**Ход решения**

**Часть 1. Основные команды Git.**

Задание 1. Установите и настройте клиент git на своей рабочей станции.

После установки git на рабочее устройство нам потребуется его настроить для этого добавим имя и почту для добавления имени используем команду: *git config --global user.name “Shilo Yuriy”*, а для изменения почты *git config --global user.email “shilo.y.s@edu.mirea.ru”*, а для проверки наших изменений введем *git config --list*. Выполнение данных команд предоставлено на рисунке 1.

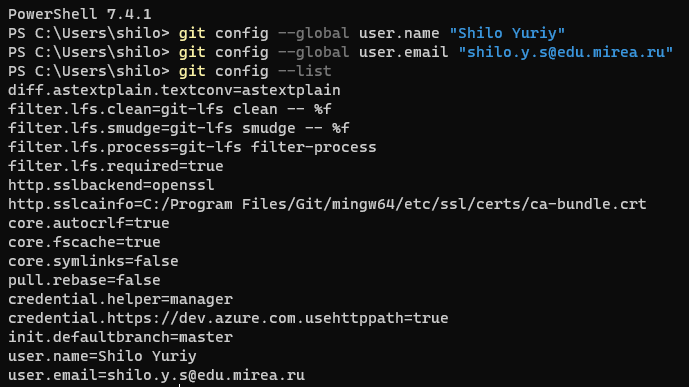


Рисунок 1 – Изменение глобальных настроек информации о пользователе

Задание 2. Создайте локальный репозиторий и добавьте в него несколько файлов

Для создание локальной репозитории мы должны ввести команду *git init*, находясь в папке, которую мы хотим сделать нашим репозиторием. После ввода данной команды консоль вывела следующее рисунок 2.



Рисунок 2 – Создание локального репозитория

Задание 3. Внесите изменения в один из файлов.

В пустой текстовый файл была написана фраза “Какой-то текст, который был добавлен. После создания репозитория.” рисунок 3.

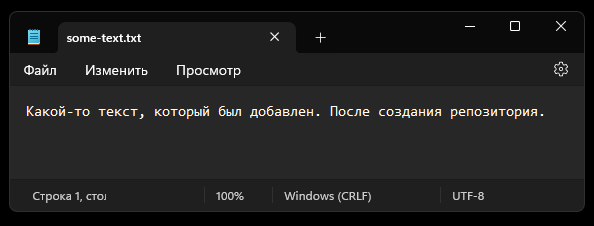


Рисунок 3 – Содержание текстового файла

Задание 4. Проиндексируйте изменения и проверьте состояние.

Для того что бы добавить изменный индекс в Git мы должны воспользоваться командой *git add .*, а чтобы проверить состояние репозитории мы должны использовать команду *git status*. После выполнения данных команд консоль выведет следующие рисунок 4.

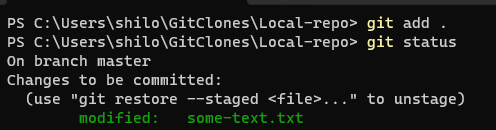


Рисунок 4 – Добавление индекса в Git и получение статуса локального репозитория

Задание 5. Сделайте коммит того, что было проиндексировано в репозиторий. Добавьте к коммиту комментарий.

Для того чтобы за коммитить изменения мы воспользуемся командой *git commit -m “…”* (за место трех точек мы пишем наш комментарий) выполнение данной команды показано на рисунке 5.

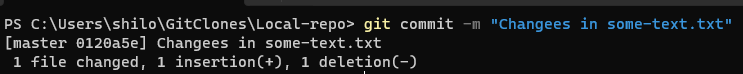


Рисунок 4 – Выполнение команды для коммита с комментарием

Задание 6. Измените еще один файл. Добавьте это изменение в индекс git. Измените файл еще раз. Проверьте состояние и произведите коммит проиндексированного изменения. Теперь добавьте второе изменение в индекс, а затем проверьте состояние с помощью команды git status. Сделайте коммит второго изменения.

Для выполнения данного задания поочередно введем команды, которые мы разбирали в предыдущих заданиях. Все введённые команды и их результат предоставлены на рисунке 6.

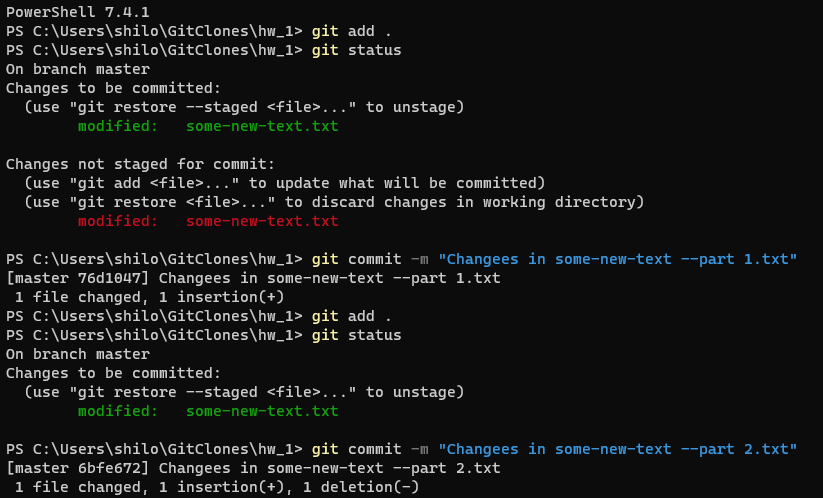


Рисунок 6 – Все операции произведенные в задание 6

Задание 7. Просмотрите историю коммитов с помощью команды *git log*. Ознакомьтесь с параметрами команды и используйте некоторые из них для различного формата отображения истории коммитов.

На рисунке 7 предоставлены три различных формата вывода коммитов: *git log, git log --inline, git log --graph*. Первый это вывод без форматирования, второй выводит каждый коммит в виде одной строчки и третий выводит список в виде графов.

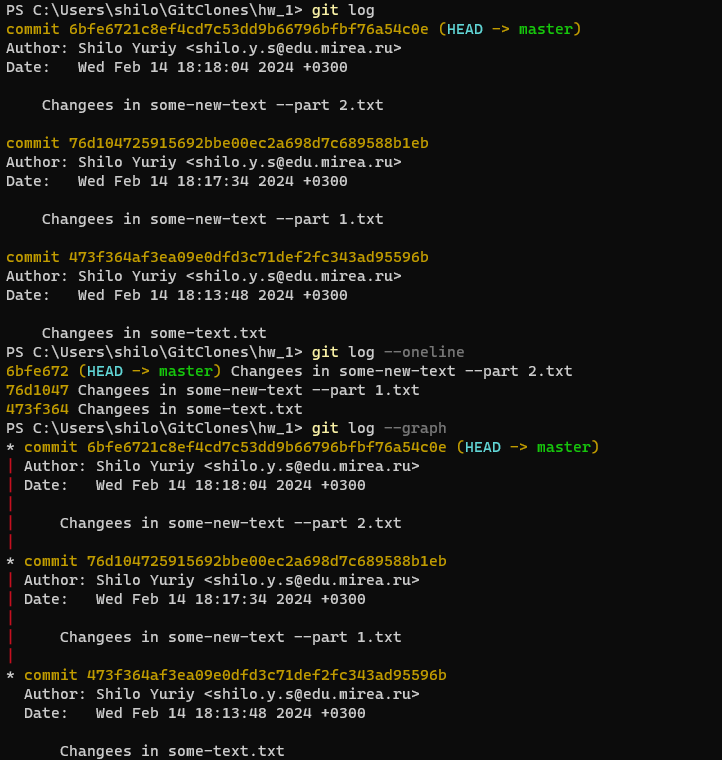


Рисунок 7 – Три варианта форматирования вывода коммитов

Задание 8. Верните рабочий каталог к одному из предыдущих состояний.

Для возврата к предыдущему коммиту мы воспользуемся командой *git checkout HEAD~*, также за место *HEAD~* мы можем написать хэш код коммита, имя ветки или имя ветки. Показ работы команды по возврату на один коммит назад предоставлена на рисунке 8.

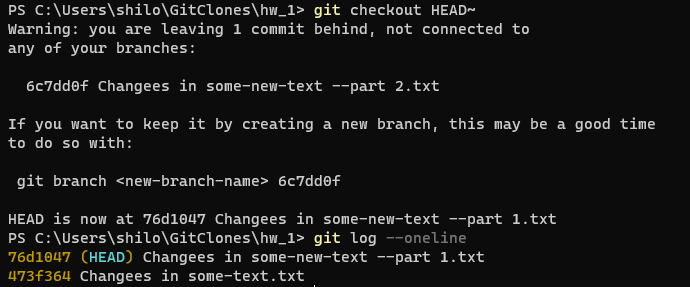


Рисунок 8 – Действия по возврату к предыдущему коммиту

Задание 9. Изучите, как создавать теги для коммитов для использования в будущем.

Для создания простого тега используется команда *git tag …*, за место троеточия нужно написать имя нашего тега. А для просмотра списка существующих тегов используется команда *git tag*. После добавления тега он будет отображаться в истории коммитов для удобного возврата. Отработка данных команд предоставлена на рисунке 9.

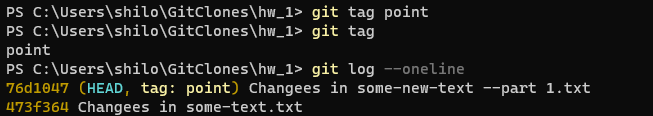


Рисунок 9 – Создание тега point

Задание 10. Отмените некоторые изменения в рабочем каталоге (до и после индексирования).

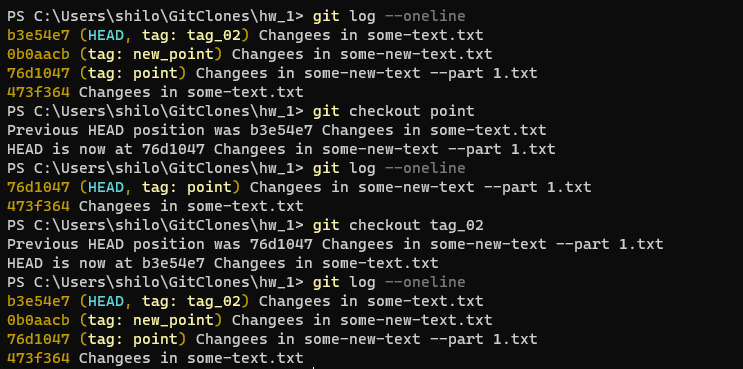


Рисунок 10 – Отмена изменений в рабочем каталоге