GitLab Walkthrough 3b
Developing with the Remote Repository

Mark Eramian and Anthony Kusalik

October 22, 2020

Developing with the Remote Repository

In this walkthrough, we will play the role of a second developer (Programmer B) who will be working on a software project independently from the developer in Gitlab Walkthrough 3a (Programmer A).

Step 0: Fake a Different Identity

Since we want to pretend we are now a different developer on the team, rather the developer we are simulating in Gitlab Walkthrough 3a, we will change our Git identity again. Execute the following commands:

```
git config --global user.name "programmer-b"
git config --global user.email "ProgB@cs.usask.ca"
```

Step 1: Clone the Remote Repository

In a **different directory** from where you cloned the remote repository in Step 1 of Gitlab Walkthrough 3a, clone the remote repository again. You should now have a working copy that includes the changes that Programmer A made in Step 2 of Gitlab Walkthrough 3a.

You should now have two separate, but identical copies of the repository. One for Programmer A, and one for Programmer B.

Step 2: Modify Programmer B's Working Copy

In your newly cloned copy of the repository, modify `bar.c` by adding an extra line to it using the following command:

```
echo "Give bar.c another line" >> bar.c
```

Now, check the status of the working copy using `git status`"

```
bash-3.2$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory

    modified:   bar.c
```

As you should by now be expecting, the status shows that there are uncommitted changes in `bar.c`. If you want to see what they are, you can use `git diff`:

```
bash-3.2$ git diff
diff --git a/bar.c b/bar.c
index 2bd7984..4cdbbef 100644
--- a/bar.c
+++ b/bar.c
@@ -1 +1,2 @@
 this is file bar.c
+Give bar.c another line
```

The last line here that begins with + shows that a new line has been added tp `bar.c` after the line `this is file bar.c`.

Now commit this change using:

```
git commit -a -m "appended a line to bar.c"
```

The result will be:

```
bash-3.2$ git commit -a -m "appended a line to bar.c"
[master 2ee3f5f] appended a line to bar.c
 1 file changed, 1 insertion(+)
```

Now let's view the change log using `git log`:

```
bash-3.2$ git log
commit 2ee3f5f57c26da51be90abc4d29da39f20d6b77e (HEAD -> master)
Author: programmer-b <ProgB@cs.usask.ca>
Date:   Mon Nov 25 09:53:09 2019 -0600

    appended a line to bar.c

commit ae6dc03471876f8bc68055f0b05891b1b6af7ec8 (origin/master, origin/HEAD)
```

```
Author: programmer-a <ProgA@cs.usask.ca>
Date:   Mon Nov 25 09:26:57 2019 -0600

    appended line to foo.c and replaced content of Makefile

commit d68c1c7a7db4fc65944be623d5cc832d626c8aea
Author: team-lead <TeamLead@cs.usask.ca>
Date:   Mon Nov 25 08:52:14 2019 -0600

    Initial commit by TeamLead
```

You can see that latest commit was done by programmer-b, the one before that by programmer-a, and the first commit by Teamlead. We are already building up a history of who did what! Notice how the first commit is tagged with `(HEAD -> master)` which shows that this commit is the current version of our local repository. The second commit is labeled with `(origin/master, origin/HEAD)` showing that this is the current version in the remote repository.

Now we are ready to push Programmer B's changes to the remote repository for others to use. `git push` should do the trick:

```
bash-3.2$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 316 bytes | 316.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To gitusask:mark.eramian/cmpt-214-test-repo
   c16fc13..294d0f6  master -> master
```

If we run `git log` one more time we see this:

```
bash-3.2$ git log
commit 294d0f6b0b37e04ddaf7866f3dba60ffc832440a (HEAD -> master, origin/master,
     origin/HEAD)
Author: programmer-a <ProgA@cs.usask.caend{lstlisting}>
Date:   Thu Jun 25 10:15:30 2020 -0600

    appended a line to bar.c

commit c16fc13dd5b1f0631bf1d4721aec653eb1cf4085
Author: programmer-a <ProgA@cs.usask.caend{lstlisting}>
Date:   Thu Jun 25 09:52:07 2020 -0600

    appended line to foo.c and replaced content of Makefile

commit 412971010eb800e279bdfe5561f98ad61805cf89
Author: team-lead <TeamLead@cs.usask.ca>
Date:   Thu Jun 25 09:45:16 2020 -0600
```

```
    Initial commit by TeamLead
```

Notice how the first, and most recent commit, is now tagged with all of `(HEAD -> master, origin/master, origin/HEAD)`, indicating that this is now the current version for both the local and remote repositories.

Conclusion

We are now finished simulating the activities of Programmer B.

Take another break! You've earned it. Once you've rested your eyes and brain again, return to GitLabWalkthrough 3a and resume it at Step 6.