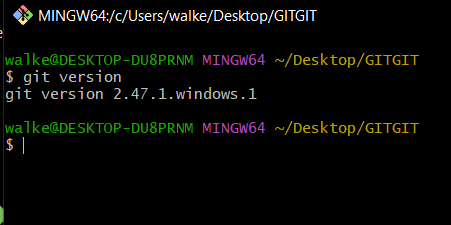
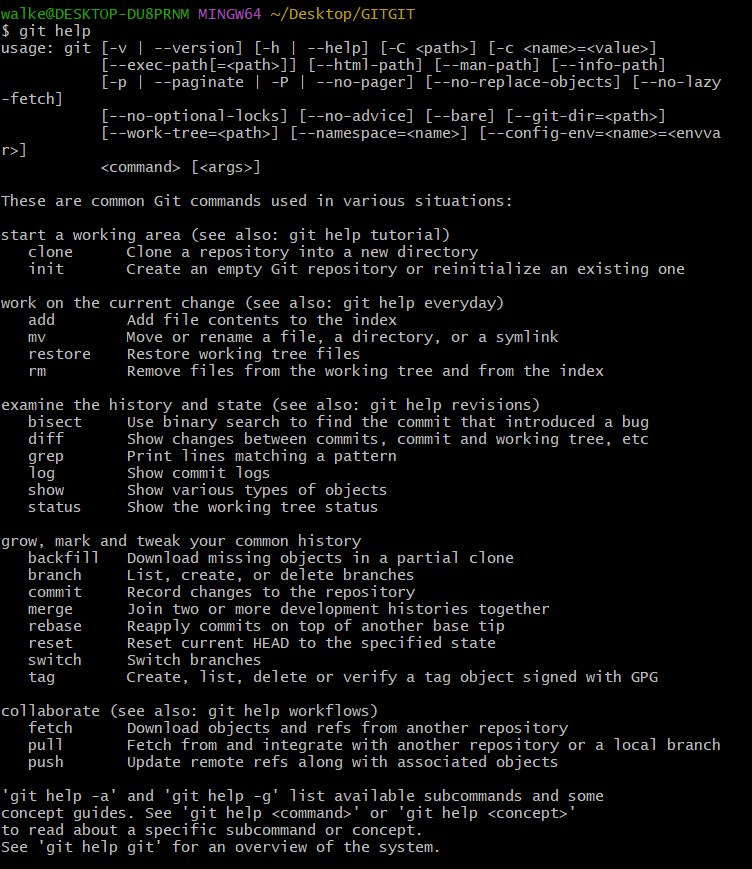
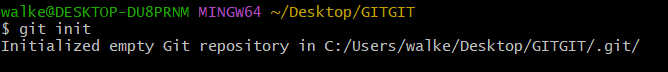
**Проверка установленной версии**



**Для получения справки по Git используем команду git help и получим:**



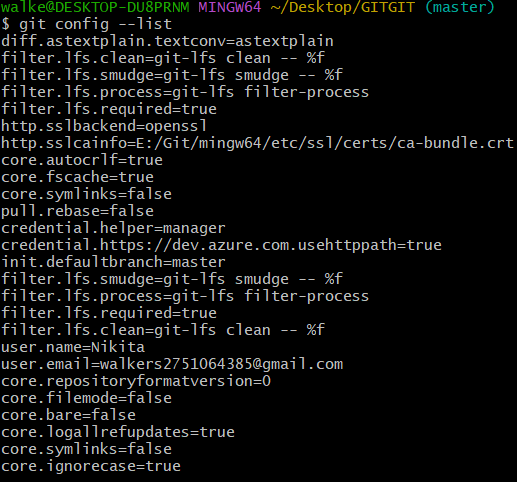
Создание локального репозитория с помощью команды git init







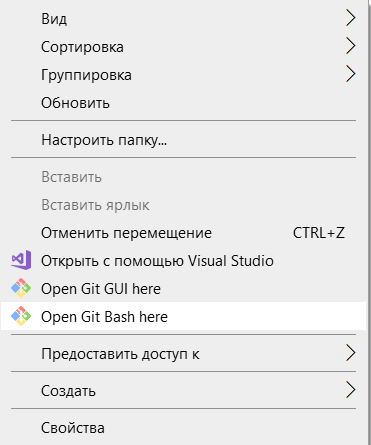
Проверка настройки конфигурационного файла командой **git config –list**



Инициализация локального репозитория в рабочей папке, где он будет размещён. Исследование изменения, которые произошли в папке:

1) Переходим в рабочую папку

2) Открываем **Git Bash here**



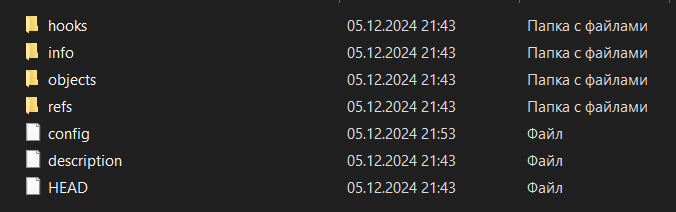


В папке появилась новая скрытая папка: .git – локальный репозиторий

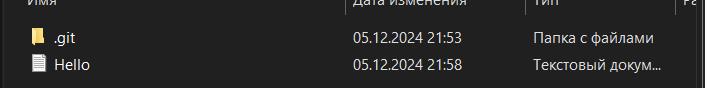
Вот она:

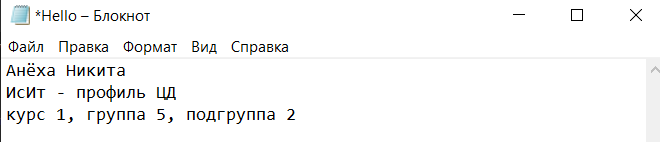


Внутри:

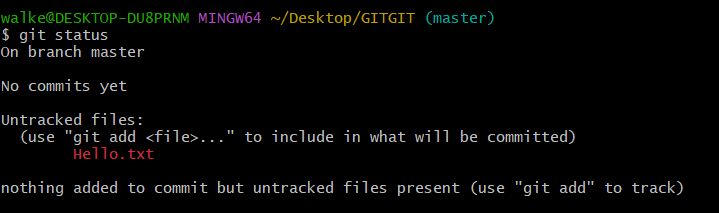


Создание текстового файла Hello:





Получение информации о статусе репозитория с помощью **git status:**



Можно заметить, что создана ветка **master**

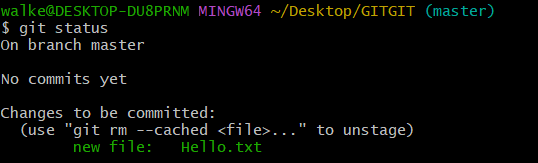
Красным цветом отмечены новые и модифицированные файлы и папки

Индексация файла Hello.txt и изменения в репозитории:

1) Добавим файл Hello.txt в репозиторий с помощью команды git add Hello.txt

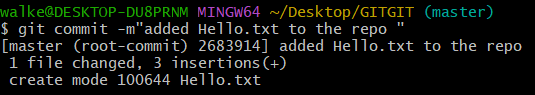


2) Проверим обновлённое состояние репозитория с помощью команды **git status**:

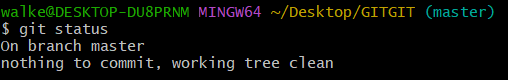


Теперь наш файл Hello.txt подсвечивается зелёным цветом, это значит, что он проиндексирован

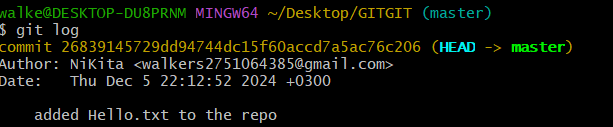
Зафиксируем изменения индексированных файлов в репозитории с помощью команды **git commit –m “added Hello.txt to the repo”**:

 Ключ –m позволяет добавить комментарий, описывающий, что именно было изменено в коммите ("added Hello.txt to the repo"). Файл добавлен в ветвь **master**

Проверим изменения в репозитории:

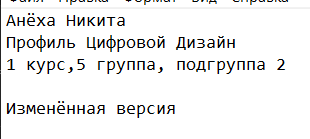


Просмотр журнала с помощью команды **git log**:

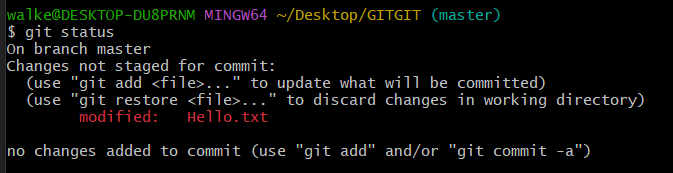


Получение информации о проиндексированных и неиндексированных файлах.

Изменим уже существующий файл Hello.txt и сохраним изменения



Просмотрим статус нашего репозитория



Hello.txt теперь modified: Hello.txt, что указывает на изменения файла

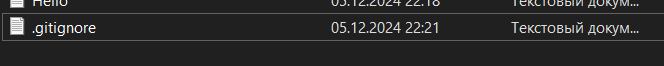
При добавлении этого изменённого файла получим

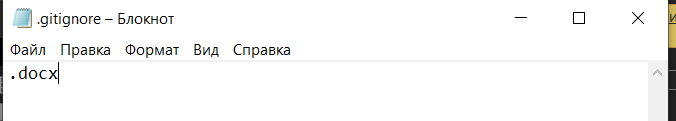


Теперь оно подсвечивается зелёным цветом

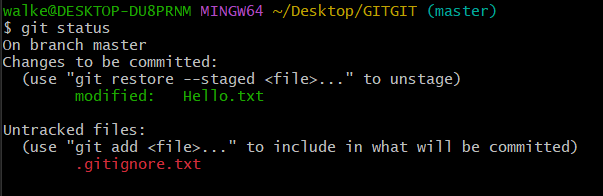
Настройка файла .gitignore

Создаём текстовый файл и прописываем внутри данного файла \*[расширения файлов, которые не хотим, чтобы они отображались в репозитории]

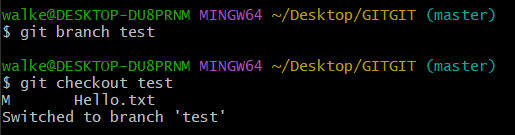




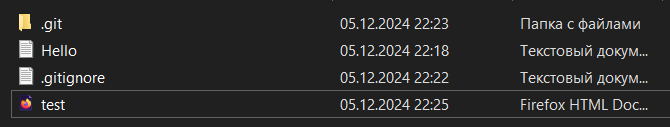
Далее просматриваем статус нашего репозитория и видим, что файл с расширением **.docx** больше не отображаются

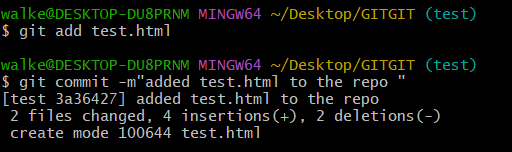
 Создание новой ветки

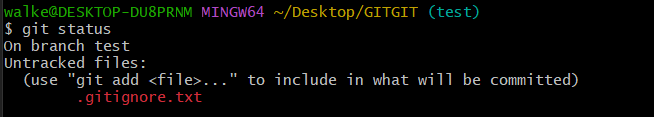
Создание (branch) новой ветки, например, test и переключитесь (checkout) в нее. Переключим ветки с помощью команды **git checkout test**



Создаём новый файл **test.html** и фиксируем его







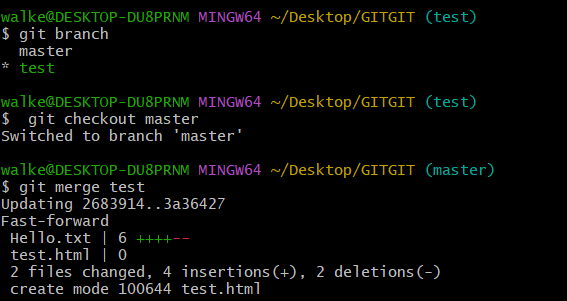
C помощью команды **git branch** просмотрим список всех веток

Текущей веткой являет **test**

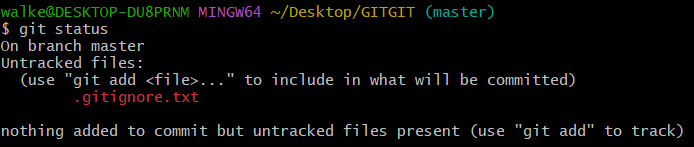
Выполним слияние веток с помощью команды **git merge**

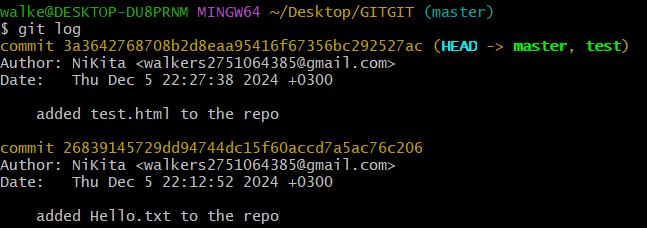
Переходим в ветку в которой будем выполнять слияние, а именно **master**

Для этого используем команду **git checkout master**

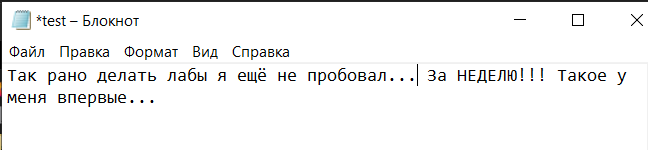


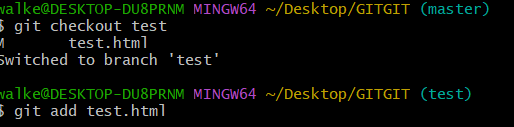
Просмотрим статус репозитории





По очереди отредактируем файл test.html в ветках master и test.





Разрешим ситуацию конфликта слияний.

Изменим наш текстовый документ и оставим тот вариант, который нас устраивает, сохраним