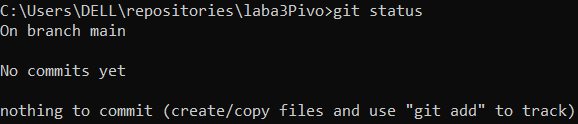
1. Создать локальный репозиторий в текущей папке:



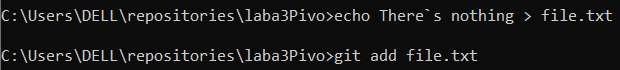
2. посмотреть статус текущего репозитория:



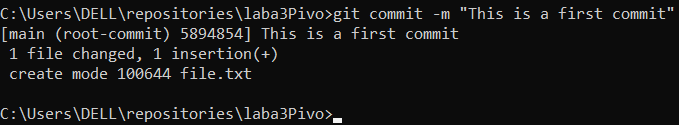
3. Что такое ветка?

Ветка – это независимая параллельная линия разработки, позволяющая работать над изменениями, не затрагивая основной проект.

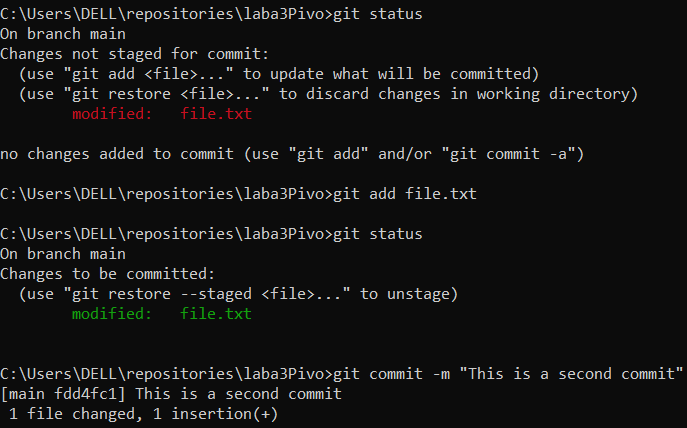
4. добавить файл в контекст, который будет коммититься:



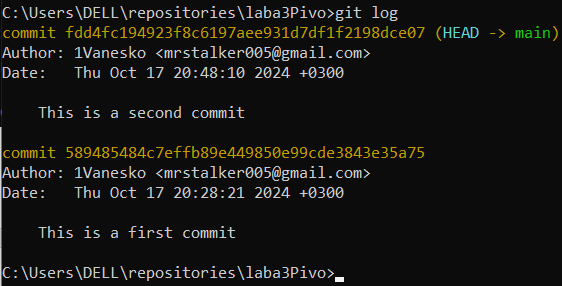
5.  создать коммит на основе текущего контекста и указать для него комментарий:



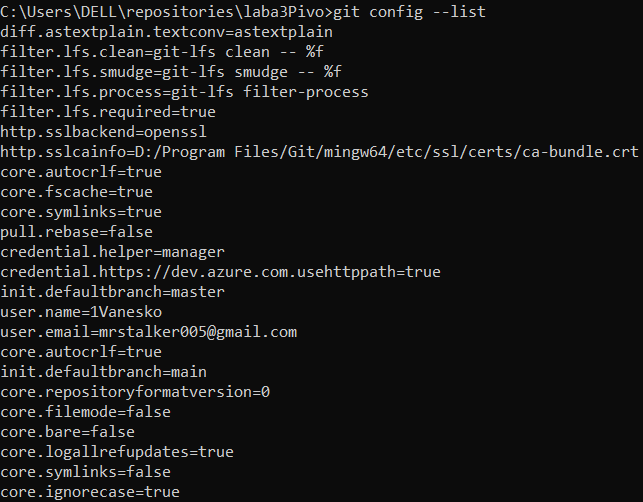
6. создать коммит, включающий изменения всех наблюдаемых файлов и указать для него комментарий:



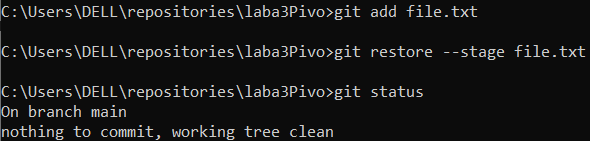
7. посмотреть протокол(лог) коммитов:



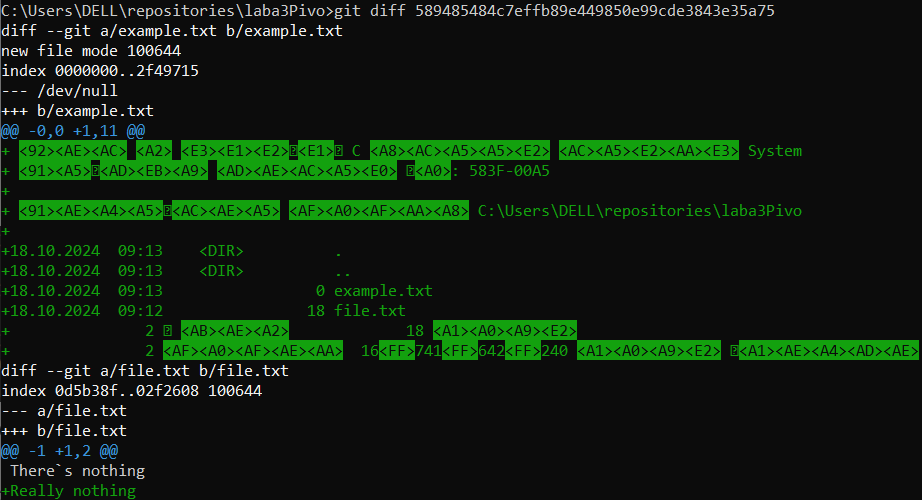
8. посмотреть информацию о текущих настройках:



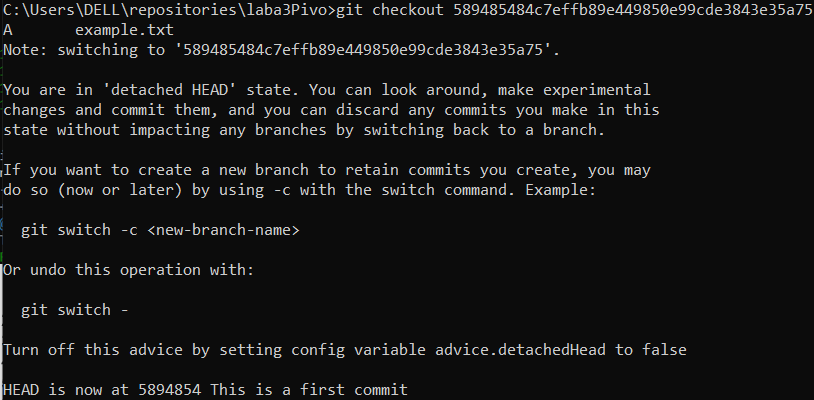
9. убрать файл из контекста:



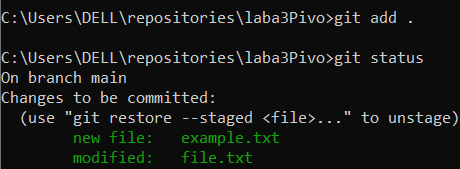
10. посмотреть изменения в файле по сравнению с последним коммитом:



11. убрать изменения относительно последнего коммита из файла



12. добавить в контекст коммита все измененные и созданный файлы



13. изменить глобальные/локальные настройки

Было:





Стало:

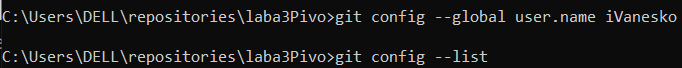


14. переписать имя пользователя



Было





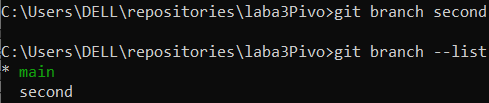
Стало:



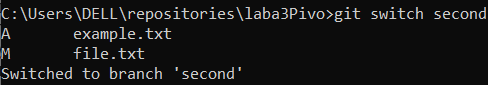
15. просмотреть существующие ветки



16. создать новую ветку



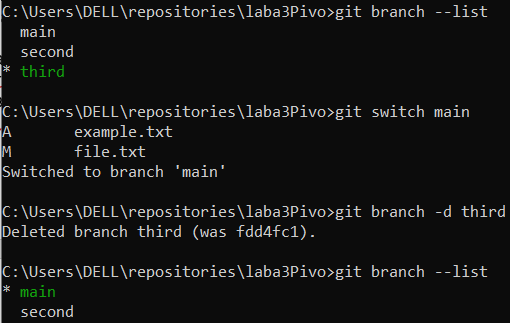
17. переключиться на другую ветку



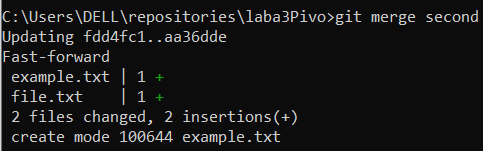
18. создать новую ветку и сразу же переключиться на неё



19. удалить ветку/удалить ветку, даже если она не примержена

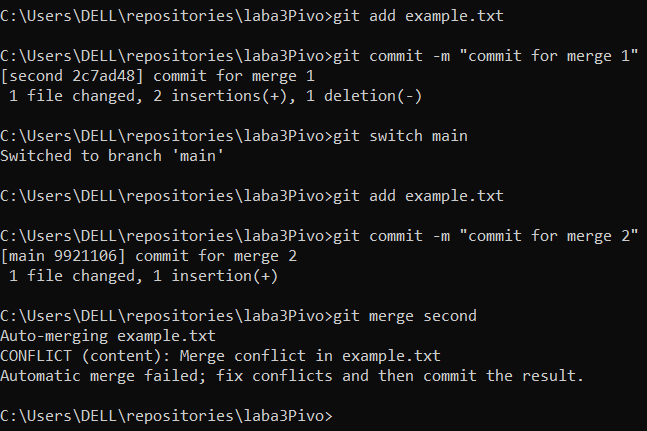


20. примержить изменения из указанной ветки в текущую



21. в каком случае могут появиться конфликты? сделать конфликт **/T**





22. как посмотреть в каких файлах конфликты:

При появлении конфликта в консоль пишет название конфликтного файла, либо можно прописать **git status** и увидеть: в каком возник файле конфликт.

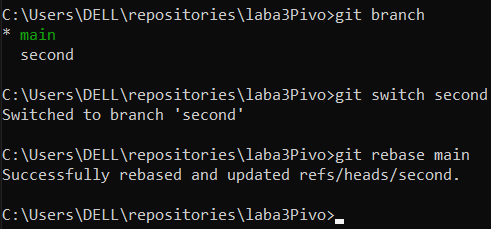
23. как устранить конфликты?

Открыть конфликтный файл из изменить его: оставить одно из изменений, либо грамотно соединить два и более ищменений.

24. как переключиться на указанный коммит:

Прописать в терминал команду: git checkout <хэш коммита>.

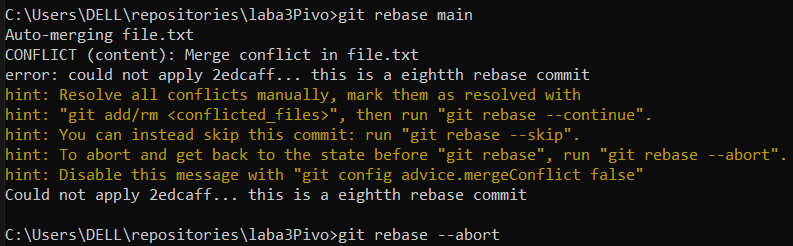
25. сделать ребазирование(rebase) текущей ветки



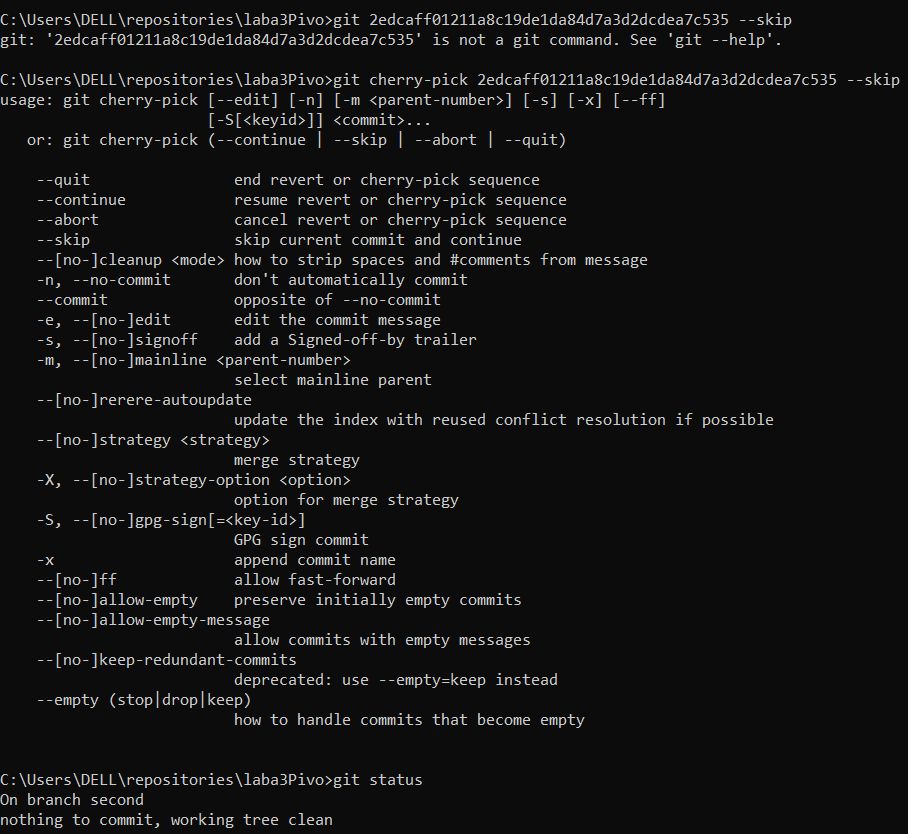
26. устранение конфликтов во время ребазирования?

Такое же, как и при мёрже.

27. отменить ребазирование во время конфликтов

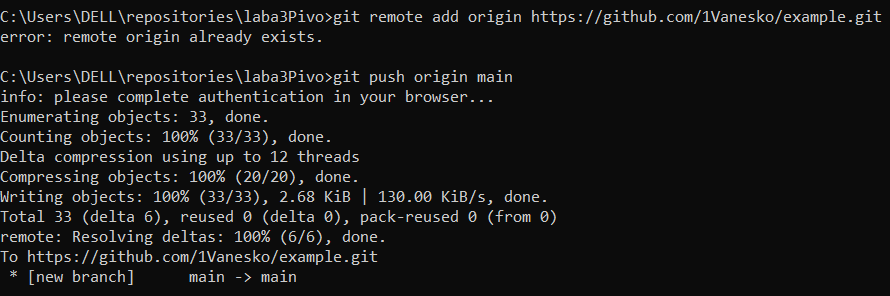


28. пропустить текущий конфликтный коммит и перейти к следующему



29. отправить изменения из локального репозитория для указанной ветки в удалённый(дистанционный) репозиторий

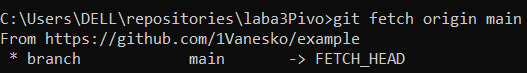




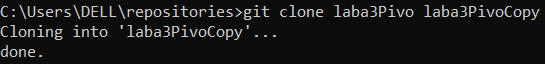
30. забрать изменения из репозитория, для которого были созданы удалённые ветки по умолчанию



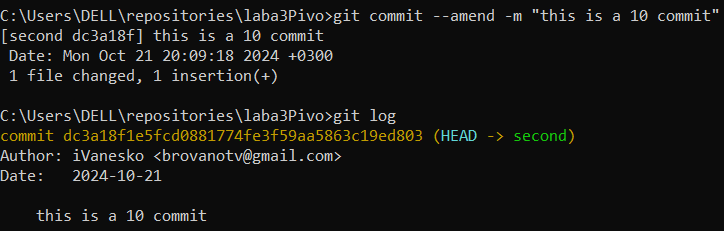
31. забрать изменения удалённой ветки из репозитория по умолчанию, основной ветки



32. создание копии репозитория



33. переименовать последний коммит



34. переименовать не последний коммит

Git rebase -i HEAD~N. N-номер коммита с конца.

35. скрыть изменения по сравнению с последним коммитом

