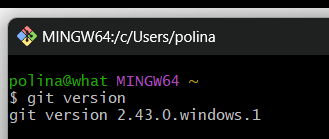
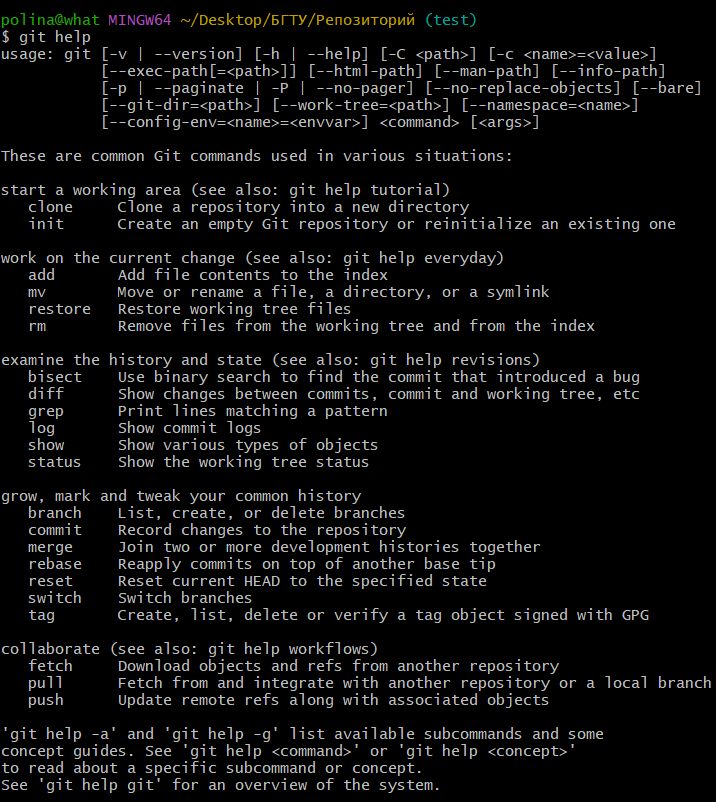
**Лабораторная работа 6**

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



4. Установленная версия

5. Список команд с кратким их описанием можно получить, написав git help



6. Для создания репозитория нужно вписать git init

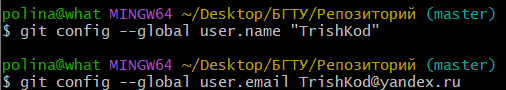
По умолчанию репозиторий создастся по пути C:/Users/user/.git



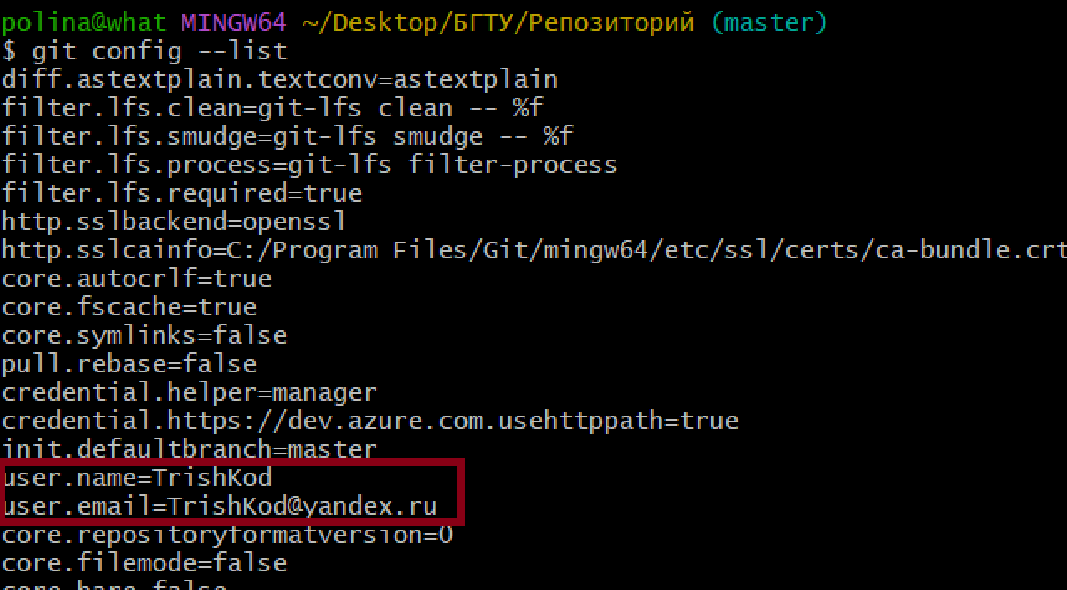
Чтобы удалить репозиторий можно использовать команду rm -rf .git



7. Выполнение конфигурации выполняется следующими командами

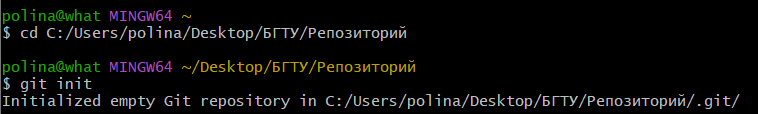


8. Проверка введенных данных проходит через команду git config –list

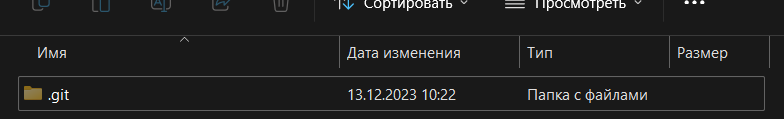


9. Для инициализации локального репозиторий в рабочей папки потребуется вписать cd /путь до рабочей папки/, в моем случае команда будет выглядеть так: C:\Users\polina\Desktop\БГТУ\Репозиторий

После прописать git init

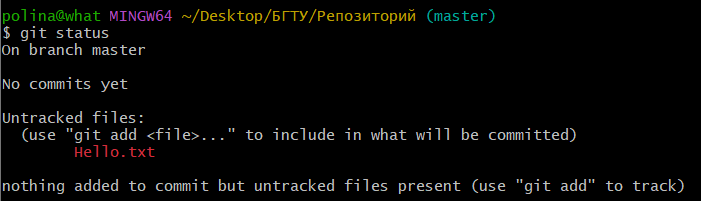


В папке создастся скрытая папка .git



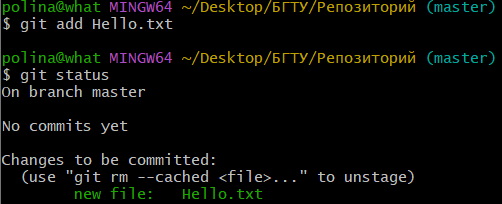
10. Создаем текстовый файл в папке, в которой лежит папка .git

11. Пропишем git status, чтобы получить информацию о состоянии репозитория (git status)

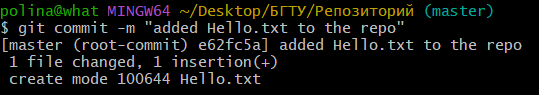


Создалась ветка master. Красным подписаны новые файлы.

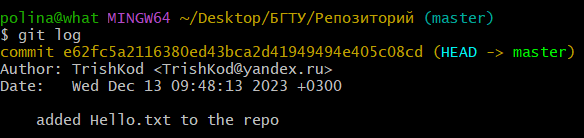
12-13. Для добавления файла в репозиторий прописываем команду git add /Имя файла/



Наш текстовый файл стал зеленым, значит, он был проиндексирован. Можем добавить комментарий через команду git commit -m «Текст комментария»

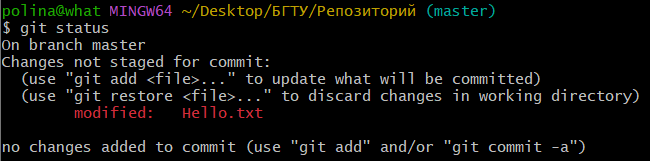


14. Через команду git log получаем информацию



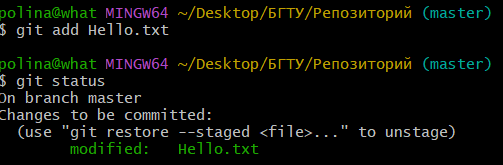
Здесь мы можем увидеть информацию о проиндексированных и не индексированных файлах, свой никнейм и почту, дату, комментарий.

15-16. Изменим текст в файле Hello.txt, чтобы узнать, как программа среагирует на их изменение. Прописываем git status

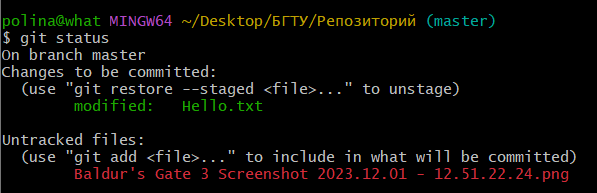


Файл опять стал красным и появилась приписка «модифицирован»

Добавляем файл через git add

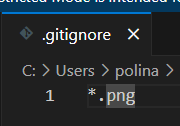


17. Добавим какой-то файл в папку. В моем случае картинку. Проверяем статус

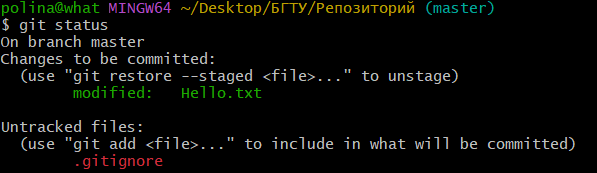


Чтобы добавить файл в игнорируемое создаем текстовый файл с расширением .gitignore

В нем ставим \* и пишем нужное нам расширение.

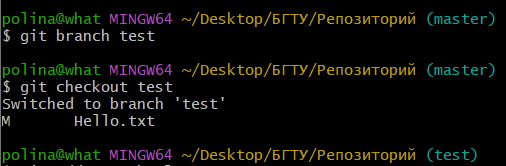


Проверяем статус

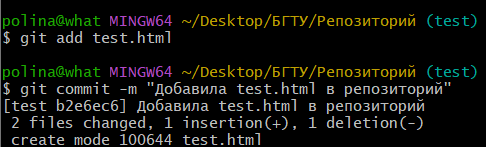


18. Для создания новой ветки используем branch, а для перехода между ветками checkout

Прописываем git branch test и после переходим в созданную ветку

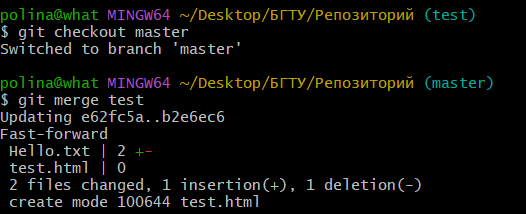


19. Создаем файл test.html, добавляем его и даем комментарий

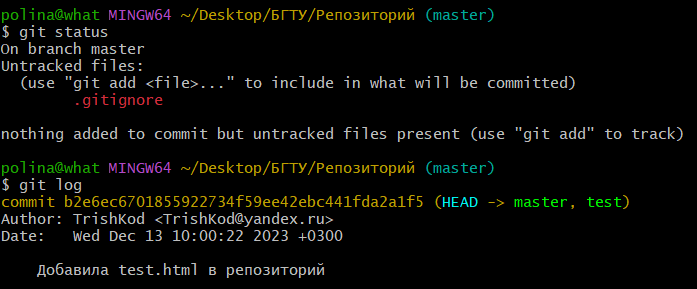


20. Проверяется ветка по бирюзовому тексту в скобочках после пути репозитория. У меня их две: master и test. На скриншотах видно, что был произведен переход из одной ветки в другую.

21. Выполним слияние веток. Переходим в ветку master и после производим слияние с помощью команды git merge /название другой ветки/

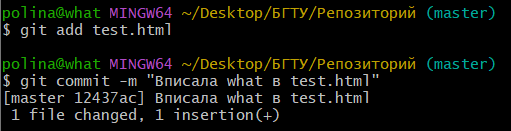


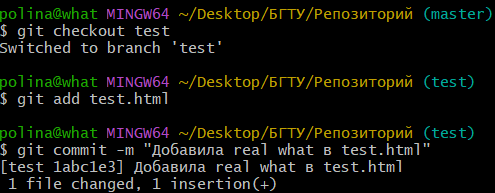
Просмотрим статус и лог:

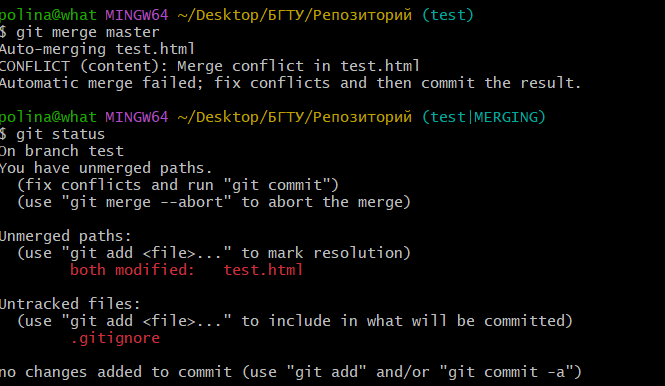


22. Переключаясь из одной ветки в другую, изменим содержание файла в test.html и посмотрим на результат. После изменения добавляем файл и комментарий

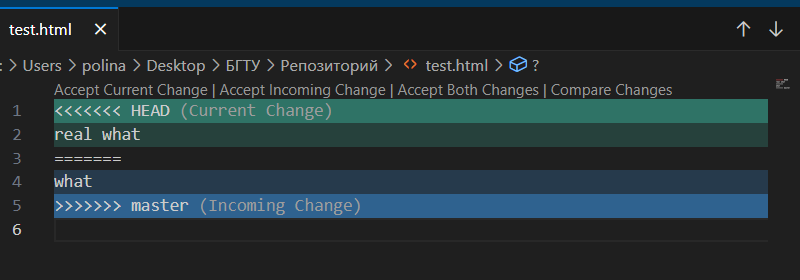
Производим слияние и получаем ошибку







При слиянии произошел конфликт. При проверке статуса показано, что файл был изменен в обоих ветках. В самом файле ошибка выглядит следующим способом:



Для решения необходимо просто оставить интересующий нас вариант и добавить его в репозиторий:

