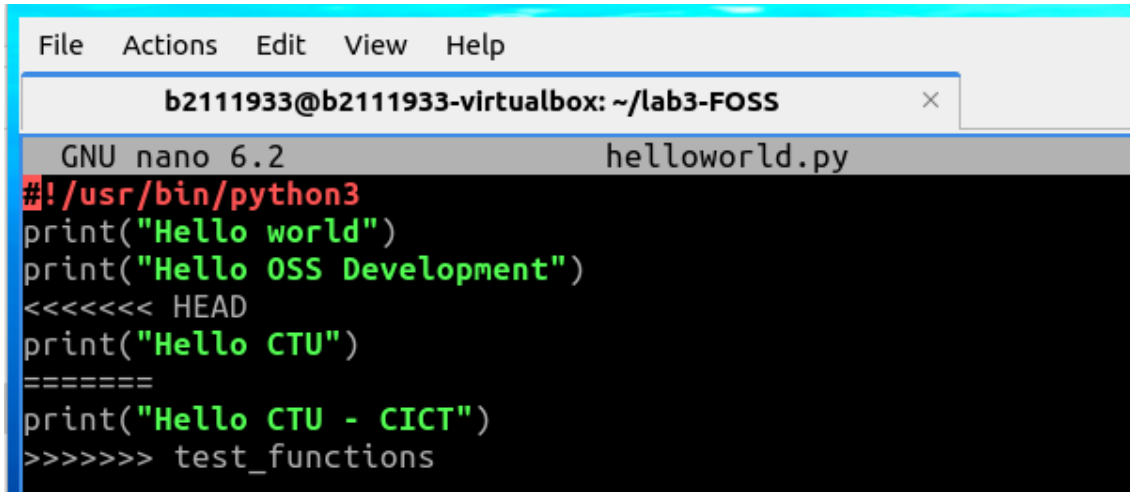
Switched back to branch **master**

```
$git merge test_functions
```

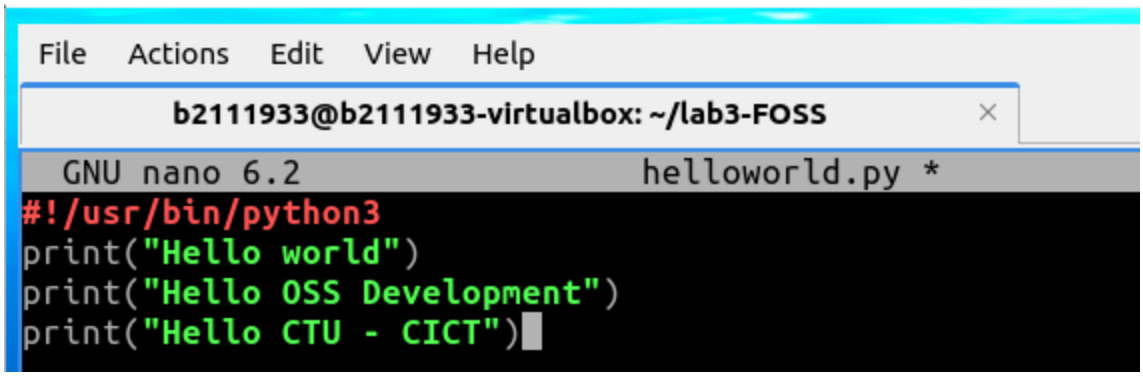


There are conflicts

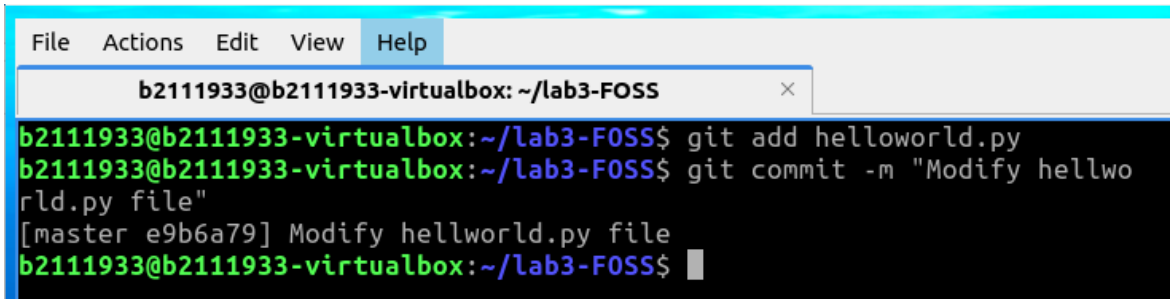
- Are there any conflicts? If YES, resolve the conflict (i.e. edit the conflict markers to match how you want the file to look - like) and commit the result. Use git log to see the resulting commits on the master branch.



When there are **conflicts**, the file will look like this



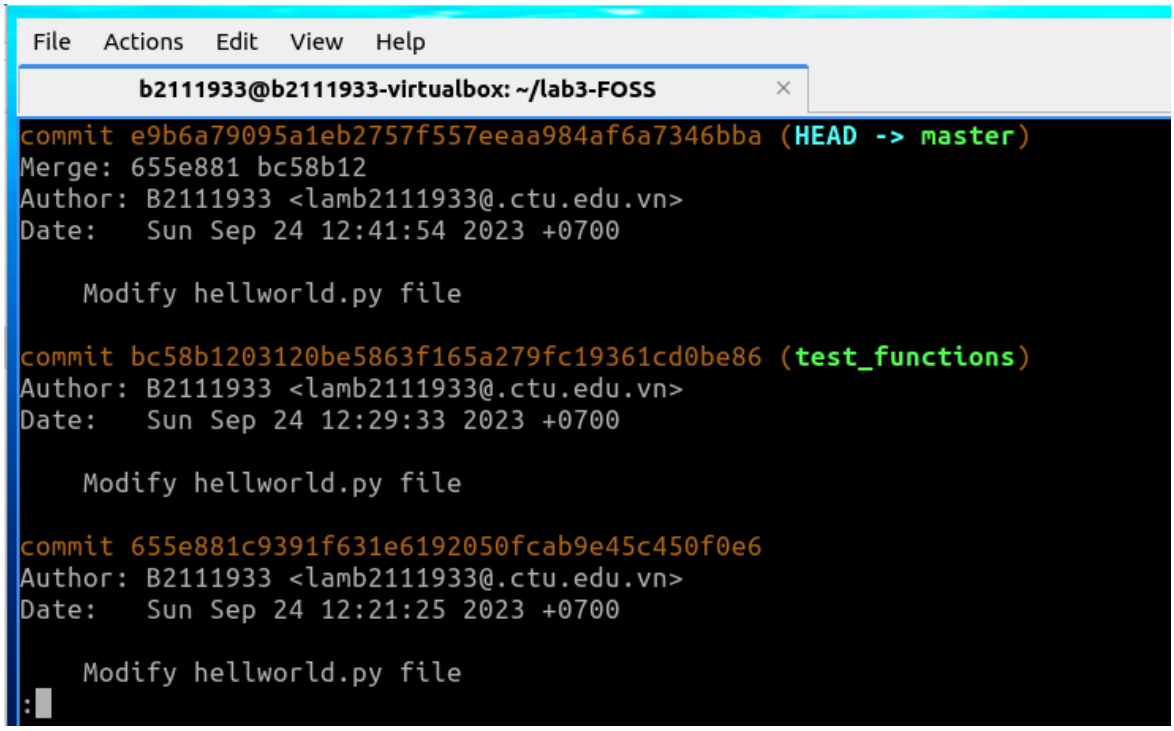
Edited file **helloworld.py**



Committed the result

- View the log to see the resulting commits on the master branch.

```
$git log
```

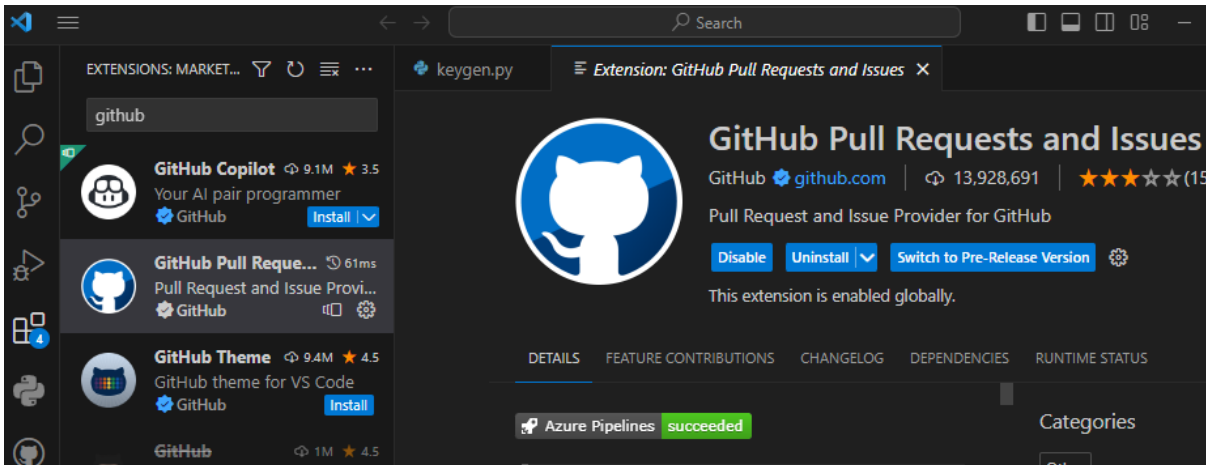


View the **log** to see the **resulting commits** on the **master** branch

(take screenshots to show that you finish this exercise)

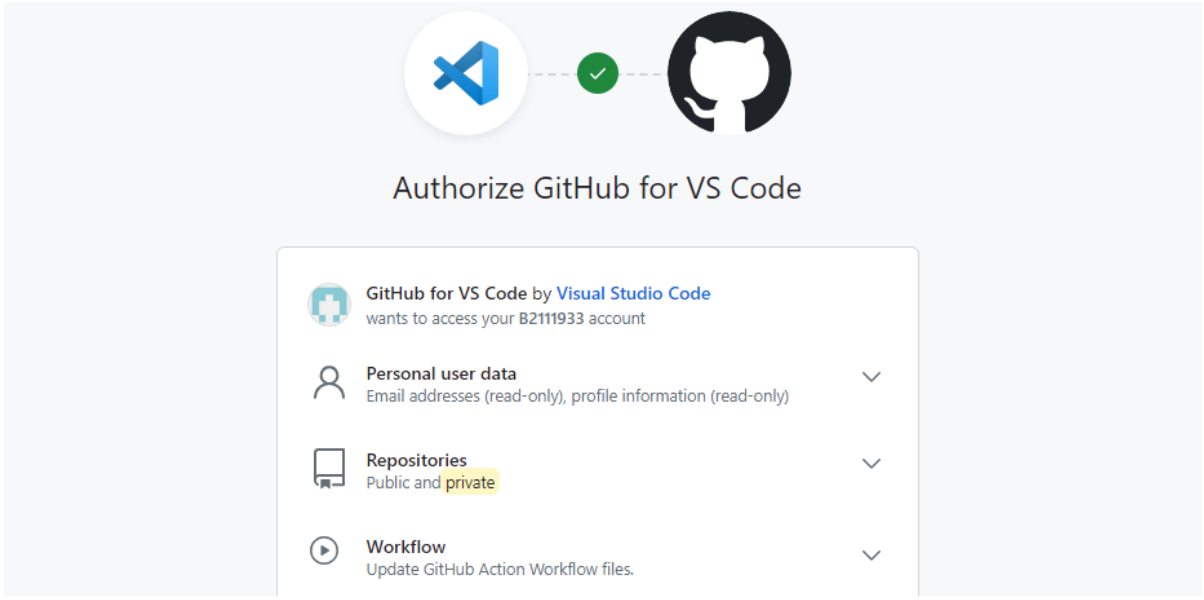
3. Working with GitHub in VS Code

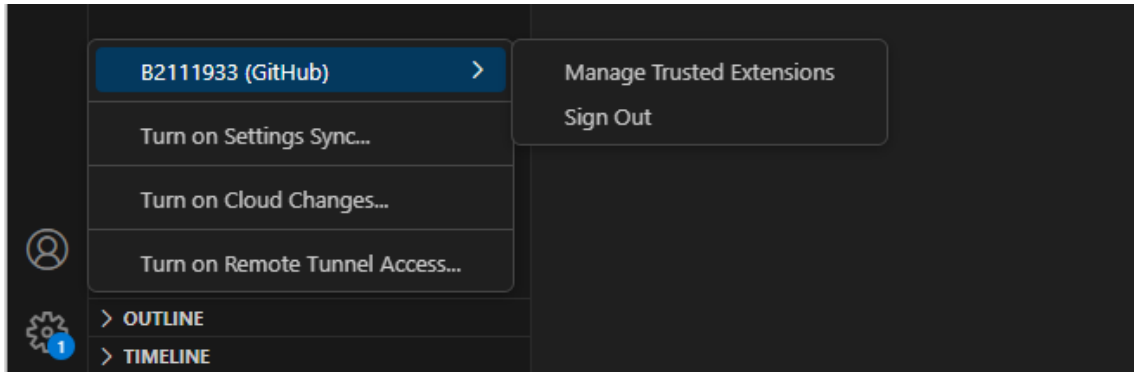
- Install VS Code and its GitHub Pull Requests-Issues extension to your computer.



VS Code and its **GitHub Pull Requests-Issues** extension was installed successfully

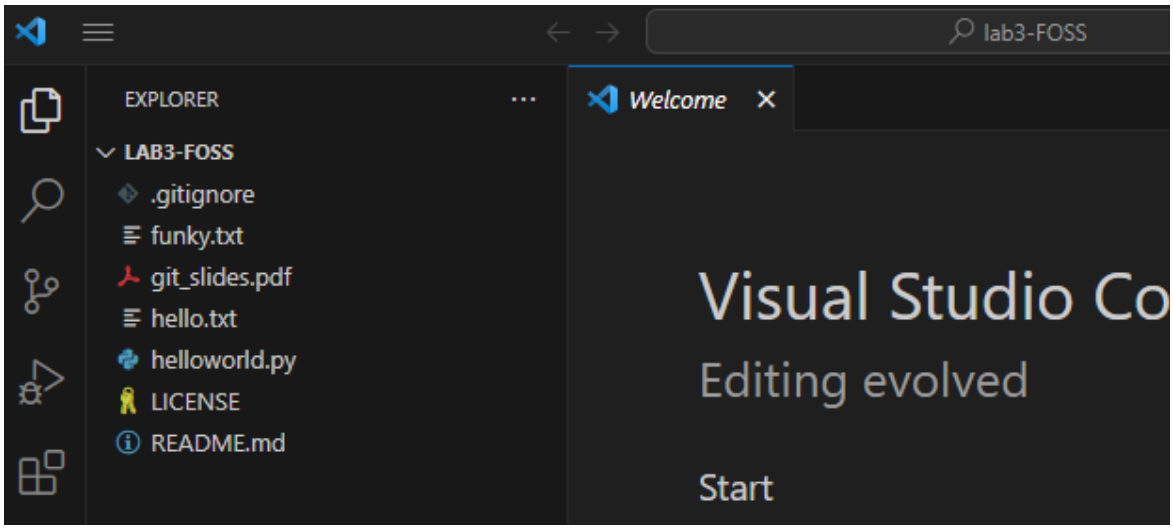
- Sign in GitHub using GitHub Pull Requests and Issues extension.





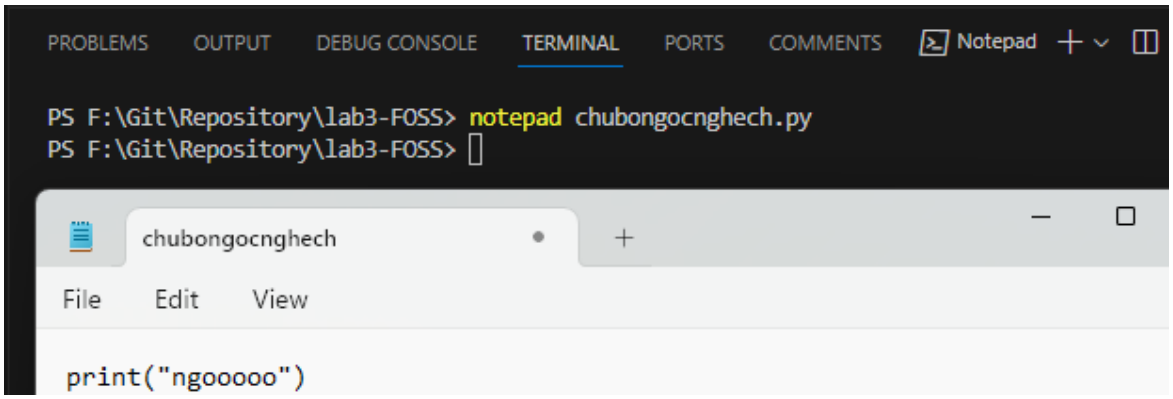
Signed in **GitHub** using **GitHub Pull Requests and Issues** extension

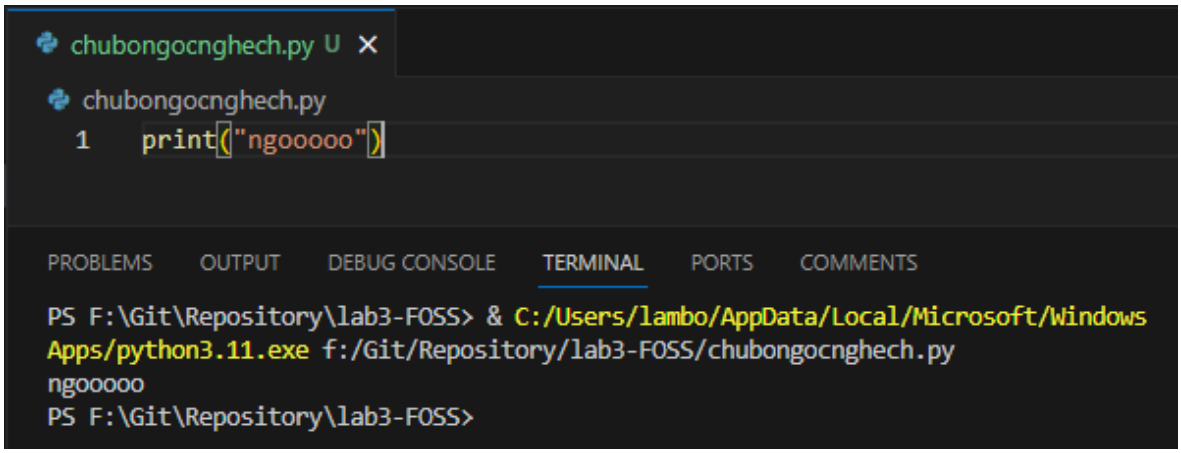
- Clone a remote repository to your computer.



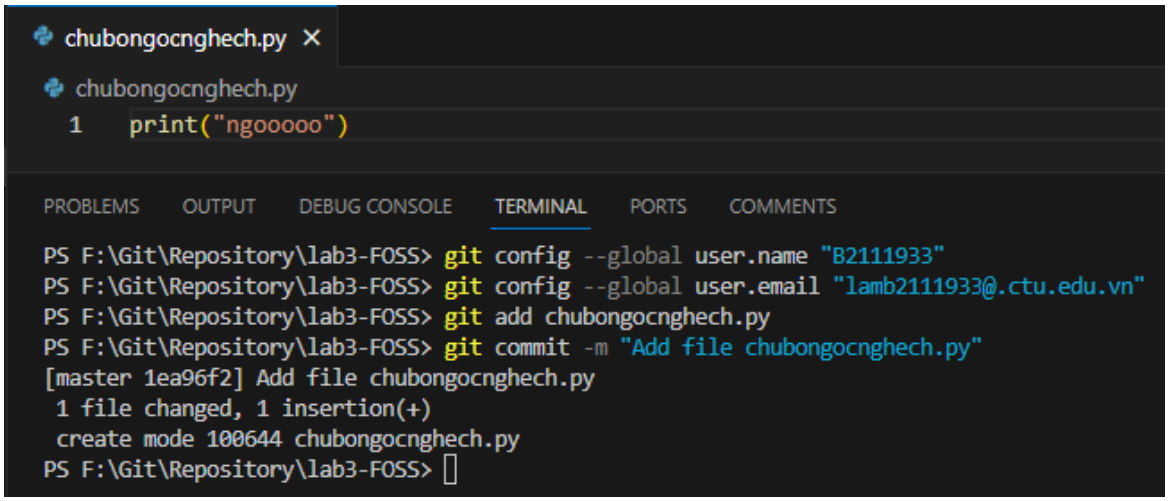
Cloned the **remote repository** in exercises above to my computer

- Create and add a new file to the local repository, then commit everything you have done so far.



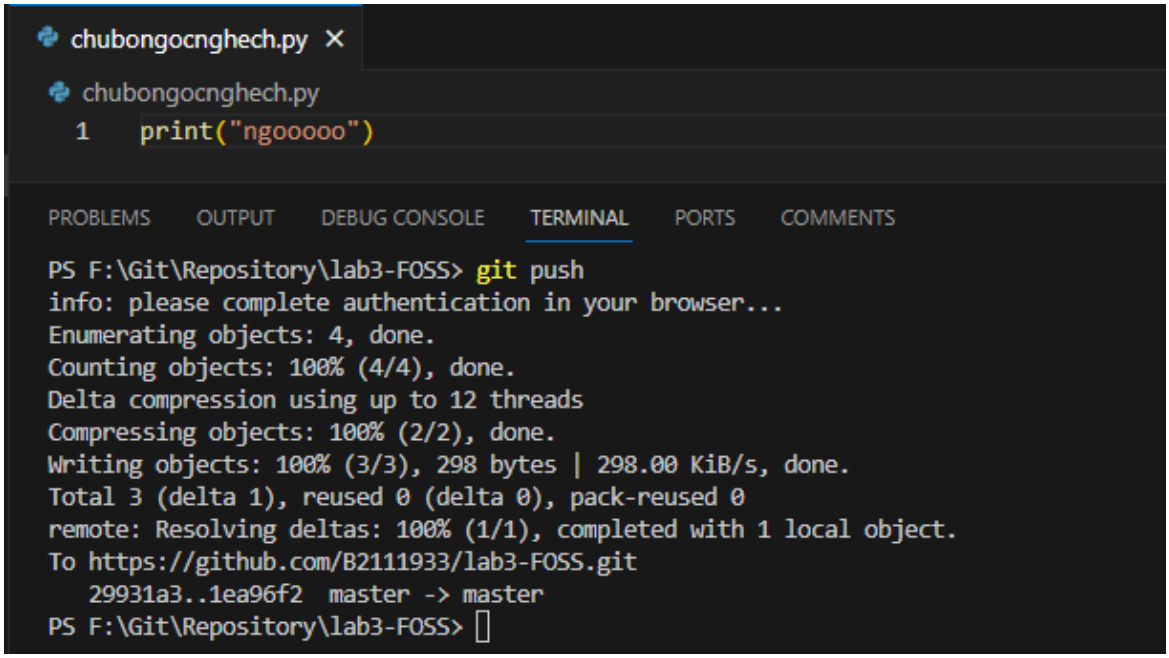


Create file **chubongocnghech.py**



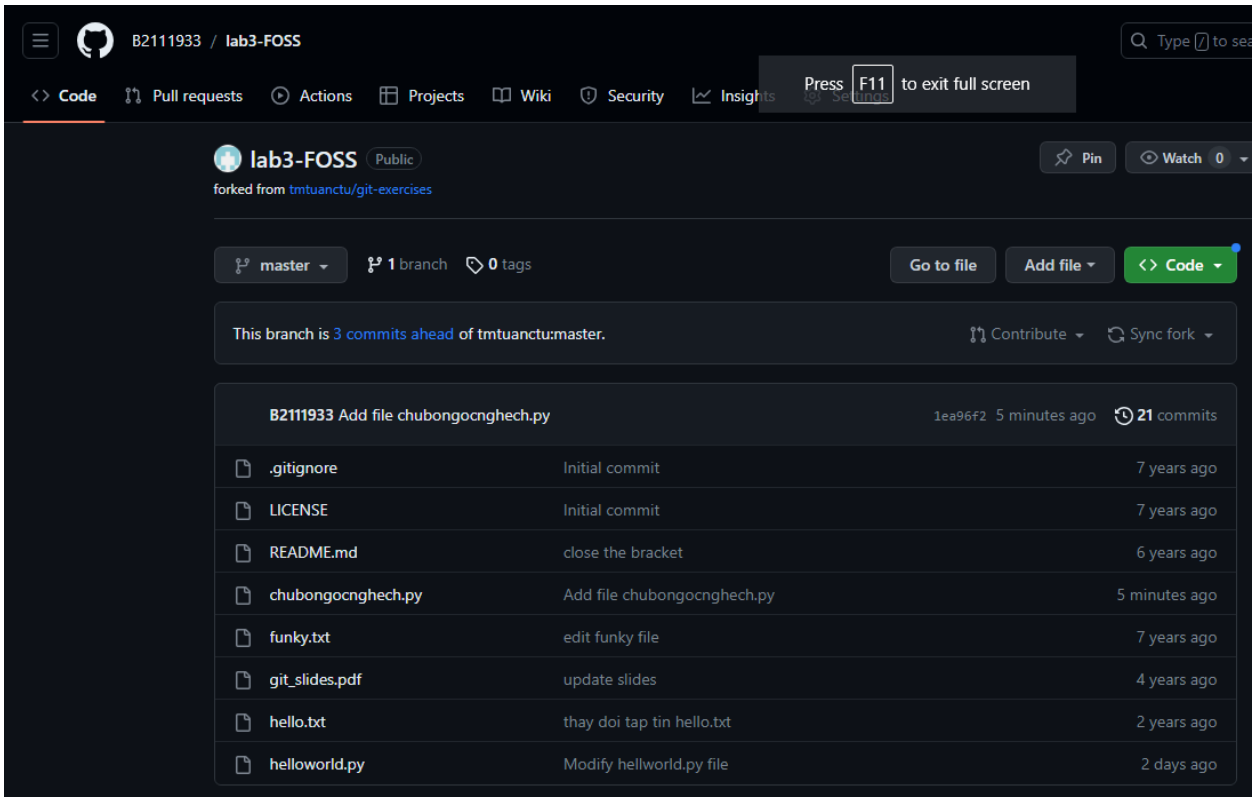
Committed the result

- Push the commits to the remote repository.



Push the commits to the remote repository

Now let's check the result



Pushed successfully

(take screenshots to show that you finish this exercise)

---END---