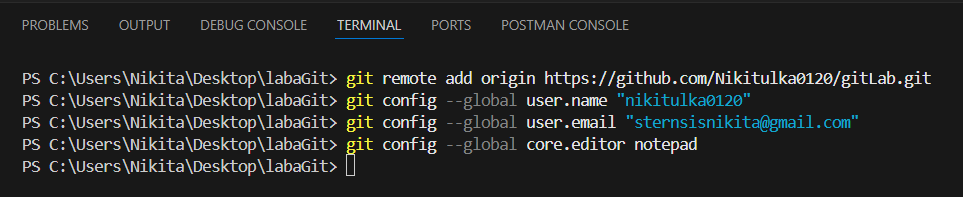
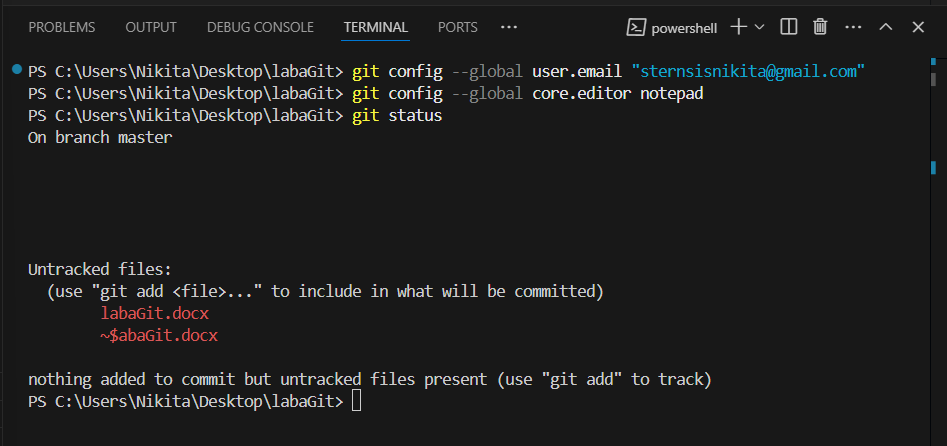
Логинимся



Проверяем ветку через git status



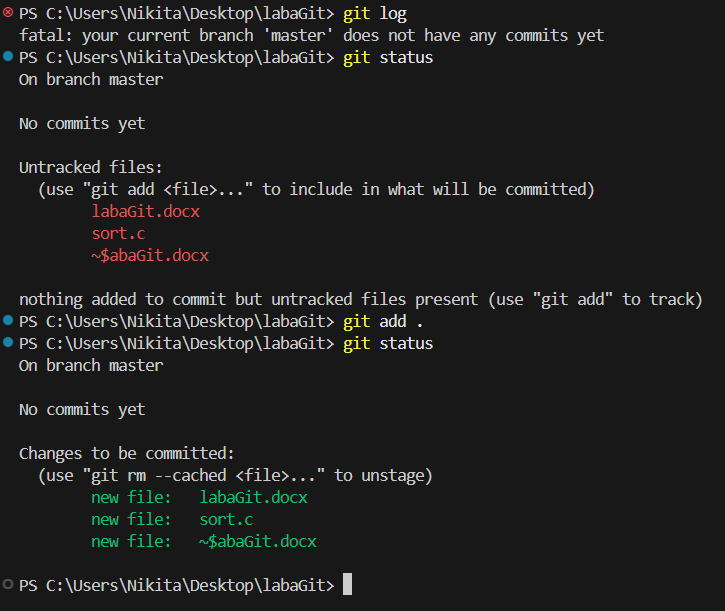
Как выглядит git log?

Создайте файл sort.c и вставьте туда код функции любой сортировки (только ф-ия сортировки)

Как сейчас выглядит вывод git status?

Добавьте файл в область stage (add)

Как сейчас выглядит git status?



Закоммитить файл в репозиторий

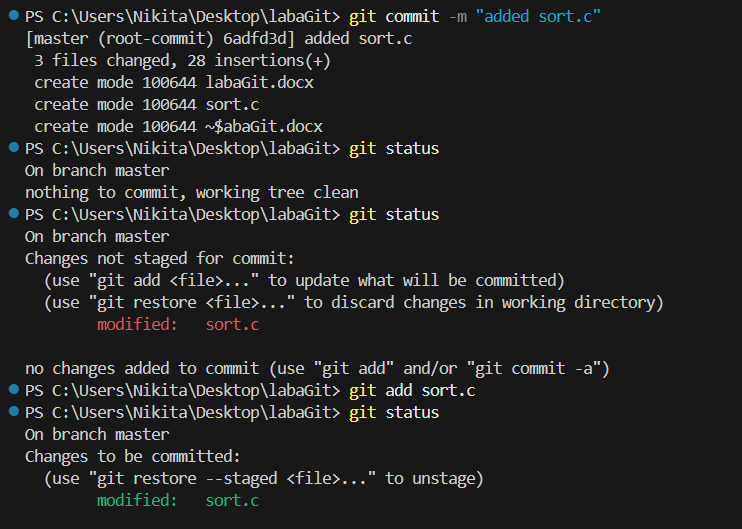
Как сейчас выглядит git status?

Добавить комментарий с любым текстом в этот же файл

Как сейчас выглядит git status?

Добавьте (add) изменение файла

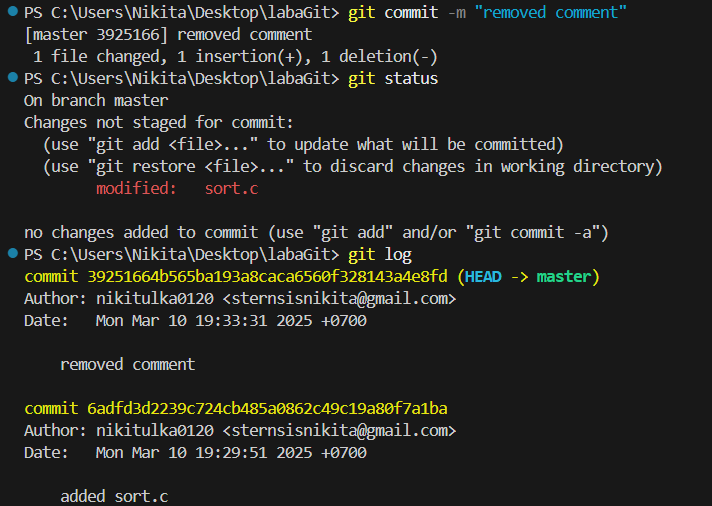
Как сейчас выглядит git status?



Измените файл еще раз (можно добавить еще комментарий или убрать старый)

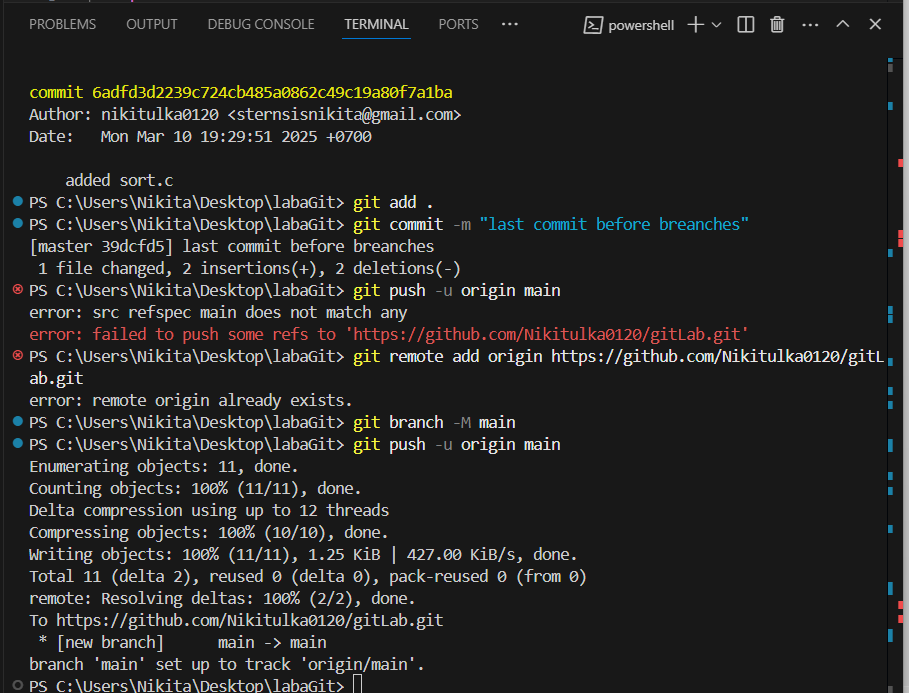
Сделайте коммит

Как сейчас выглядит status? Журнал (log)?



Добавьте в stage и закоммитьте последнее изменение

Запушим на удаленный репо (git push)



Теперь мы немного поиграем с ветками.

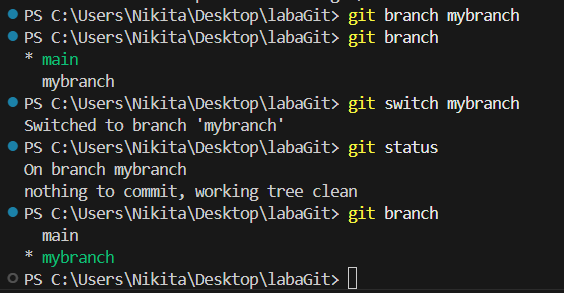
Используйте git branch mybranch (или git checkout -b mybranch), чтобы создать новую ветку с именем mybranch.

Снова используйте git branch, чтобы увидеть новую созданную ветку.

Используйте git switch mybranch (или git checkout mybranch), чтобы переключиться на новую ветку.

Как изменяется вывод git status при переключении между master и новой веткой, которую вы создали?

Убедитесь, что вы находитесь на своей ветке mybranch, прежде чем продолжить.



Создайте файл с именем file1.txt и своим именем.

Добавьте файл и закоммитьте это изменение.

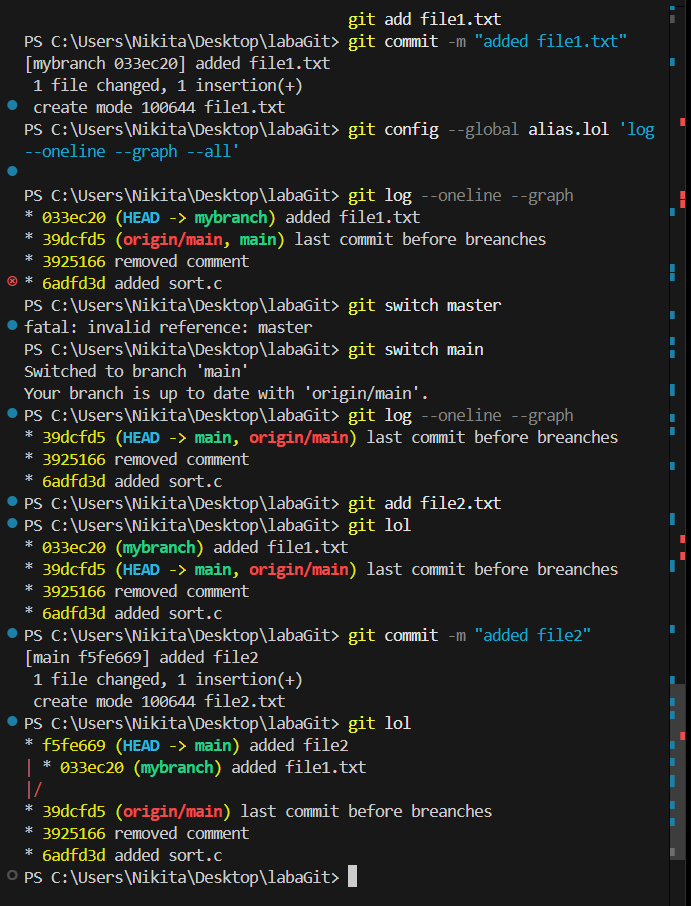
Используйте git log --oneline --graph, чтобы увидеть, что ваша ветка указывает на новый коммит.

Вернитесь к ветке с именем master.

Используйте git log --oneline --graph, что изменилось?

Создайте новый файл с именем file2.txt и закоммитьте его.

Используйте git log --oneline --graph --all, чтобы увидеть, что ваша ветка указывает на новый коммит, и что теперь у двух веток разные коммиты.



Переключитесь на вашу ветку mybranch.

Наш file2.txt пропал?

Используйте git diff mybranch master, чтобы увидеть разницу между двумя ветками.

