

Лабораторная работа №8.

Слияние веток и получение изменений

Цель: научиться проводить слияние веток и получать изменения.

1. Клонировать репозиторий <https://github.com/for-you-nastya/test.git>, с которым мы будем работать:

```
git clone https://github.com/for-you-nastya/test.git
```

Команда `cd test` перейдет в каталог `test`.

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit
$ git clone https://github.com/for-you-nastya/test.git
Cloning into 'test'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0
Receiving objects: 100% (7/7), done.

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit
$ cd test

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/test (master)
$
```

2. Далее создадим новую ветку `mergevetka` и перейдем на нее:

```
git branch mergevetka
```

```
git checkout mergevetka
```

Команда `git branch` без указания имени ветки выводит список всех локальных веток в текущей директории:

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit /test (master)
$ git branch mergevetka

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit /test (master)
$ git checkout mergevetka
Switched to branch 'mergevetka'

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit /test (mergevetka)
$ git branch
  master
* mergevetka
```

3. Создадим новый текстовый файл, добавим его в индекс и проверим статус. Все команды уже известны.

```
git add file.txt
```

```
git commit -m "Added file"
```

`git status`

```
пользователь@LAPTOP-27FOP5IS MINGW64 /c/LabGit/test (mergevetka)
$ git add file.txt

пользователь@LAPTOP-27FOP5IS MINGW64 /c/LabGit/test (mergevetka)
$ git commit -m "Added file"
[mergevetka aa336bc] Added file
1 file changed, 1 insertion(+)
create mode 100644 file.txt

пользователь@LAPTOP-27FOP5IS MINGW64 /c/LabGit/test (mergevetka)
$ git status
On branch mergevetka
Your branch is ahead of 'origin/mergevetka' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Далее введем следующую команду, с которой мы позже познакомимся. Она позволяет внести изменения в удаленный репозиторий

`git push`

```
пользователь@LAPTOP-27FOP5IS MINGW64 /c/LabGit/mergepapka/test (mergevetka)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 327 bytes | 163.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/for-you-nastya/test.git
  35d940f..aa336bc  mergevetka -> mergevetka
```

4. Переключимся на ветку master. Именно она станет принимающей изменения. Далее следует убедиться, что в обеих ветка содержатся последние изменения из удаленного репозитория. Выполните команду, позволяющую получить из него последние изменения:

`git fetch`

```
пользователь@LAPTOP-27FOP5IS MINGW64 /c/LabGit/test (mergevetka)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

пользователь@LAPTOP-27FOP5IS MINGW64 /c/LabGit/test (master)
$ git fetch
```

Также могут пригодиться команды и опции `git fetch`:

`git fetch <репозиторий>` – извлечение всех веток из репозитория, коммитов и файлов

`git fetch <репозиторий> <ветка>` – данные извлекаются из указанной ветки

`git fetch --all` – извлекает все зарегистрированные удаленные репозитории и их ветки

`git fetch --dry-run` – выполняет демонстрационный прогон команды

5. После указанных выше действий можно делать слияние.

Для этого существует команда

`git merge <название сливаемой ветки>`

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit /test (master)
$ git merge mergevetka
Updating 35d940f..aa336bc
Fast-forward
 file.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 file.txt
```

Все прошло успешно: мы сделали слияние ветки mergevetka с веткой master.

6. Проверим команды:

`git branch --merged main` – показывает ветки, объединенные с указанной веткой

`git branch --no-merged` – перечисляет ветки, которые не были объединены

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/test (master)
$ git branch --merged master
* master
  mergevetka

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/test (master)
$ git branch --no-merged
```

Так видно, что ветка mergevetka слита с веткой master.