

Создание и удаление группы файлов:

touch файл1.txt файл2.txt файл3.txt пустые файлы

echo "Текст для файла 1" > файл1.txt

echo "Текст для файла 2" > файл2.txt файлы с текстом

rm файл1.txt файл2.txt

mv пример.txt новый\_пример.txt переименовать

Команда git log (рис. 9) используется для просмотра истории коммитов в репозитории, позволяя анализировать внесённые изменения, авторов и временные рамки разработки. Флаги в команде git log позволяют фильтровать и настраивать вывод коммитов: -n <number> ограничивает количество выводимых коммитов, --since <date> и --after <date> показывают коммиты, сделанные после указанной даты, в то время как --until <date> и --before <date> отображают коммиты до определённой даты. Флаги --author <name> (рис. 11) и --committer <name> фильтруют коммиты по автору и коммиттеру соответственно, а --grep <pattern> позволяет искать коммиты по сообщениям, отображая только те, которые содержат указанный текст.

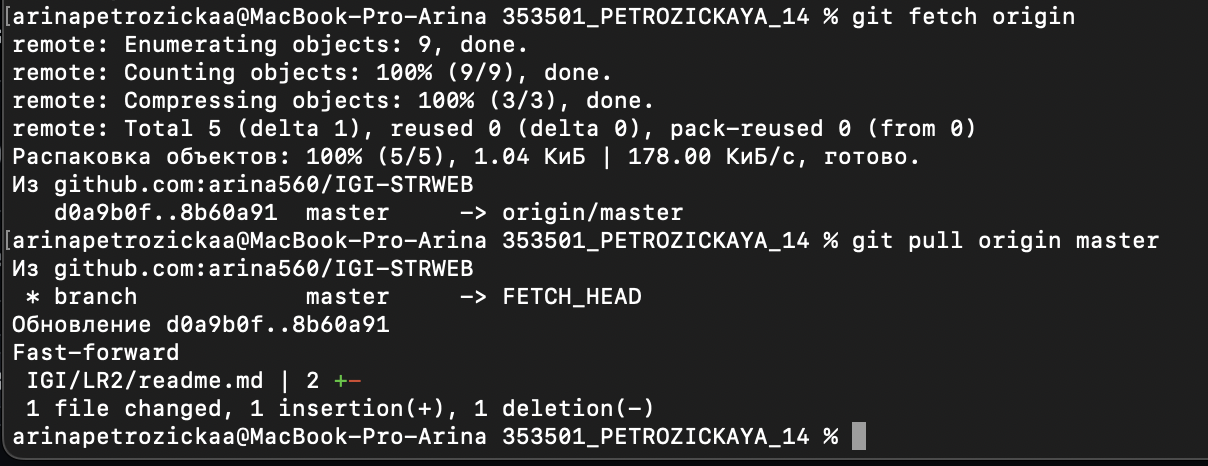
Следующие флаги команды git log позволяют настраивать вывод информации: -p отображает изменения (патчи) для каждого коммита, --stat показывает статистику изменений в файлах, --shortstat даёт краткую статистику изменений, --name-only выводит только имена изменённых файлов, а --name-status показывает имена файлов с их статусами. Флаг --relative-date отображает дату коммита в относительном формате , --pretty позволяет настраивать формат вывода коммитов, а --graph визуализирует структуру ветвления и слияний в истории.

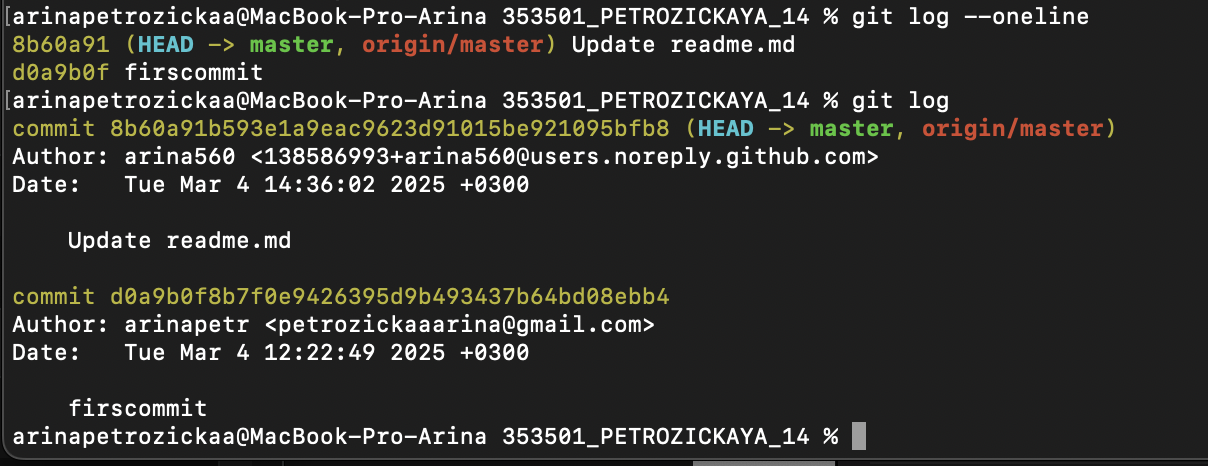
Для нахождения разницы между коммитами сначала были определены их хэши с помощью команды git log –oneline, а затем была найдена разница межд коммитом rename и del hello с помощью команды git diff <name> <name>

Индексация файла test.txt (git add text.txt) и её отмена (git reset test.txt)

Git push origin - - delete «название ветки» удалить удаленную ветку

git branch -D имя\_ветки удалить локальную ветку

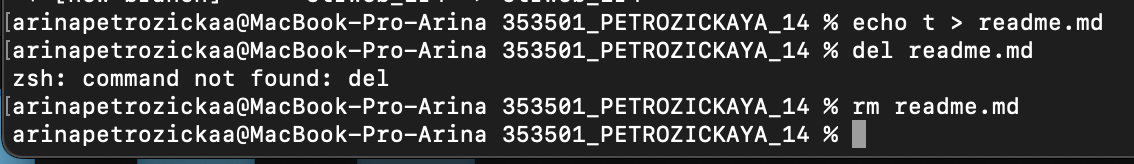
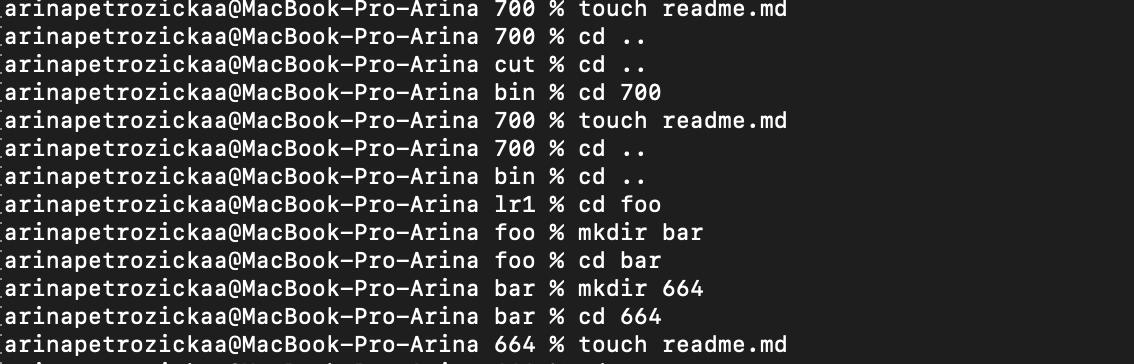
Получите данные из удаленного репозитория:

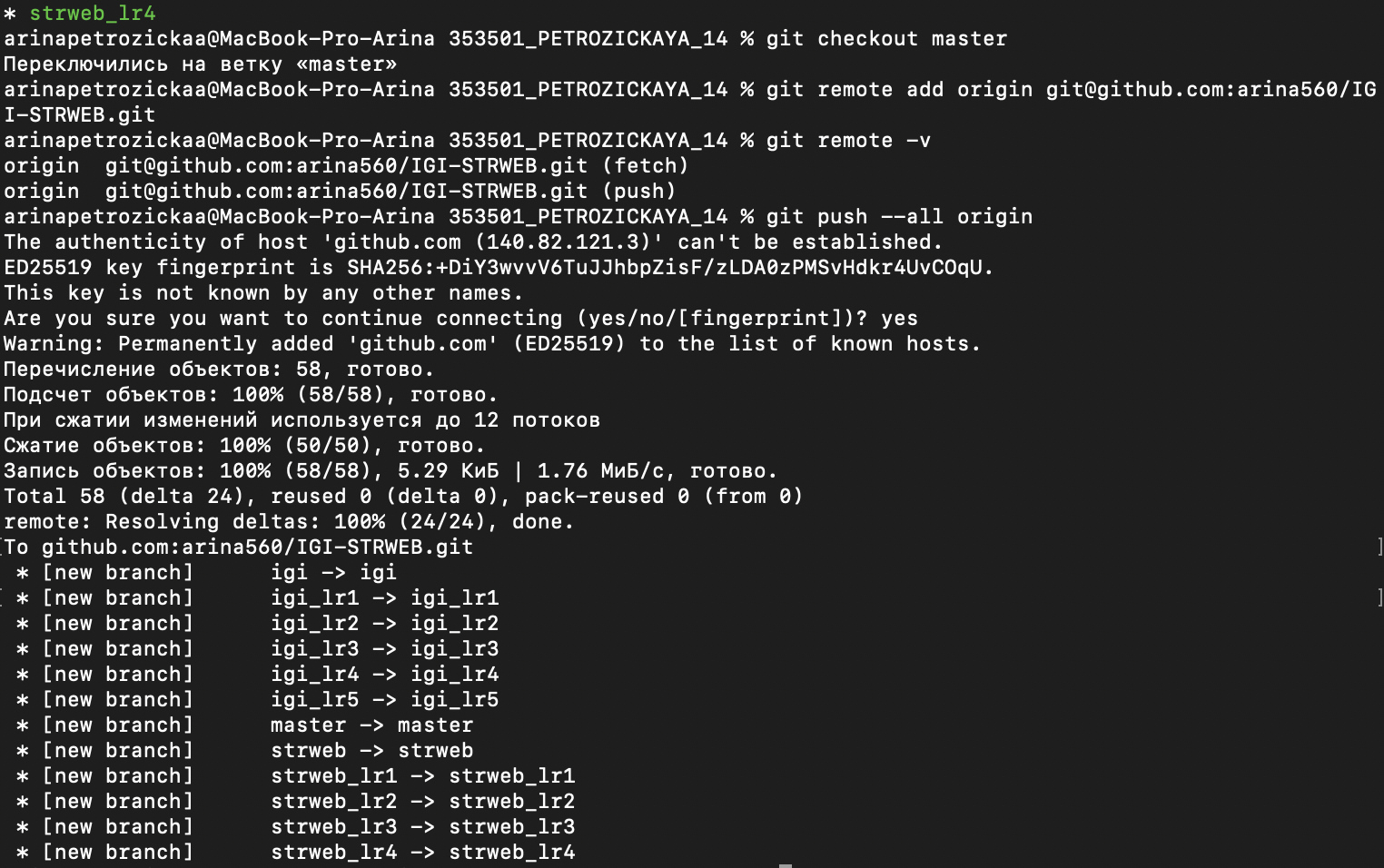
Просмотреть историю коммитов:

как перенести файлы из одной ветки в другую:

git checkout имя-целевой-ветки // переходим на нужную ветку

git checkout имя-исходной-ветки -- путь/к/файлу1 путь/к/файлу2 // имя ветки из которой мы переносим файлы, путь до файлов

git merge new-branch-2 слияние ветки new branch 2 с веткой в которой сейчас находятся

Git rm -rf . Сбрасывание индексации

вставленное-изображение.tiff