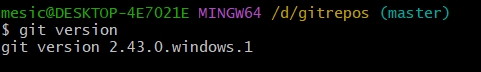
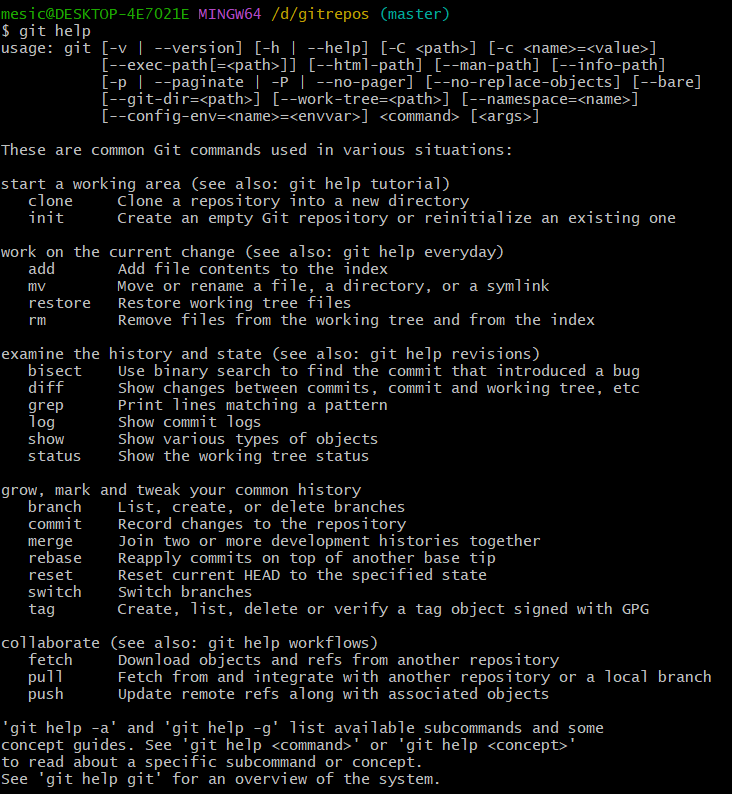
Лабараторная работа №6 ВРПО

Отчёт о проделанной работе

Мы проверили работу программы



Получили весь список команд с помощью команды $ git help



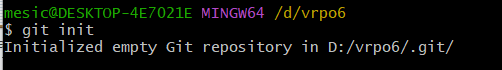
Создаём локальный репозиторий:

1.Создаём папку, в которой не инициализирован репозиторий

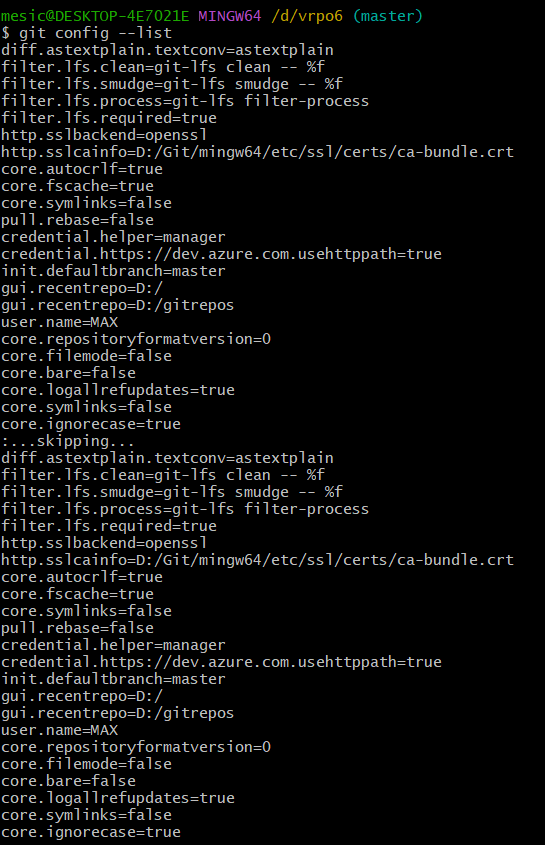
2. Затем открываем папку с помощью Git Bush (Git Bush Here)

3. Далее вписываем $git init

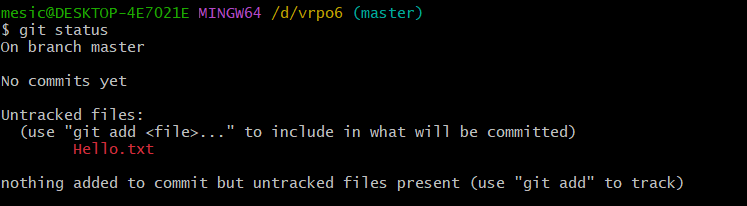
Готово



Проверяем настройки конфигурационного файла командой $git config --list

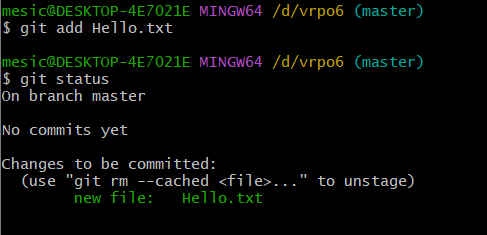


Проверили работу команды $git status



На данном снимке мы видим, что у нас в репозитории появился файл “Hello.txt” он отмечен красным, т.к. он не был ранее инициализирован либо он был недавно создан

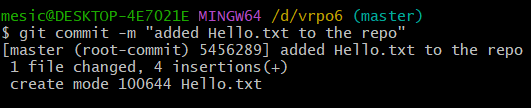
Далее нас просят проиндексировать файл Hello.txt



Вуаля! Файл у нас проиндексирован, об этом у нас говорит зелёный цвет нашего файла

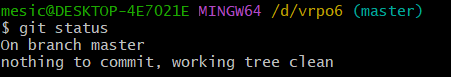
Далее просят зафиксировать в репозитории данный файл

Выполняем с помощью $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"))

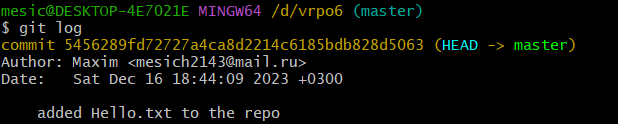


Git информирует об успешном создании нового коммита, в ветку master добавлен 1 файл

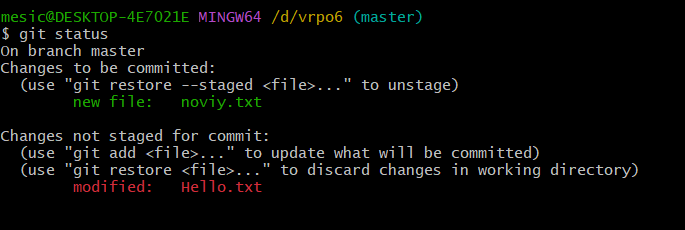
Файл зафиксирован в репозитории и мы можем узнать с помощью команды $git status изменение индексированных файлов



Посмотрели журнал гита



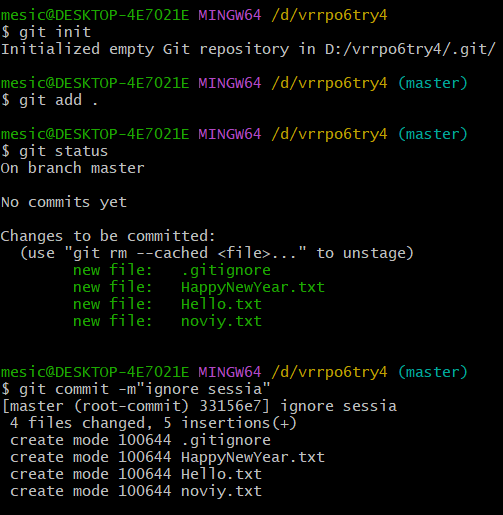
Далее просят изменить уже существующуий и добавить новый файл, добавить в репозиторий и посмотреть изменения

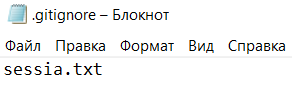


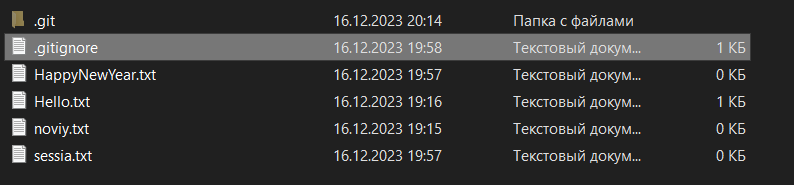
Собственно наш первоначальный файл стал снова красным цветом, а новый - зелёным

Затем просят внесни файл .gitignore в репозиторий, в который мы заранее внесли файл, который хотим исключить

На картинке ниже можно увидеть другой репозиторий, это всё связанно с тем, что я изначально сделал не правильно и получилась какая-то каша, поэтому сделал заного данную операцию, и в той же картинке виден результат операции

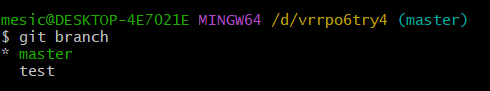




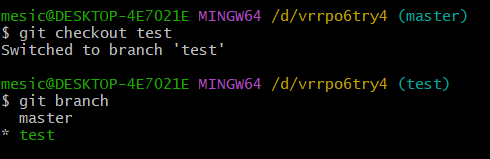


Далее нас просят создать ветку test и переключится на неё



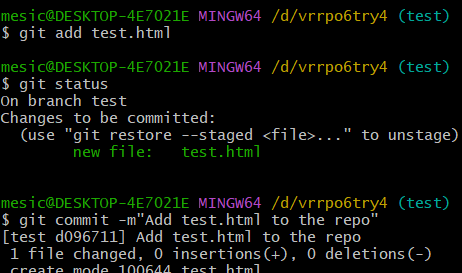


Переход на новую ветвь происходить благодаря $ git checkout <branch\_name>

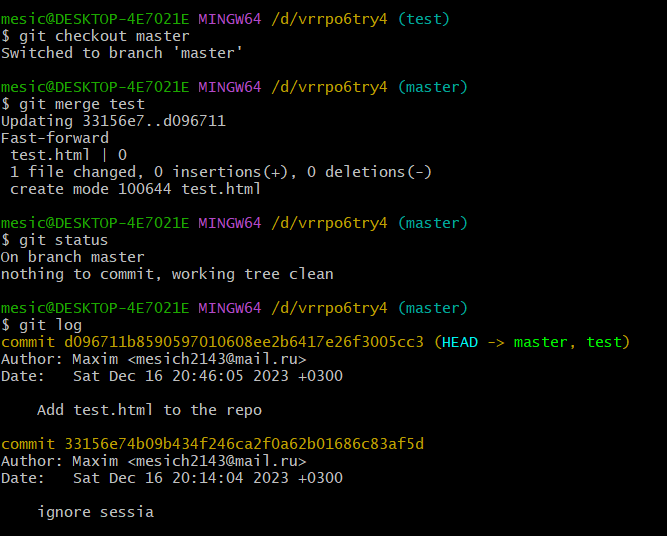


Таким образом далее мы работаем с веткой test

И нас просят создать файл test.html и зафиксировать его



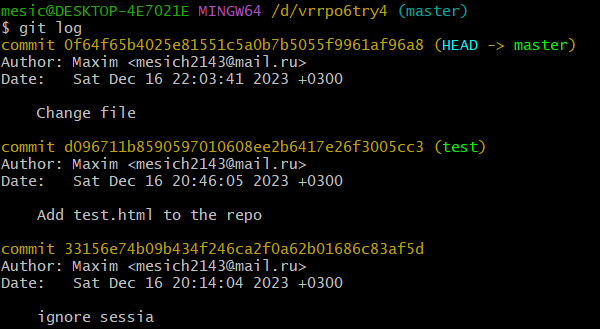
Выполняем слияние веток



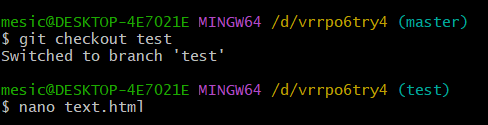
И так, нас просят изменить файл в разных ветках



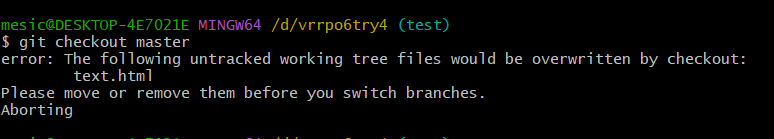
После мы видим что у нас меняется значение в нашем документе, добавляем и коммитим его



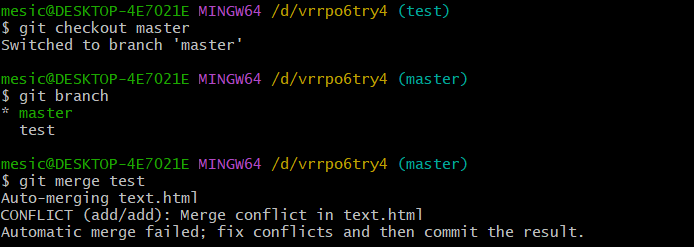
Далее переходим в другую ветку и делаем тоже самое, только изменяем на другой текст, к примеру.



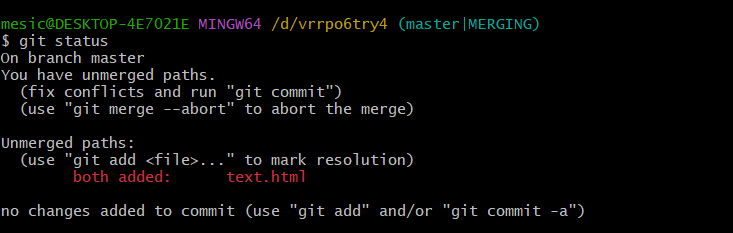
Потом снова переходим в ветку master и видим что у нас вылазит проблема слияния



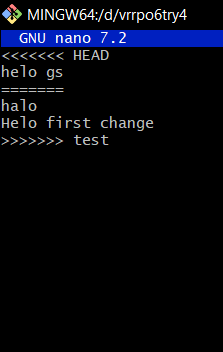
Исходя из этого мы вынуждены добавить данный элемент и закоммитить его в нашу ветку master. Находясь в ветке master выполним слияние снова. И вновь видим что у нас конфликт изменённых файлов.



Вводя git status мы увидим что у нас есть конфликт

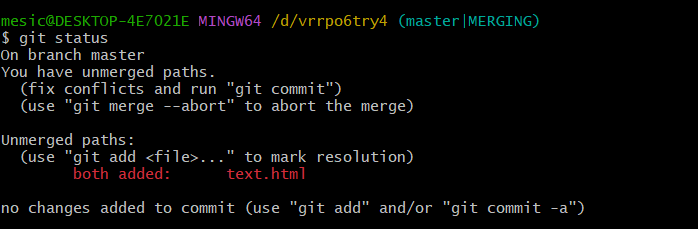


Давайте зайдём в файл и посмотрим как выглядит данный конфликт



В верхней части мы видим изменения главной ветки, а после знаков (============) у нас идёт изменение ветки test, и нам нужно отредактировать так, как мы хотим видеть документ в конце.

После редактирование прописывая git status мы видим следующее



Нам предлагают либо убрать слияние ( git merge –abort), либо добавить на индексацию и закоммитить. Коммит выполняем следующей командой: $git commit

Дальше нам предлагают комментарий который описывает ситуацию которая у нас происходила



Далее для того чтобы принять этот коментарий мы пишем :wq(write and quit)

КОНЕЦ.

