Заходим в аккаунт GitHub. Создаем новый публичный репозиторий "laba1" (заранее меняем название главной ветки на "master"). Открываем командную строку через Win+R, используя команду cmd. Создаем папку и текствовый файл test.txt с именем в ней. Переходим в созданный каталог в командной строке при помощи команды cd. Дальше в командной строке настраиваем имя пользователя и адрес электронной почты

git config --global user.name "Mikhael"

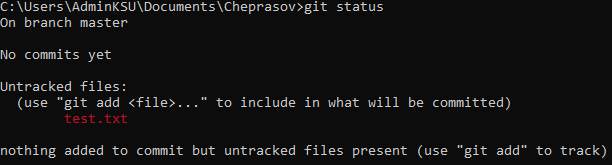
git config --global user.email "cheprasov-03@mail.ru"

Инициализируем созданную папку с помощью команды git init

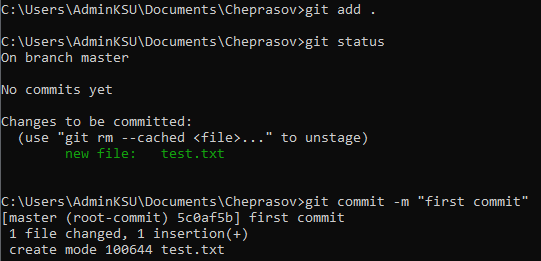


Командная строка возвращает сообщение, что проинициализирована пустая git директория.

Вводим команду git status

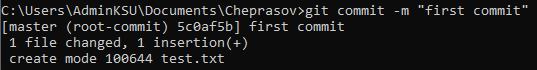


Отобразилось сообщение, что есть новый файл, но он пока не отслеживается. Для этого используем команду git add . , добавляющую все файлы из папки( у нас он 1).



Файл готов к коммиту. Сделаем наш первый коммит, чтобы зафиксировать изменения командой

git commit -m "first commit"



Команда создала новый коммит со всеми изменениями из области подготовки. Вызовим git status и посмотрим на изменения



Чтобы связать наш локальный репозиторий с репозиторием на GitHub, выполним следующую команду

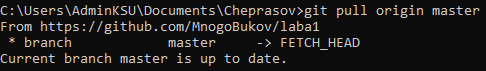
git remote add origin <https://github.com/MnogoBukov/laba1>

Вызовем команду git remote -v для просмотра подлюченного проекта



При создании репозитория на GitHub мы указали создать в нем файл README. Можно скачать его в наш локальный репозиторий камандой

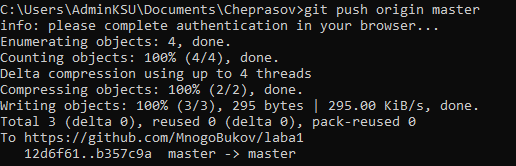
git pull origin master





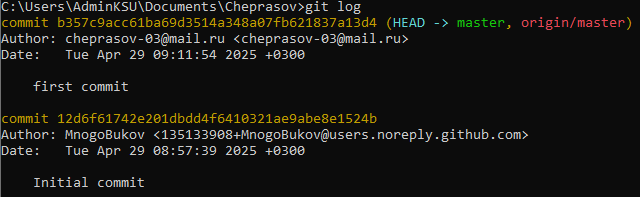
Теперь отправим наш коммит на сервер командой

git push origin master

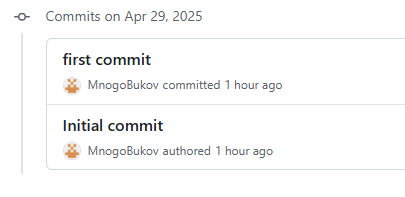


Посмотрим наши изменения с помощью команды

git log



Изменения на GitHub



Создадим новую ветку second

git branch second

Можно посмотреть в какой ветке нахождимся командой

git branch

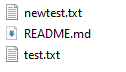


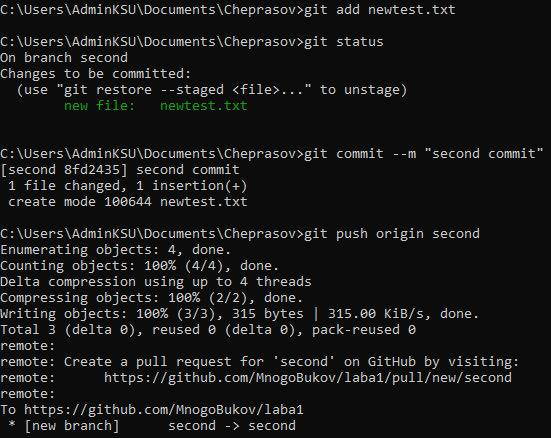
Чтобы переключиться на другую ветку, используем команду

git checkout second

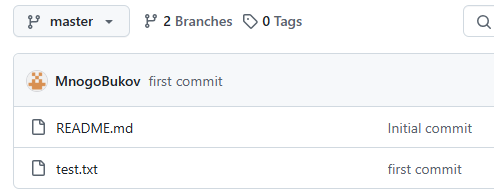


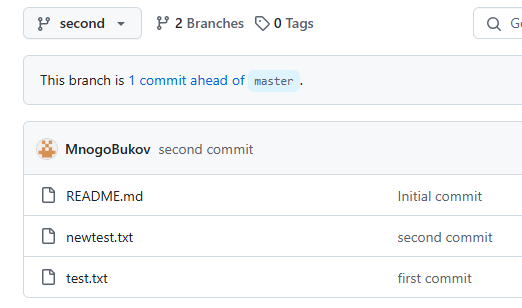
Создадим новый файл newtest.txt и добавим его на сервер, используя известные команды



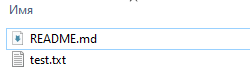


Проверяем результат на GitHub



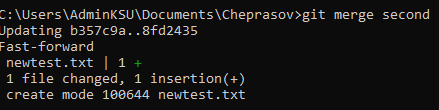


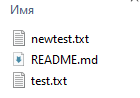
Перключимся на ветку master. Фиксируем изменения в проводнике Windows



Файла newtest.txt не видно, т.к. он в ветке second. Для его отображения используем команду для объединения веток

git merge second



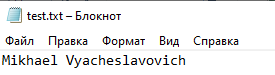


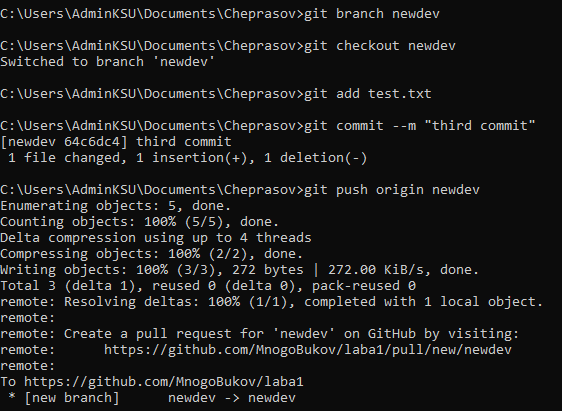
Теперь можно удалить ветку second командой

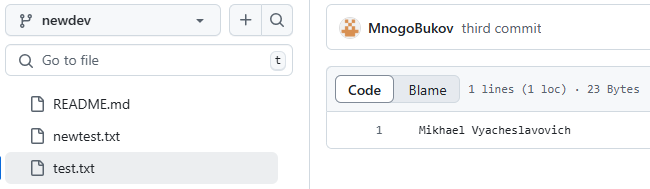
git branch -d second



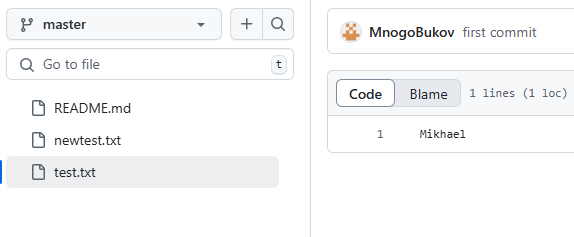
Создадим ветку newdev, переключимся на нее, добавим в файл с именем отчество,зафиксируем изменения и закоммитим их.





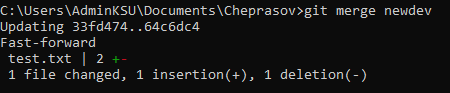


В ветке master осталось только имя

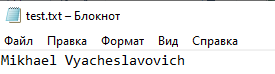


Объединим ветки

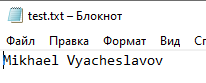
git merge newdev

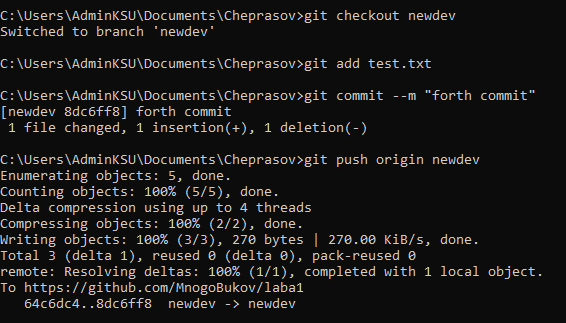


Файл test.txt в ветке master

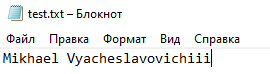


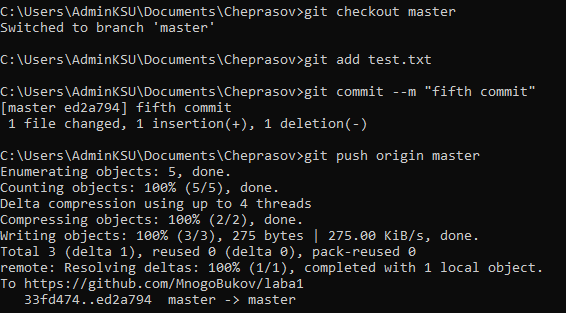
Теперь в ветке newdev удалим в отчестве несколько букв, зафиксируем и закоммитим



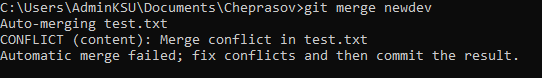


В ветке master добавим в отчестве несколько букв, зафиксируем и закоммитим



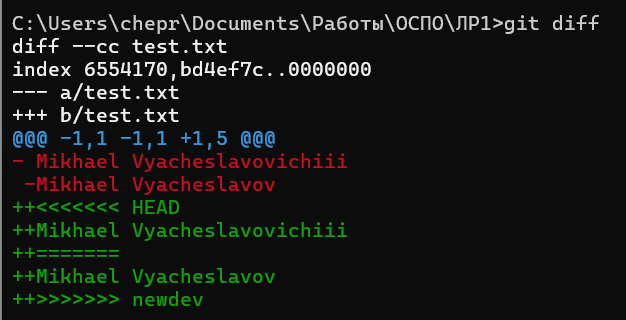


Объединим ветки

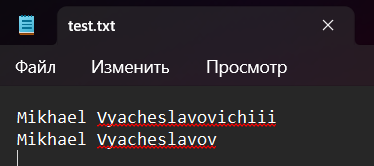


Нужно разрешить конфликт, для этого наберем команду, для просмотра изменений

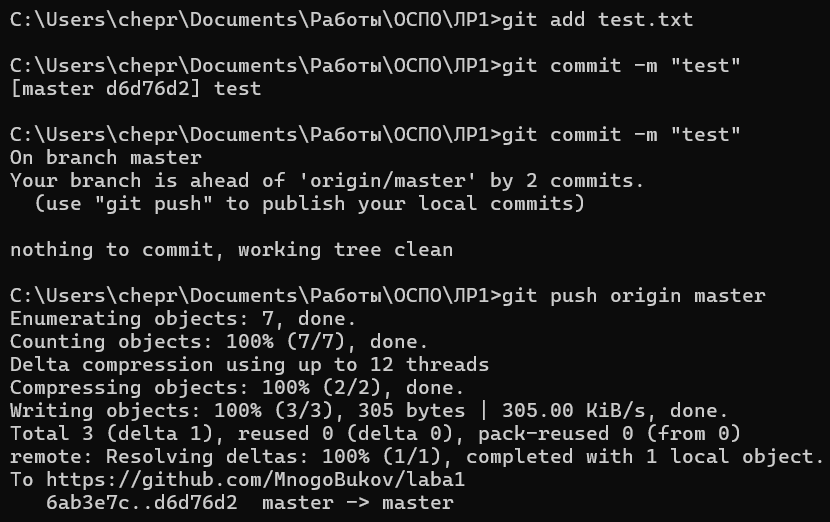
git diff

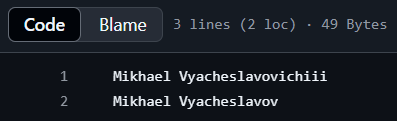


Разрешаем конфликт вручную, для этого в файле test.txt убираем разделители (<<<<<<<, =======,>>>>>>>):

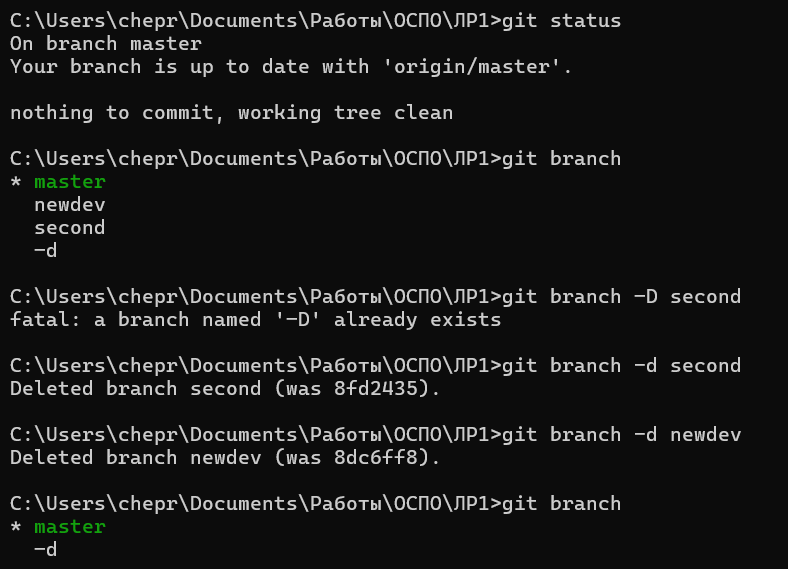


Коммитим и отправляем на сервер конечный вариант

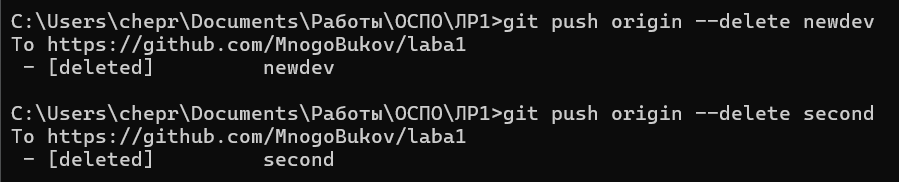


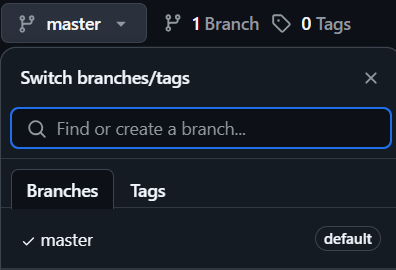


Теперь удалим ветки



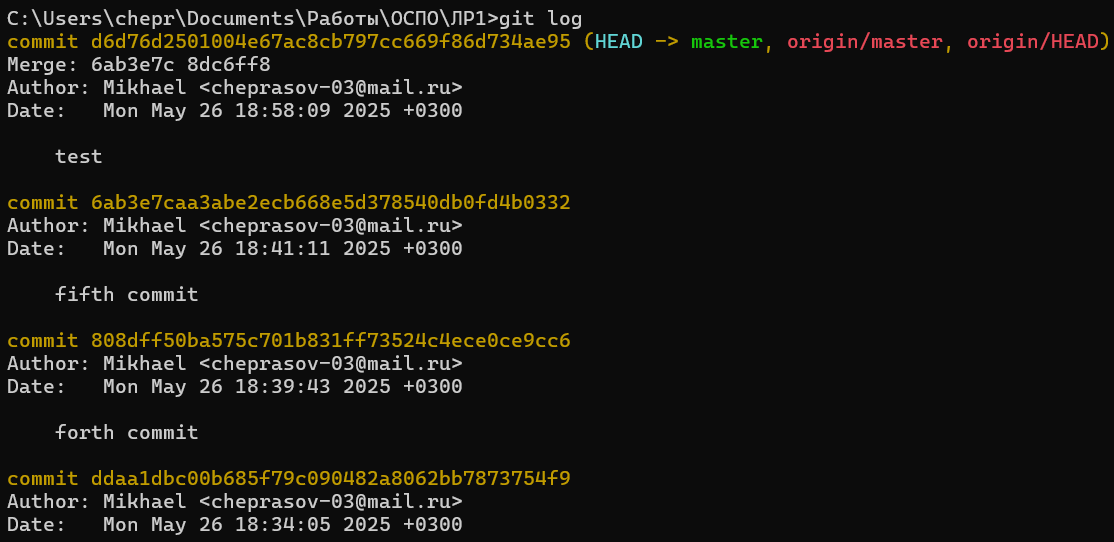
На сервере

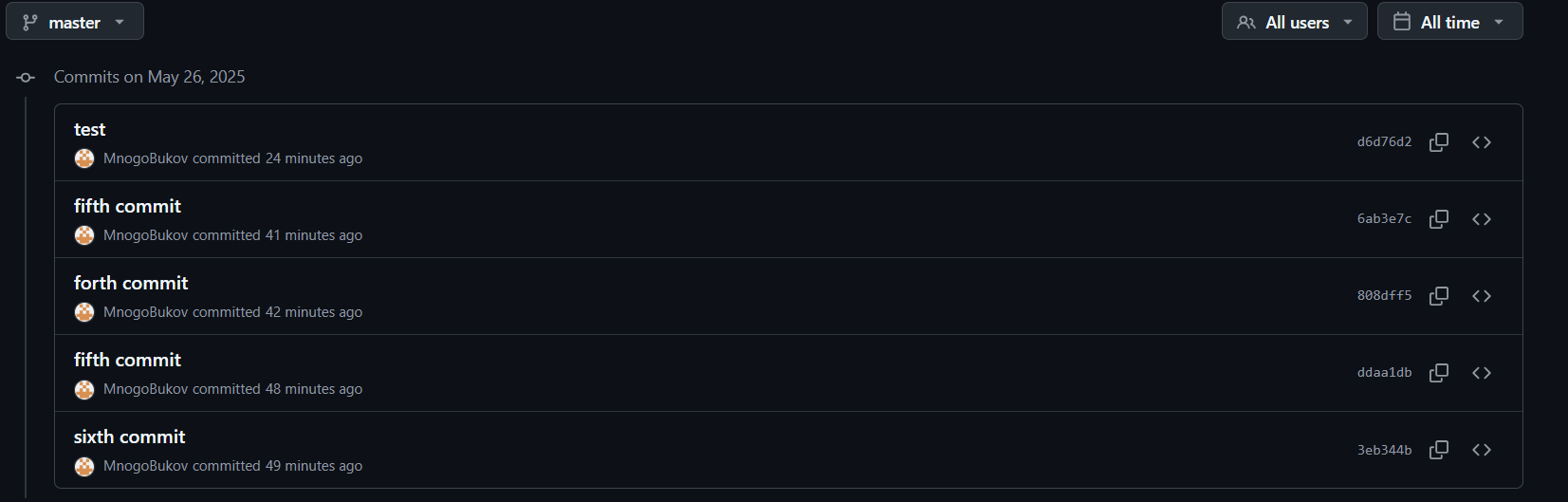




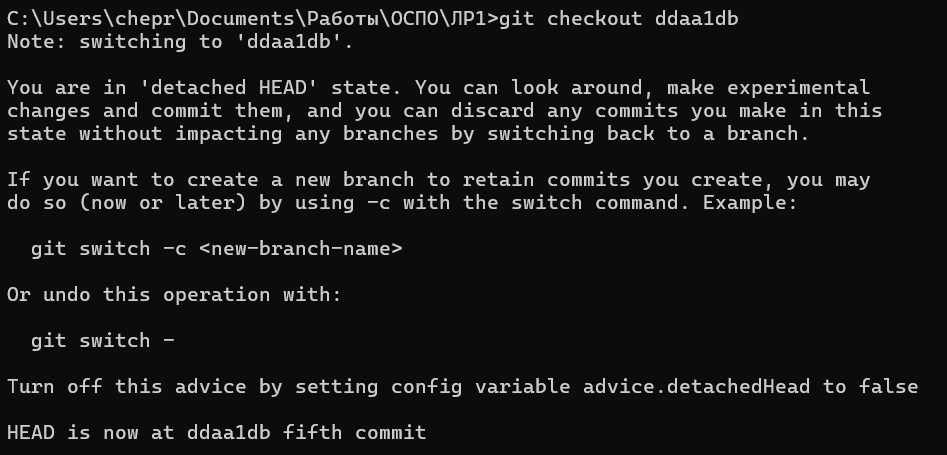
Все ветки удалены.

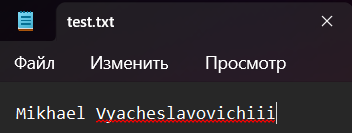
Попробуем откатиться на какой-нибудь коммит. Для этого можно использовать git log или посмотреть коммиты на GitHub



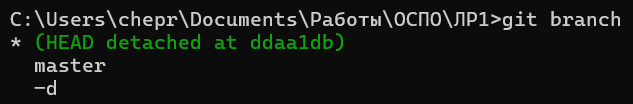


Для отката достаточно указать его первые несколько символов(например, fifth commit - ddaa1db)





Создалась псевдо-ветка

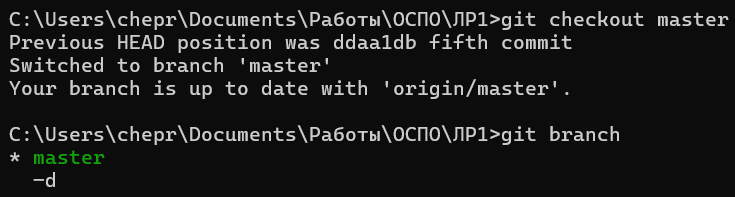


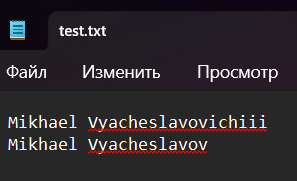
Также коммиты можно удалять при помощи команд

git reset --hard HEAD – удаляет все, что не закоммичено

git reset --hard HEAD~n – удаляет n коммитов

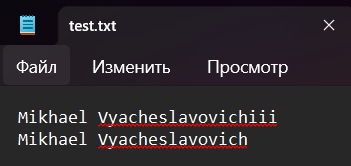
Откатимся обратно на ветку master





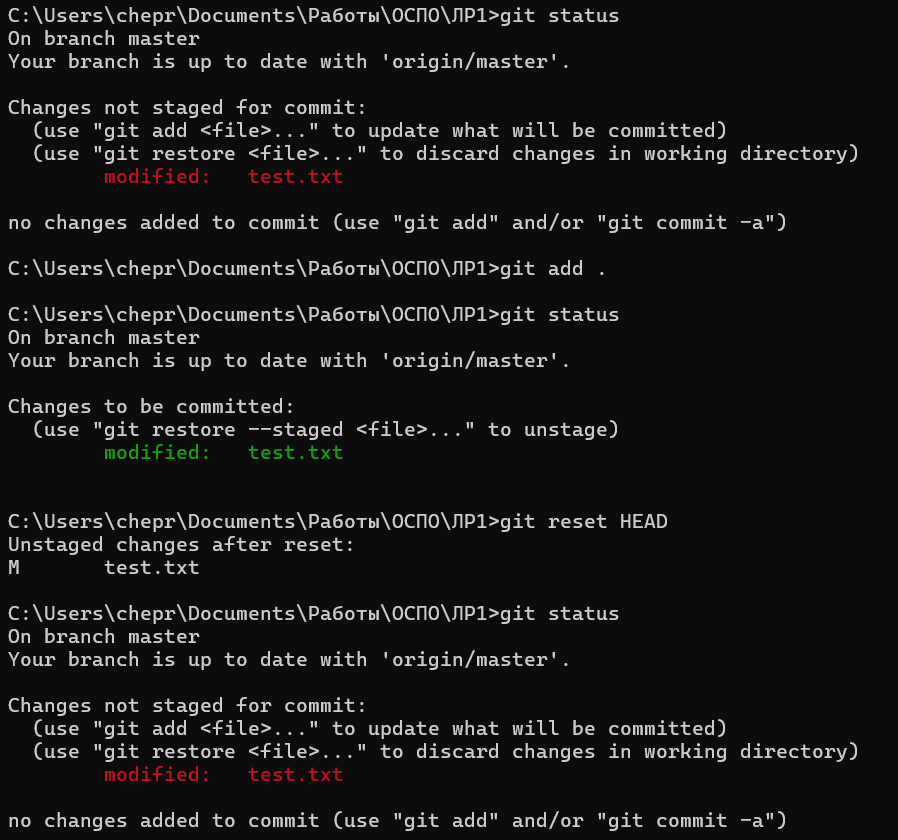
Все вернулось обратно, а псевдо-ветка удалилась, т.к. мы ее не коммитили и не отправляли на сервер.

Поменяем данные в файле test.txt

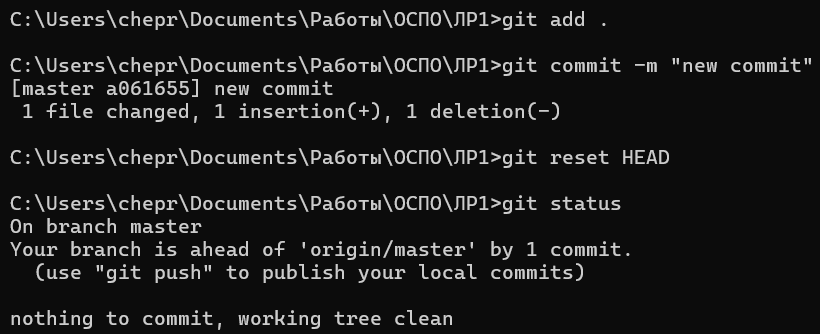


Можем добавить, а потом убрать файл из области закрепления, без коммита при помощи команды

git reset HEAD

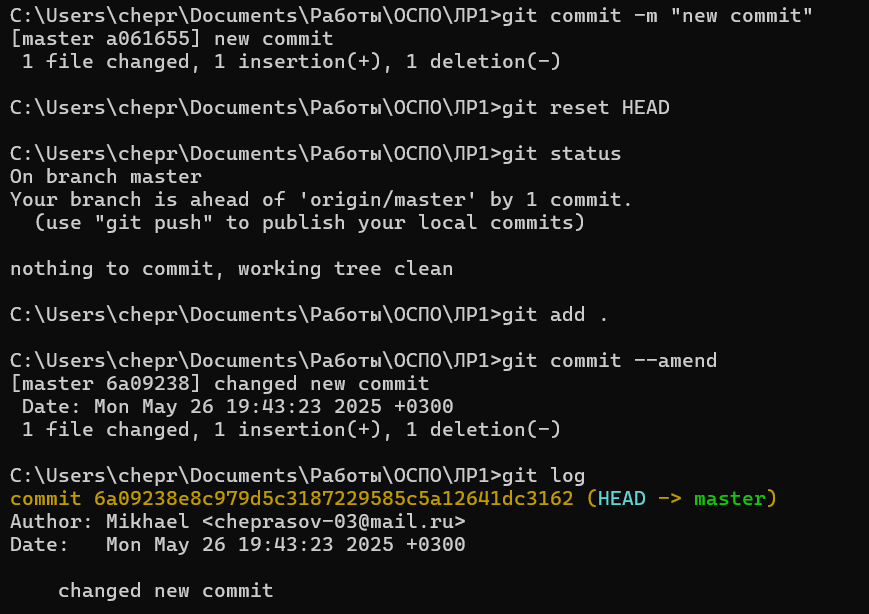


Теперь снова добавим в область закрепления и закомитим, но еще не отправляем на сервер, можно вызвать ту же команду и тогда все, что осталось незакомиченным будет удалено



Если захотим изменить комментарий можно вызвать команду:

git commit --amend



Для того, чтобы не коммитить файлы в репозитории(которые там присутствуют, но мы не хотим их коммитить), добавим файл .gitignore, закоммитим его и отправим на сервер

