Лабораторная работа 3.

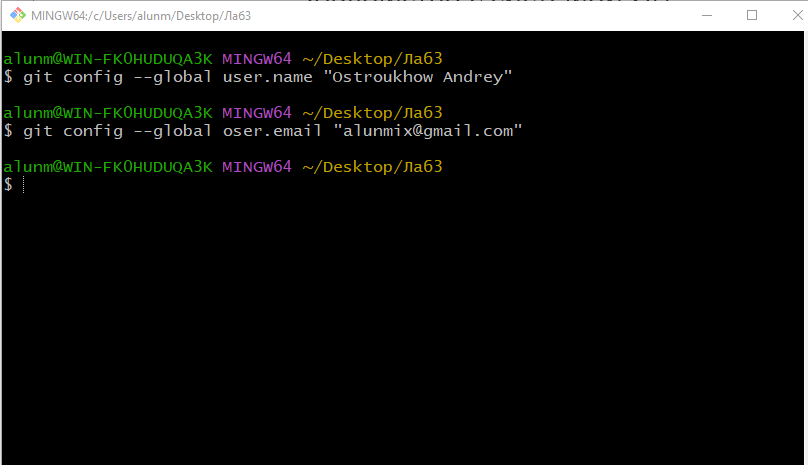
Знакомство с системой GIT.

6 Вариант

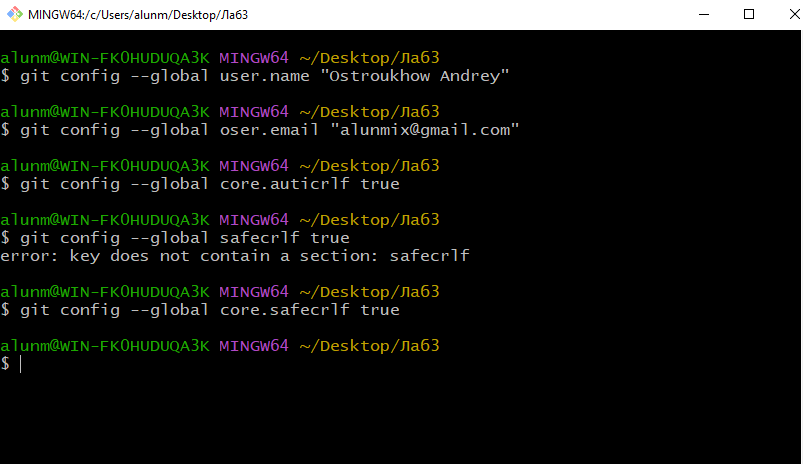
https://github.com/lunmix/lab3

Использовать в работе 3 разных файла, и количество версий каждого файла не должно быть меньше 3.

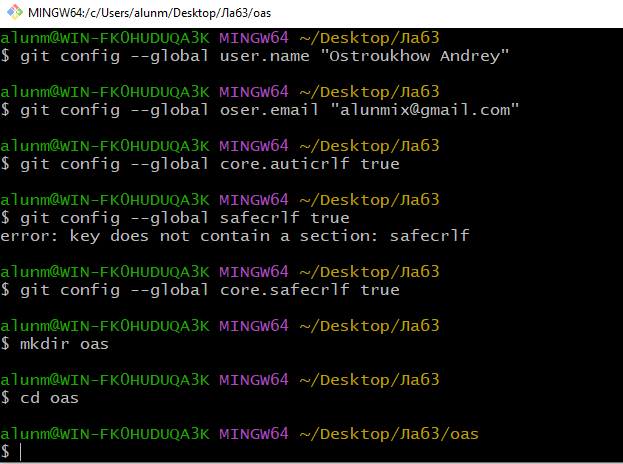
1. Установил имя и адрес электронной почты.



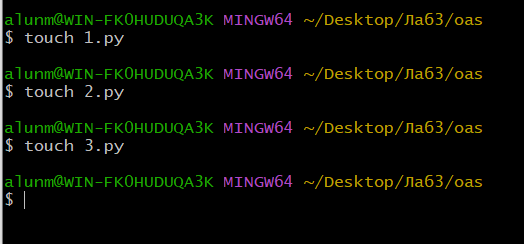
2.Установил параметры окончания строк.



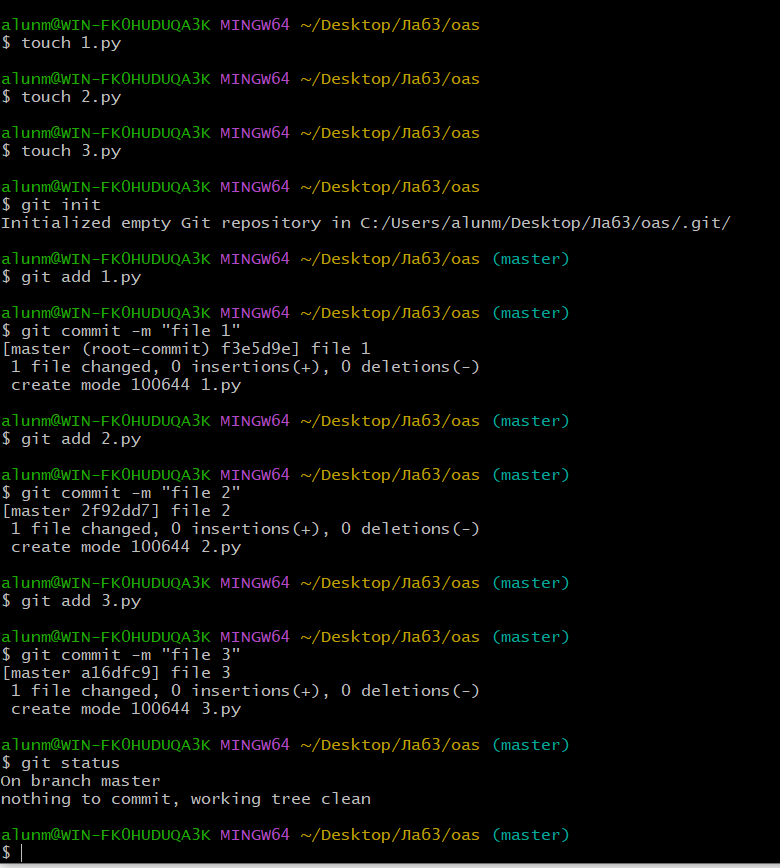
3.Создал каталог с именем, определяемый инициалами и выбрал его в качестве рабочего:



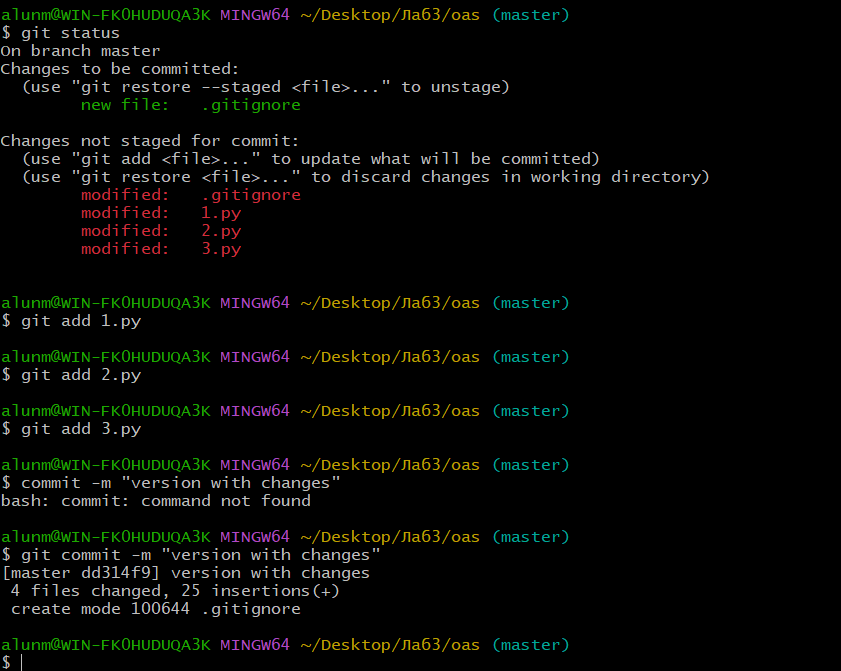
4.В рабочем каталоге создали три файла, согласно варианту.



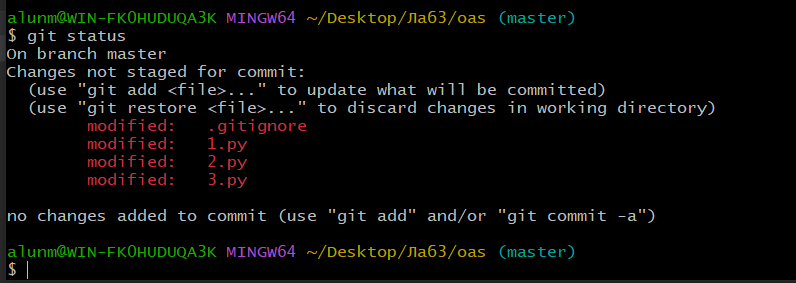
5.Создал репозиторий и добавил файлы в репозиторий. Добавил коммиты. Проверил состояние репозитория.



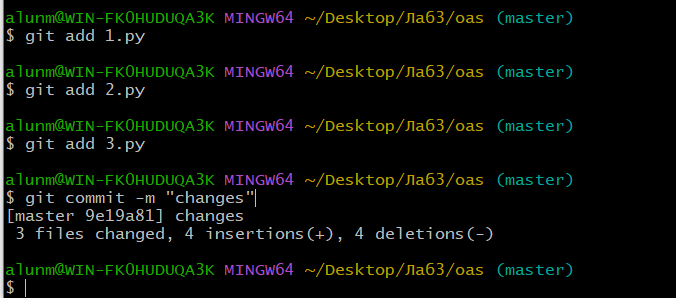
6. Проверил состояние репозитория после изменений в трех файлах. Проиндексировали изменения и добавили коммит.



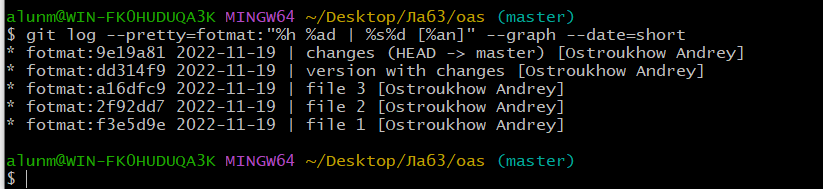
7. Внесли еще изменения и проверили состояние репозитория.



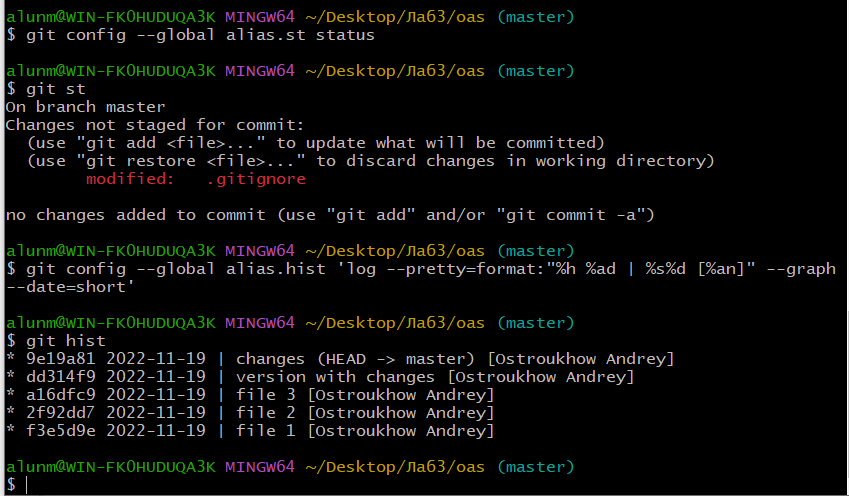
8. Проиндексируем изменения и добавляем коммит.



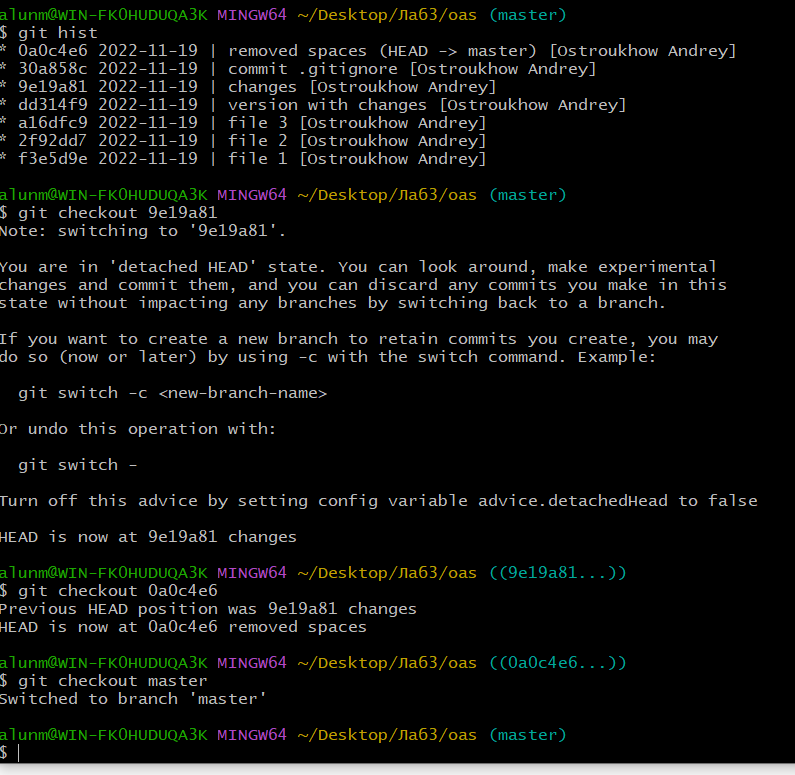
9.Выводим историю версий в форматированном виде.



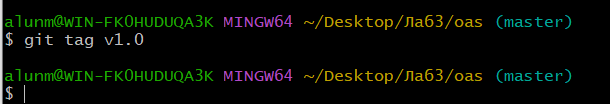
10. Создаем алиасы для удобства.



11. Вернулись к прошлой версии программы и к последней.

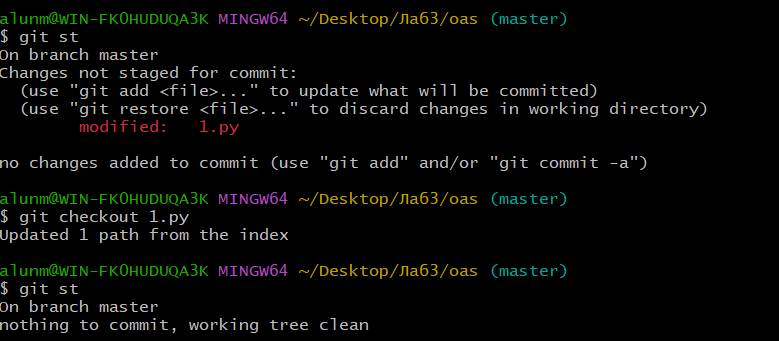


12. Добавили тегирование.

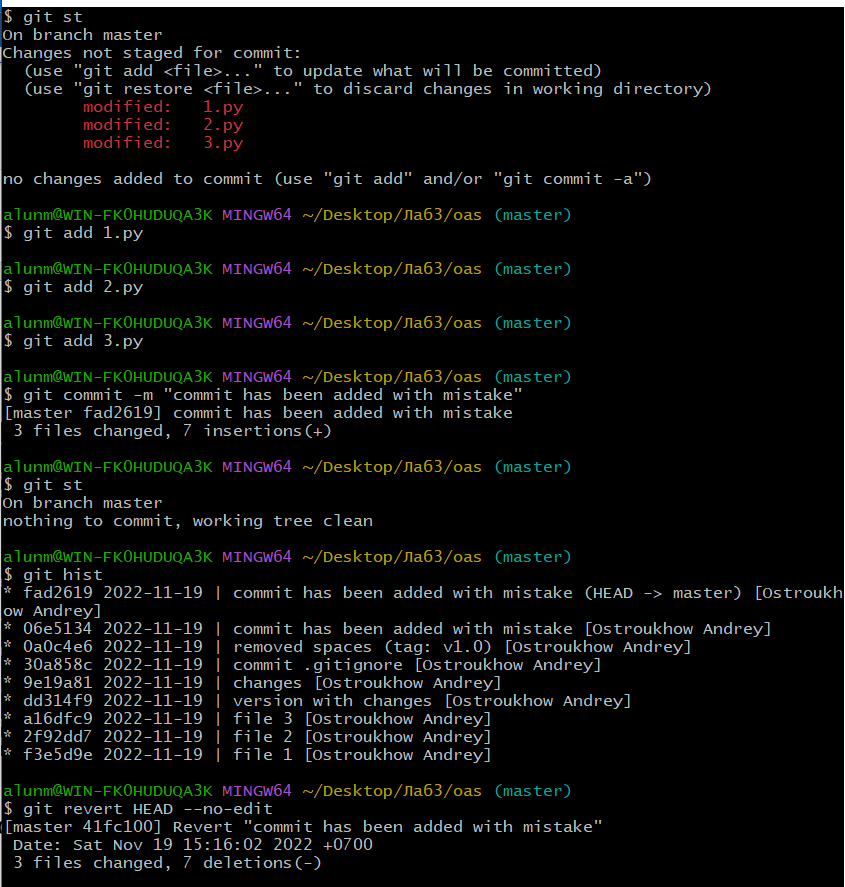


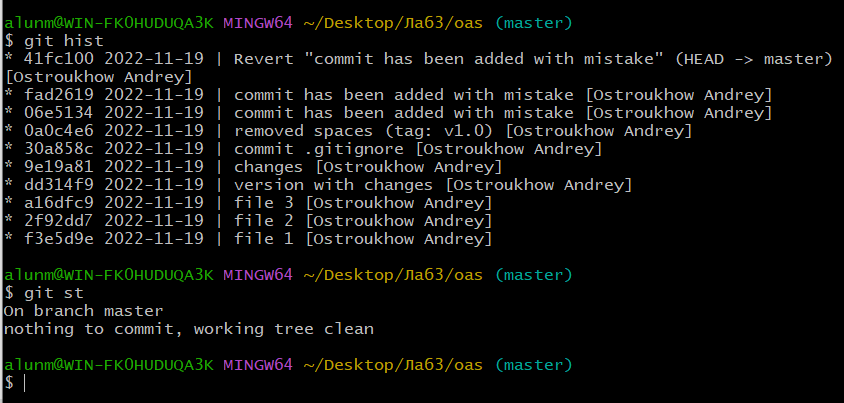
13. Отмена изменений.

А. К последней проиндексированной версии

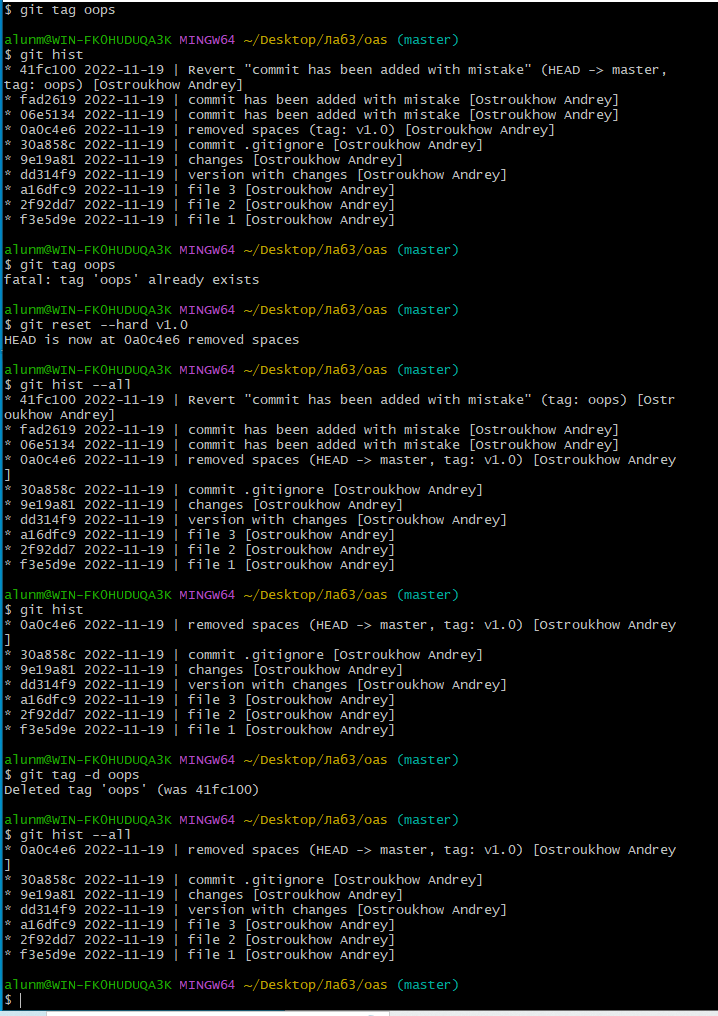


Б. Если файл с ошибкой был проиндексирован, но не добавлен коммит.

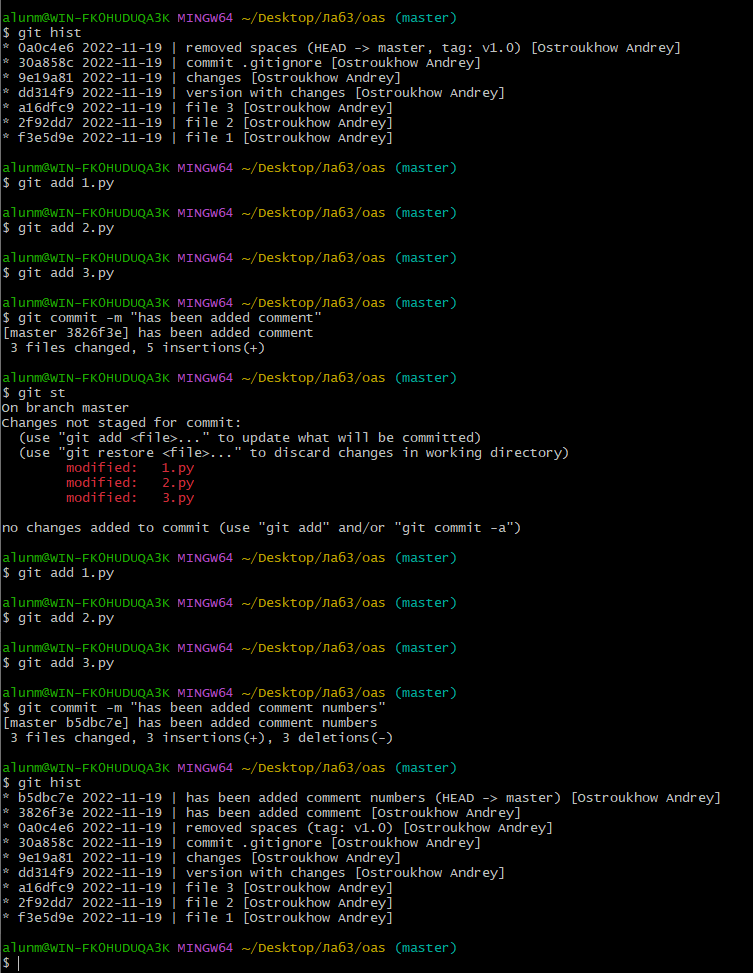




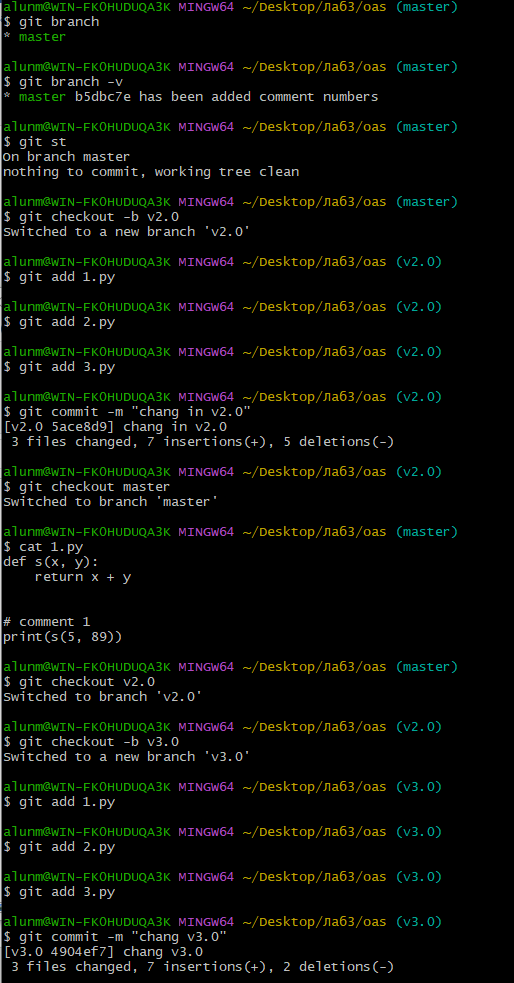
Отменили коммит и выполнили сброс к предыдущей версии.

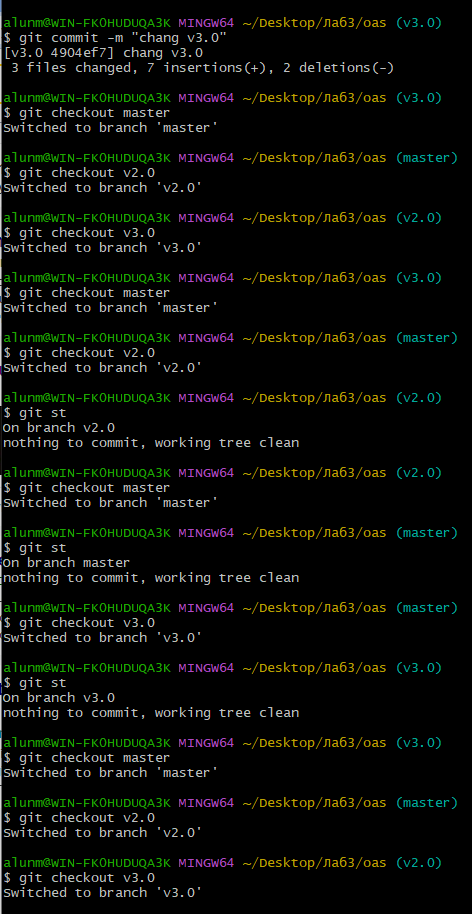


В. Исправил коммит без его отмены и удаления.

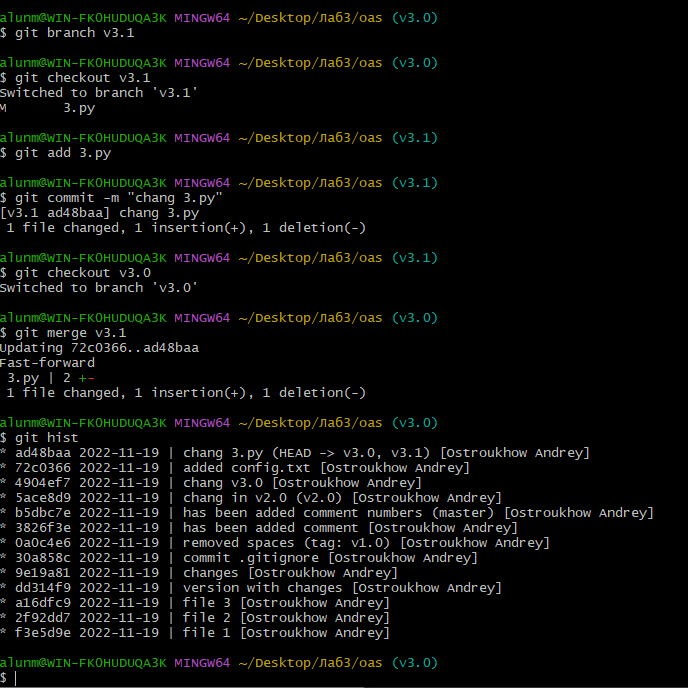


14. Создаем три ветки и создаем три версии файлов согласно варианту.

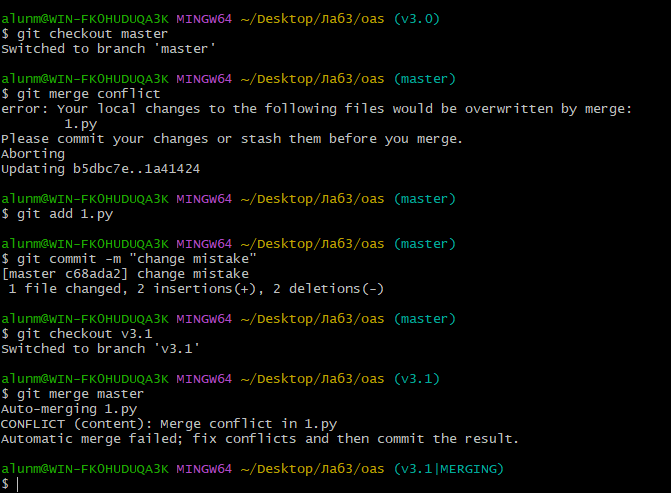


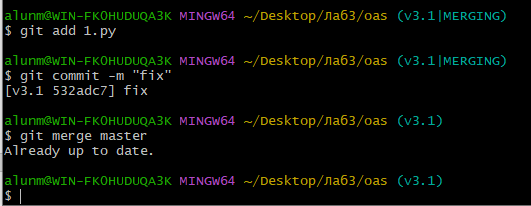


15. Произвели слияние веток v3.0 и v3.1.

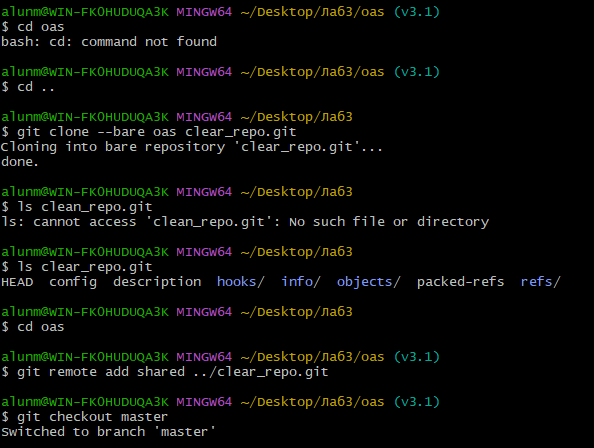


16. Создание и разрешение конфликта





17. Создание чистого репозитория.



**Контрольные вопросы**

1. Система GIT представляет систему контроля версий (Version Control System, VCS) позволяющая отслеживать изменения в документах, при необходимости производить их откат, определять, кто и когда внес исправления

2. Система контроля версий позволяет создавать несколько версий одного документа, отслеживать изменения в нем, возвращаться к ранним версиям, вести параллельную разработку изменений.

1. Это команда Git для записи индексированных изменений в репозиторий. Вызывается он командой (git commit” Текст сделанного изменения”).

4.Историю коммитов можно посмотреть с помощью команды Git hist.

5. Зайти в папку нашего будущего проекта, нажать правую кнопку мыши и нажать Git Bash Here, затем ввести команду в консоль git init, у нас создастся пустой репозиторий, который мы будем заполнять нашим проектом.

6. Какую архитектуру имеет система контроля версий GIT?

-Хранение данных

-Отслеживание изменений содержимого (хранение истории изменений, включающей метаданные слияния)

-Распространение данных и истории изменений между разработчиками

7. Как создать новую ветку в GIT и соединить ветки?

Новая ветка создаётся командой Git checkout –b <название ветки> -создание Git merge <название ветки> -слияние

8. Поясните понятие «staging area».

Файл, хранящийся в каталоге GIT, который содержит информацию о том, что должно войти в следующий коммит

9. Как сравнить версии файла в GIT? Командой Git diff --cached