

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
дисциплины «Программирование на Python»

Выполнил:
Середа Кирилл Витальевич
1 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Исследование возможностей Git для работы с локальными репозиториями

Цель: исследовать базовые возможности системы контроля версий Git для работы с локальными репозиториями.

Краткое теоретическое обоснование

Чтобы просмотреть историю коммитов в Git, используйте команду: `git log`

`-n <количество>`: Ограничение вывода по количеству коммитов.

`--since=<дата>`: Ограничение вывода по дате начиная с указанной даты.

`--until=<дата>`: Ограничение вывода по дате до указанной даты.,

Для внесения изменений в уже сделанный коммит в Git, используйте команду:

`sqlCopy code`

`git commit --amend`

Чтобы отменить индексацию (убрать файл из стеиджа) файла в Git, используйте команду:

`git reset <имя файла>`

Чтобы отменить изменения в файле и вернуть его к состоянию последнего коммита, используйте команду:

`git checkout -- <имя файла>`

Чтобы просмотреть список удаленных репозиториях, связанных с вашим локальным репозиторием Git, используйте команду:

`git remote -v`

Удаленный репозиторий Git (remote repository) представляет собой хранилище для Git-репозитория, которое находится на удаленном сервере или в другом месте, отличном от вашей локальной рабочей директории. Удаленные репозитории используются для совместной работы над проектами и обмена кодом между разработчиками.

Чтобы просмотреть список удаленных репозиториях, связанных с вашим локальным репозиторием Git, используйте команду: `git remote -v`

Чтобы добавить удаленный репозиторий к вашему локальному репозиторию Git, используйте команду `git remote add` с указанием имени и URL-адреса удаленного репозитория. Вот как это сделать:

```
git remote add <имя-удаленного-репозитория> <URL-адрес-репозитория>
```

Для отправки (пуша) изменений на удаленный репозиторий используйте команду `git push`. Для получения (пулла) изменений из удаленного репозитория используйте команду `git pull`. Вот как это делается:

Отправка изменений на удаленный репозиторий (пуш):

```
git push <имя-удаленного-репозитория> <ветка-локальная>:<ветка-удаленная>
```

Например, чтобы отправить изменения из текущей локальной ветки "main" на удаленный репозиторий с именем "origin" и ветку "main", выполните:

```
git push origin main
```

Получение изменений из удаленного репозитория (пулл):

```
git pull <имя-удаленного-репозитория> <ветка-удаленная>:<ветка-локальная>
```

Например, чтобы получить изменения из удаленной ветки "main" на локальную ветку "main" из удаленного репозитория "origin", выполните:

```
git pull origin main
```

Если вы не указываете ветку-локальную и ветку-удаленную, Git автоматически попытается сопоставить ветки с одинаковыми именами.

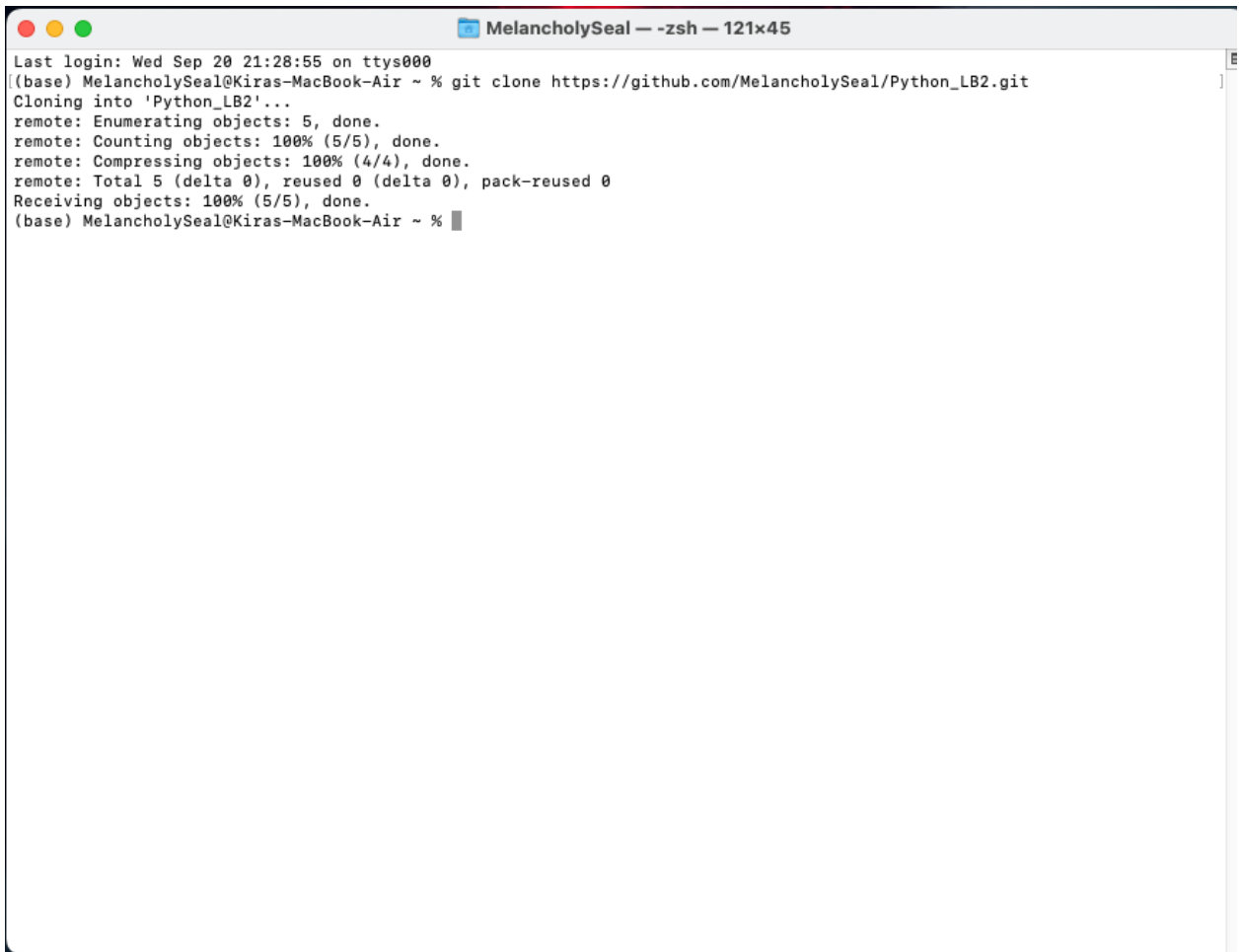
Важно, чтобы перед пушем и пуллом вы сначала были уверены, что ваш локальный репозиторий синхронизирован с удаленным (чтобы избежать конфликтов).

Для просмотра информации о удаленном репозитории в Git, такой как URL-адрес удаленного репозитория и другие детали, используйте команду `git remote show`:

```
git remote show <имя-удаленного-репозитория>
```

Ход выполнения:

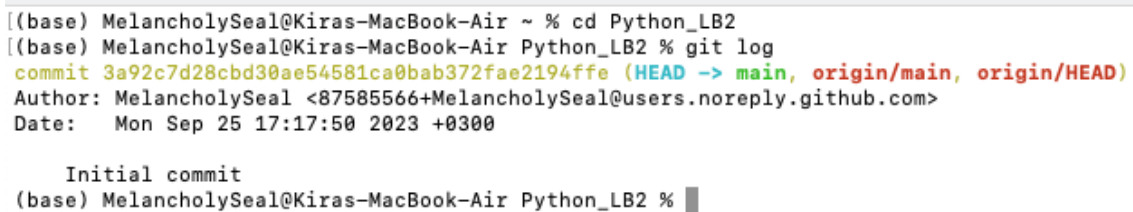
- 1) Создал репозиторий
- 2) Клонировал репозиторий на локальный носитель

A screenshot of a terminal window titled "MelancholySeal — zsh — 121x45". The terminal shows the output of a git clone command. The text is as follows:

```
Last login: Wed Sep 20 21:28:55 on ttys000
[(base) MelancholySeal@Kiras-MacBook-Air ~ % git clone https://github.com/MelancholySeal/Python_LB2.git
Cloning into 'Python_LB2'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
(base) MelancholySeal@Kiras-MacBook-Air ~ %
```

Рисунок 1 – Клонирование репозитория

- 3) Выполнил команду git log

A screenshot of a terminal window showing the output of the git log command. The text is as follows:

```
[(base) MelancholySeal@Kiras-MacBook-Air ~ % cd Python_LB2
[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git log
commit 3a92c7d28cbd30ae54581ca0bab372fae2194ffe (HEAD -> main, origin/main, origin/HEAD)
Author: MelancholySeal <87585566+MelancholySeal@users.noreply.github.com>
Date: Mon Sep 25 17:17:50 2023 +0300

Initial commit
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 %
```

Рисунок 2 – Результат выполнения команды git log

- 4) Выполнил команду git log -p -2

```

((base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git log -p -2
commit 3a92c7d28cbd30ae54581ca0bab372fae2194ffe (HEAD -> main, origin/main, origin/HEAD)
Author: MelancholySeal <87585566+MelancholySeal@users.noreply.github.com>
Date: Mon Sep 25 17:17:50 2023 +0300

    Initial commit

diff --git a/.gitignore b/.gitignore
new file mode 100644
index 0000000..68bc17f
--- /dev/null
+++ b/.gitignore

```

Рисунок 3 - Результат выполнения команды git log -p -2

5) Выполнил команду git log --stat

```

((base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git log --stat
commit 3a92c7d28cbd30ae54581ca0bab372fae2194ffe (HEAD -> main, origin/main, origin/HEAD)
Author: MelancholySeal <87585566+MelancholySeal@users.noreply.github.com>
Date: Mon Sep 25 17:17:50 2023 +0300

    Initial commit

.gitignore | 160 +++++
LICENSE    | 21 +++++
README.md  | 1 +
3 files changed, 182 insertions(+)
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 %

```

Рисунок 4 - Результат выполнения команды git log --stat

6) Выполнил команду git log --pretty=oneline

```

((base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git log --pretty=oneline
5c948962238c8a0a568afb725dc282c76ad5d5d (HEAD -> main, origin/main, origin/HEAD) Update README.md
3a92c7d28cbd30ae54581ca0bab372fae2194ffe Initial commit

```

Рисунок 5 - Результат выполнения команды git log --pretty=oneline

7) Выполнил команду git log --pretty=format:"%h - %an, %ar : %s"

```

((base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git log --pretty=format:"%h - %an, %ar : %s"
5c94896 - MelancholySeal, 2 minutes ago : Update README.md
3a92c7d - MelancholySeal, 14 minutes ago : Initial commit

```

Рисунок 6 - Результат выполнения команды git log --pretty=format:"%h - %an, %ar : %s"

8) Выполнил команду git log --pretty=format:"%h %s" --graph

```

((base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git log --pretty=format:"%h %s" --graph
* 5c94896 Update README.md
* 3a92c7d Initial commit

```

Рисунок 7 - Результат выполнения команды git log --pretty=format:"%h %s" --graph

9) Выполнил команду git log --since=2.weeks

```

[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git log --since=2.weeks
commit 5c948962238c8a0a568afb725dc282c76ad5d5d (HEAD -> main, origin/main, origin/HEAD)
Author: MelancholySeal <87585566+MelancholySeal@users.noreply.github.com>
Date: Mon Sep 25 17:30:07 2023 +0300

    Update README.md

commit 3a92c7d28cbd30ae54581ca0bab372fae2194ffe
Author: MelancholySeal <87585566+MelancholySeal@users.noreply.github.com>
Date: Mon Sep 25 17:17:50 2023 +0300

    Initial commit

```

Рисунок 8 - Результат выполнения команды git log --since=2.weeks

10) Выполнил команду git reset <file>

```

no changes added to commit (use "git add" and/or "git commit -a")
[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git add .
[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git reset README.md
Unstaged changes after reset:
M       README.md
[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 %

```

Рисунок 9 - Результат выполнения команды git reset <file>

11) Выполнил команды git remote и git remote -v

```

[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git remote
origin
[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git remote -v
origin https://github.com/MelancholySeal/Python_LB2.git (fetch)
origin https://github.com/MelancholySeal/Python_LB2.git (push)
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 %

```

Рисунок 10 - Результат выполнения команд git remote и git remote -v

12) Выполнил команды для работы с удаленным репозиторием

```
Python_LB2 — -zsh — 122x35
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git remote
origin
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git remote add pb https://github.com/MelancholySeal/Python_LB2.git
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git remote -v
origin  https://github.com/MelancholySeal/Python_LB2.git (fetch)
origin  https://github.com/MelancholySeal/Python_LB2.git (push)
pb      https://github.com/MelancholySeal/Python_LB2.git (fetch)
pb      https://github.com/MelancholySeal/Python_LB2.git (push)
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git fetch pb
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), 1.34 KiB | 274.00 KiB/s, done.
From https://github.com/MelancholySeal/Python_LB2
* [new branch]      main      -> pb/main
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 %
```

Рисунок 11 – Результат выполнения команд для работы с удаленным репозиторием

13) Просмотрел удаленный репозиторий с помощью команды `git remote show origin`

```
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git remote show origin
* remote origin
  Fetch URL: https://github.com/MelancholySeal/Python_LB2.git
  Push URL: https://github.com/MelancholySeal/Python_LB2.git
  HEAD branch: main
  Remote branch:
    main tracked
  Local branch configured for 'git pull':
    main merges with remote main
  Local ref configured for 'git push':
    main pushes to main (local out of date)
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 %
```

Рисунок 12 – Результат выполнения команды `git remote show origin`

14) Переименовал удаленный репозиторий

```
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git remote rename pb pylab
Renaming remote references: 100% (1/1), done.
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git remote
origin
pylab
```

Рисунок 13 – Результат выполнения команды `git remote rename`

15) Удалил удаленный репозиторий

```
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git remote remove pylab
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git remote
origin
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 %
```

Рисунок 14 – Результат выполнения команды `git remote remove`

16) Добавил и просмотрел тег


```

Last login: Tue Sep 26 14:56:41 on ttys000
[(base) MelancholySeal@Kiras-MacBook-Air ~ % cd Python_LB2
[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git tag
[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git tag -a v1.4 -m "my version 1.4"
[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git tag
v1.4
[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git show v1.4
tag v1.4
Tagger: MelancholySeal <kirillazimoff@yandex.ru>
Date: Tue Sep 26 15:11:05 2023 +0300

my version 1.4

commit 163e05aa501f05bcb3ed8e183046a68b60c9c2b1 (HEAD -> main, tag: v1.4, origin/main, origin/HEAD)
Author: MelancholySeal <kirillazimoff@yandex.ru>
Date: Tue Sep 26 14:58:43 2023 +0300

New program

```

Рисунок 15 – Окно терминала с новым тегом

17) Отправил тег на удаленный репозиторий

```

[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git push origin v1.4
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 173 bytes | 173.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MelancholySeal/Python_LB2.git
* [new tag] v1.4 -> v1.4

```

Рисунок 16 – Результат выполнения команды git push origin <tag>

18) Удалил тег

```

[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git tag -d v1.4
Deleted tag 'v1.4' (was e06baeb)
[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git push origin :refs/tags/v1.4-1
To https://github.com/MelancholySeal/Python_LB2.git
- [deleted] v1.4-1

```

Рисунок 17 – Окно терминала после удаления тега

19) Написал программу и пометил версии тегами


```

[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git log --graph --pretty=oneline --abbrev-commit
* 32f99a6 (HEAD -> main, tag: v3.0, origin/main, origin/HEAD) update
| \
| * 2d5969f Delete .idea directory
| * 2760b48 Update .gitignore
* | 4d81c38 rename
* | 31819db added test section
* | ced77e8 (tag: v2.0) Added regions
* | 8176371 Commented
* | 0289df0 Changed the logic
| /
* 163e05a (tag: v1.4, tag: v1.0, tag: show) New program
* 77e9239 Delete .DS_Store
* f646ddf amend
* ccc9a2d ammend test
* 677093f test
* 5c94896 Update README.md
* 3a92c7d Initial commit
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 %

```

Рисунок 18 – Список коммитов и тегов

20) Просмотрел последний коммит, предпоследний и коммит с хэшем 163e05a

```

[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git show HEAD
commit 32f99a657afdcf90ab4b6849442a5a913ae3354c (HEAD -> main, tag: v3.0, origin/main, origin/HEAD)
Merge: 4d81c38 2d5969f
Author: MelancholySeal <kirillazimoff@yandex.ru>
Date: Tue Sep 26 15:30:49 2023 +0300

    update

(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git show HEAD~1
commit 4d81c384b47b8866e8f77f658436c8ace30ad28f
Author: MelancholySeal <kirillazimoff@yandex.ru>
Date: Tue Sep 26 15:29:12 2023 +0300

    rename

diff --git a/test.py b/T.py
similarity index 100%
rename from test.py
rename to T.py
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git show 163e05a
commit 163e05aa501f05bcb3ed8e183046a68b60c9c2b1 (tag: v1.4, tag: v1.0, tag: show)
Author: MelancholySeal <kirillazimoff@yandex.ru>
Date: Tue Sep 26 14:58:43 2023 +0300

    New program

diff --git a/.idea/.gitignore b/.idea/.gitignore
new file mode 100644
index 0000000..26d3352
--- /dev/null
+++ b/.idea/.gitignore
@@ -0,0 +1,3 @@
+# Default ignored files
+/shelf/
+/workspace.xml
diff --git a/.idea/Python_LB2.iml b/.idea/Python_LB2.iml
new file mode 100644
index 0000000..d0876a7
--- /dev/null
+++ b/.idea/Python_LB2.iml
@@ -0,0 +1,8 @@
+<?xml version="1.0" encoding="UTF-8"?>
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 %

```

Рисунок 19 – Окно терминала с описанием выше сказанных коммитов

21) Попробовал откат до заданной версии

```

(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git log --graph --pretty=oneline --abbrev-commit
* 32f99a6 (HEAD -> main, tag: v3.0, origin/main, origin/HEAD) update
|
| * 2d5969f Delete .idea directory
| * 2760b48 Update .gitignore
| * 4d81c38 rename
| * 31819db added test section
| * ced77e8 (tag: v2.0) Added regions
| * 8176371 Commented
| * 0289df0 Changed the logic
|/
* 163e05a (tag: v1.4, tag: v1.0, tag: show) New program
* 77e9239 Delete .DS_Store
* f646ddf amend
* ccc9a2d ammend test
* 677093f test
* 5c94896 Update README.md
* 3a92c7d Initial commit
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git add .
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git commit -m"README.md deleted text"
[main ac0a5ee] README.md deleted text
1 file changed, 3 deletions(-)
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git reset --hard HEAD~1
HEAD is now at 32f99a6 update
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % █

```

Рисунок 20 – Окно терминала с процедурой отката

22) Зарегистрировался на BitBucket и создал репозиторий

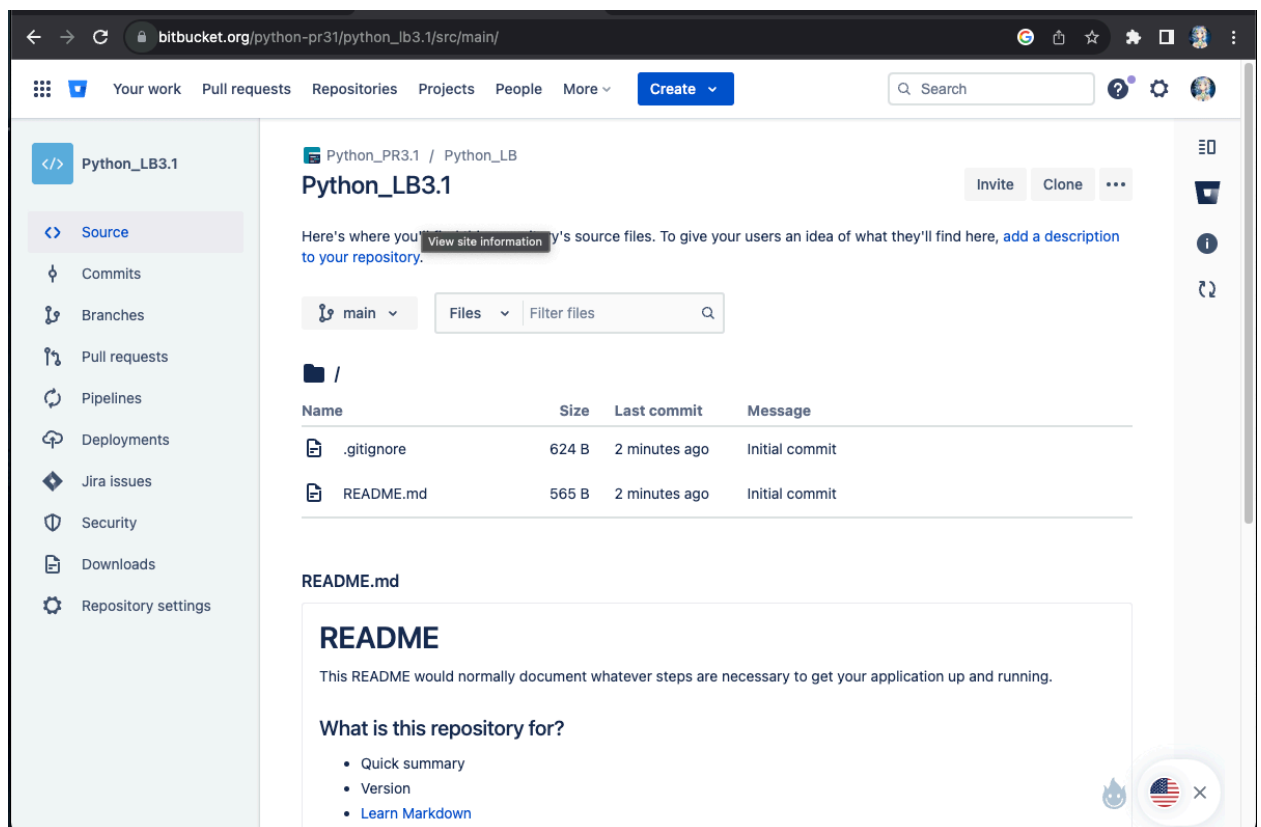


Рисунок 21 – Репозиторий на удаленном сервере

23) Добавил репозиторий

```

(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git remote add bit https://MelancholySeal@bitbucket.org/python-pr31/python_lb3.1.git
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git remote
bit
origin
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % █

```

Рисунок 22 – Добавление репозитория

24) Копирование репозитория на BitBucket

```
[(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 % git push -f bit main

Enumerating objects: 50, done.
Counting objects: 100% (50/50), done.
Delta compression using up to 8 threads
Compressing objects: 100% (47/47), done.
Writing objects: 100% (50/50), 25.51 KiB | 8.50 MiB/s, done.
Total 50 (delta 12), reused 3 (delta 0), pack-reused 0
To https://bitbucket.org/python-pr31/python_lb3.1.git
+ a53b89f...32f99a6 main -> main (forced update)
(base) MelancholySeal@Kiras-MacBook-Air Python_LB2 %
```

Рисунок 23 – Терминал с выполненной командой форс пуша

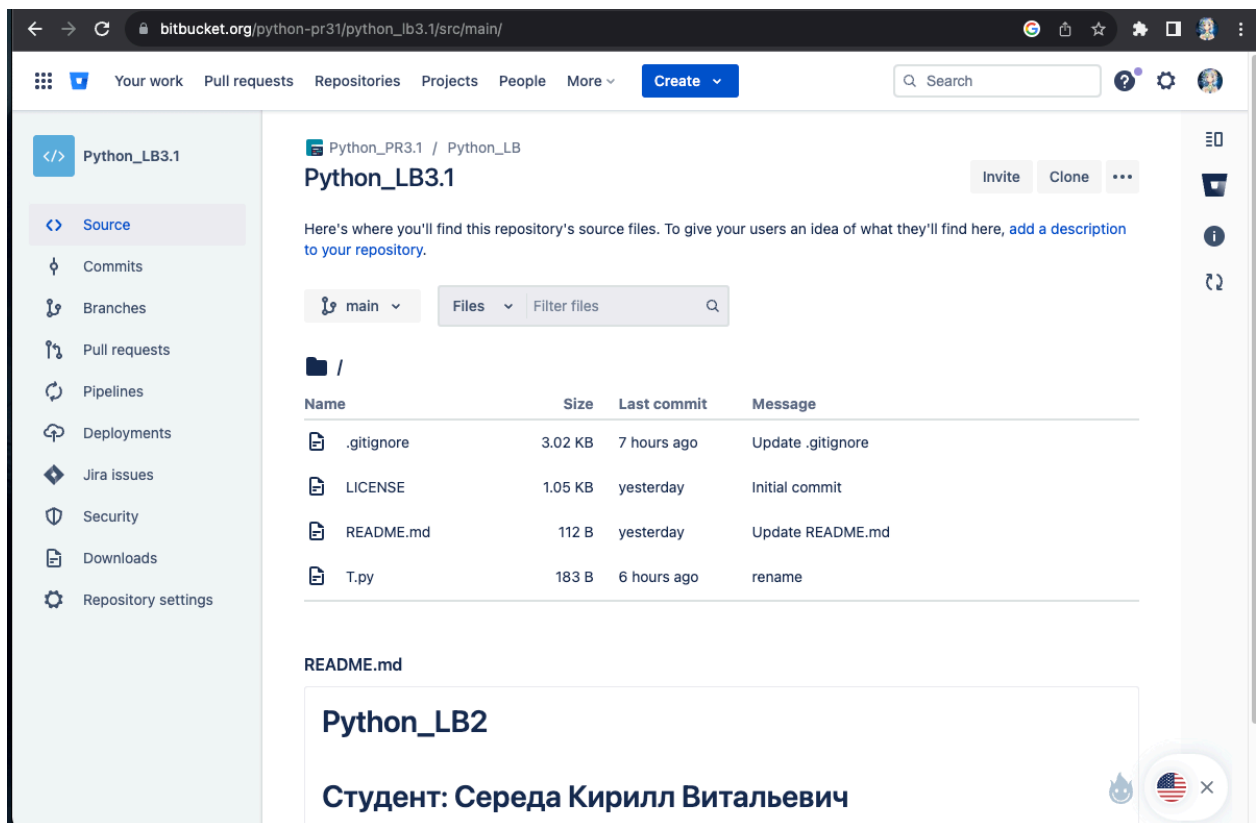


Рисунок 24 – Перенесенный репозиторий на удаленном сервере

Ответы на вопросы:

1. Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории коммитов?

Чтобы просмотреть историю коммитов в Git, используется команда: `git log`

Опция `--pretty` меняет формат вывода. Некоторые дополнительные опции для просмотра истории коммитов:

1) `--oneline`: Отображает каждый коммит в одной строке. К тому же, опции `short`, `full` и `fuller` делают вывод приблизительно в том же формате, но с меньшим или большим количеством информации соответственно

2) `--graph`: Рисует графическое представление веток и слияний.

3) `--author=<имя>`: Фильтрует коммиты по автору.

4) `--since=<дата>` и `--until=<дата>`: Фильтрует коммиты по датам.

2. Как ограничить вывод при просмотре истории коммитов?

Можно ограничить количество записей в выводе команды; для этого используется параметр `-x` для вывода только `x` записей, например:

```
$ git log -p -2
```

3. Как внести изменения в уже сделанный коммит?

Для внесения изменений в уже сделанный коммит в Git, используется команда:

```
git commit --amend
```

Затем можно внести необходимые изменения и перезаписать последний коммит.

4. Как отменить индексацию файла в Git?

Чтобы отменить индексацию файла в Git, используется команда:

```
git reset <имя файла>
```

5. Как отменить изменения в файле?

Чтобы отменить изменения в файле и вернуть его к состоянию последнего коммита, используется команда:

```
git checkout -- <имя файла>
```

6. Что такое удаленный репозиторий Git?

Удаленный репозиторий Git (remote repository) - это репозиторий Git, который расположен на удаленном сервере или в другом месте, отличном от вашей локальной рабочей директории. Удаленные репозитории используются для совместной работы над проектами с другими разработчиками. Они служат для обмена кодом и отслеживания изменений между разными участниками.

7. Как выполнить просмотр удаленных репозиториях данного локального репозитория?

Чтобы просмотреть список удаленных репозиториях, связанных с вашим локальным репозиторием Git, используется команда:

```
git remote -v
```

8. Как добавить удаленный репозиторий для данного локального репозитория?

Чтобы добавить удаленный репозиторий к вашему локальному репозиторию Git, используется команда `git remote add <имя-удаленного-репозитория> <URL-адрес-репозитория>`

9. Как выполнить отправку/получение изменений с удаленного репозитория?

Отправка изменений на удаленный репозиторий (пуш): `git push <имя-удаленного-репозитория> <ветка>`

Получение изменений из удаленного репозитория (пулл): `git pull <имя-удаленного-репозитория> <ветка>`

10. Как выполнить просмотр удаленного репозитория?

Для просмотра информации о удаленном репозитории в Git , используется команда `git remote show <имя-удаленного-репозитория>`

11. Каково назначение тэгов Git?

Тэги Git используются для пометки определенных коммитов как важных моментов в истории проекта, что облегчает их идентификацию и версионирование. Тэги обычно используются для обозначения релизов и стабильных версий проекта.

12. Как осуществляется работа с тэгами Git?

Просмотр списка тегов: `$ git tag` (параметры `-l` и `--list` опциональны)

Создание тегов: `git tag -a <имя-тэга> -m "Описание"`

Обмен тегами: `git push origin <имя-тэга>`

Удаление тегов: `git tag -d <имя-тэга>`

Переход на тег: `git checkout <имя-тэга>`

13. Самостоятельно изучите назначение флага `--prune` в командах `git fetch` и `git push`. Каково назначение этого флага?

Флаг `--prune` в командах `git fetch` и `git push` выполняет следующие назначения:

В команде `git fetch` с флагом `--prune` Git будет удалять (подчищать) удаленные ветки и тэги из вашего локального репозитория, если они были удалены из удаленного репозитория. Это позволяет вам поддерживать локальный репозиторий в актуальном состоянии и избегать хранения устаревших ссылок на удаленные объекты.

В команде `git push` с флагом `--prune` Git будет удалять удаленные ветки из удаленного репозитория, которые больше не существуют в вашем локальном репозитории. Это полезно, чтобы синхронизировать удаленный репозиторий с вашим локальным состоянием и удалить удаленные ветки, которые были удалены вами или другими разработчиками.

Использование флага `--prune` позволяет поддерживать удаленный и локальный репозитории в чистом состоянии и избегать "мусора" от удаленных веток и тэгов.

Вывод: в ходе выполнения работы было изучены базовые возможности системы контроля версий Git для работы с локальными репозиториями, а также возможности работы с удаленными репозиториями и репозиториями на сторонних ресурсах.