

Лабораторная работа 4

Зыкун Е.А.

Основы работы с ветками с Git

Клонируем репозиторий - *git clone*

```
zykun@zykun-VirtualBox:~/repository$ git clone git@github.com:andrewbudo/git-clone-lesson.git
Клонирование в «git-clone-lesson»...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Получение объектов: 100% (3/3), готово.
```

Убедились, что репозитории связаны

```
git remote -v
```

```
zykun@zykun-VirtualBox:~/repository$ git remote -v
origin  git@github.com:Zykunchik/repository.git (fetch)
origin  git@github.com:Zykunchik/repository.git (push)
```

Клонируем репозиторий и выводим содержимое файла task.txt

```
zykun@zykun-VirtualBox:~$ git clone git@github.com:andrewbudo/git-clone-practice.git
Клонирование в «git-clone-practice»...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
Получение объектов: 100% (10/10), готово.
zykun@zykun-VirtualBox:~$ cd git-clone-practice/
zykun@zykun-VirtualBox:~/git-clone-practice$ cat task.txt
Страшный-пугающий секрет
```

Потренируйтесь выполнять «форк». Перейдите [по этой ссылке](#) и нажмите на кнопку Fork в правом верхнем углу. Нажимаем *New Fork*. Далее клонируем удаленный репозиторий.

```
zykun@zykun-VirtualBox:~$ git clone git@github.com:Zykunchik/git-basics.git
Клонирование в «git-basics»...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Получение объектов: 100% (4/4), готово.
zykun@zykun-VirtualBox:~$ cd repository/
zykun@zykun-VirtualBox:~/repository$ touch test.txt
zykun@zykun-VirtualBox:~/repository$ git add test.txt
zykun@zykun-VirtualBox:~/repository$ git commit -m 'my first commit'
[master 9865afa] my first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test.txt
```

Задание для самостоятельной работы


Выбираем понравившийся репозиторий и переходим по ссылке, создаем *New Fork* и клонируем репозиторий на компьютер.

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)


Required fields are marked with an asterisk ().*

Owner *

 Zykunchik

Repository name *


git-basics

 git-basics is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

☒ **Copy the `main` branch only**
Contribute back to [andrewbudo/git-basics](#) by adding your own branch. [Learn more.](#)

 You are creating a fork in your personal account.

Create fork

```
zykun@zykun-VirtualBox:~/repository$ git clone git@github.com:Zykunchik/awesome.git
Клонирование в «awesome»...
remote: Enumerating objects: 3166, done.
remote: Counting objects: 100% (118/118), done.
remote: Compressing objects: 100% (56/56), done.
remote: Total 3166 (delta 72), reused 103 (delta 61), pack-reused 3048
Получение объектов: 100% (3166/3166), 1.43 МиБ | 1.08 МиБ/с, готово.
Определение изменений: 100% (1797/1797), готово.
```

Практическая работа №1

Открываем репозиторий Феди. Клонировем его к себе на локальный репозиторий. Видим исполняемый файл `check.sh`. Вводим команды

```
$ chmod +x check.sh # эта команда сделает файл исполняемым
$ ./check.sh # эта команда исполнит скрипт
```

Секретный ключ: practicum

```
zykun@zykun-VirtualBox:~/repository$ git clone git@github.com:Zykunchik/git-fork-case.git
Клонирование в «git-fork-case»...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Получение объектов: 100% (4/4), готово.
zykun@zykun-VirtualBox:~/repository$ cd git-fork-case/
zykun@zykun-VirtualBox:~/repository/git-fork-case$ chmod +x check.sh
zykun@zykun-VirtualBox:~/repository/git-fork-case$ ./check.sh
Секретный ключ: practicum
```

Создайте тестовый проект `learn_branches`. Добавьте в него файл `README.md`, проиндексируйте изменения и сделайте коммит:

```
git commit -m "Выполнить первый коммит".
```

Вспомните, что несколько команд можно объединять в одной строке терминала символом логического «и» — `&&`.

```
$ mkdir learn_branches && cd learn_branches && git init # создали новый репозиторий
$ touch README.md # создали файл
$ git add . # команда git add с флагом-точкой подготовит к сохранению текущую папку;
# вместо этого можно вызвать git add -all
$ git commit -m "Выполнить первый коммит"
```

Репозиторий создан, файлы добавлены и закоммичены. В репозитории появилась главная ветка. Это можно проверить командой для просмотра веток `git branch`.

```
$ git branch
* main # мы в основной ветке
# чтобы выйти из просмотра веток, может понадобится Q!
```

При вызове `git branch` выводятся ветки, которые есть в проекте. Звёздочкой (*) отмечено, в какой ветке вы находитесь в текущий момент.

Создать ветку - `git branch "название_ветки"`

Для создания веток в Git есть команда `git branch` с параметром в виде названия ветки:

```
git branch <название_ветки>.
```

Шагаем с ветки на ветку

Чтобы начать работу в `feature/add-branch-info`, перейдите в неё с помощью команды `git checkout` с флагом — названием ветки: `git checkout feature/add-branch-info`.

```
$ git checkout feature/add-branch-info # перешли в новую ветку
Switched to branch 'feature/add-branch-info'
$ git branch # проверили
* feature/add-branch-info # теперь находимся тут
main
```

Строчка `Switched to branch...` (англ. «переключено на ветку...») сообщает, на какую ветку вы только что переключились.

Создать ветку и сразу переключиться на неё – `git checkout -b <название_ветки>`

Можно создать ветку и сразу начать в ней работать. За это отвечает команда `git checkout` с флагом `-b` (от англ. `branch`) и названием ветки. Эта команда делает то же самое, что и последовательность команд `git branch %название-ветки% && git checkout %название-ветки%`. Убедитесь, что вы в основной ветке. Затем создайте ещё одну ветку для исправления ошибок `bugfix/fix-branch` и сразу переключитесь на неё.

Сравниваем ветки — `git diff <название_ветки1> <название_ветки2>`

```
diff --git a/README.md b/README.md
index 86b1ff4..fff4920 100644
--- a/README.md
+++ b/README.md
@@ -1,3 +1,5 @@
# Ветки в Git
```

Чтобы посмотреть все активные ветки в проекте, нужно вызвать команду `git branch` без аргументов. + Для сравнения веток есть команда `git diff`.

При сравнении вы также можете использовать название ветки и хеш коммита. Для этого сначала выполните команду `git log --oneline`, чтобы вывести список коммитов.

```
git log --oneline
2ea56ab (HEAD -> feature/diff) Добавить git diff в README
de8b09b (main) Добавить git branch в README
7ad18bd Добавить файл README
```

Практическая работа №2. Помогаем другу разобраться в ветках



```
zykun@zykun-VirtualBox:~$ cd git-branch-case/
zykun@zykun-VirtualBox:~/git-branch-case$ git log --oneline
6b5c26d (HEAD -> main, origin/main, origin/HEAD) fix codePg
c8b0002 Merge pull request #1 from andrewbudo/merge-main
09c03a5 Добавить параграф в readme.md
ce1a489 Создать readme.md
f8f2dba Initial commit
zykun@zykun-VirtualBox:~/git-branch-case$ git diff cela489 09c03a5
fatal: неоднозначный аргумент «cela489»: неизвестная редакция или не путь в рабочем каталоге.
Используйте «--» для отделения путей от редакций, вот так:
«git <команда> [<редакция>...] -- [<файл>...]»
zykun@zykun-VirtualBox:~/git-branch-case$ git diff ce1a489 09c03a5
diff --git a/README.md b/README.md
index 8f326b4..76021df 100644
--- a/README.md
+++ b/README.md
@@ -1 +1,3 @@
-#
+#
\ No newline at end of file
+<CB><EE><EA><E0><EB><FC><ED><FB><E5> <E2><E5><F2><EA><EB> <F1><E2><FF><E7><E0><ED><FB> <F1> <F3>
<F3><E4><E0><EB><B8><ED><ED><FB><EC><E8>.
```


Задание для самостоятельной работы

```
zykun@zykun-VirtualBox:~/MergeProject$ touch README.md
zykun@zykun-VirtualBox:~/MergeProject$ git add -all
error: имелось в виду '--all' (с двумя дефисами)?
zykun@zykun-VirtualBox:~/MergeProject$ git add --all
error: '.local/share/Trash/files/git-branch-case/' does not have a commit checked out
fatal: ошибка при добавлении файлов
zykun@zykun-VirtualBox:~/MergeProject$ git commit -m 'Создать README.md'
Текущая ветка: master
```

Начальный коммит

Неотслеживаемые файлы:

(используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)

```
../.bash_history
../.bash_logout
../.bashrc
../.cache/
../.config/
../.gitconfig
../.gnupg/
../.lessht
../.local/
../.profile
../.ssh/
../.sudo_as_admin_successful
../
../del
../git-basics/
../git-branch-case/
../git-clone-practice/
../hello.cpp
../learn_branches/
../my-new-file.txt
../repository/
../snap/
```

индекс пуст, но есть неотслеживаемые файлы

(используйте «git add», чтобы проиндексировать их)

```
zykun@zykun-VirtualBox:~/MergeProject$ git add README.md
zykun@zykun-VirtualBox:~/MergeProject$ git commit -m 'Создать README.md'
[master (корневой коммит) 23f762e] Создать README.md
1 file changed, 1 insertion(+)
create mode 100644 MergeProject/README.md
zykun@zykun-VirtualBox:~/MergeProject$ git branch
* master
zykun@zykun-VirtualBox:~/MergeProject$ git branch feature/new-line
zykun@zykun-VirtualBox:~/MergeProject$ git checkout feature/new-line
Переключились на ветку «feature/new-line»
zykun@zykun-VirtualBox:~/MergeProject$ git branch
* feature/new-line
  master
zykun@zykun-VirtualBox:~/MergeProject$ git add README.md
```

```
zykun@zykun-VirtualBox:~/MergeProject$ git add README.md
zykun@zykun-VirtualBox:~/MergeProject$ git commit -m 'Добавить новую строку в README.md'
[feature/new-line b1dd1df] Добавить новую строку в README.md
1 file changed, 1 insertion(+)
zykun@zykun-VirtualBox:~/MergeProject$ git checkout master
Переключились на ветку «master»
zykun@zykun-VirtualBox:~/MergeProject$ git branch
  feature/new-line
* master
zykun@zykun-VirtualBox:~/MergeProject$ git merge feature/new-line
git: «merge» не является командой git. Смотрите «git --help».
```

Самые похожие команды:

merge

```
zykun@zykun-VirtualBox:~/MergeProject$ git merge feature/new-line
```

Обновление 23f762e..b1dd1df

Fast-forward

```
MergeProject/README.md | 1 +
1 file changed, 1 insertion(+)
zykun@zykun-VirtualBox:~/MergeProject$ git log
commit b1dd1dfd89c38c31a890706af6c6888fcbf6cb (HEAD -> master, feature/new-line)
Author: zykun <zykunkelly@gmail.com>
Date: Sun Dec 17 13:46:21 2023 +0300
```

Добавить новую строку в README.md

```
commit 23f762e078355e58308b184edd12497ab999dd47
Author: zykun <zykunkelly@gmail.com>
Date: Sun Dec 17 13:42:38 2023 +0300
```

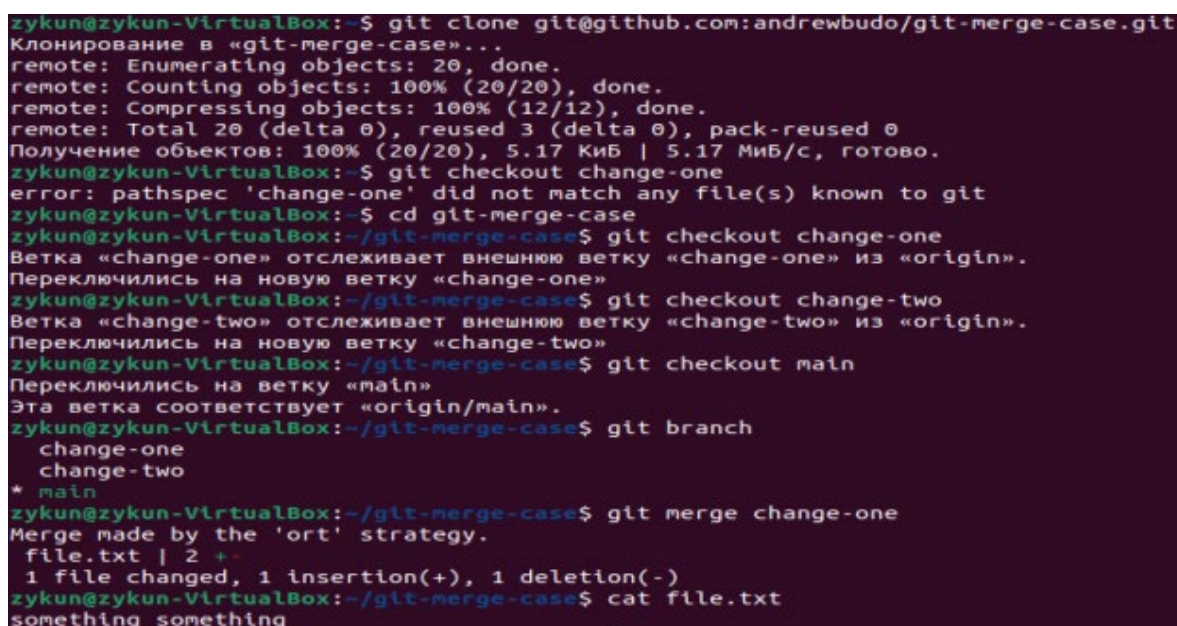
Создать README.md

```
zykun@zykun-VirtualBox:~/MergeProject$ git log -oneline
fatal: неопознанный аргумент: -oneline
zykun@zykun-VirtualBox:~/MergeProject$ git log --oneline
b1dd1df (HEAD -> master, feature/new-line) Добавить новую строку в README.md
23f762e Создать README.md
```

Практическая №3 Учим друга «мёржить» ветки

Склонируйте [этот проект](#). Создайте локальные версии всех веток с помощью команды `git checkout`. Затем снова переключитесь на ветку `main`.

```
$ git checkout change-one
branch 'change-one' set up to track 'origin/change-one'.
Switched to a new branch 'change-one'
# создали локальную версию ветки change-one
$ git checkout change-two
branch 'change-two' set up to track 'origin/change-two'.
Switched to a new branch 'change-two'
# создали локальную версию ветки change-two
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
# переключились на ветку main
```



```
zykun@zykun-VirtualBox:~$ git clone git@github.com:andrewbudo/git-merge-case.git
Клонирование в «git-merge-case»...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 20 (delta 0), reused 3 (delta 0), pack-reused 0
Получение объектов: 100% (20/20), 5.17 КиБ | 5.17 МБ/с, готово.
zykun@zykun-VirtualBox:~$ git checkout change-one
error: pathspec 'change-one' did not match any file(s) known to git
zykun@zykun-VirtualBox:~$ cd git-merge-case
zykun@zykun-VirtualBox:~/git-merge-case$ git checkout change-one
Ветка «change-one» отслеживает внешнюю ветку «change-one» из «origin».
Переключились на новую ветку «change-one»
zykun@zykun-VirtualBox:~/git-merge-case$ git checkout change-two
Ветка «change-two» отслеживает внешнюю ветку «change-two» из «origin».
Переключились на новую ветку «change-two»
zykun@zykun-VirtualBox:~/git-merge-case$ git checkout main
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
zykun@zykun-VirtualBox:~/git-merge-case$ git branch
  change-one
  change-two
* main
zykun@zykun-VirtualBox:~/git-merge-case$ git merge change-one
Merge made by the 'ort' strategy.
 file.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
zykun@zykun-VirtualBox:~/git-merge-case$ cat file.txt
something something
```

Практическая работа №4. Выходим из отпуска — и узнаём, что проект изменился

Время действовать — скачиваем изменения

1. Скачайте архив с проектом, который уже привязан к удалённому репозиторию. Именно в таком состоянии был проект Феди на его локальном компьютере, когда герой вернулся из отпуска.
2. Перейдите в проект.
3. Подтяните в ветку `main` изменения, которые внёс Боря, пока Федя отдыхал.
4. Выполните «мёрж» ветки `main` в ветку Феди `feature/feodors-new-project`.

Создаём пул-реквест Настало время создать пул-реквест! Чтобы не засорять репозиторий Бори, выполните «форк» проекта. В дальнейшем всю работу вам нужно будет продолжить в проекте, который вы открыли из архива. Сделайте следующее:

1. Удалите текущий привязанный origin.

```
$ git remote rm origin # эта команда удалит текущий origin
```

2. Добавьте новый origin. В качестве ссылки используйте ваш репозиторий. Ссылка будет доступна в вашем репозитории по клику на кнопку Code в интерфейсе GitHub.

```
$ git remote add origin git@github.com:%ВАШ_АККАУНТ%/git-github-case.git
```

3. «Запушьте» ветку в удалённый репозиторий. Если всё правильно, после выполнения команды Git выведет ссылку, по которой вы сможете открыть пул-реквест.
4. Перейдите по ссылке и создайте пул-реквест в свой «форк» проекта. Чтобы сделать это именно в свой репозиторий, не забудьте выбрать его в окне создания пул-реквеста.

```
zykun@zykun-VirtualBox:~$ cd git-github-case/
zykun@zykun-VirtualBox:~/git-github-case$ git clone git@github.com:andrewbudo/git-github-case.git
Клонирование в «git-github-case»...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
Получение объектов: 100% (10/10), готово.
zykun@zykun-VirtualBox:~/git-github-case$ git branch
* feature/feodors-new-project
  main
zykun@zykun-VirtualBox:~/git-github-case$ git checkout main
M   README.md
Переключились на ветку «main»
Ветка отстает от «origin/main» на 2 коммита и может быть быстро перемотана.
(используйте «git pull», чтобы обновить вашу локальную ветку)
zykun@zykun-VirtualBox:~/git-github-case$ git pull
Обновление 952dacc..e531162
Fast-forward
 boris-project.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 boris-project.txt
zykun@zykun-VirtualBox:~/git-github-case$ git merge main
Уже актуально.
zykun@zykun-VirtualBox:~/git-github-case$ git remote set-url origin git@github.com:Zykunchik/git-github-case.git
bash: Zykunchik/git-github-case.git: Нет такого файла или каталога
zykun@zykun-VirtualBox:~/git-github-case$ git remote set-url origin git@github.com:Zykunchik/git-github-case.git
bash: Zykunchik/git-github-case.git: Нет такого файла или каталога
zykun@zykun-VirtualBox:~/git-github-case$ git remote set-url origin git@github.com:Zykunchik/git-github-case.git
zykun@zykun-VirtualBox:~/git-github-case$ git remote -v
origin  git@github.com:Zykunchik/git-github-case.git (fetch)
origin  git@github.com:Zykunchik/git-github-case.git (push)
zykun@zykun-VirtualBox:~/git-github-case$ git push -u feature/feodors-new-project
```