

**Федеральное государственное образовательное бюджетное  
учреждение высшего образования  
«Ульяновский государственный университет»  
Факультет математики информационных и авиационных технологий**

**Лабораторная работа: Git, Git LFS  
По курсу «Системы искусственного интеллекта»**

**Выполнил студент группы МОАИС-О-21/1**

**Кузина Н. С.**

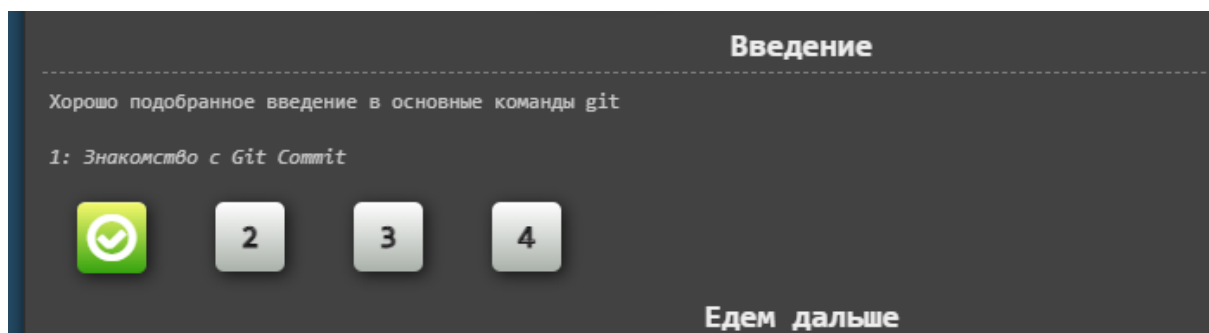
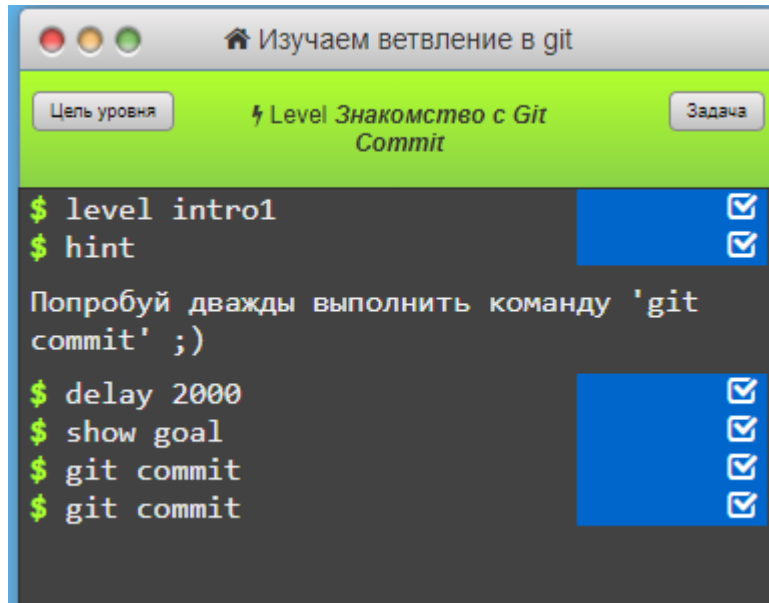
**Проверил**

Ульяновск 2023

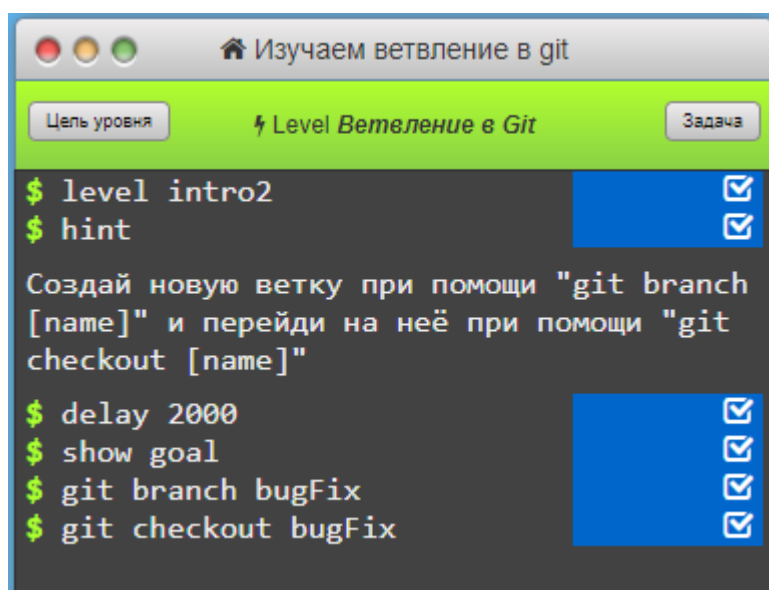
# Git

## ОСНОВЫ

1: Знакомство с Git Commit



2: Ветвление в Git



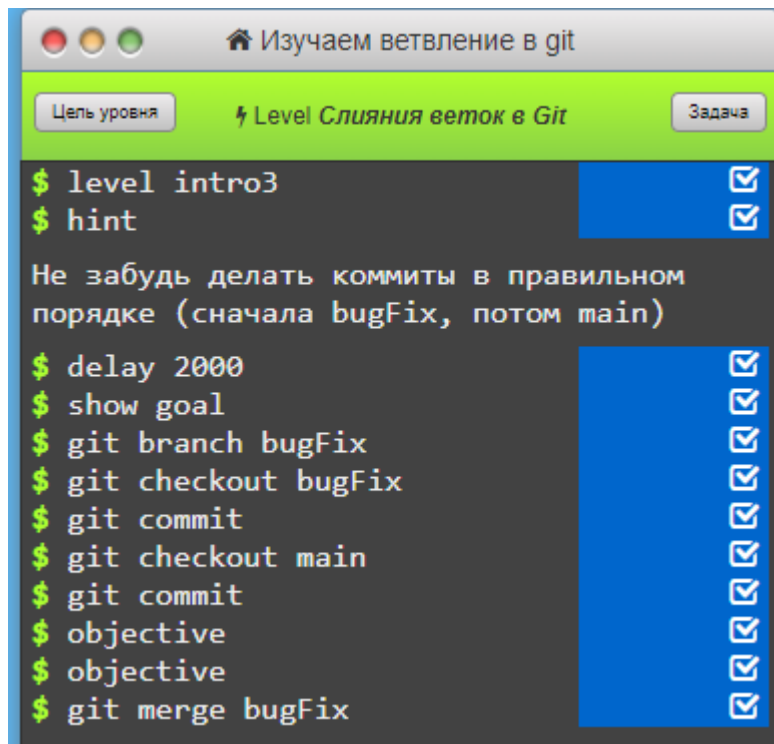
Введение

Хорошо подобранное введение в основные команды git

1: Знакомство с Git Commit


✓ ✓ 3 4


### 3: Слияния веток в Git





Хорошо подобранное введение в основные команды git

1: Знакомство с Git Commit













Хорошо подобранное введение в основные команды git

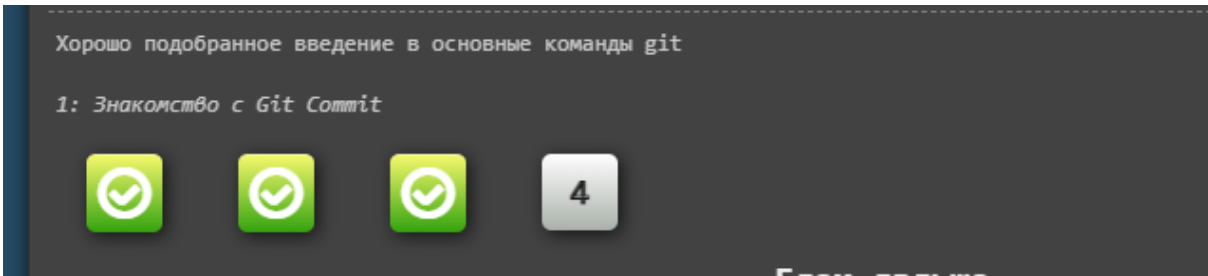
1: Знакомство с Git Commit



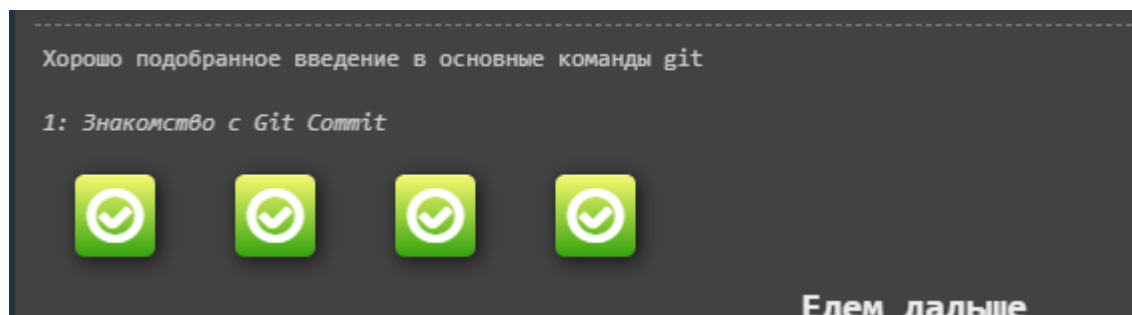
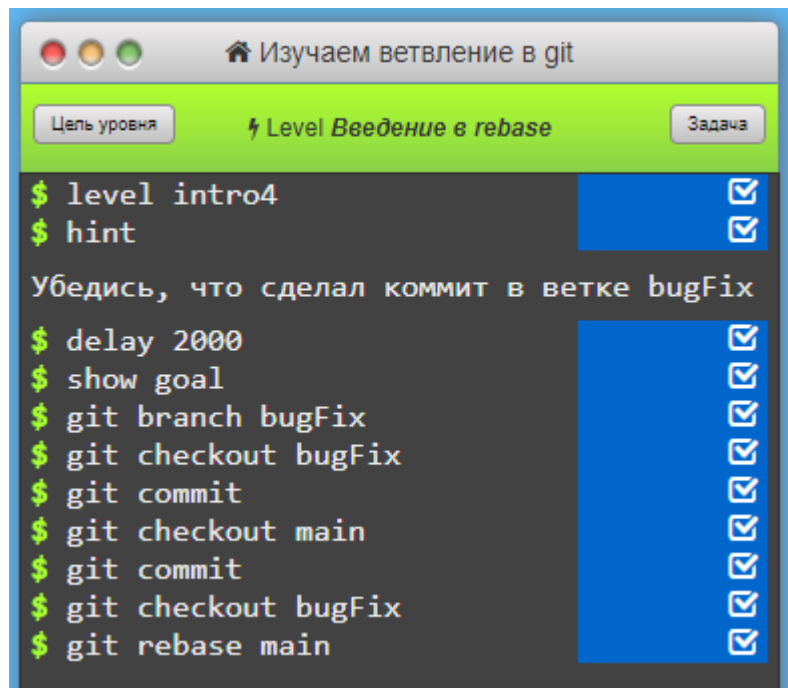




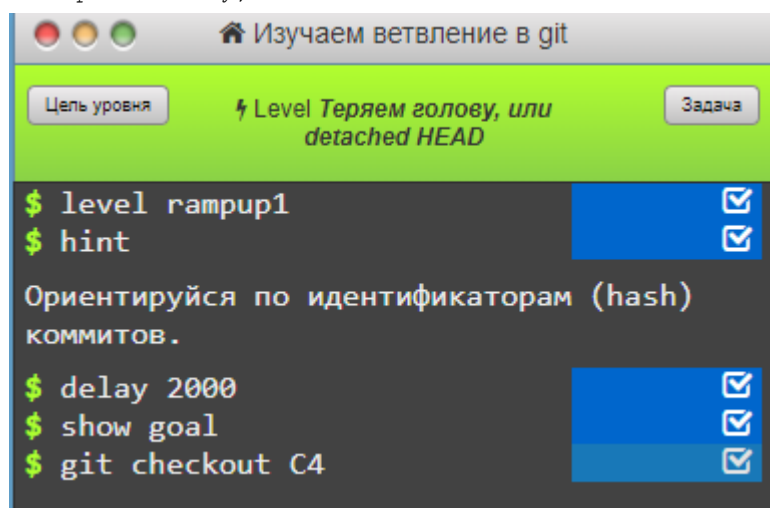




#### 4: Введение в rebase



#### 1: Теряем голову, или detached HEAD



Следующая порция абсолютной git-крутотенюшки. Проголодались?

1: Теряем голову, или *detached HEAD*



Перемещаем труды туда-сюда

2: Относительные ссылки (^)

Изучаем ветвление в git

Цель уровня

⚡ Level Относительные ссылки (^)

Задача

\$ level rampup2

\$ hint

Не забудь оператор ^^^

\$ delay 2000

\$ show goal

\$ objective

\$ git checkout bugFix

\$ git checkout C4

\$ git checkout HEAD^

☒

☒

☐

☒

☒

☒

☒

☒

☒


Следующая порция абсолютной git-крутотенюшки. Проголодались?

1: Теряем голову, или *detached HEAD*



Перемещаем труды туда-сюда

### 3: Относительные ссылки №2

 Изучаем ветвление в git

Цель уровня ⚡ Level Относительные ссылки №2 Задача

```
$ level rampup3
$ hint
```

Понадобится использовать как минимум одну прямую ссылку (хеш), чтобы пройти этот уровень

```
$ delay 2000
$ show goal
$ git branch -f bugFix HEAD~3
```

Коммит C2 не содержит ~3

```
$ git checkout C5
$ git branch -f bugFix HEAD~3
$ git checkout C4
$ git branch -f main C6
$ git checkout HEAD~2
```

The command "git checkout HEAD~2" isn't supported, sorry!

```
$ git checkout C1
```

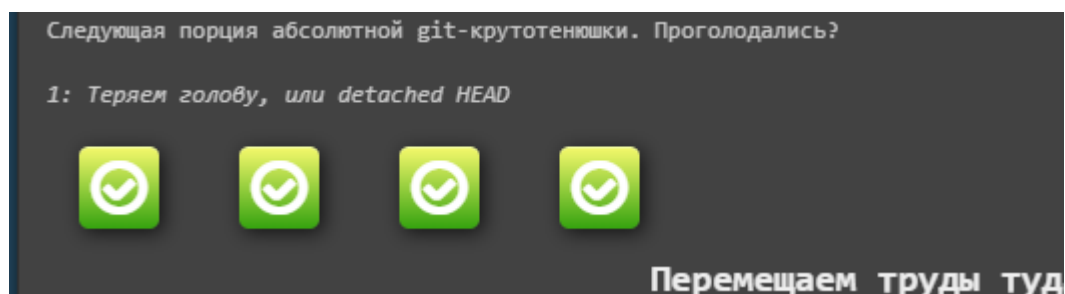
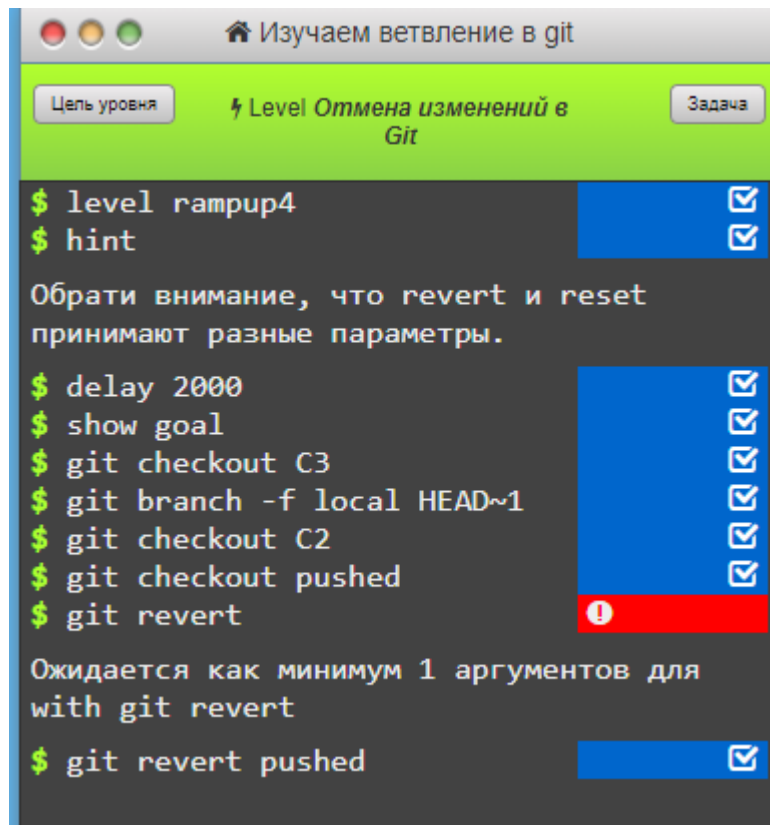
Следующая порция абсолютной git-крутотенюшки. Проголодались?

1: Теряем голову, или detached HEAD

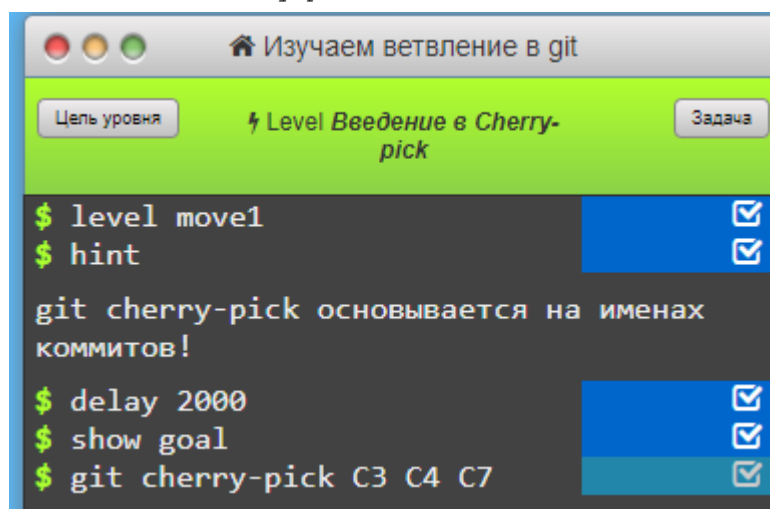


Перемещаем труды туда

#### 4: Отмена изменений в Git



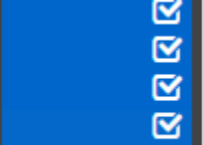
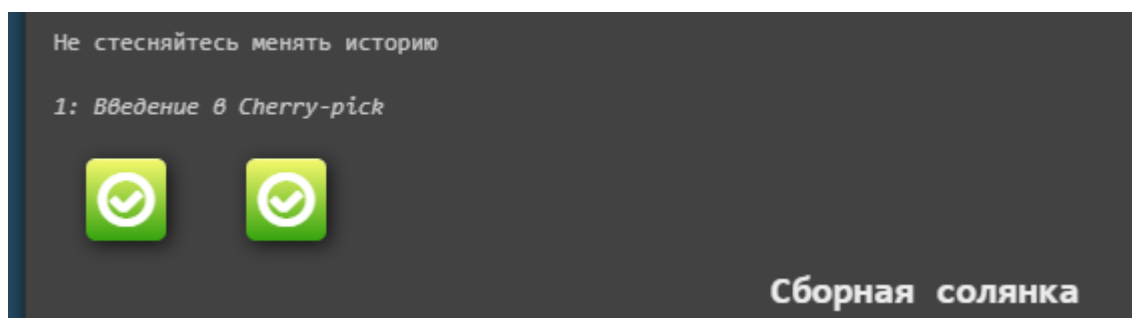
#### 1: Введение в Cherry-pick





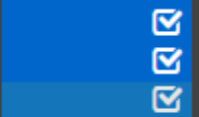
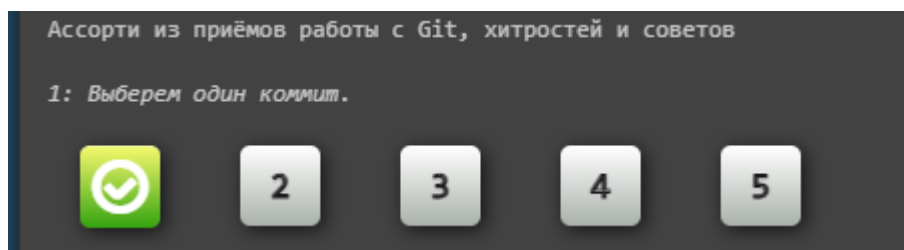
2: Введение в интерактивный Rebase

```
$ reset
$ git rebase -i HEAD~4
$ git rebase -i main C4
$ git branch -f main C4'
```

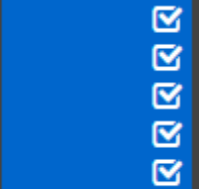
1: Выберем один коммит.

```
$ reset
$ git rebase -i HEAD~3
$ git branch -f main C4'
```

2: Жонглируем коммитами

```
$ reset
$ git rebase -i HEAD~2
$ git rebase -i HEAD~1
$ git rebase -i HEAD~2
$ git branch -f main C3''
```





Ассорти из приёмов работы с Git, хитростей и советов

1: Выберем один коммит.



3: Жонглируем коммитами №2

```
$ reset
$ git checkout main
$ git cherry-pick C2
$ git cherry-pick C2'
```

Коммит C2' существует, отменяю!

```
$ git branck -f main C1
```

The command "git branck -f main C1" isn't supported, sorry!

```
$ git branch -f main C1
$ git cherry-pick C2'
$ git cherry-pick C3
```

Ассорти из приёмов работы с Git, хитростей и советов

1: Выберем один коммит.



4: git tag

```
$ reset
$ git tag v0 C1
$ git tag v1 C2
$ git checkout C2
```

Ассорти из приёмов работы с Git, хитростей и советов

1: Выберем один коммит.








5: *Git describe*

```
$ git describe main
v0_2_gC2
$ git describe side
v1_1_gC4
$ git describe bugFix
v1_2_gC6
$ git commit
```

Ассорти из приёмов работы с Git, хитростей и советов

1: *Выберем один коммит.*




    

1: *Rebase over 9000 раз*

```
$ reset
$ git rebase main bugFix
$ git rebase bugFix side
$ git rebase side another
$ git branch -f main C7'
```

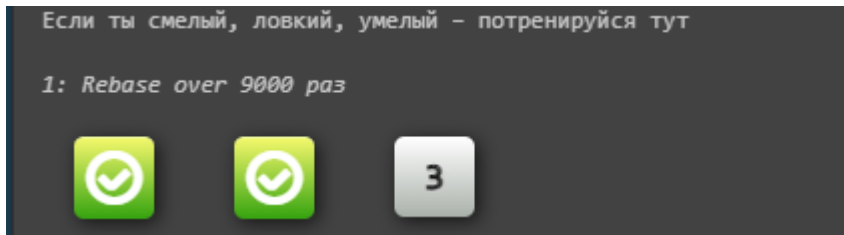
Если ты смелый, ловкий, умелый – потренируйся тут

3: *Спутанные ветки*

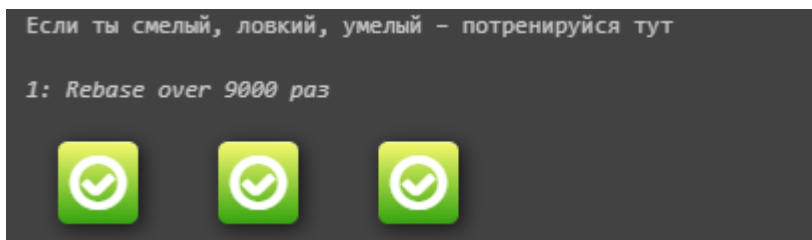
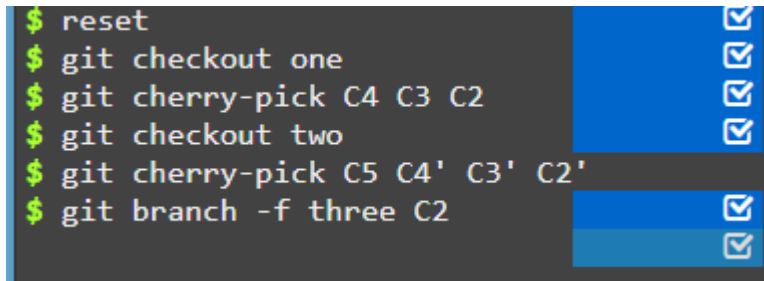
  

2: *Здоровая семья, или несколько родителей*

```
$ reset
$ git checkout HEAD~^2~
$ git branch bugWork
$ git checkout main
```

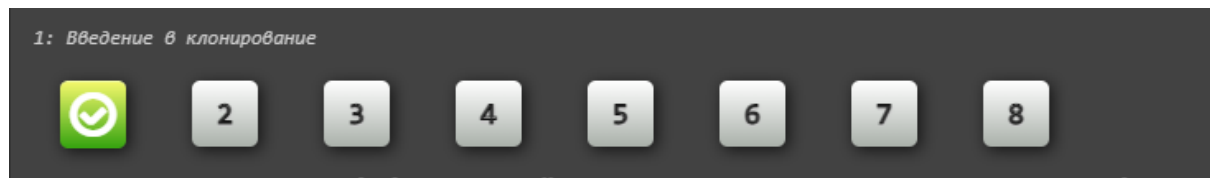
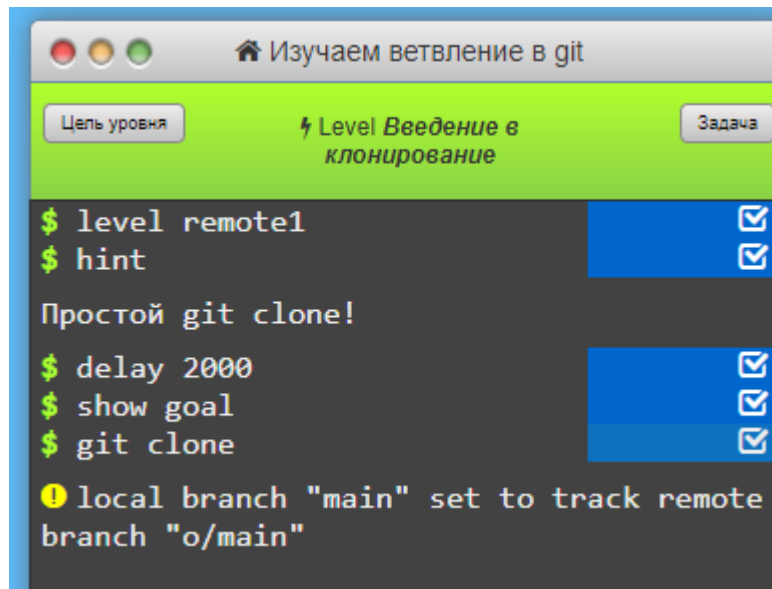


3: Спутанные ветки

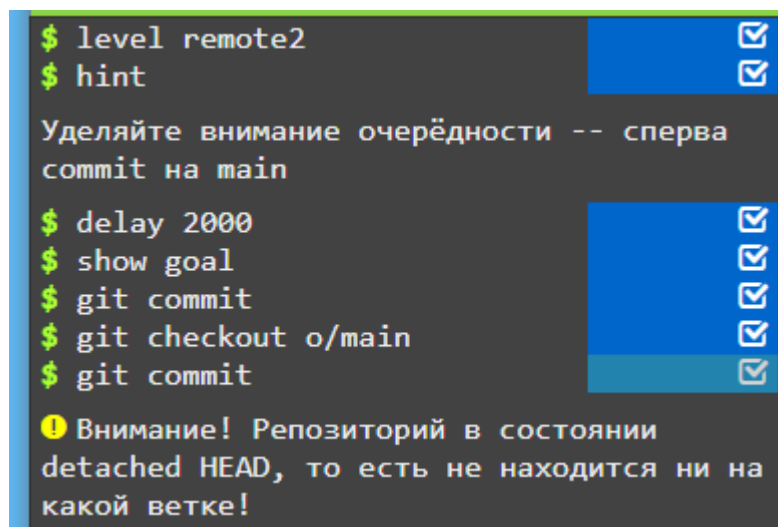


## Удаленные репозитории

1: Введение в клонирование



2: Удалённые ветки



### 3: Git fetch

```
$ level remote3
$ hint
Просто выполните git fetch!
$ delay 2000
$ show goal
$ git fetch
```

1: Введение в клонирование



### 4: Git pull

```
$ level remote4
$ hint
Запустите команду git pull !
$ delay 2000
$ show goal
$ git pull
```

1: Введение в клонирование



### 5: Коллективная работа

```
$ reset
$ git clone
! local branch "main" set to track remote
branch "o/main"
$ git fakeTeamwork main 2
$ git commit -m "C4"
$ git pull
```

1: Введение в клонирование



6: Git push

```
$ delay 2000
$ show goal
$ git commit
$ git commit
$ git push
```

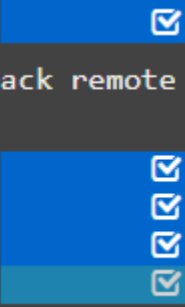


6: Git push



7: Расхождение в истории

```
$ git clone
! local branch "main" set to track remote
branch "o/main"
$ git fakeTeamwork
$ git commit
$ git pull --rebase
$ git push
```

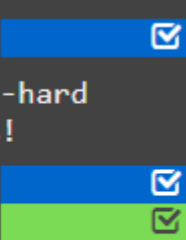


7: Расхождение в истории



8: Заблокированная ветвь main

```
$ git reset --hard o/main
! По умолчанию будет выполнен --hard
reset, эту опцию можно опускать!
$ git checkout -b feature C2
$ git push origin feature
! local branch "feature" set to track
remote branch "o/feature"
```

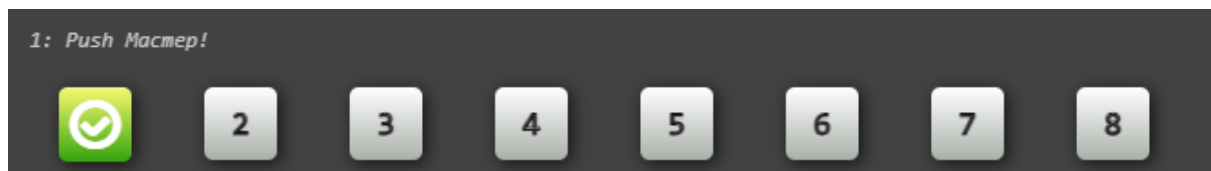


8: Заблокированная ветвь main



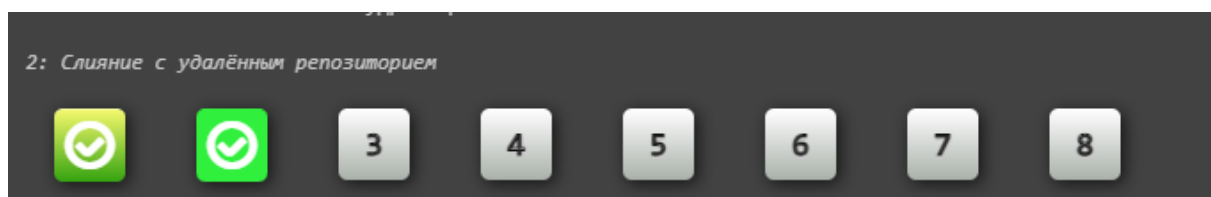
1: Push Мастер!

```
$ git rebase side1 side2
$ git rebase side2 side3
$ git rebase side3 main
Выполняю Fast forward...
$ git pull --rebase
$ git push
```



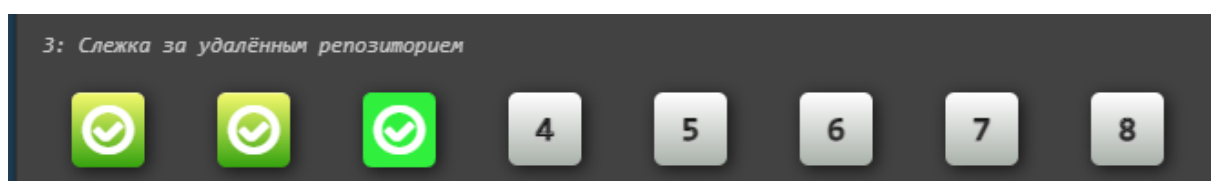
2: Слияние с удалённым репозиторием

```
$ git checkout main
$ git pull
Выполняю Fast forward...
$ git merge side1
$ git merge side2
$ git merge side3
$ git push
```



3: Слежка за удалённым репозиторием

```
$ git checkout -b side o/main
! local branch "side" set to track remote
branch "o/main"
$ git commit
$ git pull --rebase
$ git push
```



4: Аргументы git push

```
$ git push origin main
$ git push origin foo
```

1: Push Мастер!



5: Аргументы для push -- расширенная версия!

```
$ git push origin main~1:foo
$ git push origin foo:main
```

5: Аргументы для push -- расширенная версия!



6: Аргументы для fetch

```
$ git fetch origin main~1:foo
$ git push origin foo:main
```

Удаленный репозиторий разошелся с вашим локальным репозиторием, поэтому выгрузка ваших изменений не может быть в режиме fast forward (и следовательно ваш push будет отклонён). Пожалуйста, удалите изменения в удаленном репозитории которые, объедините их в эту ветку и попробуйте еще раз. Вы можете сделать это с помощью git pull или git pull --rebase

```
$ git fetch origin foo:main
$ git checkout foo
$ git merge main
```

1: Push Мастер!



7: Пустой источник

```
$ reset
$ git push origin :foo
$ git fetch origin :bar
```

7: Пустой источник





8: Аргументы для pull

```
$ git pull origin bar:foo  
$ git pull origin main:side
```

8: Аргументы для pull




Выбери уровень

Основы Удаленные репозитории

### Push & Pull - удалённые репозитории в Git!

Настало время поделиться своими единицами и нулями. Время коллективного программирования


1: Введение в клонирование



### Через origin - к звёздам. Продвинутое использование Git Remotes

Весело было быть всеильным мудрым правителем...

6: Аргументы для fetch



## Git LFS

1. Установите git-lfs. Например, используя менеджер пакетов в Ubuntu\Deb:  
`sudo apt-get install git-lfs`

```
yatoshk@yatoshk: ~  
yatoshk@yatoshk:~$ sudo apt-get install git-lfs  
[sudo] password for yatoshk:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  bridge-utils containerd pigz ubuntu-fan wmdocker  
Use 'sudo apt autoremove' to remove them.  
The following NEW packages will be installed:  
  git-lfs  
0 upgraded, 1 newly installed, 0 to remove and 270 not upgraded.  
Need to get 3 503 kB of archives.  
After this operation, 10,4 MB of additional disk space will be used.  
Get:1 http://ru.archive.ubuntu.com/ubuntu/jammy-updates/universe amd64 git-lfs a  
md64 3.0.2-1ubuntu0.2 [3 503 kB]  
Fetched 3 503 kB in 5s (675 kB/s)  
Selecting previously unselected package git-lfs.  
(Reading database ... 188866 files and directories currently installed.)  
Preparing to unpack .../git-lfs_3.0.2-1ubuntu0.2_amd64.deb ...  
Unpacking git-lfs (3.0.2-1ubuntu0.2) ...  
Setting up git-lfs (3.0.2-1ubuntu0.2) ...  
Processing triggers for man-db (2.10.2-1) ...  
yatoshk@yatoshk:~$
```

2. Запустите следующую команду. Ее потребуется запустить лишь однажды, для каждого юзера в системе `bashlearn git branching.js.org`

`git lfs install`

```
yatoshk@yatoshk:~$ git lfs install  
Git LFS initialized.  
yatoshk@yatoshk:~$
```

3. Создайте репозиторий или перейдите в уже готовый. Если у вас нет "большого" файла для теста - можете создать файл(ы) фиксированного размера с помощью команды:

`truncate -s 10M ten_megabyte_file.txt`

```
yatoshk@yatoshk:~$ git clone https://github.com/Yatoshk/BigData.git  
Клонирование в «BigData»...  
remote: Enumerating objects: 7, done.  
remote: Counting objects: 100% (7/7), done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 7 (delta 0), reused 0 (delta 0), pack-reused 0  
Получение объектов: 100% (7/7), 11.25 КиБ | 65.00 КиБ/с, готово.  
yatoshk@yatoshk:~/BigData$
```

4. Теперь нужно создать правило для слежения за большими файлами. Например, мы тренируем нейронку и храним картинки\звук\таблички и т.п. в папке Data. Тогда пропишем

`git lfs track Data`

```
yatoshk@yatoshk:~/BigData$ git lfs track Data  
Tracking "Data"
```

5. Или мы знаем, что все наши таблички с данными будут в формате CSV. В таком случае можно использовать такую команду:

```
git lfs track "*.csv"
```

6. Убедитесь, что ваши большие данные не были предварительно исключены из индексации Гитом в файле .gitignore

```
git check-ignore -v *
```

Если вы видите правило, которое заставляет Гит игнорировать ваши большие данные - отредактируйте файл.

7. Убедимся, что файл с правилами для Git LFS индексируется. Добавим его.

```
git add .gitattributes
```

8. Теперь всё готово. Просто работайте с файлами так, как это делали раньше. Например:

```
git add Data
```

```
git commit -m "added Data"
```

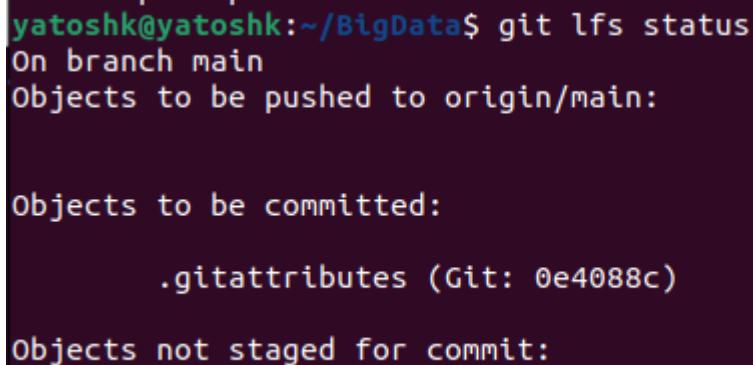
```
git push
```

## Отчет

Прикрепите следующие скриншоты в ЭИОС

1. Результат команды

```
git lfs status
```



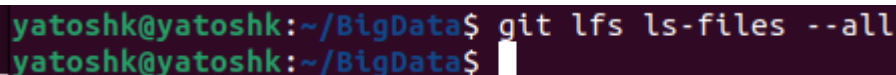
```
yatoshk@yatoshk:~/BigData$ git lfs status
On branch main
Objects to be pushed to origin/main:

Objects to be committed:
    .gitattributes (Git: 0e4088c)

Objects not staged for commit:
```

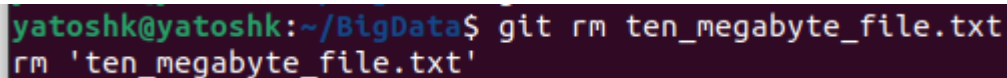
2. Результат команды

```
git lfs ls-files --all
```

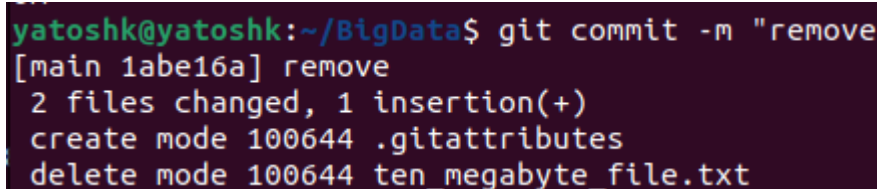


```
yatoshk@yatoshk:~/BigData$ git lfs ls-files --all
yatoshk@yatoshk:~/BigData$
```

3. Самостоятельно удалить большой файл и запустить изменения

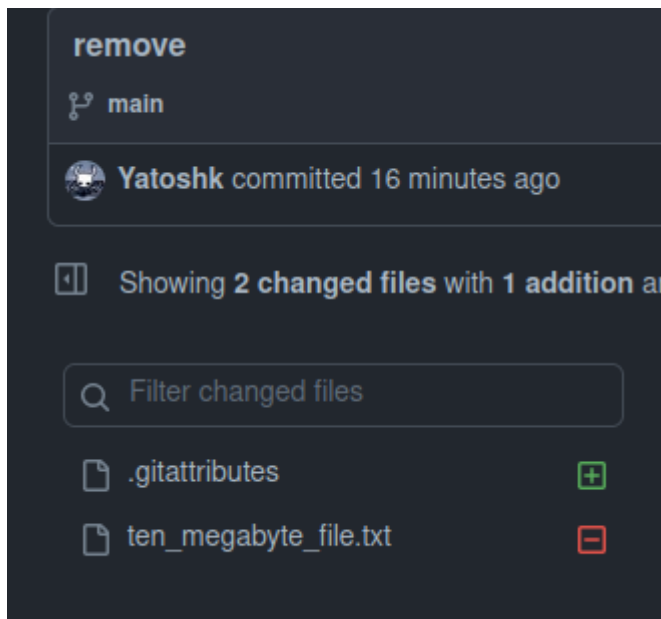


```
yatoshk@yatoshk:~/BigData$ git rm ten_megabyte_file.txt
rm 'ten_megabyte_file.txt'
```



```
yatoshk@yatoshk:~/BigData$ git commit -m "remove
[main 1abe16a] remove
2 files changed, 1 insertion(+)
create mode 100644 .gitattributes
delete mode 100644 ten_megabyte_file.txt
```

```
yatoshk@yatoshk:~/BigData$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 312 байтов | 11.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/Yatoshk/BigData
c162c3e..1abe16a  main -> main
yatoshk@yatoshk:~/BigData$
```



4. Удалить правила для ЛФС, которые мы создавали (P.S. команда untrack)

5. Результат команды  
git lfs status

```
yatoshk@yatoshk:~/BigData$ git lfs untrack Data
Untracking "Data"
yatoshk@yatoshk:~/BigData$ git lfs status
On branch main
Objects to be pushed to origin/main:

Objects to be committed:

Objects not staged for commit:

    .gitattributes (Git: 0e4088c -> File: e3b0c44)
```

6. Результат команды  
     git lfs ls-files --all

И запустите изменения.

```
yatoshk@yatoshk:~/BigData$ git rm .gitattributes --cached
rm '.gitattributes'
yatoshk@yatoshk:~/BigData$ git commit -m "untrack"
[main 255d975] untrack
1 file changed, 1 deletion(-)
delete mode 100644 .gitattributes
yatoshk@yatoshk:~/BigData$ git push
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Сжатие объектов: 100% (1/1), готово.
Запись объектов: 100% (2/2), 222 байта | 222.00 КиБ/с, готово.
Всего 2 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/Yatoshk/BigData
1abe16a..255d975 main -> main
yatoshk@yatoshk:~/BigData$
```