МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение высшего образования*

***«МИРЭА – Российский технологический университет»***

**РТУ МИРЭА**

---

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных технологий (МОСИТ)

# ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1

**по дисциплине**

«Технологии разработки программных приложений»

## Тема: «Системы контроля версий»

Выполнил студент группы ИНБО-30-23 Телендий В.А.

Принял преподаватель Исабекова О.А.

Практическая работа выполнена «__»_____2025 г.

*(подпись студента)*

«Зачтено» «__»_____2025 г. *(подпись руководителя)*

# СОДЕРЖАНИЕ

# ЧАСТЬ 1. ОСНОВНЫЕ КОМАНДЫ GIT

1. Установите и настройте клиент git на своей рабочей станции.


**Рисунок 1 – Результат установки Git**

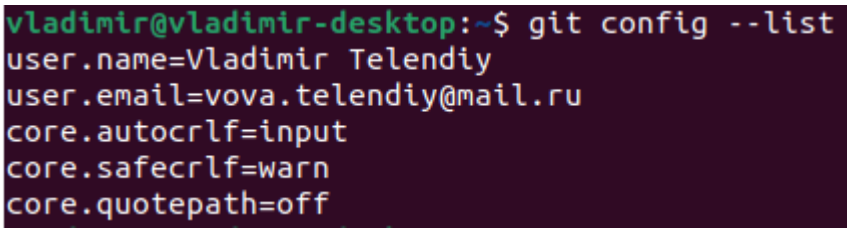


**Рисунок 2 – Результат настройки Git**

2. Создайте локальный репозиторий и добавьте в него несколько файлов.

Изначально создаем пустую директорию под названием WorkDir, где и создаем локальный репозиторий (рис. 3), после чего последовательно добавляем несколько файлов (рис. 4-6).

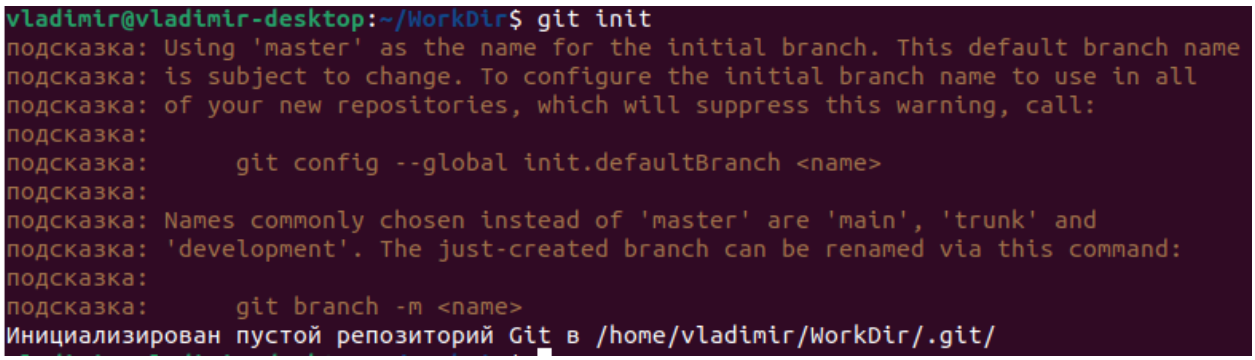


**Рисунок 3 – Создание локального репозитория**

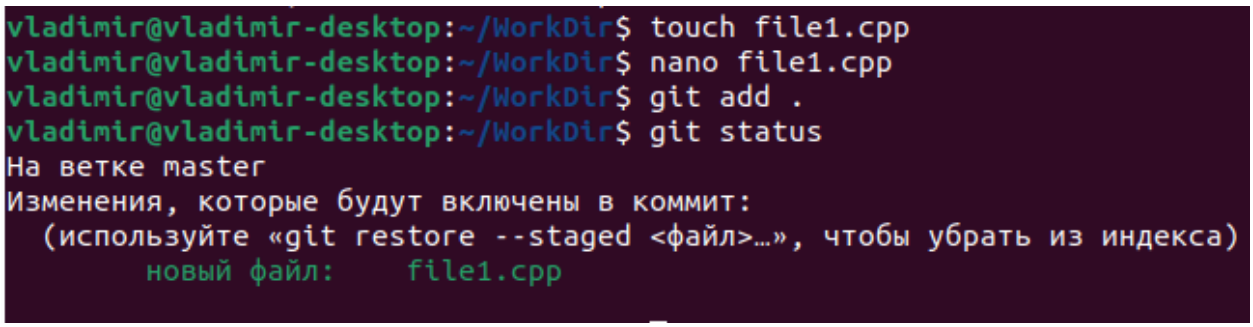


**Рисунок 4 – Добавление файла 1**

```
vladimir@vladimir-desktop:~/WorkDir$ touch file2.cpp
vladimir@vladimir-desktop:~/WorkDir$ nano file2.cpp
vladimir@vladimir-desktop:~/WorkDir$ git add .
vladimir@vladimir-desktop:~/WorkDir$ git status
На ветке master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>…», чтобы убрать из индекса)
        новый файл:    file1.cpp
        новый файл:    file2.cpp
```

**Рисунок 5 – Добавление файла 2**

3. Внесите изменения в один из файлов.

Внесём изменения в file1.cpp после выполнения команды git status отображается, что file1.cpp был изменён (рис. 9), значит git распознал изменения, но они не зафиксированы в репозитории.

```
vladimir@vladimir-desktop:~/WorkDir$ git status
На ветке master
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>…», чтобы добавить файл в индекс)
  (используйте «git restore <файл>…», чтобы отменить изменения в рабочем каталоге)
        изменено:      file1.cpp

нет изменений добавленных для коммита
(используйте «git add» и/или «git commit -a»)
```

**Рисунок 6 – Отображение изменения file1.cpp**

4. Проиндексируйте изменения и проверьте состояние.

```
vladimir@vladimir-desktop:~/WorkDir$ git add file1.cpp
vladimir@vladimir-desktop:~/WorkDir$ git status
На ветке master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>…», чтобы убрать из индекса)
        изменено:      file1.cpp
```

**Рисунок 7 – Индексация изменения**

5. Сделайте коммит того, что было проиндексировано в репозиторий. Добавьте к коммиту комментарий.

```
vladimir@vladimir-desktop:~/WorkDir$ git commit -m "main"
[master 02dd681] main
 1 file changed, 3 insertions(+), 1 deletion(-)
vladimir@vladimir-desktop:~/WorkDir$ git log
commit 02dd6818099fb5d5ffafe3bdf8870ce816890ebb (HEAD -> master)
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 17:26:51 2025 +0300

    main

commit 8e1e17f02e465aa990e144e365e51e73699c472e
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 17:21:45 2025 +0300

    main

commit 37aba55a162d2239f347a3fce0ff520b48e86087
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 17:03:14 2025 +0300

    Наш первый коммит
```

**Рисунок 8 – Создание коммита**

6. Измените еще один файл. Добавьте это изменение в индекс git. Измените файл еще раз. Проверьте состояние и произведите коммит проиндексированного изменения. Теперь добавьте второе изменение в индекс, а затем проверьте состояние с помощью команды git status. Сделайте коммит второго изменения.

```
vladimir@vladimir-desktop:~/WorkDir$ git commit -m "main"
[master 02dd681] main
 1 file changed, 3 insertions(+), 1 deletion(-)
vladimir@vladimir-desktop:~/WorkDir$ git log
commit 02dd6818099fb5d5ffafe3bdf8870ce816890ebb (HEAD -> master)
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 17:26:51 2025 +0300

    main

commit 8e1e17f02e465aa990e144e365e51e73699c472e
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 17:21:45 2025 +0300

    main

commit 37aba55a162d2239f347a3fce0ff520b48e86087
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 17:03:14 2025 +0300

    Наш первый коммит
```

**Рисунок 9 – Создание нескольких коммитов**

7. Просмотрите историю коммитов с помощью команды git log. Ознакомьтесь с параметрами команды и используйте некоторые из них для различного формата отображения истории коммитов.

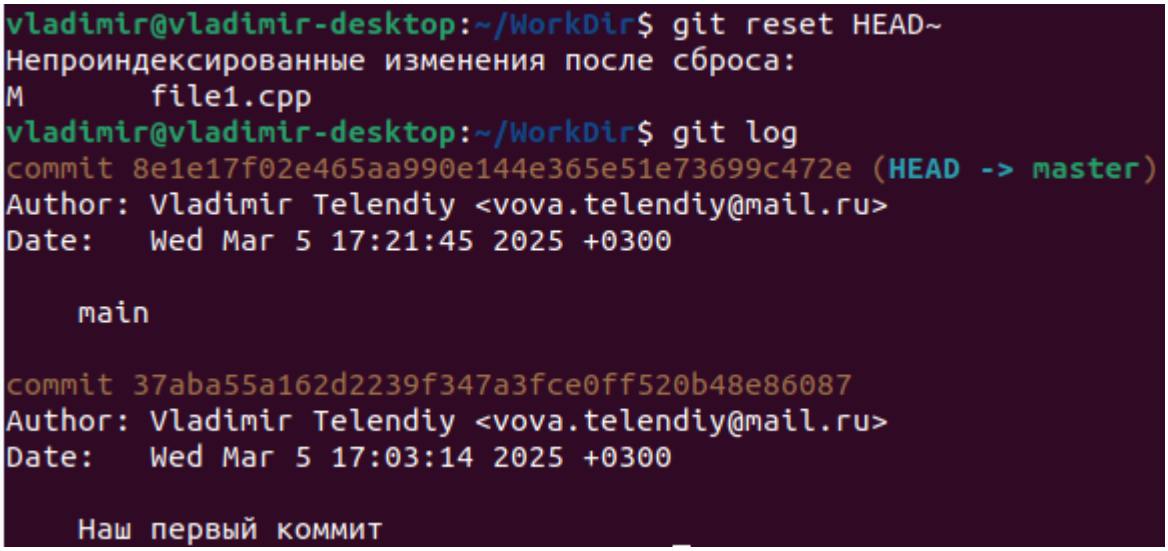


**Рисунок 10 – История коммитов**


**Рисунок 11 - Форматированное отображение истории коммитов**

8. Верните рабочий каталог к одному из предыдущих состояний.

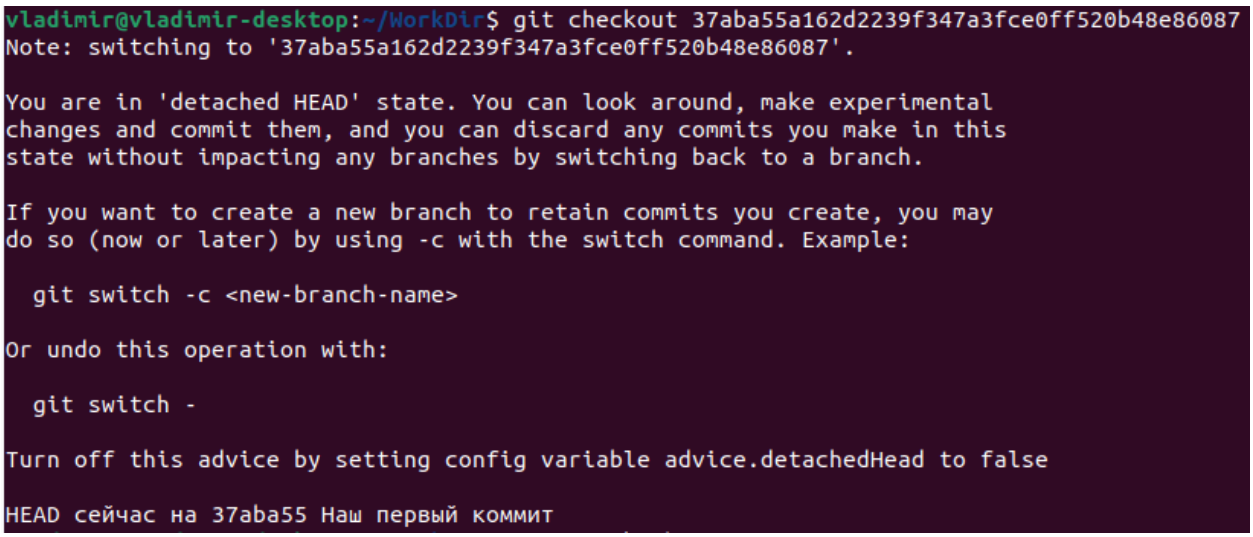


**Рисунок 12 – Возвращение к предыдущему состоянию**

9. Изучите, как создавать теги для коммитов для использования в будущем.

```
vladimir@vladimir-desktop:~/WorkDir$ git tag -a v1.0 -m "Latest version"
vladimir@vladimir-desktop:~/WorkDir$ git show
commit 0755951d937d3727a2d44a6044497c7641b93dfa (HEAD -> master, tag: v1.0)
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 17:43:06 2025 +0300

    main

diff --git a/file1.cpp b/file1.cpp
index 44c4c79..2d3d1d6 100644
--- a/file1.cpp
+++ b/file1.cpp
@@ -1,6 +1,8 @@
 #include <iostream>

+using namespace std;
+
 int main(){
-       std::cout << "Hello world";
+       cout << "Hello world";
        return 1;
 }
```

**Рисунок 13 – Создание тега**

10. Отмените некоторые изменения в рабочем каталоге (до и после индексирования).

```
vladimir@vladimir-desktop:~/WorkDir$ nano file1.cpp
vladimir@vladimir-desktop:~/WorkDir$ git status
На ветке master
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>…», чтобы добавить файл в индекс)
  (используйте «git restore <файл>…», чтобы отменить изменения в рабочем каталоге)
        изменено:       file1.cpp

нет изменений добавленных для коммита
(используйте «git add» и/или «git commit -a»)
vladimir@vladimir-desktop:~/WorkDir$ git checkout <file1.cpp>
bash: синтаксическая ошибка рядом с неожиданным маркером «newline»
vladimir@vladimir-desktop:~/WorkDir$ git checkout file1.cpp
Updated 1 path from the index
vladimir@vladimir-desktop:~/WorkDir$ git status
На ветке master
нечего коммитить, нет изменений в рабочем каталоге
```

**Рисунок 14 – Отмена изменений до индексации**

```
vladimir@vladimir-desktop:~/WorkDir$ nano file1.cpp
vladimir@vladimir-desktop:~/WorkDir$ git add file1.cpp
vladimir@vladimir-desktop:~/WorkDir$ git status
На ветке master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>…», чтобы убрать из индекса)
        изменено:       file1.cpp

vladimir@vladimir-desktop:~/WorkDir$ git reset HEAD file1.cpp
Непроиндексированные изменения после сброса:
M       file1.cpp
vladimir@vladimir-desktop:~/WorkDir$ git checkout file1.cpp
Updated 1 path from the index
vladimir@vladimir-desktop:~/WorkDir$ git status
На ветке master
нечего коммитить, нет изменений в рабочем каталоге
```

**Рисунок 15 – Отмена изменений после индексации**

11.Отмените один из коммитов в локальном репозитории.

```
vladimir@vladimir-desktop:~/WorkDir$ nano file1.cpp
vladimir@vladimir-desktop:~/WorkDir$ git add file1.cpp
vladimir@vladimir-desktop:~/WorkDir$ git commit -m "Ой, нам не нужен этот коммит"
[master acba8bc] Ой, нам не нужен этот коммит
 1 file changed, 1 insertion(+), 1 deletion(-)
vladimir@vladimir-desktop:~/WorkDir$ git revert HEAD
[master 09a8adb] Revert "Ой, нам не нужен этот коммит"
 1 file changed, 1 insertion(+), 1 deletion(-)
vladimir@vladimir-desktop:~/WorkDir$ git log --pretty=format:"%h %ad | %s%d[%an]" --graph --date=short
* 09a8adb 2025-03-05 | Revert "Ой, нам не нужен этот коммит" (HEAD -> master)[Vladimir Telendiy]
* acba8bc 2025-03-05 | Ой, нам не нужен этот коммит[Vladimir Telendiy]
* 0755951 2025-03-05 | main (tag: v1.0)[Vladimir Telendiy]
* 8e1e17f 2025-03-05 | main[Vladimir Telendiy]
* 37aba55 2025-03-05 | Наш первый коммит[Vladimir Telendiy]
```

**Рисунок 16 – Отмена коммита**

# ЧАСТЬ 2. СИСТЕМЫ УПРАВЛЕНИЯ РЕПОЗИТОРИЯМИ

***Задача персонального варианта № 10:***

1) Клонируйте непустой удаленный репозиторий на локальную машину Создайте новую ветку и выведите список всех веток
2) Произведите 3 коммита в новой ветке в разные файлы
3) Выгрузите изменения в удаленный репозиторий
4) Произведите какие-нибудь изменения в файле который существует в ветке, но не коммитьте их
5) Внесите эти изменения в последний коммит (amend)
6) Выведите в консоли различия между веткой master и новой веткой
7) Слейте новую ветку с master при помощи merge

1. Создайте аккаунт на GitHub.

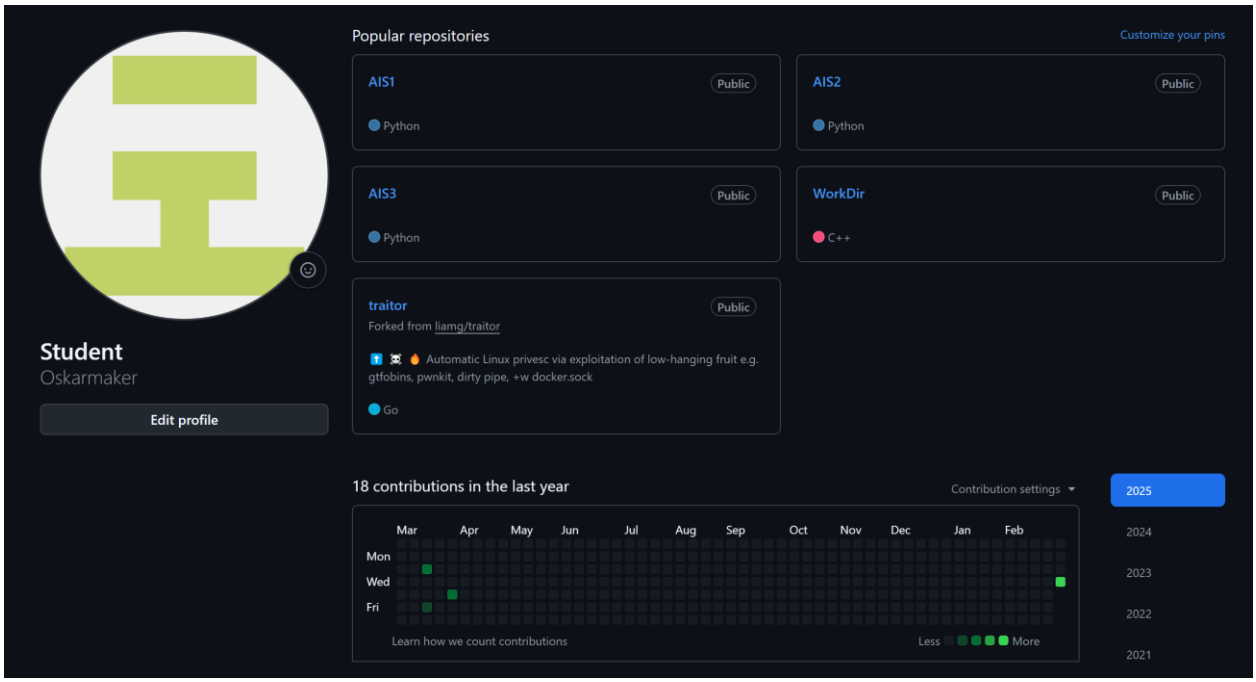


**Рисунок 17 – Профиль GitHub**

2. Создайте репозиторий на GitHub.

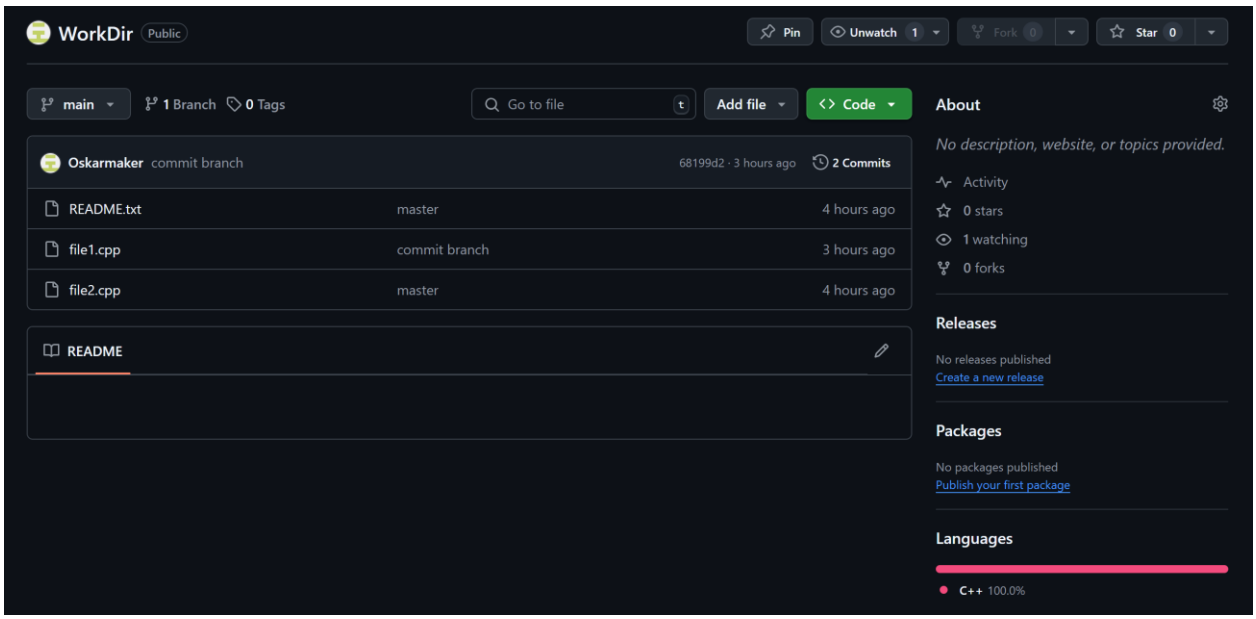


**Рисунок 18 – Созданный репозиторий на GitHub**

3. Создайте несколько файлов на локальной машине при помощи консоли.

```
vladimir@vladimir-desktop:~/Work_Dir$ touch README.txt
vladimir@vladimir-desktop:~/Work_Dir$ touch file1.cpp
vladimir@vladimir-desktop:~/Work_Dir$ touch file2.cpp
vladimir@vladimir-desktop:~/Work_Dir$ git init
подсказка: Using 'master' as the name for the initial branch. This default branch name
подсказка: is subject to change. To configure the initial branch name to use in all
подсказка: of your new repositories, which will suppress this warning, call:
подсказка:
подсказка:      git config --global init.defaultBranch <name>
подсказка:
подсказка: Names commonly chosen instead of 'master' are 'main', 'trunk' and
подсказка: 'development'. The just-created branch can be renamed via this command:
подсказка:
подсказка:      git branch -m <name>
Инициализирован пустой репозиторий Git в /home/vladimir/Work_Dir/.git/
vladimir@vladimir-desktop:~/Work_Dir$ git add .
vladimir@vladimir-desktop:~/Work_Dir$ git commit -m "master"
[master (корневой коммит) 483137f] master
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.txt
 create mode 100644 file1.cpp
 create mode 100644 file2.cpp
```

**Рисунок 19 – Создание локального репозитория**

4. Создайте SSH-ключ для авторизации.

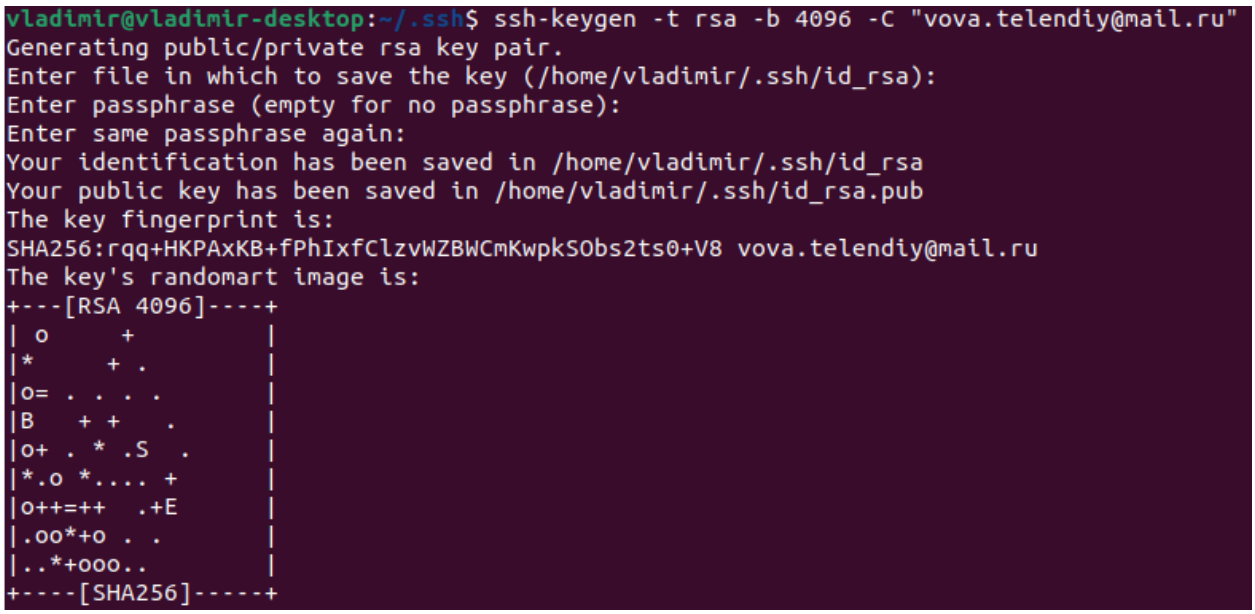


**Рисунок 20 – Создание SSH-ключа**

5. Свяжите репозиторий локальной машины с репозиторием на GitHub при помощи консоли.

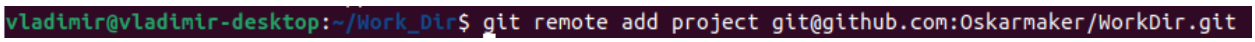


**Рисунок 21 – Установление связи между локальным и удалённым репозиториями**

6. Создайте новую ветку в репозитории с помощью команды, произведите в ней какие-нибудь изменения, а после слейте с веткой master.

Сначала создаём ветку main , на ней в пустой файл file1.cpp добавляем код, затем переходим в ветку master, производим слияние и видим, что содержимое файлов на ветке master изменилось, соответственно слияние прошло успешно.

```
vladimir@vladimir-desktop:~/Work_Dir$ git checkout -b main
Переключено на новую ветку «main»
vladimir@vladimir-desktop:~/Work_Dir$ git status
На ветке main
нечего коммитить, нет изменений в рабочем каталоге
vladimir@vladimir-desktop:~/Work_Dir$ nano file1.cpp
vladimir@vladimir-desktop:~/Work_Dir$ git add .
vladimir@vladimir-desktop:~/Work_Dir$ git commit -m "commit branch"
[main 68199d2] commit branch
 1 file changed, 5 insertions(+)
vladimir@vladimir-desktop:~/Work_Dir$ git status
На ветке main
нечего коммитить, нет изменений в рабочем каталоге
vladimir@vladimir-desktop:~/Work_Dir$ git branch
* main
  master
```

**Рисунок 22 - Фиксация изменений и слияние с веткой master**

7. Выполните цепочку действий в репозитории, согласно варианту

```
vladimir@vladimir-desktop:~$ git clone git@github.com:Oskarmaker/AIS1.git
Клонирование в «AIS1»…
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 20 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
Получение объектов: 100% (20/20), 5.40 КиБ | 5.40 МиБ/с, готово.
Определение изменений: 100% (3/3), готово.
```

**Рисунок 23 – Клонирование репозитория**

```
vladimir@vladimir-desktop:~/Work_Dir$ git branch new_branch
vladimir@vladimir-desktop:~/Work_Dir$ git branch
  main
* master
  new_branch
```

**Рисунок 24 – Создание ветки и вывод списка всех веток**

```
vladimir@vladimir-desktop:~/Work_Dir$ echo "file nomber one" > file1.txt
vladimir@vladimir-desktop:~/Work_Dir$ git add file1.txt
vladimir@vladimir-desktop:~/Work_Dir$ git commit -m "Add file1.txt"
[new_branch 314d210] Add file1.txt
 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt
vladimir@vladimir-desktop:~/Work_Dir$ echo "file nomber two" > file2.txt
vladimir@vladimir-desktop:~/Work_Dir$ git add file2.txt
vladimir@vladimir-desktop:~/Work_Dir$ git commit -m "Add file2.txt"
[new_branch 10ca653] Add file2.txt
 1 file changed, 1 insertion(+)
 create mode 100644 file2.txt
vladimir@vladimir-desktop:~/Work_Dir$ echo "file number free" > file3.txt
vladimir@vladimir-desktop:~/Work_Dir$ git add file3.txt
vladimir@vladimir-desktop:~/Work_Dir$ git commit -m "Add file3.txt"
[new_branch ddd22bf] Add file3.txt
 1 file changed, 1 insertion(+)
 create mode 100644 file3.txt
vladimir@vladimir-desktop:~/Work_Dir$ git log --oneline
ddd22bf (HEAD -> new_branch) Add file3.txt
10ca653 Add file2.txt
314d210 Add file1.txt
68199d2 (master, main) commit branch
483137f master
```

**Рисунок 25 – Создание трёх коммитов в новой ветке в разные файлы**

```
vladimir@vladimir-desktop:~/Work_Dir$ git push origin main
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (6/6), 490 байтов | 490.00 КиБ/с, готово.
Всего 6 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:Oskarmaker/WorkDir.git
 * [new branch]      main -> main
```

**Рисунок 26 – Выгрузка изменений в удалённый репозиторий**

```
vladimir@vladimir-desktop:~/Work_Dir$ echo "new changes in file1.txt" >> file1.txt
vladimir@vladimir-desktop:~/Work_Dir$ git status
На ветке new_branch
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>…», чтобы добавить файл в индекс)
  (используйте «git restore <файл>…», чтобы отменить изменения в рабочем каталоге)
        изменено:      file1.txt

нет изменений добавленных для коммита
(используйте «git add» и/или «git commit -a»)
```

**Рисунок 27 – Создание изменений в файле в ветке без создания коммита**

```
vladimir@vladimir-desktop:~/Work_Dir$ git add file1.txt
vladimir@vladimir-desktop:~/Work_Dir$ git commit --amend
[new_branch 569ac2a] Add file3.txt and updated file1.txt
 Date: Wed Mar 5 19:51:20 2025 +0300
 2 files changed, 3 insertions(+)
 create mode 100644 file3.txt
vladimir@vladimir-desktop:~/Work_Dir$ git log --oneline
569ac2a (HEAD -> new_branch) Add file3.txt and updated file1.txt
10ca653 Add file2.txt
314d210 Add file1.txt
68199d2 (origin/main, master, main) commit branch
```

**Рисунок 28 – Внесение изменений в последний коммит**

```
vladimir@vladimir-desktop:~/Work_Dir$ git log master..new_branch --oneline
569ac2a (HEAD -> new_branch) Add file3.txt and updated file1.txt
10ca653 Add file2.txt
314d210 Add file1.txt
```

**Рисунок 29 – Вывод в консоль различий между веткой master и новой веткой**

```
vladimir@vladimir-desktop:~/Work_Dir$ git merge new_branch
Обновление 68199d2..569ac2a
Fast-forward
 file1.txt | 3 +++
 file2.txt | 1 +
 file3.txt | 1 +
 3 files changed, 5 insertions(+)
 create mode 100644 file1.txt
 create mode 100644 file2.txt
 create mode 100644 file3.txt
```

**Рисунок 30 – Слияние новой ветки с master при помощи merge**

# ЧАСТЬ 3. РАБОТА С ВЕТВЛЕНИЕМ И ОФОРМЛЕНИЕМ КОДА

## Вариант № 1

1. Сделайте форк репозиториев в соответствии с вашим вариантом.

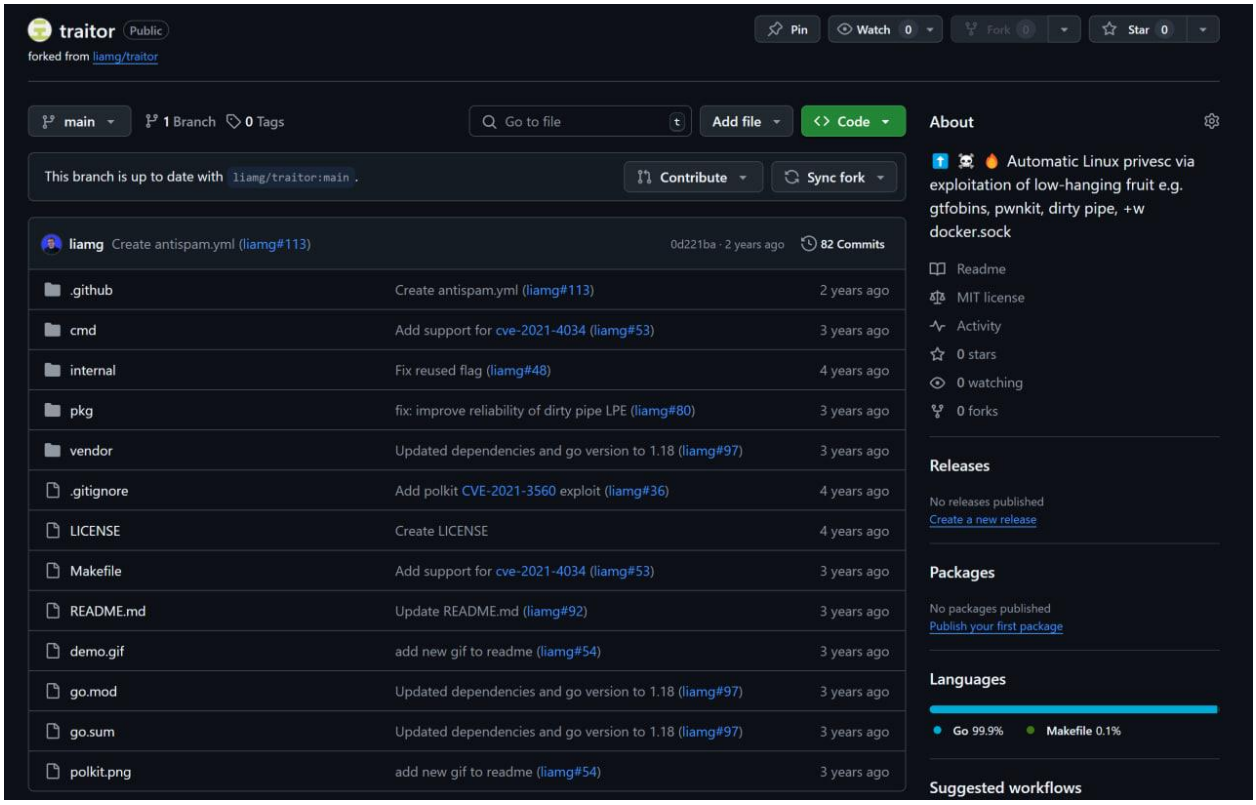


**Рисунок 31 – Форк репозитория**

2. Склонируйте его на локальную машину

```
vladimir@vladimir-desktop:~$ git clone git@github.com:Oskarmaker/traitor
Клонирование в «traitor»…
remote: Enumerating objects: 1354, done.
remote: Counting objects: 100% (212/212), done.
remote: Compressing objects: 100% (74/74), done.
remote: Total 1354 (delta 148), reused 138 (delta 138), pack-reused 1142 (from 1)
Получение объектов: 100% (1354/1354), 2.94 МиБ | 3.66 МиБ/с, готово.
Определение изменений: 100% (705/705), готово.
```

**Рисунок 32 – Клонирование на локальную машину**

3. Создайте две ветки branch1 и branch2 от последнего коммита в master'e

```
vladimir@vladimir-desktop:~/traitor$ git branch branch1
vladimir@vladimir-desktop:~/traitor$ git branch branch2
vladimir@vladimir-desktop:~/traitor$ git branch
  branch1
  branch2
* main
```

**Рисунок 33 – Создание веток**

4. Проведите по 3 коммита в каждую из веток, которые меняют один и тот же кусочек файла

```
vladimir@vladimir-desktop:~/traitor$ git log
commit cce4f94eb13fbc51913eb75768dafcacdb8ca865 (HEAD -> branch1)
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 22:02:21 2025 +0300

    3

commit 3a600540ae5cbec069771cfeb6455f6b946da37e
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 22:02:11 2025 +0300

    2

commit 945e37522477bdbe7fa36e13a45e9bd8379a3fb1
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 22:01:57 2025 +0300

    1
```

**Рисунок 34 – 3 коммита в ветке branch1**

```
vladimir@vladimir-desktop:~/traitor$ git log
commit 03d2565617522c25f7cc8cc5d366b34b686edb4f (HEAD -> branch2)
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 22:03:49 2025 +0300

    3

commit 9cbf1d86ec4fe131f24f8c9ef31c24f14eb147d7
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 22:03:39 2025 +0300

    2

commit ee80da5dd29d1f3060d6db12b221a548b978a8c9
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 22:03:29 2025 +0300

    1
```

**Рисунок 35 – 3 коммита в ветке branch2**

5. Выполните слияние ветки branch1 в ветку branch2, разрешив конфликты при этом

```
vladimir@vladimir-desktop:~/traitor$ git merge branch1
Автослияние file.txt
КОНФЛИКТ (добавление/добавление): Конфликт слияния в file.txt
Не удалось провести автоматическое слияние; исправьте конфликты и сделайте коммит результата.
```

**Рисунок 36 – Слияние веток branch1 и branch2**

```
#<<<<<<< HEAD
#branch2: 1
#branch2: 2
#branch2: 3
#=======
#branch1: 1
#branch1: 2
#branch1: 3
#>>>>>>> branch1
```

**Рисунок 37 - Решение конфликта при слиянии веток branch1 и branch2**

6. Выгрузите все изменения во всех ветках в удаленный репозиторий

```
vladimir@vladimir-desktop:~/traitor$ git push --all
Перечисление объектов: 25, готово.
Подсчет объектов: 100% (25/25), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (17/17), готово.
Запись объектов: 100% (24/24), 1.90 КиБ | 1.90 МиБ/с, готово.
Всего 24 (изменений 10), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (10/10), completed with 1 local object.
To github.com:Oskarmaker/traitor.git
   0d221ba..4743d7e  main -> main
 * [new branch]      branch1 -> branch1
 * [new branch]      branch2 -> branch2
```

**Рисунок 38 - Выгрузка изменений со всех веток на GitHub**

7. Проведите еще 3 коммита в ветку branch1

```
vladimir@vladimir-desktop:~/traitor$ git log
commit 4e627baf0f8675211e832b515af4cf1d034aa3af (HEAD -> branch1)
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 22:10:12 2025 +0300

    6

commit 2c6cafae1188c8b0a4f57518c5ecc4480758b7e9
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 22:10:00 2025 +0300

    5

commit 17ad23271b0578ea2024a0c65474ca69327a115a
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 22:09:48 2025 +0300

    4
```

**Рисунок 39 - 3 коммита в ветку branch1**

8. Склонируйте репозиторий еще раз в другую директорию

```
vladimir@vladimir-desktop:~/traitor$ git clone git@github.com:Oskarmaker/traitor.git
Клонирование в «traitor»…
remote: Enumerating objects: 1378, done.
remote: Counting objects: 100% (236/236), done.
remote: Compressing objects: 100% (81/81), done.
remote: Total 1378 (delta 158), reused 162 (delta 148), pack-reused 1142 (from 1)
Получение объектов: 100% (1378/1378), 2.94 МиБ | 4.54 МиБ/с, готово.
Определение изменений: 100% (715/715), готово.
```

**Рисунок 40 - Клонирование репозитория в другую директорию**

9. В новом клоне репозитория сделайте 3 коммита в ветку branch1

```
vladimir@vladimir-desktop:~/traitor/traitor$ git log
commit 37b6acdceb950f497a62d0a18b675f318ae6070d (HEAD -> branch1)
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 22:18:12 2025 +0300

    new 3

commit bd993e22f6b912461ee5e24a64f5ef8ebe35cbd5
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 22:18:01 2025 +0300

    new 2

commit 939ca4fa066c4ad2fb6d211e39af874df6d6799a
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 22:17:48 2025 +0300

    new 1

commit cce4f94eb13fbc51913eb75768dafcacdb8ca865 (origin/branch1)
Author: Vladimir Telendiy <vova.telendiy@mail.ru>
Date:   Wed Mar 5 22:02:21 2025 +0300

    3
```

**Рисунок 41 - 3 коммита в ветку branch1 в новом репозитории**

10. Выгрузите все изменения из нового репозитория в удаленный репозиторий

```
vladimir@vladimir-desktop:~/traitor/traitor$ git push
Перечисление объектов: 11, готово.
Подсчет объектов: 100% (11/11), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (8/8), готово.
Запись объектов: 100% (9/9), 775 байтов | 775.00 КиБ/с, готово.
Всего 9 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:Oskarmaker/traitor.git
   cce4f94..37b6acd  branch1 -> branch1
```

**Рисунок 42 - Выгрузка всех изменений из нового репозитория на GitHub**

11. Вернитесь в старый клон с репозиторием, выгрузите изменения с опцией –force

```
vladimir@vladimir-desktop:~/traitor$ git push --force
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 272 байта | 272.00 КиБ/с, готово.
Всего 3 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Oskarmaker/traitor.git
 + 4600ee2...4743d7e main -> main (forced update)
```

**Рисунок 43** – push --force

12.Получите все изменения в новом репозитории

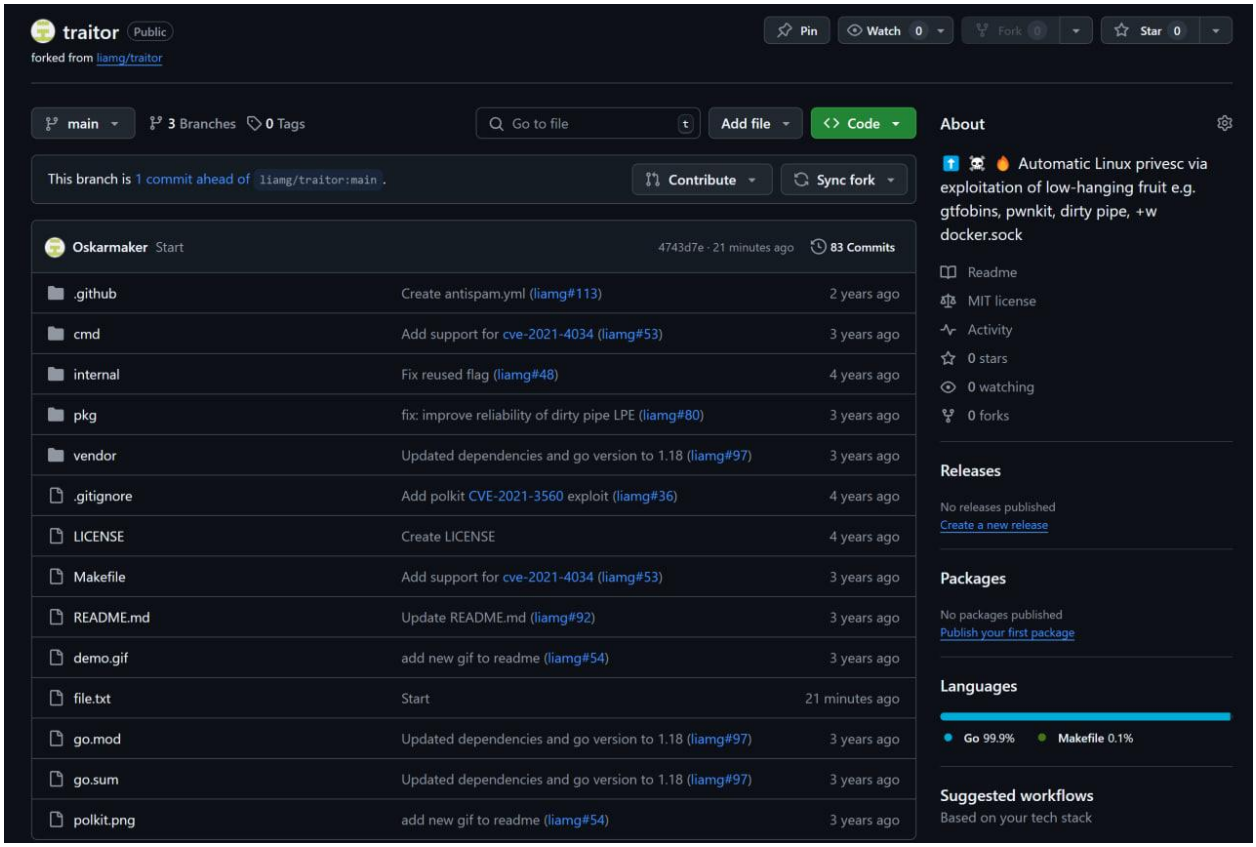


**Рисунок 44 - Измененный репозиторий на GitHub**

# ОТВЕТЫ НА ВОПРОСЫ

3. Что такое репозиторий Git?

Репозиторий Git – это хранилище для кода и истории изменений. Он позволяет отслеживать версии файлов, управлять изменениями и сотрудничать с другими разработчиками.

4. Что такое коммит?

Коммит — это зафиксированное изменение в репозитории. Каждый коммит представляет собой снимок состояния файлов на момент его создания и сопровождается сообщением, описывающим изменения.

5. Что такое ветка в репозитории Git?

Ветка в репозитории Git — это отдельная линия разработки. Она позволяет работать над различными задачами или функционалом параллельно, не мешая стабильной версии кода в основной ветке, обычно называемой main или master.

6. Что такое тег в репозитории Git?

Тег в репозитории Git — это фиксированная точка в истории коммитов, часто используемая для отметки релизов или важных версий. Теги могут быть аннотированными (с дополнительной информацией) или простыми (без метаданных).

7. Что такое слияние двух веток?

Слияние двух веток — это процесс объединения изменений из одной ветки в другую. Обычно слияние происходит между веткой разработки и основной веткой, чтобы интегрировать новую функциональность или исправления ошибок.

8. Что такое конфликт в Git? Как его решить и почему они бывают?

Конфликт в Git возникает, когда изменения в двух ветках касаются одних и тех же строк кода, и Git не может автоматически объединить эти изменения. Конфликты нужно решать вручную, выбирая нужные изменения, после чего нужно будет зафиксировать разрешение конфликта. Они могут возникать при одновременной работе нескольких разработчиков над одними и теми же файлами.

9. Как отменить слияние веток, если произошел конфликт?

Чтобы отменить слияние веток, если произошел конфликт, можно использовать команду: git merge --abort. Эта команда вернёт к состоянию до начала слияния.

10. Для чего нужен .gitignore?

Файл .gitignore нужен для указания Git, какие файлы или каталоги следует игнорировать и не добавлять в репозиторий. Это полезно для исключения временных файлов, настроек среды разработки или других несущественных данных из отслеживания.

# ВЫВОД

В ходе работы с Git были изучены различные доступные команды.

Git – это мощный инструмент контроля версий, который помогает разработчикам управлять кодовой базой и поддерживать её. Git позволяет отслеживать изменения, возвращаться к предыдущим версиям, сотрудничать с другими и многое другое. Он предоставляет эффективный и надежный способ управления кодом, а также понимание того, как его эффективно использовать, что может значительно улучшить рабочий процесс и производительность.