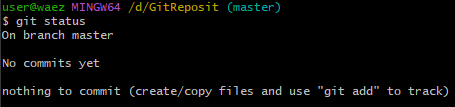
1. cоздать локальный репозиторий в текущей папке:



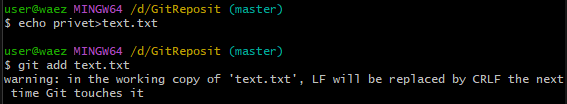
1. посмотреть статус текущего репозитория:



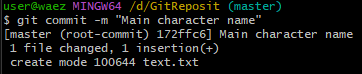
1. что такое ветка и какая ветка является обычно основной:

*Ветка в Git - это линия разработки. Обычно основной веткой является master или main.*

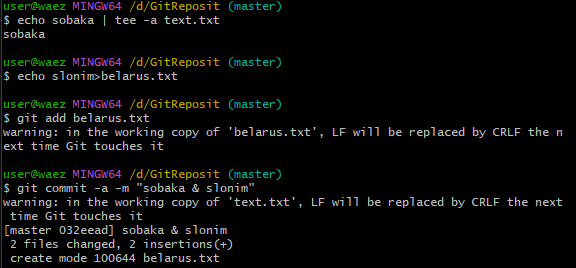
1. добавить файл в контекст, который будет коммититься:



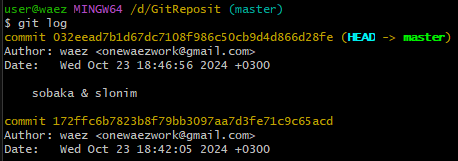
1. создать коммит на основе текущего контекста и указать для него комментарий:



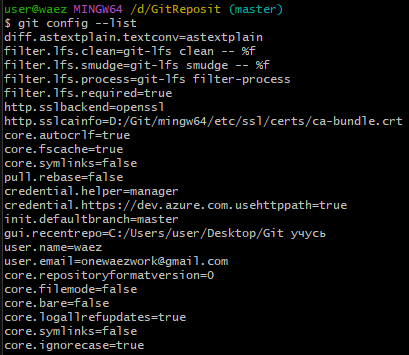
1. создать коммит, включающий изменения всех наблюдаемых файлов и указать для него комментарий:



1. посмотреть протокол(лог) коммитов:



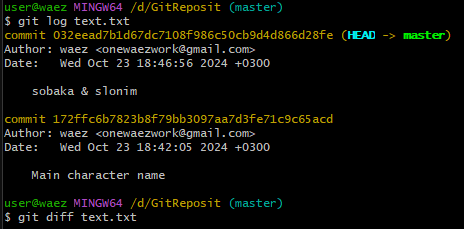
1. посмотреть информацию о текущих настройках:



1. убрать файл из контекста:

C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\9.png

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



1. убрать изменения относительно последнего коммита из файла:



1. добавить в контекст коммита все измененные и созданный файлы:



1. изменить глобальные/локальные настройки:

ГЛОБАЛЬНЫЕ



ЛОКАЛЬНЫЕ



1. переписать имя пользователя:

C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\13.2.png

1. просмотреть существующие ветки:



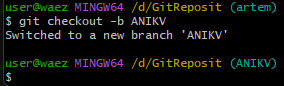
1. создать новую ветку:



1. переключиться на другую ветку:

C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\17.png

1. создать новую ветку и сразу же переключиться на неё:



1. удалить ветку/удалить ветку, даже если она не примержена:

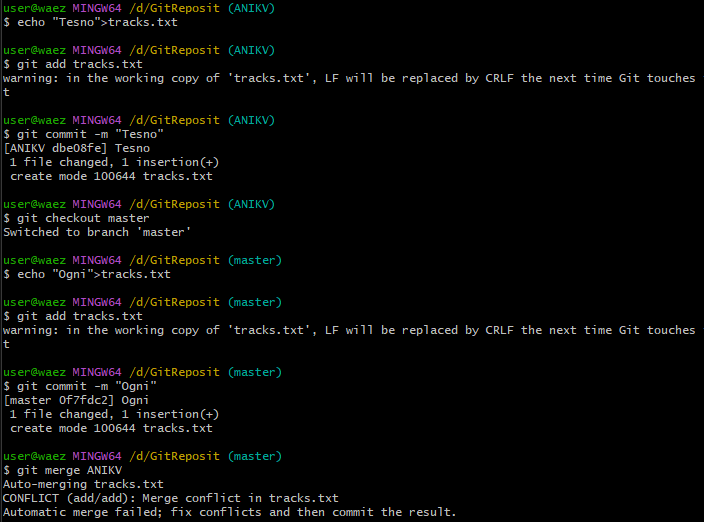


1. примержить изменения из указанной ветки в текущую:

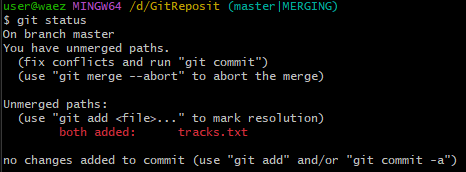


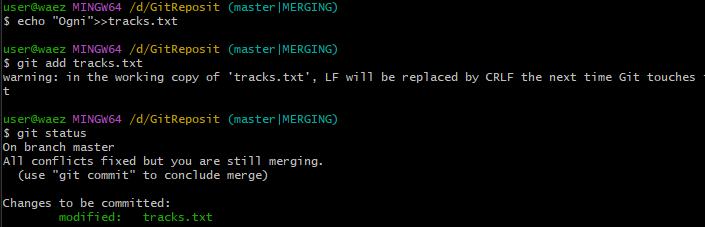
1. в каком случае могут появиться конфликты? сделать конфликт:

*Конфликты могут возникнуть в случаях, когда несколько членов команды работают над одним и тем же репозиторием Git, и каждый из них вносит изменения в одни и те же файлы. Конфликты могут возникнуть при слиянии (merge) или ребейзе (rebase) веток, а также при попытке отправить изменения в удаленный репозиторий (push), когда другой пользователь уже сделал изменения в том же файле.*



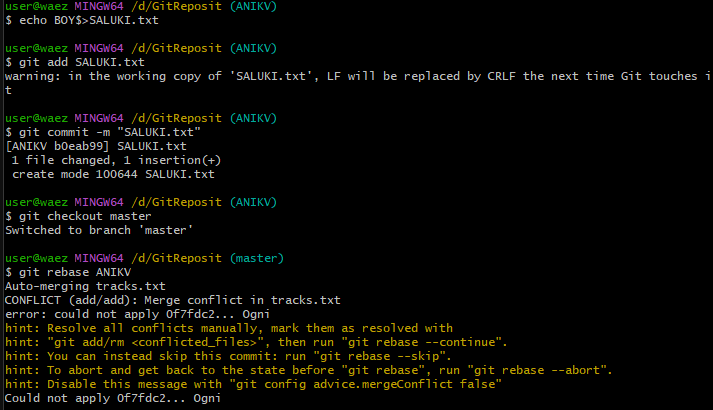
1. как посмотреть в каких файлах конфликты:



1. как устранить конфликты: 
2. как переключиться на указанный коммит:



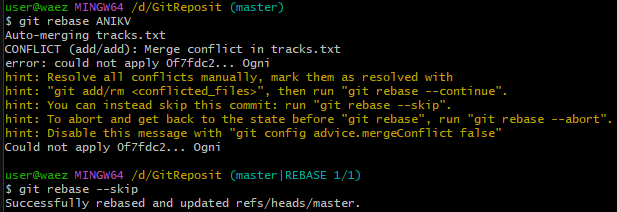
1. сделать ребазирование(rebase) текущей ветки:



27. отменить ребазирование во время конфликтов:



28. пропустить текущий конфликтный коммит и перейти к следующему:



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

29.1 Отправка изменений в указанную ветку:

Убедитесь, что вы находитесь на нужной ветке:

**bash**

Копировать код:

**git checkout имя\_ветки**

Или:

**bash**

Копировать код:

git switch имя\_ветки

Отправьте изменения на удалённый репозиторий:

**bash**

Копировать код:

**git push origin имя\_ветки**

Замените origin на имя удалённого репозитория (по умолчанию это origin) и имя\_ветки на название ветки, в которую хотите отправить изменения.

Например, если вы хотите отправить изменения из ветки **feature-branch**, выполните:

**bash**

Копировать код:

**git push origin feature-branch**

**30.** Как забрать изменения из репозитория, для котрого были созданы удаленные ветки по умолчанию.

30.1 Откройте проект в GitHub Desktop.

30.2 Переключитесь на нужную ветку.

30.3 Нажмите кнопку Fetch origin, чтобы забрать изменения.

30.4 Если в удалённой ветке есть новые изменения, которые отсутствуют в локальной, появится кнопка Pull, на которую можно нажать, чтобы забрать и объединить изменения.

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

31.1 Убедитесь, что вы находитесь на ветке main.

31.2 Нажмите кнопку Fetch origin, чтобы забрать изменения.

31.3 Если есть новые изменения в origin/main, появится кнопка Pull. Нажмите на неё, 31.4 чтобы получить и объединить изменения с вашей локальной веткой.

32.Как клонировать репозиторий

32.1 Нажмите File > Clone Repository.

32.2 Вставьте URL репозитория в поле или выберите репозиторий из списка, если он 32.3 находится в вашем аккаунте.

32.4 Выберите путь для сохранения репозитория на вашем компьютере.

Нажмите Clone.

33.Как переименовать последний коммит

Чтобы переименовать последний коммит в Git, используйте команду

**git commit --amend**

Эта команда позволяет внести изменения в последний коммит, включая изменение его сообщения.

35.Как скрыть изменения по сравнению с последним коммитом

Чтобы скрыть изменения в файлах по сравнению с последним коммитом в Git, вы можете использовать команду

**git checkout**

Или

**git restore.**

Эти команды позволят вам отменить все незакоммиченные изменения и вернуть файлы к состоянию последнего коммита.