UNIT 2: Control Version Systems. GIT

Lab 2: Working with remotes

0. INTRODUCTION

To be able to collaborate on any Git project, you need to know how to manage your remote repositories. Remote repositories are versions of your project that are hosted on the Internet or network somewhere. You can have several of them, each of which generally is either read-only or read/write for you. Collaborating with others involves managing these remote repositories and pushing and pulling data to and from them when you need to share work.

Local repositories reside on the computers of team members. In contrast, remote repositories are hosted on a server that is accessible for all team members. Technically, a remote repository doesn't differ from a local one: it contains branches, commits, and tags just like a local repository. However, a local repository has a working copy associated with it: a directory where some version of your project's files is checked out for you to work with. A remote repository doesn't have such a working directory: it only consists of the bare ".git" repository folder. A bare Git repository is typically used as a Remote Repository that is sharing a repository among several different people. You don't do work right inside the remote repository so there's no Working Tree (the files in your project that you edit), just bare repository data.

1.MAKING A REMOTE REPOSITORY

We are going to simulate a remote repository in our VM. To this end, we will make a bare repository:

- We make a new directory in Documents folder: ***cd; cd Documents; mkdir rem_rep***
- We make a bare repository: ***cd rem_rep; git init - - bare***

```
daw@daw-server:~/Documents$ cd rem_rep; git init --bare
Initialized empty Git repository in /home/daw/Documents/rem_rep/
```

Now, we have a bare repository, as we could have on a server, From now on, we will work with this repository as if it was on a server. Its content is shown in the next image:

```
daw@daw-server:~/Documents/rem_rep$ ls
branches  config  description  HEAD  hooks  info  objects  refs
```

2. Working with remotes

Now we are going to clone the “remote repository” in local, we will change it, make branches, work with them, and finally uptdate the remote.We are going to simulate we are a team work, so someone else will want to update the remote repository with his changes, and we will going to see what could happen

2.1. Cloning repositories

First, we will clone the remote repository. Every version of every file for the history of the project is pulled down by default when you run ***git clone***. In fact, if your server disk gets corrupted, you can often use nearly any of the clones on any client to set the server back to the state it was in when it was cloned. Git is a distributed source control system.

We run the following commands to clone the ***rem-rep*** repository into ***project1A*** repository:

cd;cd Documents; git clone rem-rep project1A

```
daw@daw-server:~/Documents$ git clone rem_rep project1A
Cloning into 'project1A'...
warning: You appear to have cloned an empty repository.
done.
```

That is, we have an empty repository in project1A directory:

```
daw@daw-server:~/Documents/project1A$ ls -la
total 12
drwxrwxr-x 3 daw daw 4096 ene  1 20:51 .
drwxr-xr-x 4 daw daw 4096 ene  1 20:51 ..
drwxrwxr-x 7 daw daw 4096 ene  1 20:51 .git
```

Run git remote -v in order to check the remote associated with this repository, Pay attention to the name “origin”, it is the default name of the remote, you can change this name if you want:

```
daw@daw-server:~/Documents/project1A$ git remote -v
origin  /home/daw/Documents/rem_rep (fetch)
origin  /home/daw/Documents/rem_rep (push)
```

As you can see ***origin*** is the link with the ***rem_rep*** repository.

2.2. Working in local and updating remotes

Now, we are ready to start working with our local repository ***project1A***:

a) Write the name of three modules of Web Application Development degree in a file called subjects.txt

b) Make the first *commit*. You should get something like that:

```
daw@daw-server:~/Documents/project1A$ git commit -m "first commit master"
[master (root-commit) e924714] first commit master
 1 file changed, 3 insertions(+)
 create mode 100644 subjects.txt
daw@daw-server:~/Documents/project1A$
```

c) Update the remote repository (push).

```
daw@daw-server:~/Documents/project1A$ git push
Counting objects: 3, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 282 bytes | 141.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To /home/daw/Documents/rem_rep
 * [new branch]      master -> master
```

After that, the ***git log*** command will show that remote(in red) and local(green) repositories are synchronized, they are pointing at the same commit

```
daw@daw-server:~/Documents/project1A$ git log
commit e924714f32b21977d81253d9af7a65ea8fe06c95 (HEAD -> master, origin/master)
Author: Luz Martínez <luz.martinez@iesabastos.org>
Date:   Wed Jan 2 11:30:48 2019 +0100

    first commit master
```

d) Make a branch, ***version1***.

e) Move to ***version1*** branch and add to the subjects.txt file the name of one more subject of Web Application Development degree

```
daw@daw-server:~/Documents/project1A$ git branch version1
daw@daw-server:~/Documents/project1A$ git checkout version1
Switched to branch 'version1'
```

f) Commit the ***version1*** branch changes

```
daw@daw-server:~/Documents/project1A$ git add .
daw@daw-server:~/Documents/project1A$ git commit -m "first commit version1 branch"
[version1 fe92c4d] first commit version1 branch
 1 file changed, 1 insertion(+)
```

The ***git log --graph --all*** command will show the following:

```
daw@daw-server:~/Documents/project1A$ git log --graph  --all
* commit fe92c4da9eb333513daa92cf04eb2ec6447632c7 (HEAD -> version1)
| Author: Luz Martínez <luz.martinez@iesabastos.org>
| Date:   Wed Jan 2 12:04:50 2019 +0100
|
|     first commit version1 branch
|
* commit e924714f32b21977d81253d9af7a65ea8fe06c95 (origin/master, master)
  Author: Luz Martínez <luz.martinez@iesabastos.org>
  Date:   Wed Jan 2 11:30:48 2019 +0100

      first commit master
```

We have the master branches at the same point in the remote and the local repositories, but we have one more branch, ***version1***, in local that is not yet in the remote.

g) Update the remote repository (push) with the new branch, ***version1***. To add the new branch to the remote, the command would be ***git push origin version1.***

```
daw@daw-server:~/Documents/project1A$ git push origin version1
Counting objects: 3, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 321 bytes | 321.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To /home/daw/Documents/rem_rep
 * [new branch]      version1 -> version1
```

h) Show the log information.

```
daw@daw-server:~/Documents/project1A$ git log --graph  --all
* commit fe92c4da9eb333513daa92cf04eb2ec6447632c7 (HEAD -> version1, origin/version1)
| Author: Luz Martínez <luz.martinez@iesabastos.org>
| Date:   Wed Jan 2 12:04:50 2019 +0100
|
|     first commit version1 branch
|
* commit e924714f32b21977d81253d9af7a65ea8fe06c95 (origin/master, master)
  Author: Luz Martínez <luz.martinez@iesabastos.org>
  Date:   Wed Jan 2 11:30:48 2019 +0100

      first commit master
```

As you can see, we have both branches synchronized in both, remote and local repositories.

2.3 Cloning again.

Now, it is supposed that another member of the team is going to work in the same project, so we will simulate this situation cloning again the same repository, in this case we will use the project1B directory(***cd;cd Documents; git clone rem-rep project1B)***. Then, we will do the following activities:

a) Previously, we are going to get information about the remote we have just clone with the command ***git remote show origin.***

```
daw@daw-server:~/Documents/project1B$ git remote show origin
* remote origin
  Fetch URL: /home/daw/Documents/rem_rep
  Push  URL: /home/daw/Documents/rem_rep
  HEAD branch: master
  Remote branches:
    master   tracked
    version1 tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local ref configured for 'git push':
    master pushes to master (up to date)
```

We can see that the remote has two branches: ***master*** and ***version1***. Git log - -graph - -all give us some more information

```
daw@daw-server:~/Documents/project1B$ git log --graph --all
* commit fe92c4da9eb333513daa92cf04eb2ec6447632c7 (origin/version1)
| Author: Luz Martínez <luz.martinez@iesabastos.org>
| Date:   Wed Jan 2 12:04:50 2019 +0100
|
|     first commit version1 branch
|
* commit e924714f32b21977d81253d9af7a65ea8fe06c95 (HEAD -> master, origin/master, origin/HEAD)
  Author: Luz Martínez <luz.martinez@iesabastos.org>
  Date:   Wed Jan 2 11:30:48 2019 +0100

      first commit master
```

b) If we go to ***version1*** branch, a local version of this branch will be created. You can confirm this branch running again the git log - -graph - -all command

```
daw@daw-server:~/Documents/project1B$ git checkout version1
Branch 'version1' set up to track remote branch 'version1' from 'origin'.
Switched to a new branch 'version1'
```

```
daw@daw-server:~/Documents/project1B$ git log --graph --all
* commit fe92c4da9eb333513daa92cf04eb2ec6447632c7 (HEAD -> version1, origin/version1)
| Author: Luz Martínez <luz.martinez@iesabastos.org>
| Date:   Wed Jan 2 12:04:50 2019 +0100
|
|     first commit version1 branch
|
* commit e924714f32b21977d81253d9af7a65ea8fe06c95 (origin/master, origin/HEAD, master)
  Author: Luz Martínez <luz.martinez@iesabastos.org>
  Date:   Wed Jan 2 11:30:48 2019 +0100

      first commit master
```

c) Check that the subjects.txt file in master branch has only 3 lines and the file in version1 branch has 4 lines.

d) Go to version1 branch and add one more subject to subjects.txt file, and commit the

changes.

```
daw@daw-server:~/Documents/project1B$ git add .
daw@daw-server:~/Documents/project1B$ git status
On branch version1
Your branch is up to date with 'origin/version1'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   subjects.txt

daw@daw-server:~/Documents/project1B$ git commit -m "commit in version1 from project1B"
[version1 27b4c97] commit in version1 from project1B
 1 file changed, 1 insertion(+)
```

e) Update the remote.

```
daw@daw-server:~/Documents/project1B$ git push origin version1
Counting objects: 3, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 322 bytes | 322.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To /home/daw/Documents/rem_rep
   fe92c4d..27b4c97  version1 -> version1
```

```
daw@daw-server:~/Documents/project1B$ git log
commit 27b4c971ab80c85f430df2ef8f3d02ed98ff61c6 (HEAD -> version1, origin/version1)
Author: Luz Martínez <luz.martinez@iesabastos.org>
Date:   Wed Jan 2 19:46:19 2019 +0100

    commit in version1 from project1B

commit fe92c4da9eb333513daa92cf04eb2ec6447632c7
Author: Luz Martínez <luz.martinez@iesabastos.org>
Date:   Wed Jan 2 12:04:50 2019 +0100

    first commit version1 branch

commit e924714f32b21977d81253d9af7a65ea8fe06c95 (origin/master, origin/HEAD, master)
Author: Luz Martínez <luz.martinez@iesabastos.org>
Date:   Wed Jan 2 11:30:48 2019 +0100

    first commit master
```

2.3 More changes.

a) Imagine that the programer who works in ***project1A*** repository, ***version1*** branch, change one more time subjects.txt, adding a new module or subject to it. And commit the changes in local.

```
daw@daw-server:~/Documents/project1A$ git branch
  master
* version1
```

```
daw@daw-server:~/Documents/project1A$ git add .
daw@daw-server:~/Documents/project1A$ git commit -m "segundo commit from project1A"
[version1 9af2899] segundo commit from project1A
 1 file changed, 1 insertion(+)
```

b) Update the remote with the changes in ***version1*** branch

```
daw@daw-server:~/Documents/project1A$ git push origin version1
To /home/daw/Documents/rem_rep
 ! [rejected]        version1 -> version1 (fetch first)
error: failed to push some refs to '/home/daw/Documents/rem_rep'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Something seems to be wrong, we cannot update a branch with changes when someone more has changed before us and we haven't got these changes yet. We could know this information before, running the command ***git remote show origin.***

```
daw@daw-server:~/Documents/project1A$ git remote show origin
* remote origin
  Fetch URL: /home/daw/Documents/rem_rep
  Push  URL: /home/daw/Documents/rem_rep
  HEAD branch: master
  Remote branches:
    master   tracked
    version1 tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local refs configured for 'git push':
    master   pushes to master   (up to date)
    version1 pushes to version1 (local out of date)
```

So, we must do, what the first image said, pull before pushing.

```
daw@daw-server:~/Documents/project1A$ git pull origin version1
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From /home/daw/Documents/rem_rep
 * branch            version1   -> FETCH_HEAD
   fe92c4d..27b4c97  version1   -> origin/version1
Auto-merging subjects.txt
CONFLICT (content): Merge conflict in subjects.txt
Automatic merge failed; fix conflicts and then commit the result.
```

But, the pull doesn't work, because the commit "segundo commit from project1A " and "commit from project1B are not compatible, both of them change the same line and the lines are different.

```
daw@daw-server:~/Documents/project1A$ git log --graph --all
* commit 9af28993a95a4ff44328a8649ed105104761e229 (HEAD -> version1)
| Author: Luz Martínez <luz.martinez@iesabastos.org>
| Date:   Wed Jan 2 20:09:50 2019 +0100
|
|     segundo commit from project1A
|
| * commit 27b4c971ab80c85f430df2ef8f3d02ed98ff61c6 (origin/version1)
|/  Author: Luz Martínez <luz.martinez@iesabastos.org>
|   Date:   Wed Jan 2 19:46:19 2019 +0100
|
|       commit in version1 from project1B
|
* commit fe92c4da9eb333513daa92cf04eb2ec6447632c7
| Author: Luz Martínez <luz.martinez@iesabastos.org>
| Date:   Wed Jan 2 12:04:50 2019 +0100
|
|     first commit version1 branch
|
* commit e924714f32b21977d81253d9af7a65ea8fe06c95 (origin/master, master)
  Author: Luz Martínez <luz.martinez@iesabastos.org>
  Date:   Wed Jan 2 11:30:48 2019 +0100

      first commit master
```

So, the only way is to fix the conflict manually.

```
daw@daw-server:~/Documents/project1A$ cat subjects.txt
Client-side web development
Server-side web development
Web Interface development
Web Application deployment
<<<<<<< HEAD
Programming
=======
DataBase management
>>>>>>> 27b4c971ab80c85f430df2ef8f3d02ed98ff61c6
daw@daw-server:~/Documents/project1A$
```

We are going to accept both lines: Programming and DataBase management. And then commit the changes.

```
daw@daw-server:~/Documents/project1A$ git log --graph --all
*   commit 6c289345b770c83edcff7339f203d5eb9f3c9bd9 (HEAD -> version1)
|\  Merge: 9af2899 27b4c97
| | Author: Luz Martínez <luz.martinez@iesabastos.org>
| | Date:   Thu Jan 3 01:07:38 2019 +0100
| |
| |     Commit after conflicts(changes in project1A and project1B) in version1 branch
| |
| * commit 27b4c971ab80c85f430df2ef8f3d02ed98ff61c6 (origin/version1)
| | Author: Luz Martínez <luz.martinez@iesabastos.org>
| | Date:   Wed Jan 2 19:46:19 2019 +0100
| |
| |     commit in version1 from project1B
| |
* | commit 9af28993a95a4ff44328a8649ed105104761e229
|/  Author: Luz Martínez <luz.martinez@iesabastos.org>
|   Date:   Wed Jan 2 20:09:50 2019 +0100
|
|       segundo commit from project1A
|
* commit fe92c4da9eb333513daa92cf04eb2ec6447632c7
| Author: Luz Martínez <luz.martinez@iesabastos.org>
| Date:   Wed Jan 2 12:04:50 2019 +0100
|
|     first commit version1 branch
|
* commit e924714f32b21977d81253d9af7a65ea8fe06c95 (origin/master, master)
  Author: Luz Martínez <luz.martinez@iesabastos.org>
  Date:   Wed Jan 2 11:30:48 2019 +0100

      first commit master
```

Now, we are going to try if we can now update the changes to the remote (push)

```
daw@daw-server:~/Documents/project1A$ git push origin version1
Counting objects: 6, done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 633 bytes | 105.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0)
To /home/daw/Documents/rem_rep
   27b4c97..6c28934  version1 -> version1
```

Finally we are going to merge ***master*** and ***version1*** and upload the result to the remote.

And this is the result

```
daw@daw-server:~/Documents/project1A$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
daw@daw-server:~/Documents/project1A$ git merge version1
Updating e924714..6c28934
Fast-forward
 subjects.txt | 4 ++++
 1 file changed, 4 insertions(+)
daw@daw-server:~/Documents/project1A$ git push
Total 0 (delta 0), reused 0 (delta 0)
To /home/daw/Documents/rem_rep
   e924714..6c28934  master -> master
```

To update project1B, that is, to get the last changes from the remote, you need run the "git pull" command.

```
daw@daw-server:~/Documents/project1A$ git log --graph --all
*   commit 6c289345b770c83edcff7339f203d5eb9f3c9bd9 (HEAD -> master, origin/version1, origin/master, version1)
|\  Merge: 9af2899 27b4c97
| | Author: Luz Martínez <luz.martinez@iesabastos.org>
| | Date:   Thu Jan 3 01:07:38 2019 +0100
| |
| |     Commit after conflicts(changes in project1A and project1B) in version1 branch
| |
| * commit 27b4c971ab80c85f430df2ef8f3d02ed98ff61c6
| | Author: Luz Martínez <luz.martinez@iesabastos.org>
| | Date:   Wed Jan 2 19:46:19 2019 +0100
| |
| |     commit in version1 from project1B
| |
* | commit 9af28993a95a4ff44328a8649ed105104761e229
|/  Author: Luz Martínez <luz.martinez@iesabastos.org>
|   Date:   Wed Jan 2 20:09:50 2019 +0100
|
|       segundo commit from project1A
|
* commit fe92c4da9eb333513daa92cf04eb2ec6447632c7
| Author: Luz Martínez <luz.martinez@iesabastos.org>
| Date:   Wed Jan 2 12:04:50 2019 +0100
|
|     first commit version1 branch
|
* commit e924714f32b21977d81253d9af7a65ea8fe06c95
  Author: Luz Martínez <luz.martinez@iesabastos.org>
  Date:   Wed Jan 2 11:30:48 2019 +0100

      first commit master
```

First ***version1*** branch

```
daw@daw-server:~/Documents/project1B$ git branch
  master
* version1
daw@daw-server:~/Documents/project1B$ git pull
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 2), reused 0 (delta 0)
Unpacking objects: 100% (6/6), done.
From /home/daw/Documents/rem_rep
   27b4c97..6c28934  version1   -> origin/version1
   e924714..6c28934  master     -> origin/master
Updating 27b4c97..6c28934
Fast-forward
 subjects.txt | 2 ++
 1 file changed, 2 insertions(+)
```

Second, ***master*** branch

```
daw@daw-server:~/Documents/project1B$ git checkout master
Switched to branch 'master'
Your branch is behind 'origin/master' by 4 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)
daw@daw-server:~/Documents/project1B$ git pull
Updating e924714..6c28934
Fast-forward
 subjects.txt | 4 ++++
 1 file changed, 4 insertions(+)
daw@daw-server:~/Documents/project1B$
```

Now, remote (rem-rep). project1A and project1B are synchronized, and the log of the repository is

```
daw@daw-server:~/Documents/project1B$ git log --graph --all
*   commit 6c289345b770c83edcff7339f203d5eb9f3c9bd9 (HEAD -> master, origin/version1, origin/master, origin/HEAD, version1)
|\  Merge: 9af2899 27b4c97
| | Author: Luz Martínez <luz.martinez@iesabastos.org>
| | Date:   Thu Jan 3 01:07:38 2019 +0100
| |
| |     Commit after conflicts(changes in project1A and project1B) in version1 branch
| |
| * commit 27b4c971ab80c85f430df2ef8f3d02ed98ff61c6
| | Author: Luz Martínez <luz.martinez@iesabastos.org>
| | Date:   Wed Jan 2 19:46:19 2019 +0100
| |
| |     commit in version1 from project1B
| |
* | commit 9af28993a95a4ff44328a8649ed105104761e229
|/  Author: Luz Martínez <luz.martinez@iesabastos.org>
|   Date:   Wed Jan 2 20:09:50 2019 +0100
|
|       segundo commit from project1A
|
* commit fe92c4da9eb333513daa92cf04eb2ec6447632c7
| Author: Luz Martínez <luz.martinez@iesabastos.org>
| Date:   Wed Jan 2 12:04:50 2019 +0100
|
|     first commit version1 branch
|
* commit e924714f32b21977d81253d9af7a65ea8fe06c95
  Author: Luz Martínez <luz.martinez@iesabastos.org>
  Date:   Wed Jan 2 11:30:48 2019 +0100

      first commit master
```

To conclude, in the next image you can see how local and remote repositories interact