1. Установка и настройка Git

Команды bash

git config --global user.name "MrZhr4ik"

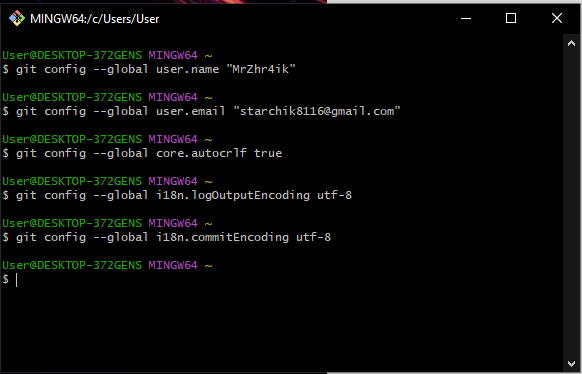
git config --global user.email "starchik8116@gmail.com"

git config --global core.autocrlf true

git config --global i18n.logOutputEncoding utf-8

git config --global i18n.commitEncoding utf-8

Эти команды задают имя и email пользователя, настраивают правильное отображение русских символов и окончания строк.



2. Создание папки проекта и файлов

Команды bash

mkdir school-management-system

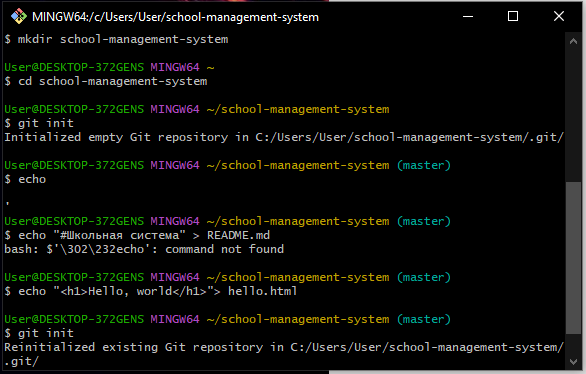
cd school-management-system

git init

Созданы файлы через echo (работает в Windows):

echo "# Школьная система" > README.md

echo "<h1>Hello, world</h1>" > hello.html



3 Инициализация репозитория Git

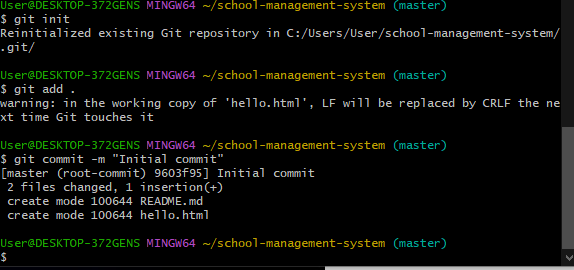
Команды bash

git init

git add .

git commit -m "Initial commit"

После этого создан локальный репозиторий с историей изменений



4. Проверка состояния репозитория

Команда bash

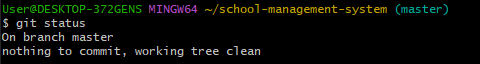
git status

Пример вывода:

On branch main

nothing to commit, working tree clean

Это означает, что все изменения уже добавлены и закоммичены



5. Добавление нового файла schedule.html

Команда:

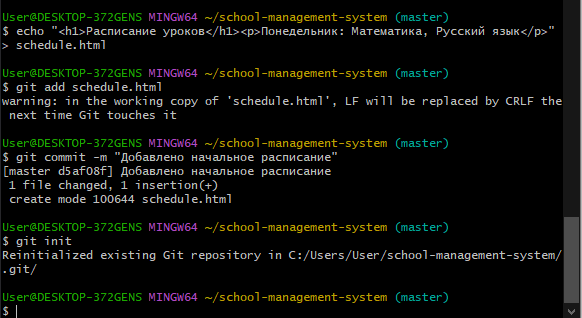
echo "<h1>Расписание уроков</h1><p>Понедельник: Математика, Русский язык</p>" > schedule.html

Добавление в Git:

git add schedule.html

git commit -m "Добавлено начальное расписание"

Так сделан первый шаг к разработке модуля расписания



6. Изменения в файле и второй коммит

Обновление содержимого schedule.html:

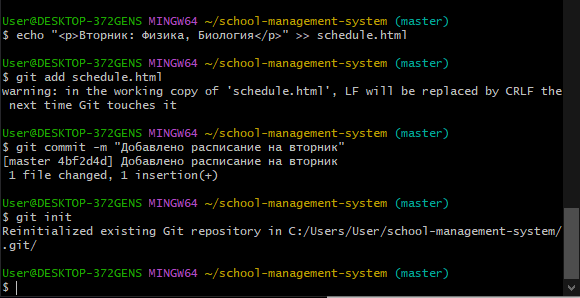
echo "<p>Вторник: Физика, Биология</p>" >> schedule.html

Добавление и коммит:

git add schedule.html

git commit -m "Добавлено расписание на вторник"

Таким образом, показано, как Git фиксирует конкретные изменения , а не просто файлы



7. Еще одно изменение и третий коммит

Добавление среды:

echo "<p>Среда: Химия, Английский</p>" >> schedule.html

git add schedule.html

git commit -m "Добавлено расписание на среду"

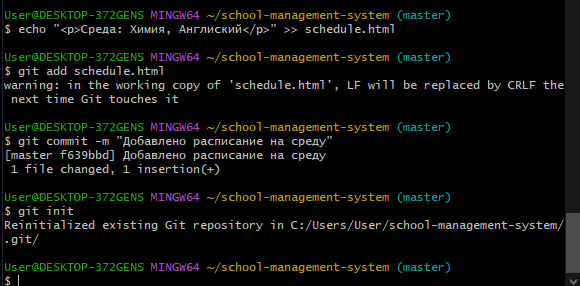
Три коммита:

Initial commit

Добавлено расписание на понедельник

Добавлено расписание на вторник

Добавлено расписание на среду



8. Просмотр истории коммитов

Команда:

git log

Пример вывода:

commit f639bbde866f63c9d9dbbcae6734b243206cf4b6 (HEAD -> master)

Author: MrZhr4ik <starchik8116@gmail.com>

Date: Thu Jun 5 13:47:56 2025 +0300

Добавлено расписание на среду

commit 4bf2d4df09e9483b9ed2d06ec3384fcaa39de761

Author: MrZhr4ik <starchik8116@gmail.com>

Date: Thu Jun 5 13:42:35 2025 +0300

Добавлено расписание на вторник

commit d5af08f461d0828ad28ae76dd1e14f58047e8c59

Author: MrZhr4ik <starchik8116@gmail.com>

Date: Thu Jun 5 13:36:26 2025 +0300

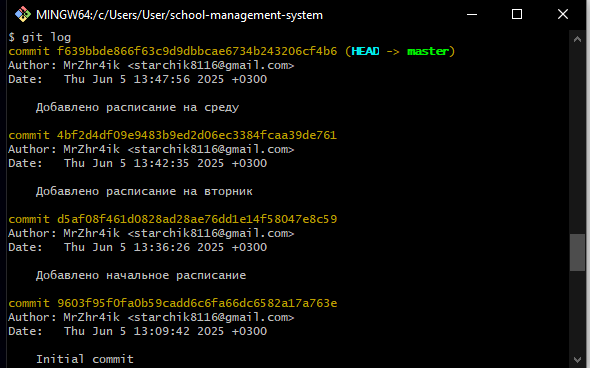
Добавлено начальное расписание

commit 9603f95f0fa0b59cadd6c6fa66dc6582a17a763e

Author: MrZhr4ik <starchik8116@gmail.com>

Date: Thu Jun 5 13:09:42 2025 +0300

Initial commit



Это позволяет отслеживать историю изменений проекта

9. Краткий формат истории

Команда:

git log --pretty=oneline

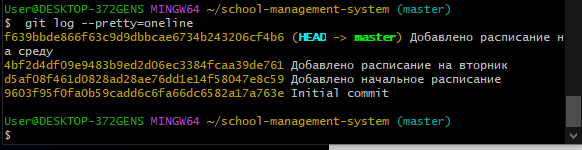
Пример вывода:

f639bbde866f63c9d9dbbcae6734b243206cf4b6 (HEAD -> master) Добавлено расписание на среду

4bf2d4df09e9483b9ed2d06ec3384fcaa39de761 Добавлено расписание на вторник

d5af08f461d0828ad28ae76dd1e14f58047e8c59 Добавлено начальное расписание

9603f95f0fa0b59cadd6c6fa66dc6582a17a763e Initial commit



Удобно для быстрого просмотра истории

10. Фильтр по времени и автору

Примеры:

git log --since="7 days ago"

git log --until="2025-05-29"

git log --author="MrZhr4ik"

Позволяет находить нужные коммиты быстро

11. Индексация и коммит по частям

Git позволяет индексировать не весь файл сразу , а только его часть.

Пример:

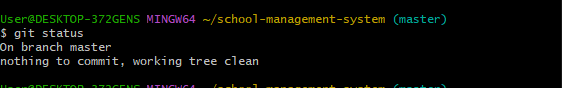
git add -p schedule.html

После этого можно выбрать, какие именно изменения зафиксировать



12. Проверка статуса после изменений

git status



Это помогает избежать случайного коммита ненужных изменений