МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Инженерно-экономический факультет

Кафедра экономической информатики

Отчёт

по лабораторной работе №2

«Работа с git»

по курсу «Современные технологии разработки web-пртложение»

Выполнил: Нтакпе Т,Д

студент гр. 172301

Проверил: А.М Рукманова

ассистент кафедры ЭИ

Минск 2023

**Лабораторная работа №2**

**Работа с git**

**Цель работы:** Изучить структуру распределенной системы контроля версий git, ее основные команды и варианты использования.

на рис. 1 мы создали репозиторий для нашего проекта git.

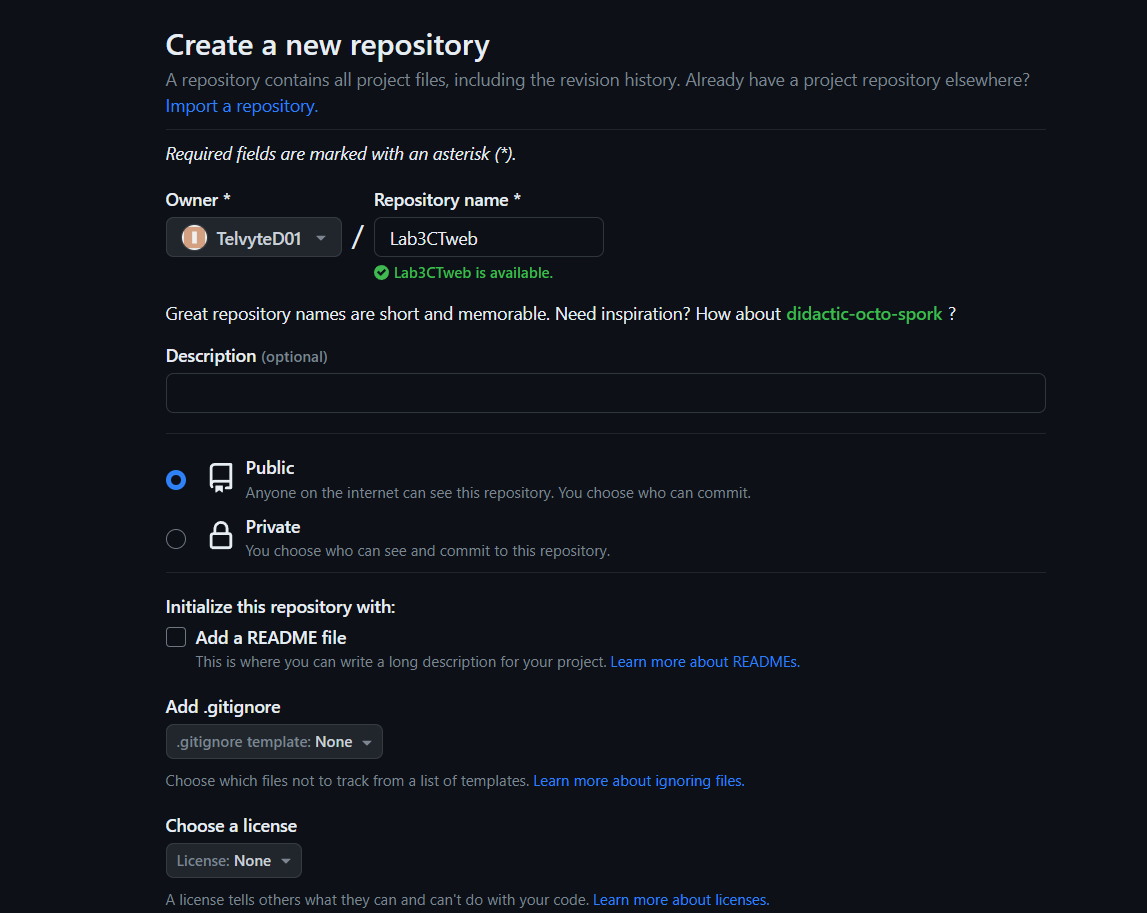


рисунок -1 git репозиторий

на рис. 2 мы создали файл read.me, который содержит некоторую информацию.

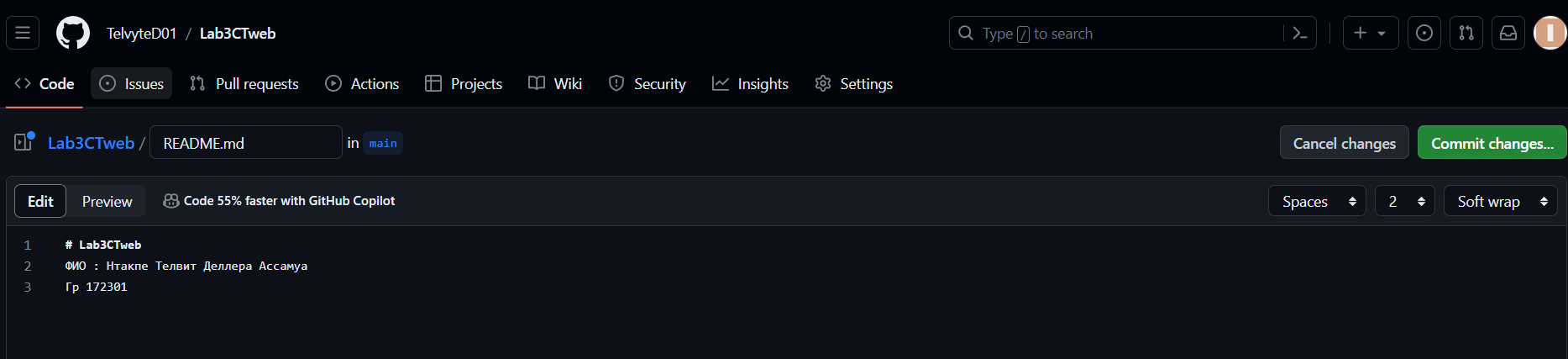


рисунок 2-создание файл read.me

затем мы приступили к настройке нашего проекта с помощью команды git config --global на рисунке 3.

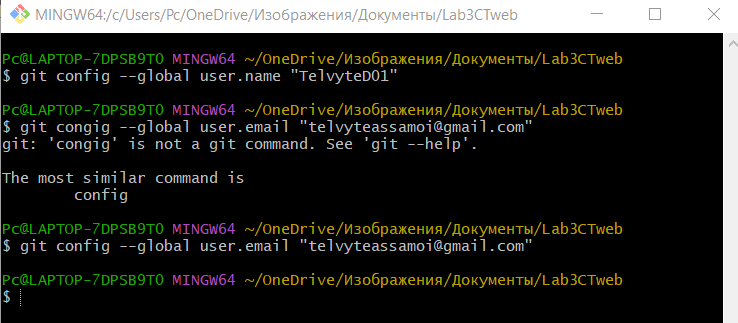


рисунок 3-команды git config

На следующем рисунке показано использование команды git config с параметром list на рисунке 4.

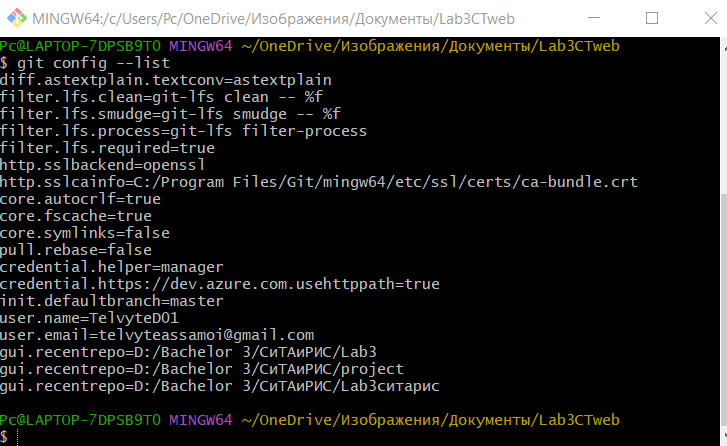


рисунок 4- показано использование команды git config с параметром list

Чтобы начать мониторинг существующего проекта, зайдите в папку этого проекта и введите команду (вы можете запустить ее, например, в нашем старом проекте. Для меня это будет проект с gson):

$git init

Эта команда создает новый скрытый подкаталог в текущем каталоге с именем .git, содержащий все необходимые файлы репозитория — основу репозитория Git.

эта команда используется на рисунке 5

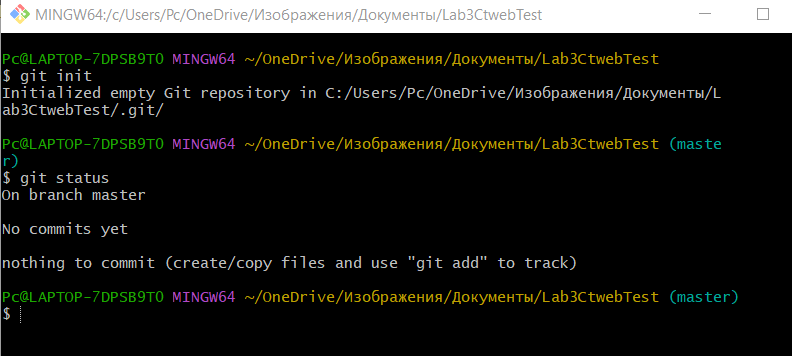


рисунок 5-git init

Команда git status отображает состояние рабочего каталога и промежуточной области. Это позволяет нам увидеть, какие изменения были внесены, а какие нет и какие файлы не отслеживаются Git. Вывод статуса не показывает нам никакой информации об истории проверенного проекта.

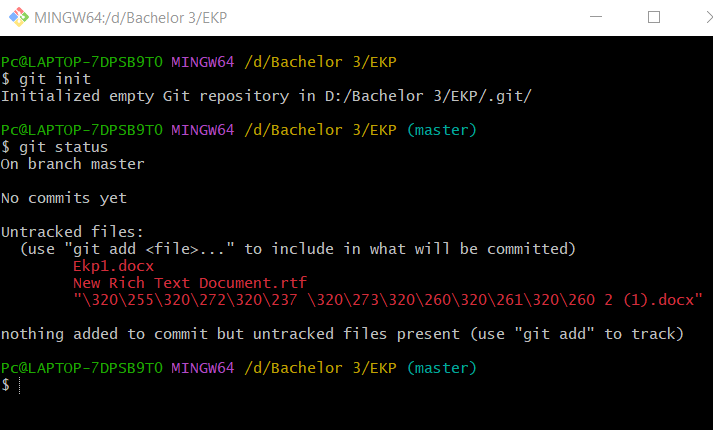


рисунок 6-git status

Git add — это команда, которая позволяет нам подготовить отдельные файлы или все файлы в каталоге проекта сразу, подготавливая их к переносу. Git add — одна из самых важных и фундаментальных команд Git, и существует множество способов ее использования.

на следующем рисунке мы увидим команду git add

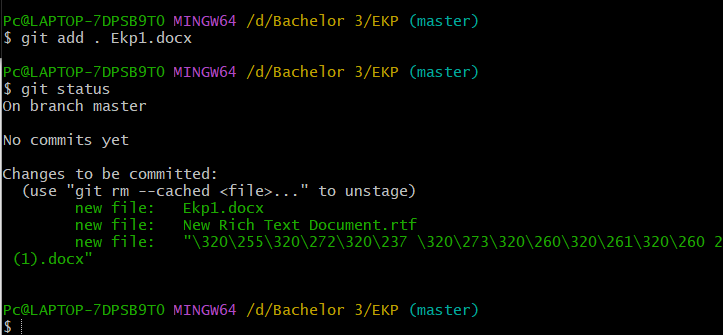


Рисунок 7- git add command

Команда git commit -m используется в Git для записи изменений, внесенных в репозиторий. Это часть процесса сохранения изменений в системе контроля версий. Вот разбивка команды:

git commit: эта команда создает новый коммит, содержащий изменения, внесенные вами в файлы в вашем репозитории. Коммиты — это снимки вашего проекта в определенный момент времени.

-m: за этим флагом следует сообщение о фиксации. Сообщение о коммите — это краткое описание изменений, внесенных вами в коммит. Хорошей практикой является написание четких и кратких сообщений, описывающих цель или причину изменений.

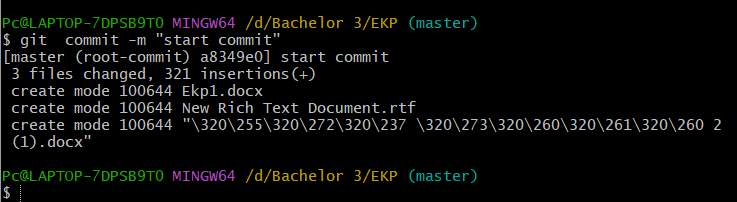
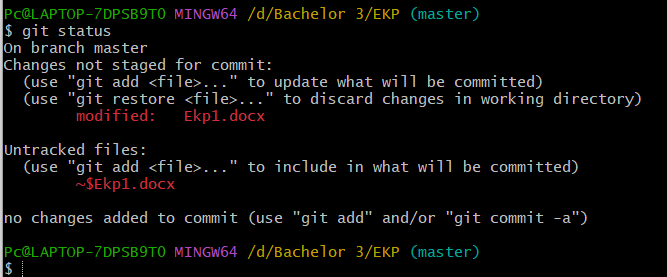


Рисунок 7 git commit -m

В Git есть флаг, который позволяет нам получать информацию в более компактной форме. Запуск git status -s или git status --short даст вам упрощенную версию вывода.



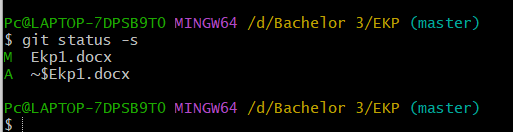


Рисунок 8 git status -s command

После сохранения нескольких версий файлов мы, вероятно, захотим взглянуть на то, что было сделано ранее. Основным инструментом в этом случае является команда git log.

Команда git log отображает в обратном хронологическом порядке список версий, зарегистрированных в данном репозитории. То есть самые последние коммиты отображаются первыми. Как видите, рядом с каждым коммитом указана его контрольная сумма SHA-1, имя и адрес электронной почты автора, дата создания и сообщение о коммите.

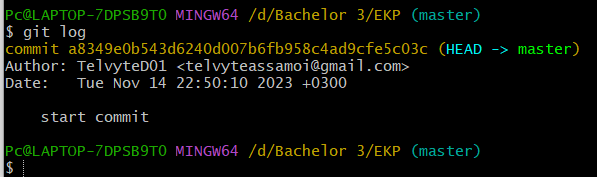
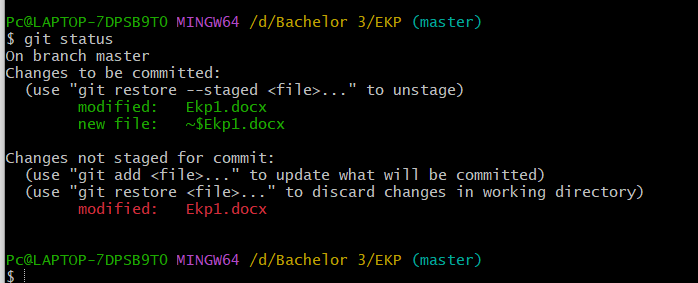


Рисунок 9 git log command

Когда мы вносим изменения в файл, отслеживаемый Git, и хотим отменить эти изменения, мы можем использовать команду git checkout с параметром --, за которым следует имя файла.

Мы проверяем текущий статус файла: мы используем команду git status, чтобы увидеть, какие файлы были изменены в вашем рабочем каталоге.



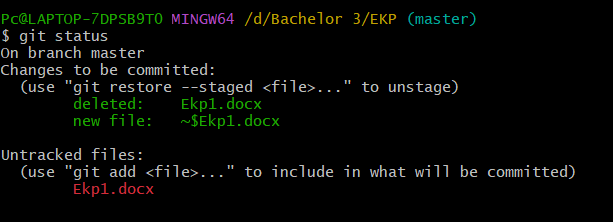


Рисунок 10- Отмена внесенных в файл изменений

git clone — это утилита командной строки Git, которая используется для выбора существующего репозитория и создания клона или копии целевого репозитория. На этой странице мы обсудим расширенные параметры конфигурации и распространенные случаи использования git clone. Здесь мы рассмотрим следующие моменты: Клонирование локального или удаленного репозитория.

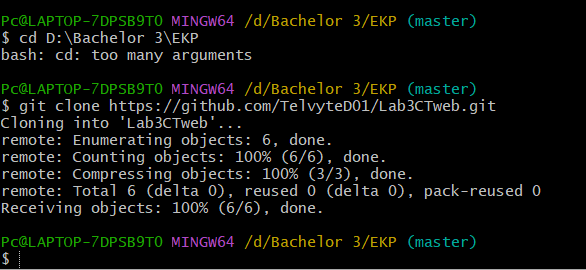
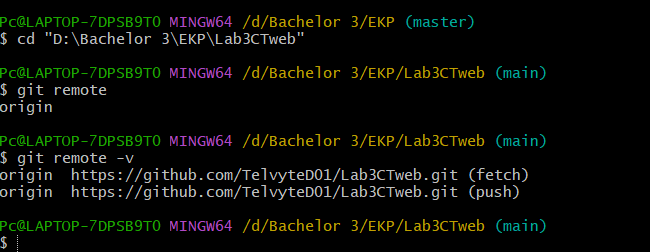
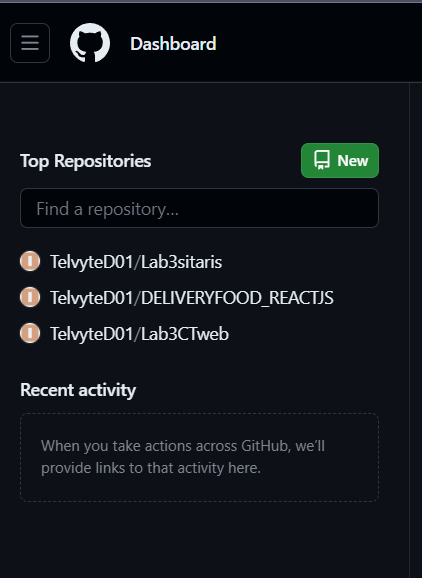


Рисунок 11- git clone

Удаленные репозитории — это версии вашего проекта, хранящиеся в Интернете или где-либо еще в сети. у нас может быть несколько удаленных репозиториев, каждый из которых может быть доступен для чтения или чтения-записи. Взаимодействие с другими пользователями предполагает управление удаленными репозиториями, а также отправку и получение данных из них. Управление репозиториями включает в себя как возможность добавления новых, так и возможность удаления устаревших репозиториев.





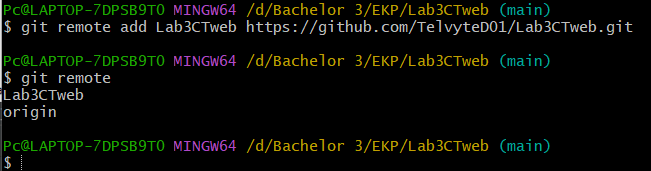
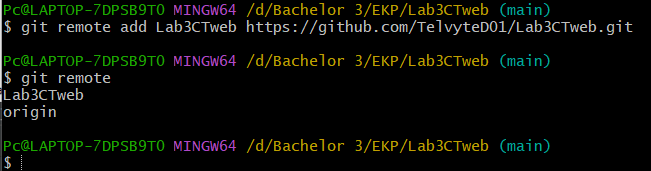


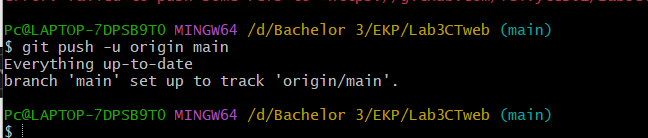
Рисунок 12 Работа с удаленными репозиториями

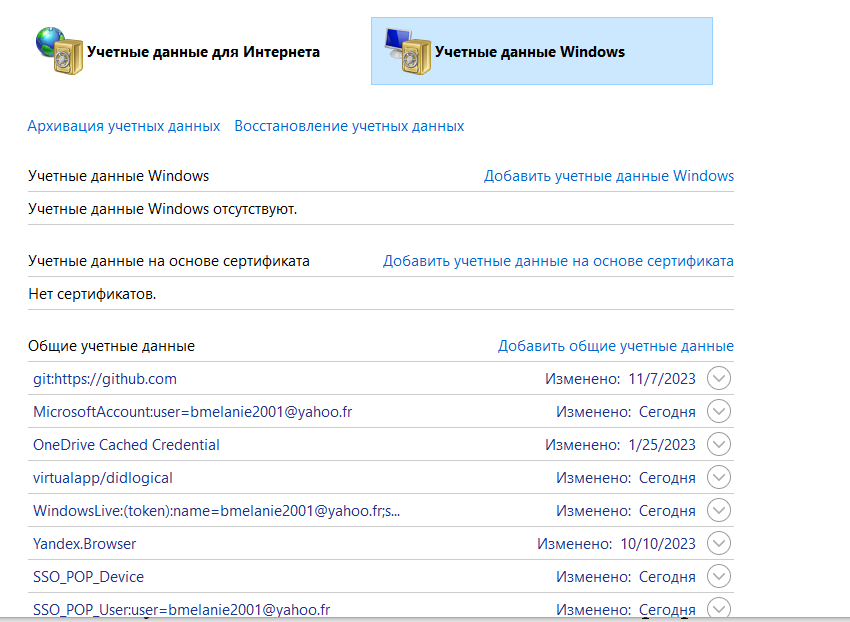
## Извлечение данных из удаленных репозиториев



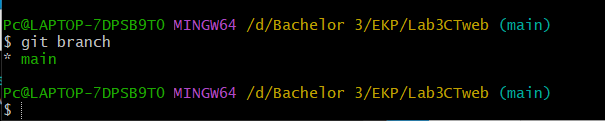
***Отправка изменений в удаленный репозиторий (Push)***

Если вы захотите поделиться своей работой, нам нужно загрузить ее в основной репозиторий. Команда для этого действия проста: git push [имя удаленного устройства] [имя ветки]. Чтобы отправить нашу главную ветку на исходный сервер (опять же, при клонировании эти два имени обычно настраиваются автоматически), мы можем запустить следующую команду, чтобы отправить задание на сервер:

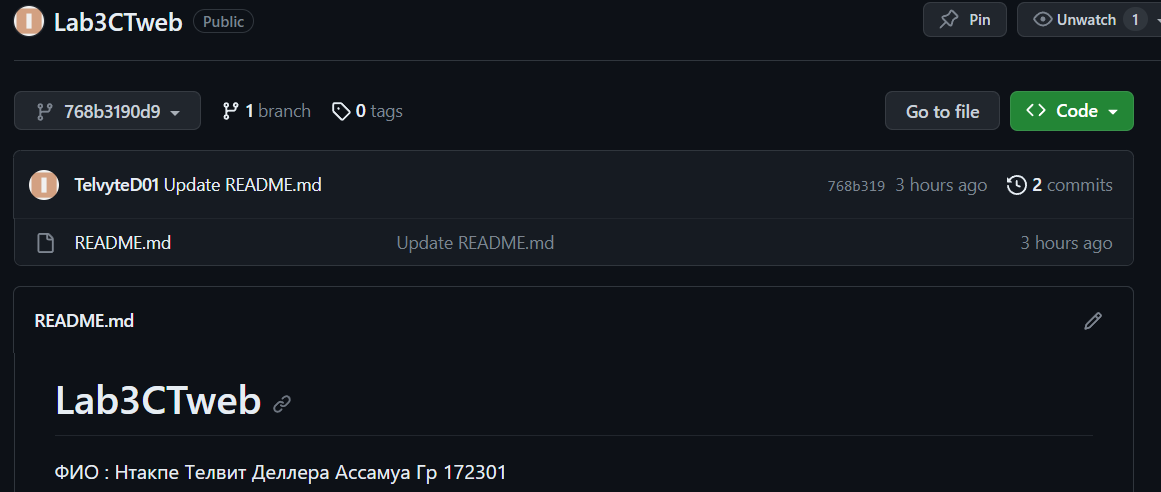


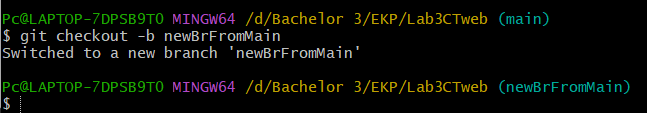


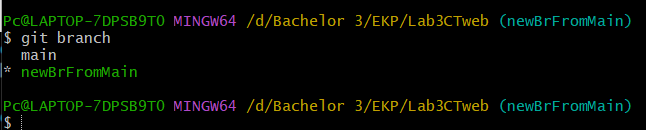
Команда git Branch в Git используется для просмотра, создания или удаления ветвей.

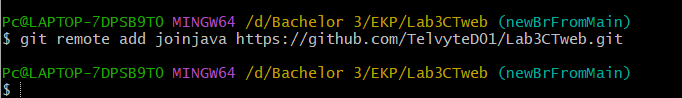












Команда git pull в Git используется для извлечения и объединения изменений из удаленного репозитория в вашу текущую ветку. Это комбинация двух команд: git fetch и git merge.

