B2 - Unix System Programming

B-PSU-200

Branches & PR

How collaborate on a repository

Branches & PR

binary name: calculator
language: C
build tool: Makefile

- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

Branches

Branches are features integrated by Git. They allow to create a duplicate of the repository to be able to code independently from the main branch. You can push and keep in memory on your repository code that will not be present on the main branch. This can allow you to code without breaking what is already functional or within a group to code on several different parts without having merge problems

Pull Request

Pull Requests in Git and on development platforms like GitHub allow developers to propose code changes to a main branch like (master or main),asking a project manager to review and merge them if necessary. They can allow to merge an entire branch with the main branch.

PRACTICE

STEP 1

Go on your Github and Create a private repository with the name of your choice

STEP 2

Clone this repository on your computer and cpoy and paste the content of the document given at the beginning of the activity. And push all of the files

STEP 3

Now try to see all of your branch
Normally you should see this :

```
Terminal
~/B-PSU-200>
main
```

STEP 4

You will now create your first branch from the main branch. Name this branch test_to_create_branch.
After that if you try to do the same of the step 3 you should see this :
Now try to see all of your branch
Normally you should see this :

```
Terminal
~/B-PSU-200>
main
test_to_create_branch
```

STEP 5

Since you have two branches you can now move to the branch you have created.

Step 6

Oh no! There is a problem with the subject. Step 4 is not correct! (you can complain to the person who wrote the pdf). The branch we created has the wrong name. Phew, we realized that before we made any changes. So we can solve this problem easily. We just have to go back to the main branch and delete the one we created.

It is forbidden to go to the next step without having done this one!

Step 7

This time it's the right one, create a branch called subtraction_function. And go on it.

Step 8

You have to implement the function that makes the substraction between the two numbers given as parameter.

Step 9

If you have succeeded in the previous step, you can push the changes on the current branch.

The current branch is not the main branch.

Step 10

Now you can go on your github. If everything went well Github will propose you to merge and create a pull request. Try to do it and have the content of your branch on the branch main.

Step 11 : To Practice

You can do the the last 4 steps with the functions “x”, “/” and “**“.

Pull Request

To put this in the context of the group you can invite one of your friends on your repository to create a pull request and put your collaborator in reviewer. This means that he will also have to validate your pull request so that it is merged with the main branch.