Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

Кафедра «ЭВМ»

**Отчет о практичкеской работе № 1**

«Работа с системами контроля версий»

**по дисциплине**

«Промышленное программирование»

Выполнили:

ст.гр. 345

Бригады №5

Костецкая В.Н.

Афонина О.А.

Проверили:

ст.преп.

ст. пр. Бастрычкин А. С.

Рязань 2025

**Цель работы**

Изучение системы версионного контроля на примере платформы GitHub и получение практических навыков работы с данной платформой.

**Практическая часть**

**Задание 1. I can win**

Установили git и сгенерировалли пару ssh ключей. Авторизовали публичный ключна github.com. Результат работы показан на рисунке 1-4.

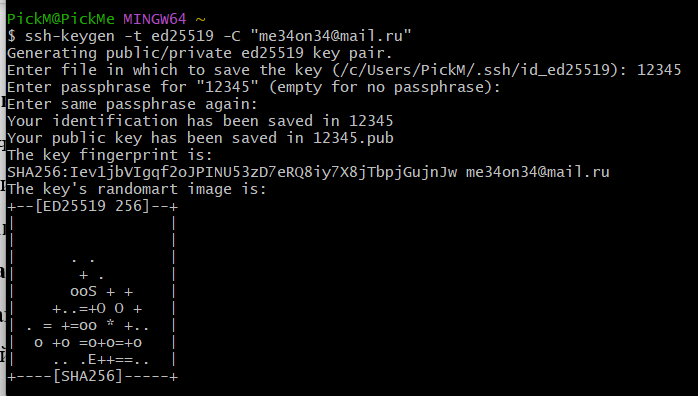


Рисунок 1 – Генерация SSH-ключей

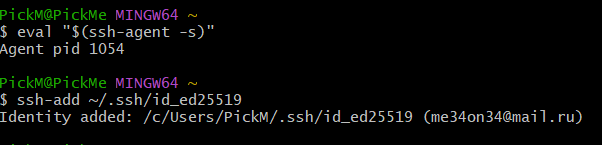


Рисунок 2 - Добавление SSH-ключа в ssh-agent

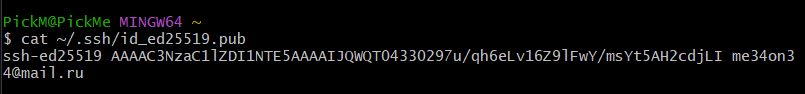


Рисунок 3 - Копирование публичного ключа

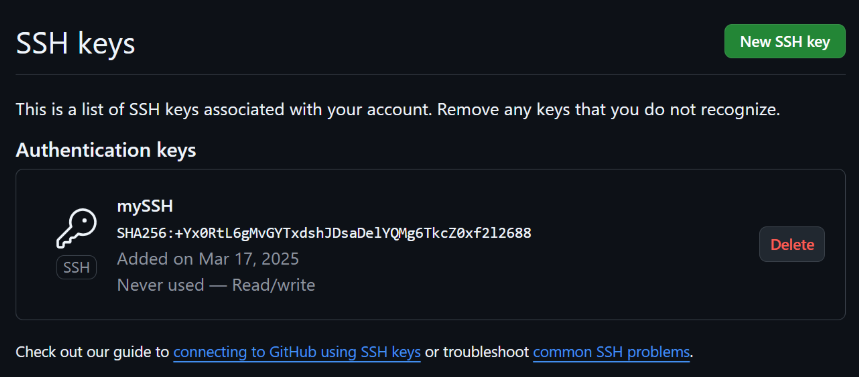


Рисунок 3 – Добавление SSH на GitHub

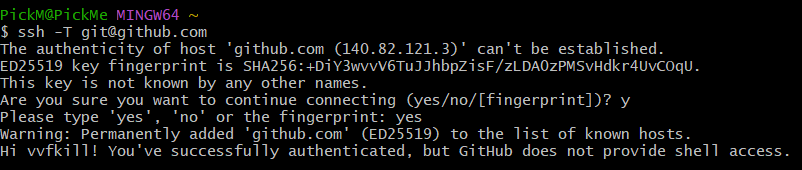


Рисунок 4 – Проверка подключения

Указали свой user.name и user.email в git (рисунок 5).

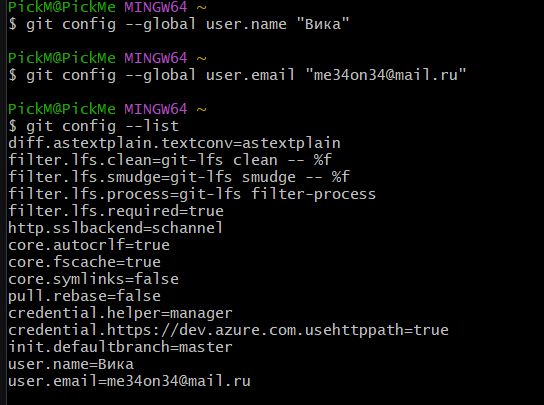


Рисунок 5 - User.name и user.email в git

Использовали веб-интерфейс github’а и создали новый public репозиторий **git-demo** содержащий README файл (Initialize with a README) на github.com. Результат можно увидеть на рисунке 6.

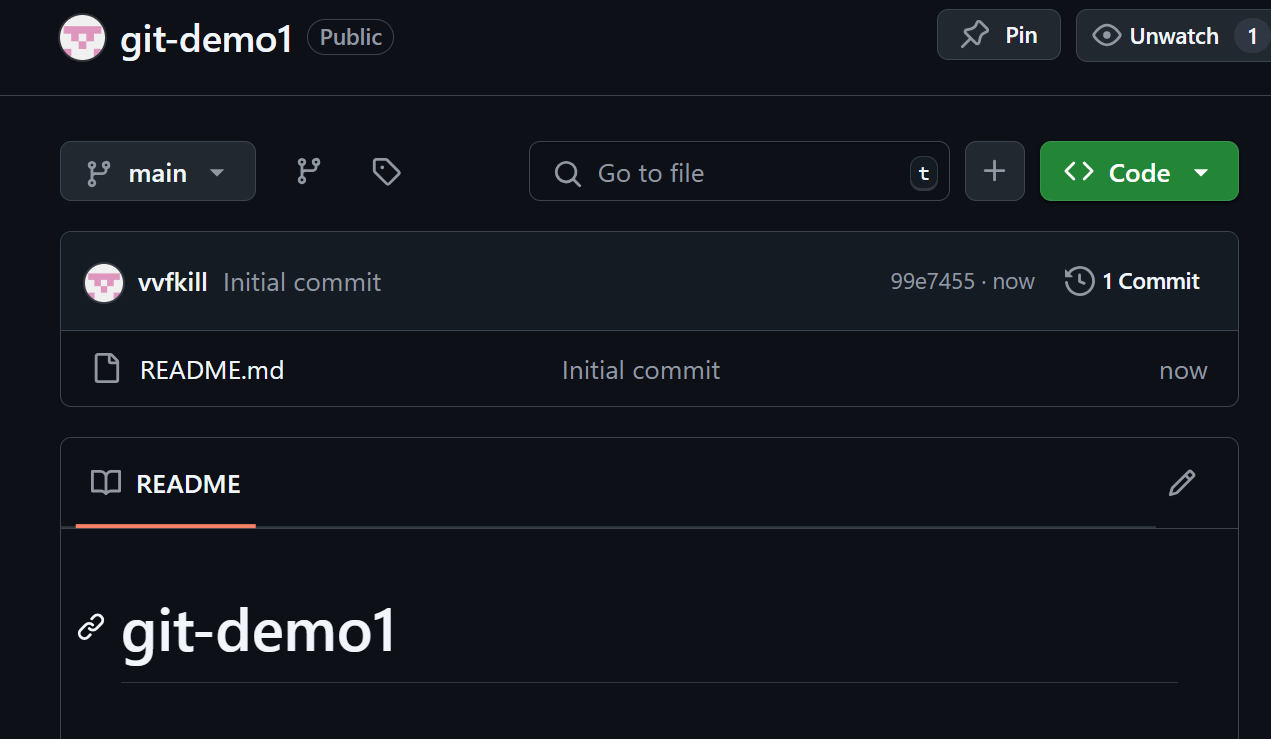


Рисунок 6 – Создание нового репозитория

Склонировали созданный репозиторий git-demo на свой компьютер (рисунок 7).

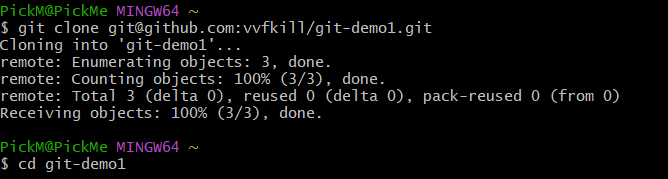


Рисунок 7 - Клонирование

Создали текстовый файл названием song.txt и поместили туда фрагмент текста песни, данный франгмент и создание показано на рисунке 8:

Sword and spear

wall of shields

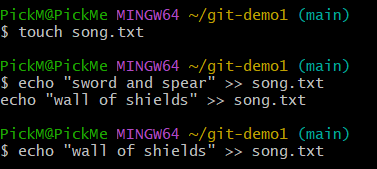


Рисунок 8 – Создание файла txt

Сделали коммит с названием "**add first two lines in the song**" (рисунок 9).

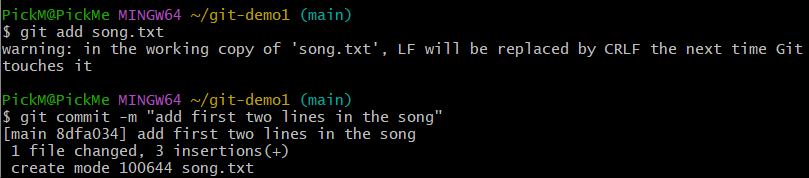


Рисунок 9 – Коммит

Отправили локальные изменения на сервер (рисунок 10).

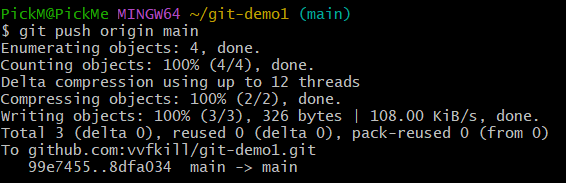


Рисунок 10 – Отправление локальных изменений

Убедились, что на github’е есть файл song.txt с текстом песни (рисунок 11).

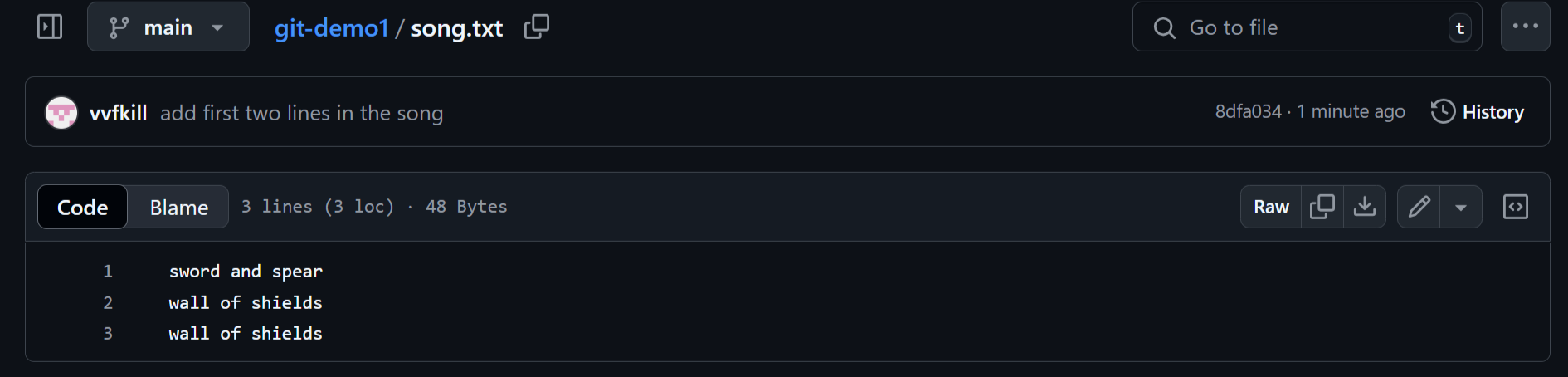


Рисунок 11 - Файл song.txt с текстом песни

Используя веб-интерфейс github’а добавили вторую половину текста песни в файл song.txt, результат показан на рисунке 12:

standing strong

on this their chosen battlefield

После сделали коммит с названием "**add two more lines to the song**".

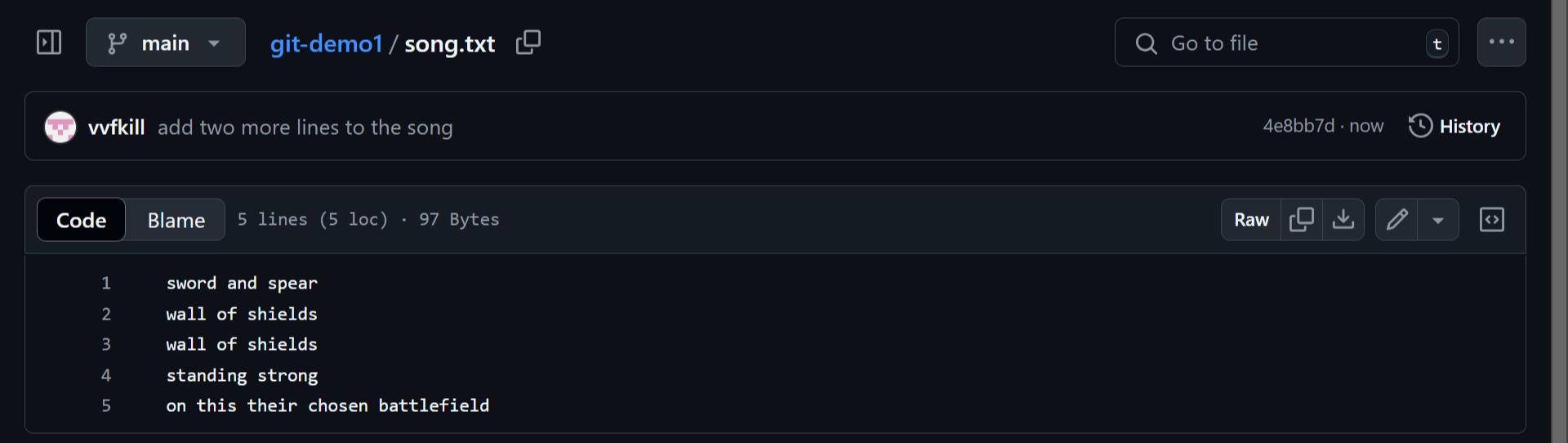


Рисунок 12 – Добавление и коммит второй половины текста

Выполнили синхронизацию локального репозитория с github’ом (рисунок 13).

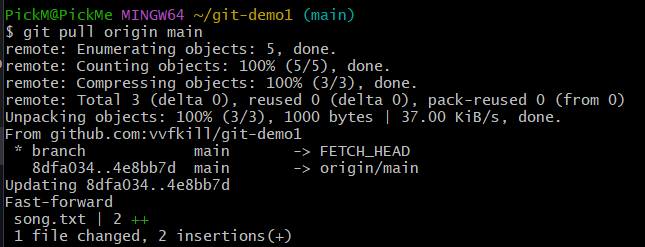


Рисунок 13 – Синхронизация

Убедились, что коммит, который вы создали на github’е, подтянулся и у вас полный текст песни в файле song.txt (рисунок 14).

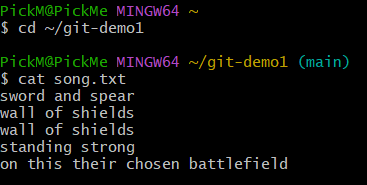


Рисунок 14 – Полный текст в файле

**Задание 2. Bring it on**

Добавили в проект файл .gitignore и настроили так чтобы скрыть файлы с расширением .db, .log и директории с названиями target или bin. Выполнение показано на риснуке 15-16.

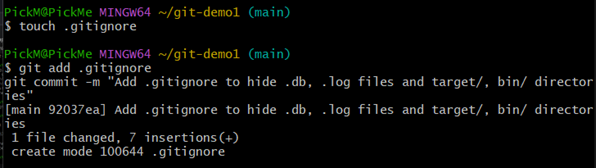


Рисунок 15 – Создание файла

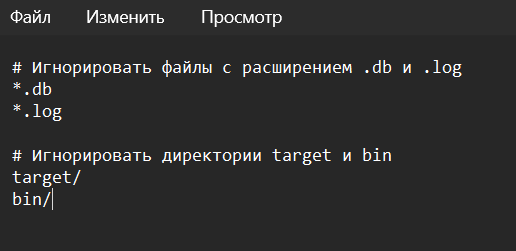


Рисунок 16 – Файл

Создали ветку feature и добавили в неё два коммита (рисунок 17).

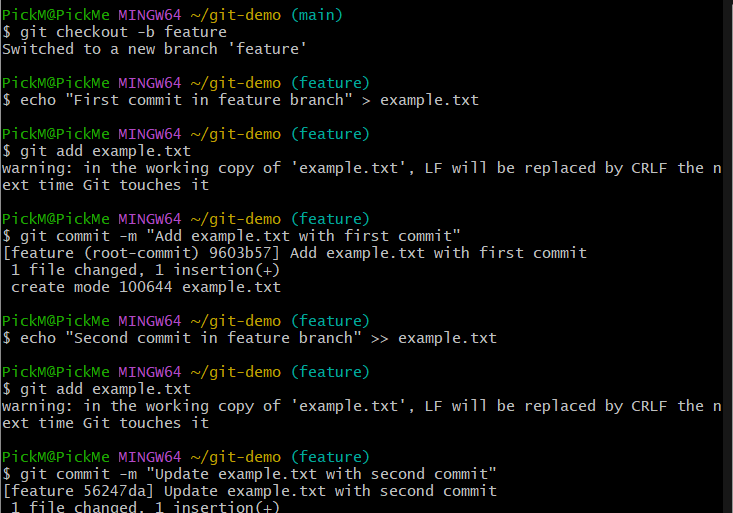


Рисунок 17 – Создание и добавление коммитов

Смержнули ветку feature в master, результат показан на риснуке 18.

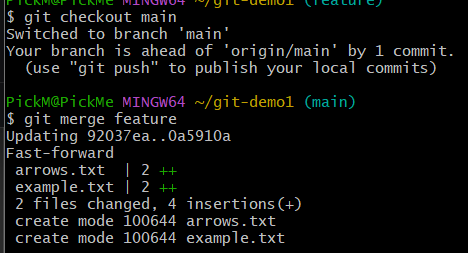


Рисунок 18 – Ветка feature в master

Вернулись в feature и создали файл arrows.txt cледующего содержания:

The ship glides gently on the waves

As day turns into night

Выполнили коммит. Данные действия показаны на рисунке 19.

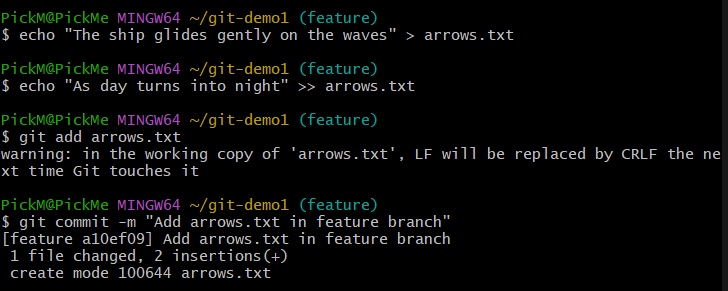


Рисунок 19 – Создание файла и выполнение коммита

Перешли в master. Создали там файл arrows.txt и добавили следующий текст:

One thousand burning arrows

Fill the starlit sky

Выполнили коммит (рисунок 20).

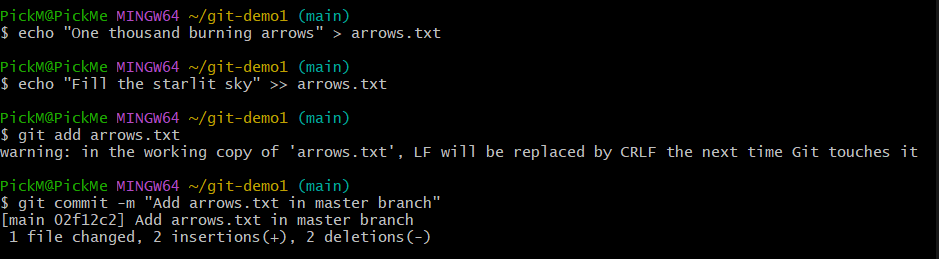


Рисунок 20 – Создание файла

Смержнули feature в master решив конфликт: сохранили все 4 строки в файле arrows.txt в порядке их добавления в предыдущих пунктах (рисунок 21-22).



Рисунок 21 - Объединение

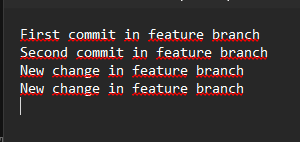


Рисунок 22 – Результат

**Задание 3. Hurt Me Plenty**

Создали ветку storm и добавили коммит в файл storm.txt:

Twenty ships with Norsemen braves

Riding the northern wind

Результат показан на рисунке 23.

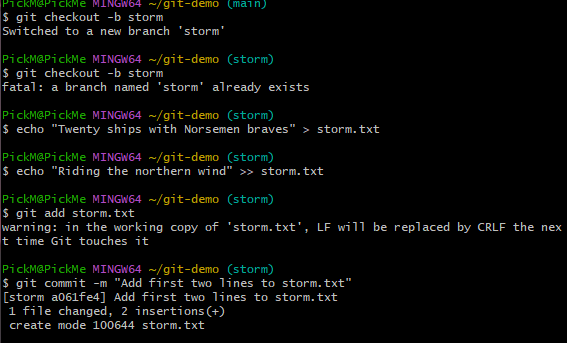


Рисунок 23 – Создание ветки и коммит

Добавили еще 2 строки в storm.txt и сделали еще один коммит (рисунок 24):

They left their shores at early dawn

As a red sun was rising in the east

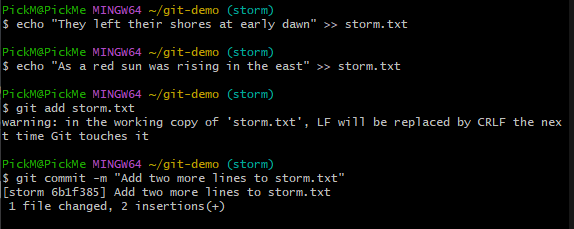


Рисунок 25 – Добавление 2-х строк

Вернулись в master и создали файл pursuit.txt с текстом ниже:

The warming sun returns again

And melts away the snow

The sea is freed from icy chainsWinter is letting go

Выполнили коммит, выполнение показано на рисунке 26.

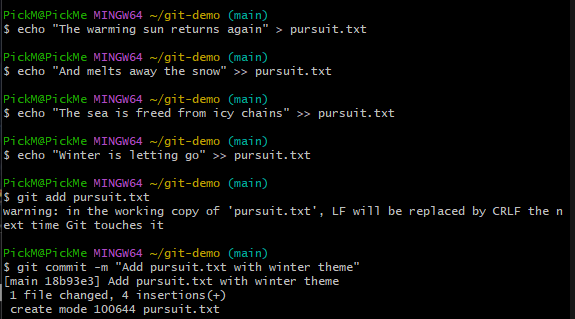


Рисунок 26 – Создание файла и коммит

Отметили коммит тегом session1 и перешли в ветку storm (рисунок 27).



Рисунок 27 - Тег

Сделали rebase ветки storm так чтобы она содержала последний коммит из мастера (рисуок 28).

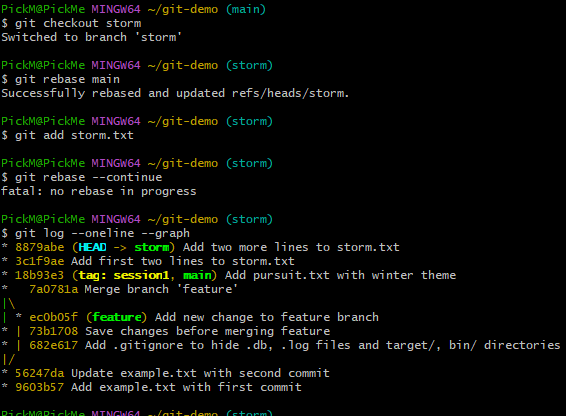


Рисунок 28 - Rebase

**Задание 4. Hardcore**

Сделали push вашего репозитория и убедились, что все коммиты есть на github (рисунок 29).

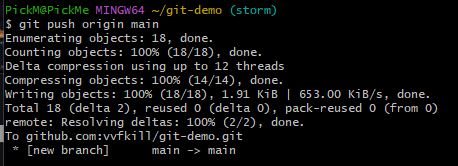


Рисунок 29 - Сделали push

Сделали новый репозиторий на github. Результат можно увидеть на рисунке 30.

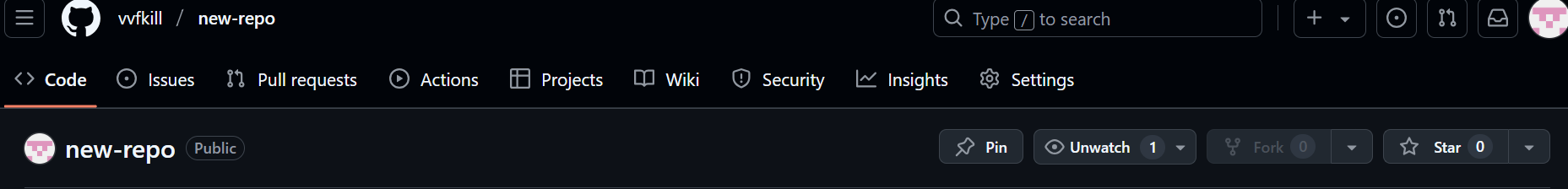


Рисунок 30 - Новый репозиторий

Сменили remote в локальном репозитории так, чтобы fetch и push шел на новый репозиторий который был создан в предыдущем шаге. Выполнение показано на риснуке 31.



Рисунок 31 – Смена remote

Сделали push и убедились, что второй репозиторий на гитхабе выглядит так же, как и первый (рисунок 32).

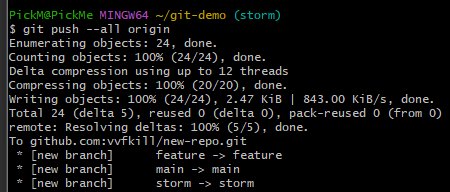


Рисунок 32 - Push

Вернули настройки remote в исходное состояние: пул и пуш первого локального репозитория ведет в один удаленный репозиторий на гитхабе. Результат показан на рисунке 33.

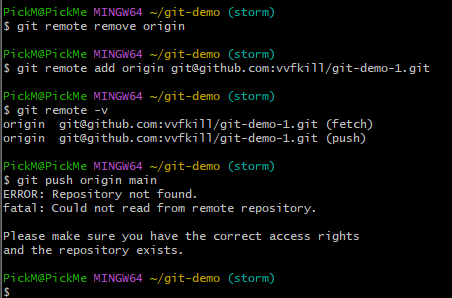


Рисунок 33 - Возвращение настроек

**Заключение**

В результате выполнения практического задания было произведено

знакомство с системами контроля версий, получены базовые

навыки работы с системой контроля версий Git, а также платформой Github,