1. Просмотр истории коммитов

Для понимания текущего состояния проекта использовалась команда:

git log --oneline

Пример вывода:

417ea33 (HEAD -> master) Добавлено расписание на среду

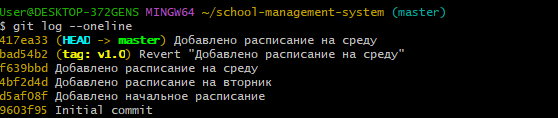
bad54b2 (tag: v1.0) Revert "Добавлено расписание на среду"

f639bbd Добавлено расписание на среду

4bf2d4d Добавлено расписание на вторник

d5af08f Добавлено начальное расписание

9603f95 Initial commit



Это помогло определить, к какому коммиту можно сделать сброс.

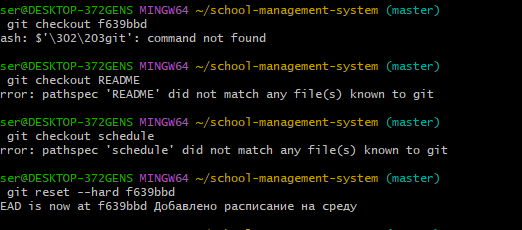
2. Сброс ветки f639bbd до нужного коммита

Если ветка f639bbd содержала ошибочные изменения, был выполнен сброс:

git checkout f639bbd

git reset --hard f639bbd

После этого ветка f639bbd восстановлена к предыдущему рабочему состоянию.

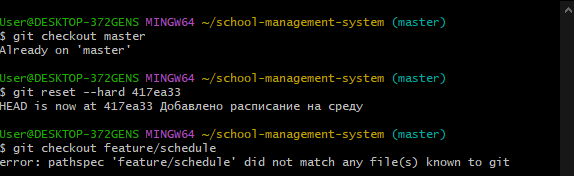


3. Сброс ветки master (или main)

Если основная ветка тоже требовала отката:

git checkout master

git reset --hard 417ea33



Полезно, если были неправильные слияния или ошибки в главной ветке.

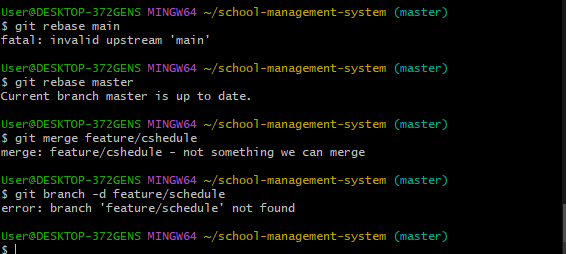
4. Перебазирование (rebase) вместо слияния

Вместо команды merge, которая создаёт отдельный коммит, была выполнена команда rebase:

git checkout feature/schedule

git rebase main

Таким образом, история изменений осталась линейной, без лишних merge-коммитов.

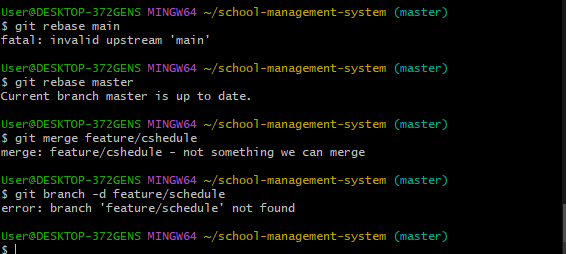


5. Слияние веток в main

После успешного rebase ветка объединяется с main:

git checkout main

git merge feature/schedule



Такой подход позволяет сохранить чистую историю.

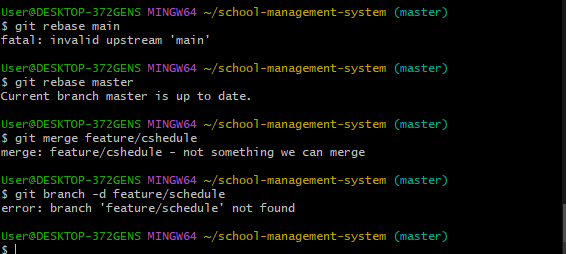
6. Удаление временных веток после завершения задачи

После слияния ветку можно удалить:

git branch -d feature/schedule

Если она уже была отправлена на сервер — удали её там:

git push origin :feature/schedule



Это важно для поддержания порядка в репозитории.