Отчет по практике Работа с Git. Основы

Выполнил студент: Болдырев Антон

Группы: 4ИС-20

Г. Иркутск

**Практическая работа №1. Работа с Git. Основы**

Что такое Git?

Git — это распределенная система управления версиями, которая используется для отслеживания изменений в исходном коде программного обеспечения и координации работы нескольких разработчиков над одним проектом. Он был создан Линусом Торвальдсом в 2005 году.

Основные цели Git:

1. **Отслеживание изменений:** Git позволяет фиксировать каждое изменение в исходном коде, что обеспечивает полный контроль над историей проекта.
2. **Коллаборация:** Git позволяет нескольким разработчикам одновременно работать над одним и тем же проектом. Каждый разработчик может вносить свои изменения, а затем объединять их с изменениями других.
3. **Ветвление и слияние:** Git позволяет создавать отдельные ветки для работы над конкретными функциональностями или исправлениями ошибок. Затем эти ветки могут быть объединены (слиты) обратно в основную ветку.
4. **Локальная работа:** Git позволяет разработчикам работать локально, не завися от сетевого соединения. Это увеличивает эффективность и удобство работы.
5. **Эффективное хранение данных:** Git использует эффективные алгоритмы хранения данных, что делает его компактным и быстрым.
6. **Откат к предыдущим версиям:** Разработчики могут легко откатываться к предыдущим версиям проекта, если что-то пошло не так или если нужно вернуться к предыдущей версии кода.

В целом, Git предоставляет мощные инструменты для управления версиями и совместной разработки программного обеспечения, делая процесс более удобным и эффективным для команд разработчиков.

**Шаг 1. Установка GIT**

Переходим на официальный сайт и устанавливаем GIT

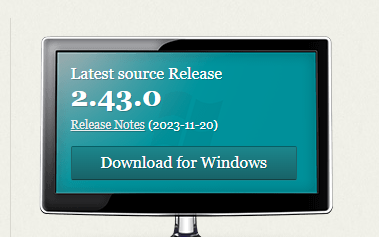


Рисунок 1 – Установка GIT

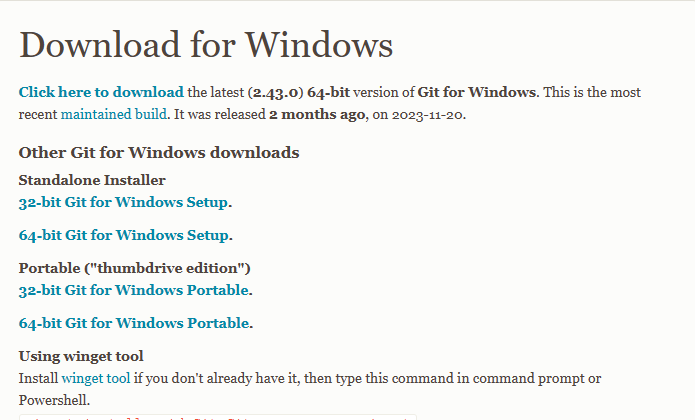


Рисунок 2 – Выбор версии при установке

Выбираем версию 64-бит в пункте “standalone installer” и запускаем установщик



Рисунок 3 – Установщик

Проводим процедуру установки и ждем завершения

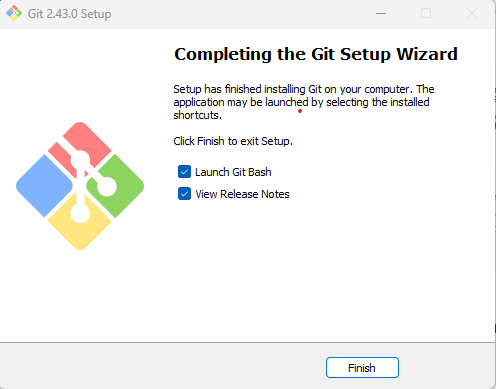


Рисунок 4 – завершение установки

После завершения запустим GIT через “Open Git Bash here”

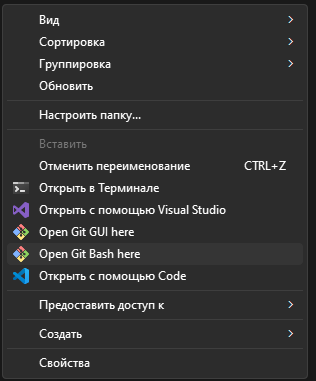


Рисунок 5 – окно Git

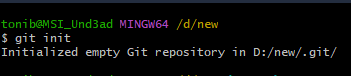
****

Рисунок 6 – git init

Git init – Команда которая нужна для создания используемого репозитория в Git в локальной файловой системе. Когда выполняется команда внутри папки создается скрытая папка с расширением. git которая содержит в себе файлы и структуру, которая отслеживает изменения в проекте.

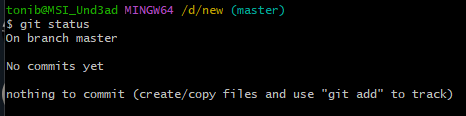


Рисунок 7 – git status

Git status – команда используемая для проверки состояния раннее созданного репозитория Git. Команда показывает информацию о ветке, измененных файлах, новых файлах, удаленных файлах и других изменениях, которые ещё не были зафиксированы (через команду git commit)

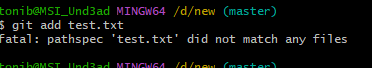


Рисунок 8 – git add

Команда git add – используется для добавления изменений в файлы в индекс готовых к сохранению (коммиту) файлов. При выполнении команды git add, Git добавляет изменения, сделанные в указанных файлах. В свою очередь команда git add (без указания файла) добавит все измененные файлы.

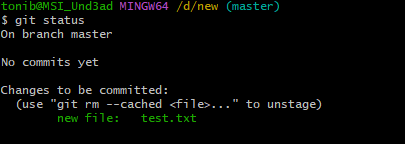


Рисунок 9 – закрепление изменений

Проверяем изменения

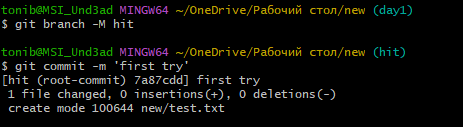


Рисунок 10 – git branch

Git branch – (ветка) это ссылка на определенный коммит.

Основной идеей ветвления является отклонение от основного кода и продолжение работы независимо от него. Также это удобно в тестировании отдельного функционала, потому что позволяет работать над новой частью кода, не беспокоясь о поломке чего-то в рабочей версии.

**Подсказки для выполнения задания:**

git init – (делается один раз в папке)

git status

git add test.txt

git status

Если добавить все файлы, то git add.

git commit -m ‘Мой первый пуш’

git branch -M day1

git remote add origin А\_ТУТ\_ССЫЛКА\_С\_GITLAB (делается один раз)

git push -u origin day1 (после одного раза можно делать git push)

touch readme.md

git checkout (поменять ветку)

Самостоятельная работа

5) Слияние используется в Git, чтобы собрать воедино разветвленную историю. Команда git merge выполняет слияние отдельных направлений разработки, созданных с помощью команды git branch, в единую ветку.

Создание различных веток

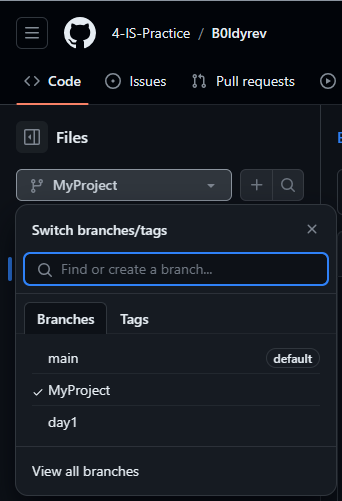


Рисунок 11 – Branches

Созданы 3 ветки, одна из которых имеет файл README



Рисунок 12 – часть README

Более подробно посмотреть на отчет и прочитать файл Readme можно в репозитории B0ldyrev

3) посмотреть видео. В данном задании рассказывалось о расширении “Makdrown Preview Enhanced”



Рисунок 13 – расширение “Makdrown Preview Enhanced”

Данное расширение мы использовали для более презентабельного редактирования файла README в GitHub.