Manage Git conflicts

Resolving a merge conflict

We will learn how to solve a merge conflict.

A merge conflict occurs when you try to merge two branches that modify the same part of the same file. Git will integrate the two versions in the same file and then let the user decide the right content.

We will have to start by causing one merge conflict intentionally.

Generate the merger conflict

- Create a new repository on gitlab, github or bitbucket
- Clone this repository locally (git clone)
- Create a data.txt file containing a text, put it in the local repository directory
- Create an initial commit and send the fiel to the remote server.

  (git init, git add, git commit, git push origin master)
- Create a branch named « fork1 » from the branch master
  (git checkout -b fork1)
- Working on the branch fork1, edit data.txt, modify the text, then add, commit,push
- Working on the branch fork1, edit data.txt, modify the text, then add, commit,push
- Working on the branch fork1, edit data.txt, modify the text, then add, commit,push
- Select to the master branch (git checkout)
- Create a branch fork2 from thebranch master
- Select the branch fork2, edit data.txt, modify the text, then add, commit,push
- Select to the master branch (git checkout)
- Merge fork1 into master (git merge)
- Merge fork2 into master (git merge)

The last git merge command ouput shows a merge conflict:

```
Auto-merging data.txt
CONFLICT (content): Merge conflict in data.txt
Automatic merge failed; fix conflicts and then commit the result.
```

The git status helps to understand the conflict:

```
On branch master You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths: (use "git add <file>..." to mark resolution)

        both modified:   data.txt
```

Resolving a merger conflict

- Open the data.txt file in its code editor,
- Analyse how git represents the conflict, and the source of each version
- Edit the file to keep only the final version you want,
- Create a commit and save;

`git status` :

```
On branch master
nothing to commit, working tree clean
```

Then apply local changes on the remote server