GIT assignments. LinhTH8@fsoft.com.vn

Installation package & document of git: [For more document/tutorial please google first]

# Assignment 1 – git

One hour to complete this assignment, using git-bash with command-line to do this assignment. **Please store all commands used for each step**.

$ git config --global user.name “To Hoang Linh”

$ git config --global user.email linhth8@fsoft.com.vn

1. Create new local repository/local bare repository named **R0 (git init --bare)**

$ cd linhth8

$ git init --bare R0

1. Clone new local repository from local repository/local bare repository created from #1, new repository named **Ra**

$ git clone R0 Ra

1. In Ra, create/edit source file(s). Add all files and commit these changes (commit **Ca0**) on branch **master**

$ cd Ra

$ git status

$ git add -A

$ git status

$ git commit -m “Ca0”

1. Create/edit multi source files. Add some of changed files (not all files), then commit added files (commit **Ca1**).

$ git status

$ git add s4.header.h

$ git commit -m “Ca1”

1. Commit other changed files (which did not add in #4) as a commit (**Ca2**).

$ git status

$ git add -A

$ git commit -m “Ca2”

1. Modify more than 2 source files, then commit 1 file and reset other files (unchanged) as commit **Ca3**.

$ git status

$ git add s4.header.h

$ git commit -m “Ca3”

$ git status

$ git checkout s4.source1.c

$ git checkout s4.source2.c

1. Push branch master to remote repository (**R0**).

$ git push origin master

1. Edit one file (one of committed file in #6), then add this change to commit **Ca3** (2 ways, commit --amend and soft reset then commit, answer how different between these ways?).

$ git status

$ git add .

$ git commit --amend

vim editor open 🡪 enter :wq

1. Push master branch to remote repository (**R0**) again.

$ git pull origin

$ git push origin master

1. Clone another local repository from local repository/local bare repository created from #1, new repository named **Rb**.

$ cd ..

$ git clone R0 Rb

1. On **Rb**, create new branch named **feature1** from commit **Ca2** of **master** branch.

$ cd Rb

$ git checkout -b feature1 ‘Ca2-Sha’

Ca2-Sha = e57950714bb9884ff7d872b1e358c2950d455c20

1. Edit 2 files then commit to **feature1** branch (commit **Cb1**).

$ git status

$ git add .

$ git commit -m “Cb1”

1. Edit 1 file which is changed in commit **Ca3** (#6 or #8), edit the same location (in file) with change from **Ca3**. After that, commit this commit as **Cb2**.

$ git status

$ git add .

$ git commit -m “Cb2”

1. Merge branch **feature1** (**Cb2**) with **master** branch (**Ca3**). Resolve conflicts then push back to **master** branch of remote repository **R0**.

$ git checkout master

$ git merge feature1

Resolve conflicts

<<<<<<< HEAD

Test\_header

=======

Step 13

>>>>>>> feature1

🡪

Step 13

$ git add .

$ git commit -m “CbMerge”

$ git push origin master

1. Create another remote repository named **R1** by copy **R0**.
2. On **master** branch of **Ra**, edit some files then commit (**Ca4**).

$ cd ..

$ cd Ra

$ git status

$ git add .

$ git commit -m “Ca4”

1. Add new remote named **rmt-r1** has URL points to **R1**.

$ git remote add rmt-r1 ../R1

1. Push new changes of **master** branch (of **Ra**) to **R1**.

$ git pull

$ git push rmt-r1 master

1. On **Ra**, create new branch named **test\_rebase** based on commit **Ca2**.

$ git checkout -b test\_rebase ‘Ca2-Sha’

Ca2-Sha = e57950714bb9884ff7d872b1e358c2950d455c20

1. On **test\_rebase** branch, edit one file which is changed in commit **Ca3** or **Ca4** (#8 or #16), edit the same location (in file) with change from **Ca3** or **Ca16**. Then commit these changes to **Cat1**.

$ git add .

$ git commit -m “Cat1”

1. Rebase the **Cat1** to the latest of **Ra/master** with conflicts resolved.

$ git rebase master

CONFLICT (content): Merge conflict in s4.source1.c

Resolve conflicts by deleting redundant text.

$ git add .

$ git commit -m “CaRebase”

$ git rebase --continue

$ git rebase --abort