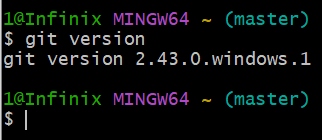
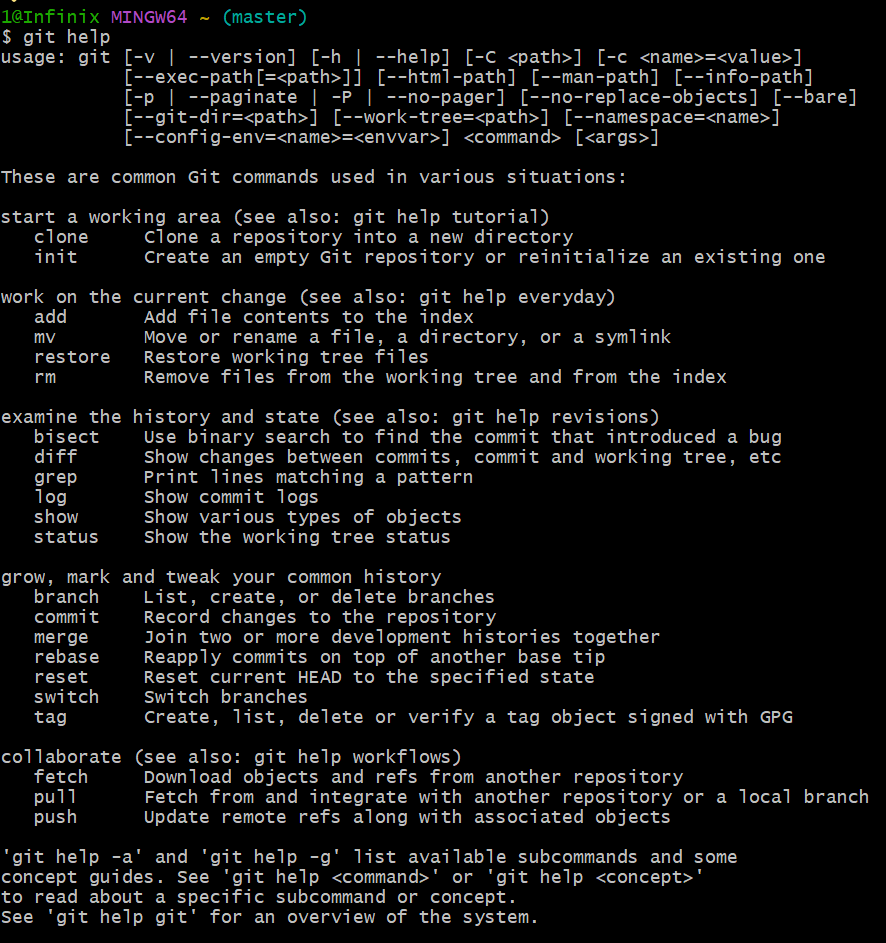
**Лабораторная работа 6**

**Системы контроля версий. Основы работы в Git**

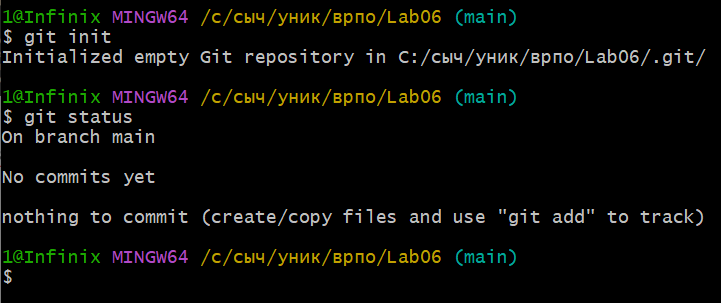
4. Проверили установленную версию.



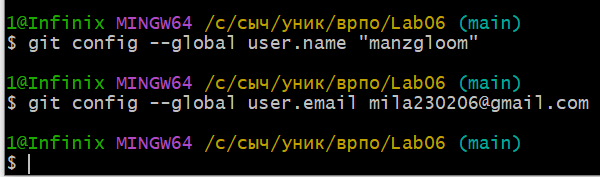
5. Получаем справку по Git с помощью git help.



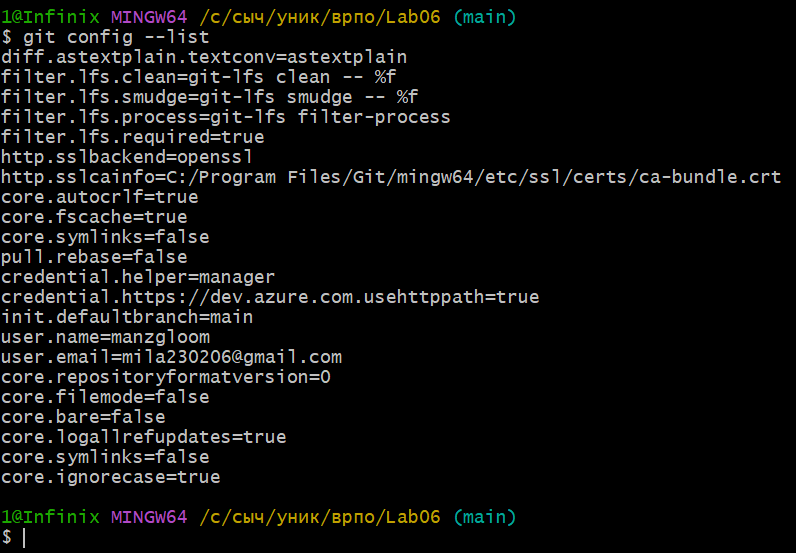
6. Открываем Git Bash через контекстное меню в нужной папке. Создаем локальный репозиторий с помощью git init.



7. Настраиваем конфигурационный файл с помощью команд git config --global user.name и git config --global user.email

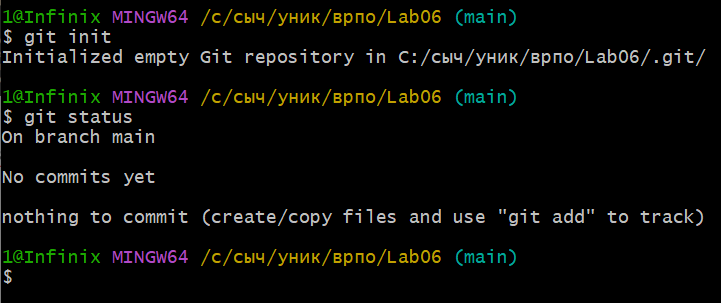


8. Проверяем настройки конфигурационного файла командой git config --list

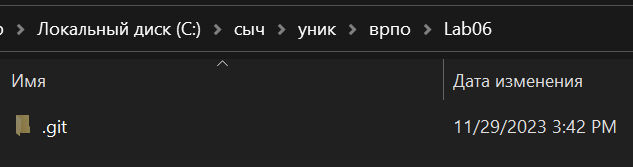


Появились значения user.name и user.email, введенные ранее.

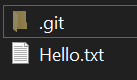
9. Для инициализации локального репозитория открываем Git Bash через контекстное меню в нужной папке. Создаем локальный репозиторий с помощью git init.



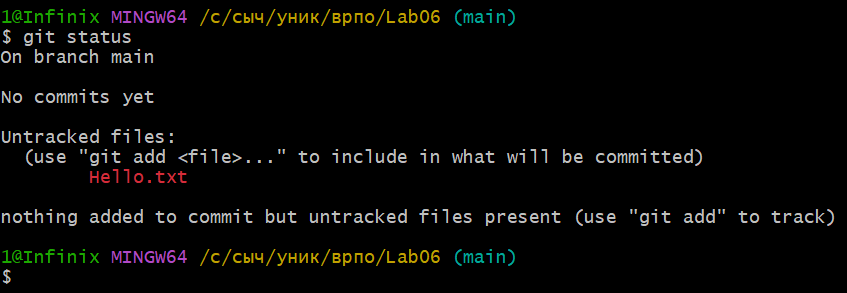
После инициализации в рабочей папке появилась новая скрытая папка .git



10. Создание текстового файла Hello.txt

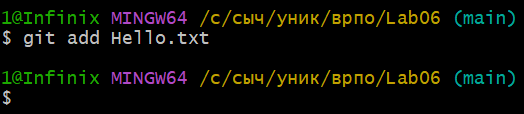


11. Для получения информации о статусе репозитория используем git status.

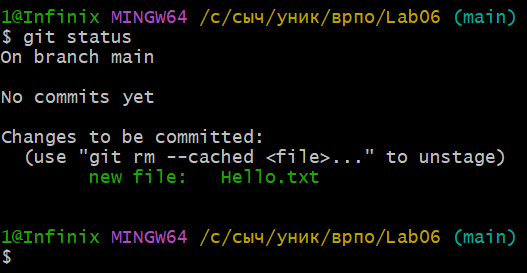


В терминальном окне GitBash отображается информация о том, на какой ветке мы находимся, информация о наличии коммитов и индексированных\неиндексированных файлов в этой папке, а также информация о возможных действиях с неиндексированными файлами.

12. Индексируем файл Hello.txt с помощью git add

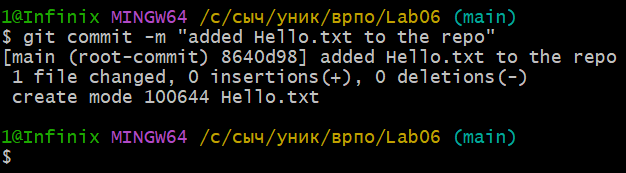


Изменение в репозитории:



Файл Hello.txt теперь проиндексирован и отслеживается репозиторием. Есть возможность закоммитить изменения.

13. Для фиксации изменений индексированных файлов в репозитории используем git commit –m



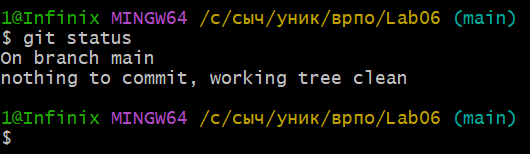
Информирование о создании нового коммита происходит командами:

[main (root-commit) 8640d98] added Hello.txt to the repo

1 file changed, 0 insertions(+), 0 deletions(-)

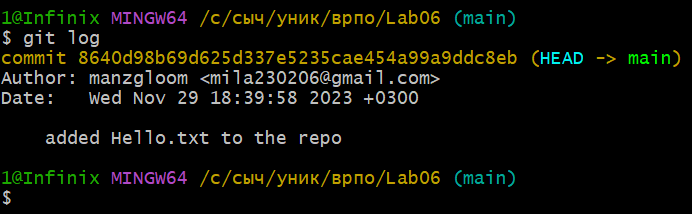
Файл добавлен в ветку main.

Изменения в репозитории:



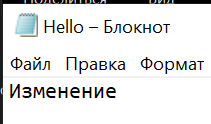
Уведомление о том, что свободных объектов для коммита нет.

14. Просмотрим журнал с помощью git log

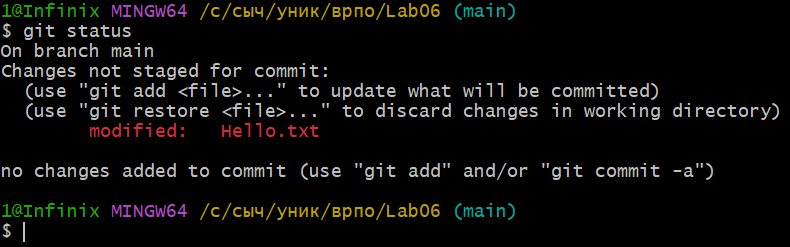


В нем можно просмотреть хэш, изменения, их время и автора, ветку, на которой находимся.

15. Изменили существующий файл Hello.txt

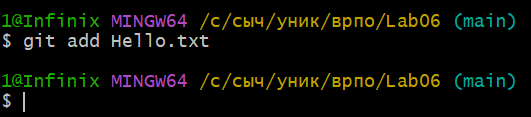


Изменения в репозитории:



Уведомление о том, что файл Hello.txt был изменен и изменения не внесены в коммит. Предложение либо закоммитить изменения, либо удалить их.

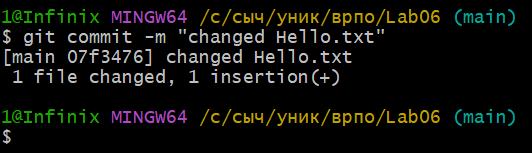
Добавляем в репозиторий новый измененный файл с помощью git add



Изменения в репозитории:



Измененный файл проиндексирован, необходимо закоммитить изменение с помощью git commit –m



Сохранили изменения в репозитории.

16. add – добавление файлов в индексированный репозиторий;

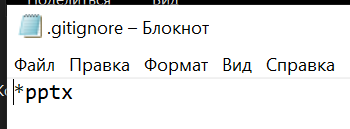
commit – фиксация изменений в репозитории;

status – просмотр текущего состояния репозитория;

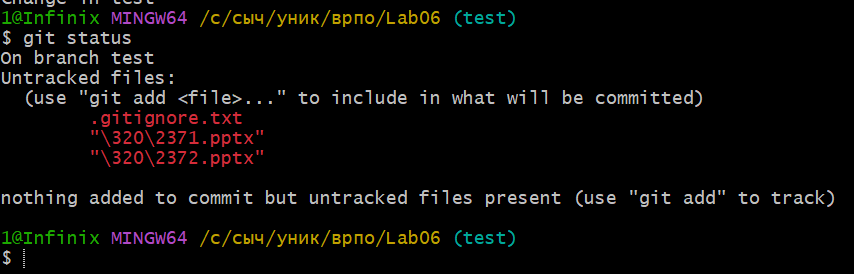
log – просмотр журнала коммитов.

17. Создаем файл .gitignore в рабочей папке и указываем в нем расширения тех файлов, которые будут игнорироваться

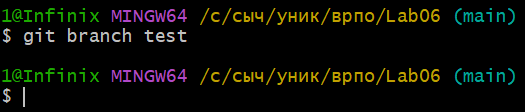


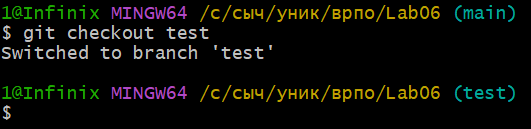


Тогда в состоянии файлов будут указаны игнорируемые файлы



18. Cоздаем ветку test с помощью git branch test

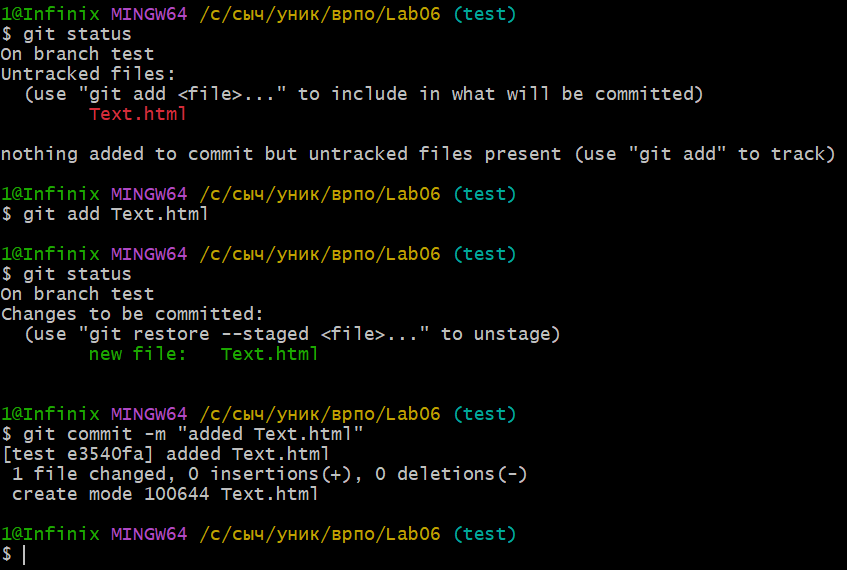


Переключаемся на нее с помощью git checkout test 

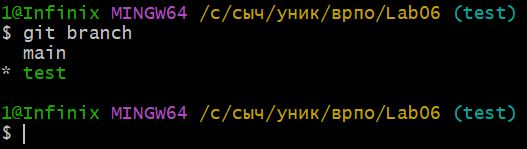
19. Создаем новый текстовый файл test.html



Индексируем файл в репозиторий и создаем коммит

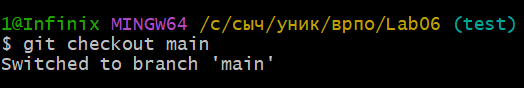


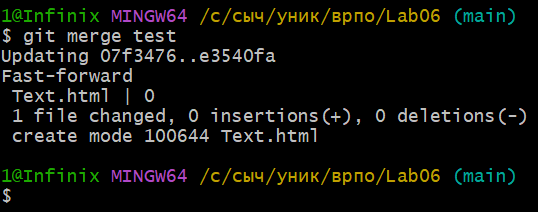
20. Смотрим все ветки с помощью git branch



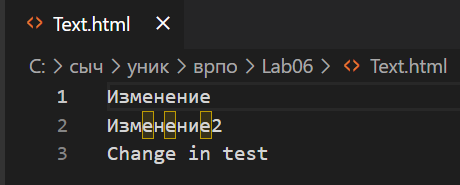
Мы на ветке test.

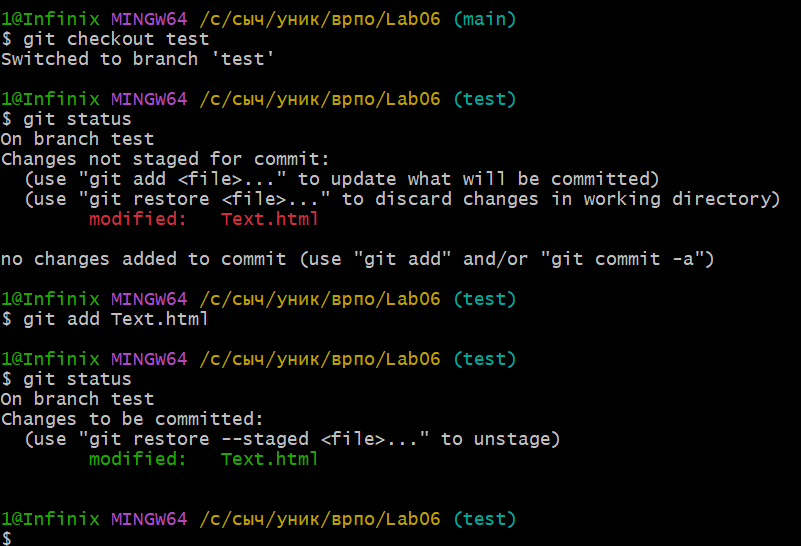
21. Для слияния переходим в ветку main и выполняем слияние с помощью git merge

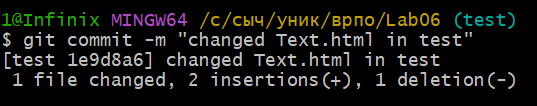




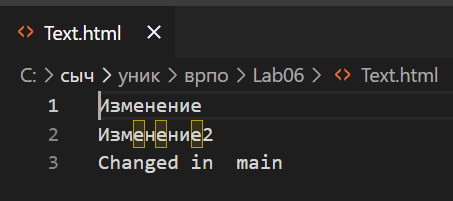
22. Изменяем файл Test.html в ветке test, индексируем и коммитим изменения

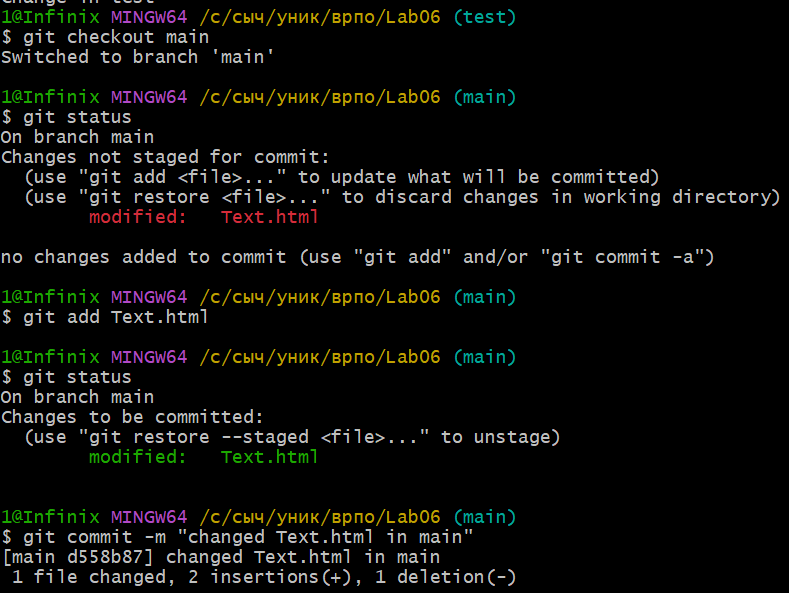




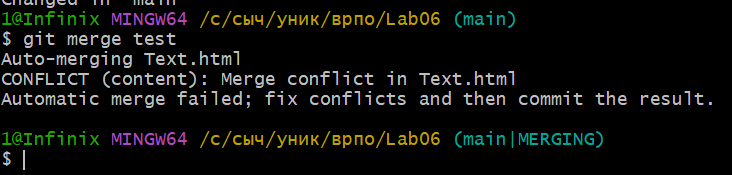


Изменяем файл Test.html в ветке main, индексируем и коммитим изменения



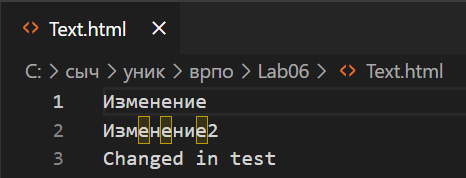


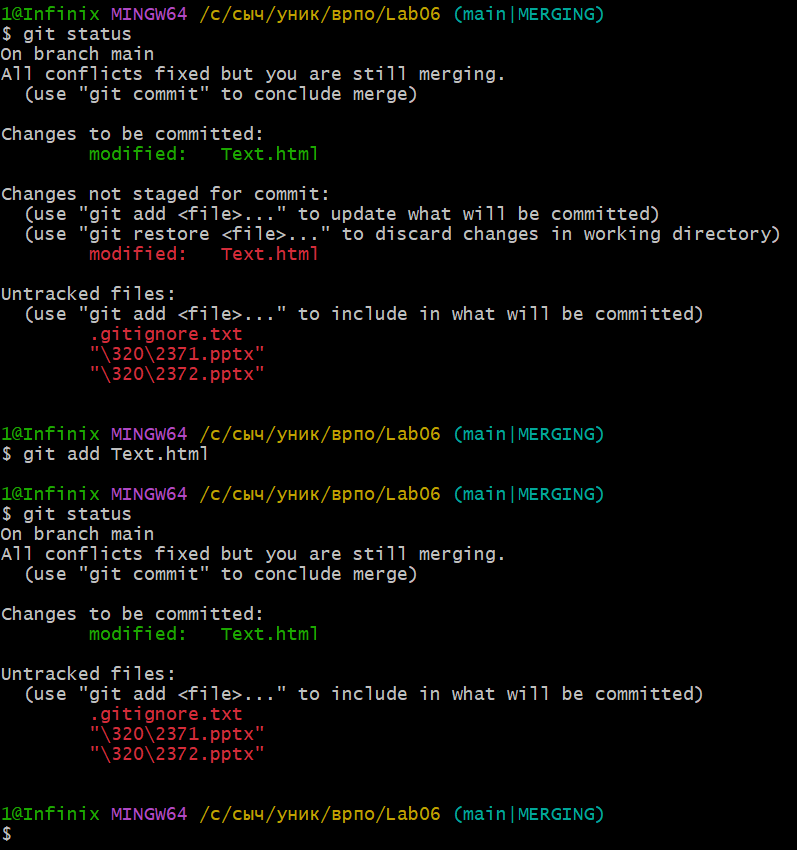
Пытаемся выполнить слияние веток main и test

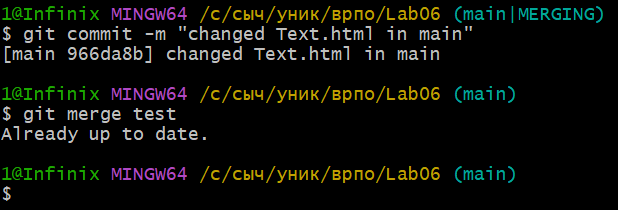


Уведомление о конфликте.

Вручную изменяем ошибку, сохраняем измененный файл и снова выполняем слияние.







Слияние успешно.