Егор@Egor MINGW32 ~/Desktop/lab02/bob

$ git clone https://github.com/Eroproralee/cs-lab03.git project

Cloning into 'project'...

remote: Enumerating objects: 18, done.

remote: Counting objects: 100% (18/18), done.

remote: Compressing objects: 100% (14/14), done.

remote: Total 18 (delta 2), reused 18 (delta 2), pack-reused 0

Unpacking objects: 100% (18/18), done.

Открываем новый терминал Git Bash в каталоге bob.

Егор@Egor MINGW32 ~/Desktop/lab02/bob

$ cd project

заходим в папку пролджект

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git config user.name 'bob(LiEA)'

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git config user.email 'bob@example.com'

настраиваем Git (git config)

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git add main.cpp

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git commit -m 'code:вывод произведения'

[master 6c71ae8] code:вывод произведения

1 file changed, 1 insertion(+)

вводим произведение и коммитим

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git push

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Delta compression using up to 4 threads

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 408 bytes | 204.00 KiB/s, done.

Total 3 (delta 1), reused 0 (delta 0)

remote: Resolving deltas: 100% (1/1), completed with 1 local object.

To https://github.com/Eroproralee/cs-lab03.git

0279feb..6c71ae8 master -> master

отправляем на git.hub

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git fetch

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0

Unpacking objects: 100% (3/3), done.

From https://github.com/Eroproralee/cs-lab002

f9eb5df..2bfaed0 master -> origin/master

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git log --oneline --decorate --all --graph

\* 2bfaed0 (origin/master, origin/HEAD) code:вывод деления

\* f9eb5df (HEAD -> master) code:вывод произведения

\* a1bf04e git:игнорирование bin и obj

\* d80ca03 code:вывод разности

\* c46e930 code:ввод суммы

\* d95d88d code:change code

\* fe14305 build: Добавлен файл проекта

\* 9f83b6c code: заготовка программы

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git pull --ff-only

Updating f9eb5df..2bfaed0

Fast-forward

main.cpp | 1 +

1 file changed, 1 insertion(+)

получаем новую версию с делением с машины Алисы на машину Боба

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git add main.cpp

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git commit -m 'code:вывод минимума'

[master 565d978] code:вывод минимума

1 file changed, 8 insertions(+)

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git push

To https://github.com/Eroproralee/cs-lab002.git

! [rejected] master -> master (fetch first)

error: failed to push some refs to 'https://github.com/Eroproralee/cs-lab002.git'

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.

ищем минимум , комитим и пытаемся отправить на git.hub

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git fetch

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0

Unpacking objects: 100% (3/3), done.

From https://github.com/Eroproralee/cs-lab002

2bfaed0..9937d27 master -> origin/master

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git log

commit 565d9782b6117bd98307435d83dce0df2f013381 (HEAD -> master)

Author: bob (LiEA) <bob@example.com>

Date: Fri Mar 27 13:01:41 2020 +0300

code:вывод минимума

commit 2bfaed0f5ed88c457d7800f96c45c8bd67c3a06f

Author: alice <alice@example.com>

Date: Fri Mar 27 12:49:42 2020 +0300

code:вывод деления

commit f9eb5dff23a0212936749db17a553dc48927b3ff

Author: bob (LiEA) <bob@example.com>

Date: Fri Mar 27 12:45:46 2020 +0300

code:вывод произведения

commit a1bf04e58244d92d7cb3b55a86066995137b47f0

Author: alice <alice@example.com>

Date: Fri Mar 27 12:29:43 2020 +0300

git:игнорирование bin и obj

загружаем коммиты из удаленного хранилища и отобразите историю всех веток

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git rebase origin/master

First, rewinding head to replay your work on top of it...

Applying: code:вывод минимума

Using index info to reconstruct a base tree...

M main.cpp

Falling back to patching base and 3-way merge...

Auto-merging main.cpp

CONFLICT (content): Merge conflict in main.cpp

error: Failed to merge in the changes.

hint: Use 'git am --show-current-patch' to see the failed patch

Patch failed at 0001 code:вывод минимума

Resolve all conflicts manually, mark them as resolved with

"git add/rm <conflicted\_files>", then run "git rebase --continue".

You can instead skip this commit: run "git rebase --skip".

To abort and get back to the state before "git rebase", run "git rebase --abort".

перемещаем коммит боба поверх коммита алисы , однако возникает ошибка , которая означает ,что возникают конфликты , которые надо удалить вручную

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git add main.cpp

Егор@Egor MINGW32 ~/Desktop/lab02/bob/project (master)

$ git rebase --countine

error: unknown option `countine'

usage: git rebase [-i] [options] [--exec <cmd>] [--onto <newbase>] [<upstream>] [<branch>]

or: git rebase [-i] [options] [--exec <cmd>] [--onto <newbase>] --root [<branch>]

or: git rebase --continue | --abort | --skip | --edit-todo

--onto <revision> rebase onto given branch instead of upstream

--no-verify allow pre-rebase hook to run

-q, --quiet be quiet. implies --no-stat

-v, --verbose display a diffstat of what changed upstream

-n, --no-stat do not show diffstat of what changed upstream

--signoff add a Signed-off-by: line to each commit

--ignore-whitespace passed to 'git am'

--committer-date-is-author-date

passed to 'git am'

--ignore-date passed to 'git am'

-C <n> passed to 'git apply'

--whitespace <action>

passed to 'git apply'

-f, --force-rebase cherry-pick all commits, even if unchanged

--no-ff cherry-pick all commits, even if unchanged

--continue continue

--skip skip current patch and continue

--abort abort and check out the original branch

--quit abort but keep HEAD where it is

--edit-todo edit the todo list during an interactive rebase

--show-current-patch show the patch file being applied or merged

-m, --merge use merging strategies to rebase

-i, --interactive let the user edit the list of commits to rebase

-p, --preserve-merges

try to recreate merges instead of ignoring them

--rerere-autoupdate allow rerere to update index with resolved conflict

-k, --keep-empty preserve empty commits during rebase

--autosquash move commits that begin with squash!/fixup! under -i

-S, --gpg-sign[=<key-id>]

GPG-sign commits

--autostash automatically stash/stash pop before and after

-x, --exec <exec> add exec lines after each commit of the editable list

--allow-empty-message

allow rebasing commits with empty messages

-r, --rebase-merges[=<mode>]

try to rebase merges instead of skipping them

--fork-point use 'merge-base --fork-point' to refine upstream

-s, --strategy <strategy>

use the given merge strategy

-X, --strategy-option <option>

pass the argument through to the merge strategy

--root rebase all reachable commits up to the root(s)

добавляем файл в индекс и продолжаем прерванную операцию rebase