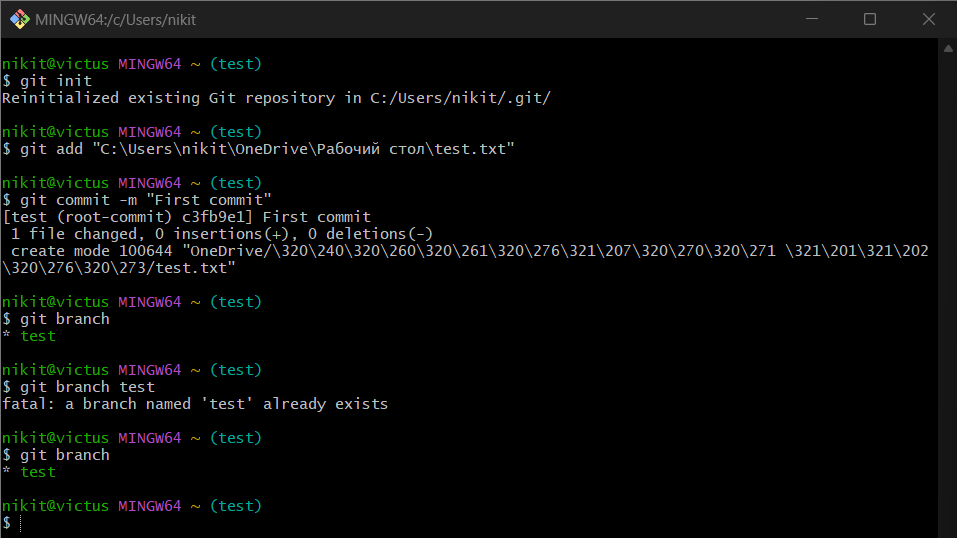
**3)** *Создайте (branch) новую ветку, например, test и переключитесь (checkout) в нее.*



**4)** *Куда указывает HEAD?*

Указывает, на какой ветке мы находимся

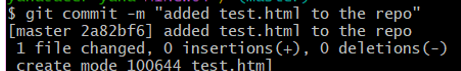
**5)**  *Просмотрите содержимое всех файлов в папке проекта, с которыми вы работали.*

**

**

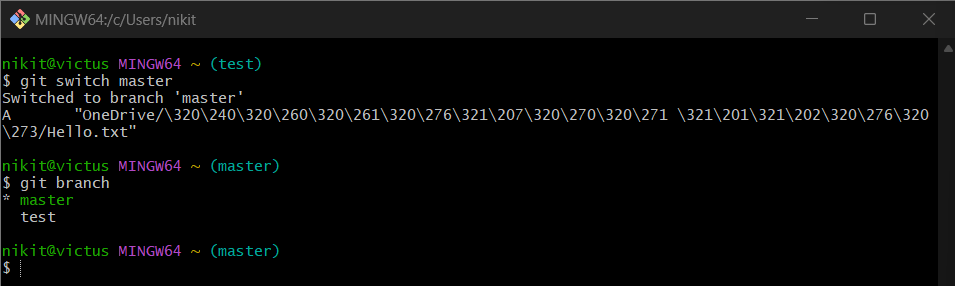
**6)** *Создайте новый текстовый файл test.html и зафиксируйте его (commit).*



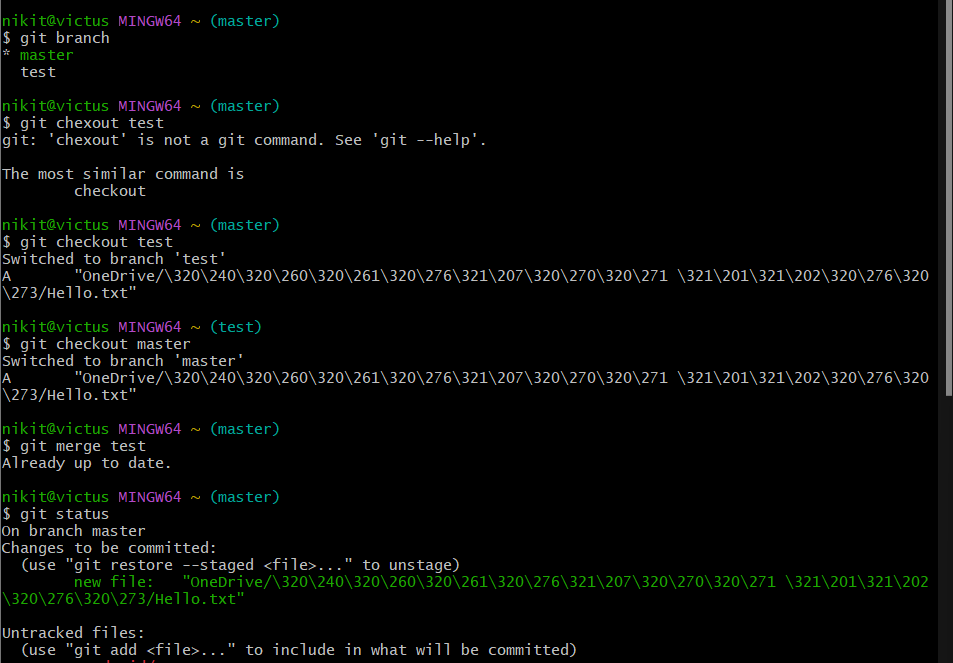


**7)** *Просмотрите список всех веток и определите текущую ветку.*

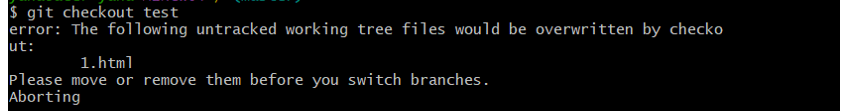
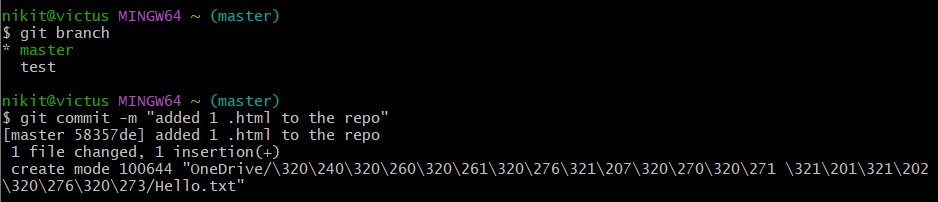
Текущая ветка – master

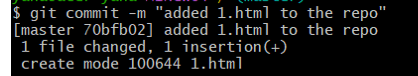


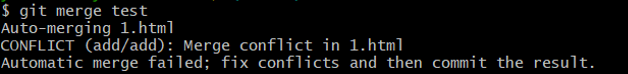
**8)** *Выполните слияние (merge) веток.*



**9)** *По очереди отредактируйте файл test.html в ветках master и test. Выполните commit. Выполните слияние в одной и второй ветках. Разрешите ситуацию конфликта слияний.*







**Вопросы.**

**1. Как используется команда git log?**

Позволяет просмотреть историю репозитория

**2. Что такое ветвление?**

отклонение от основной линии разработки, после которого работа перестает затрагивать основную линию и переходит в ветвь.

**3. Как используется команда git checkout?**

Переключение веток

**4. Что такое указатель HEAD?**

указывает на текущую ветку

**5. Продемонстрируйте создание новых веток в Git?**

$ git branch «branch\_name»

**6. Какая команда осуществляет переключение веток?**

$ git checkout «branch\_name»

**7. Для чего используется команда merge?**

Слияния веток

**8. Какая команда позволяет отобразить историю репозитория?**

$ git log

**9. Как выполнить слияние изменений из одной ветки в другую?**

$ git merge <сливаемая ветка>

**10.Какие проблемы могут быть при слиянии и как они разрешаются**

различия текста в файлах веток (несовпадающие места в файле для двух веток). Конфликт разрешается разработчиком вручную.