



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ

ГОРОДА МОСКВЫ

Государственное бюджетное профессиональное

образовательное учреждение города Москвы

«Колледж малого бизнеса № 4»

(ГБПОУ КМБ № 4)

Отчёт по лабораторной работе №5

Специальность: 09.02.07 Информационные системы и программирование

Форма обучения: очная

Студент: Межибор Ярослав Евгеньевич

Группа: ИПО-21.24

Проверил: Рыбаков Александр Сергеевич

Москва, 2025 г.

Цель работы

Научиться работать с удаленными репозиториями и освоить механизмы синхронизации веток. Изучить разницу между получением информации об изменениях и их непосредственным слиянием с локальным кодом.

Задачи

1. Изучить понятие ветки как последовательности коммитов и определить текущую активную ветку (master) с помощью команд `git status` и `git log`.
2. Создать локальный репозиторий TMP2, который будет выступать в роли удаленного источника данных.
3. Настроить связь между основным проектом и удаленным репозиторием, присвоив источнику имя `super2`.
4. Отработать алгоритм получения данных: сначала загрузить информацию об изменениях через `git fetch`, а затем настроить отслеживание веток и выполнить слияние через `git pull`.

Ход работы

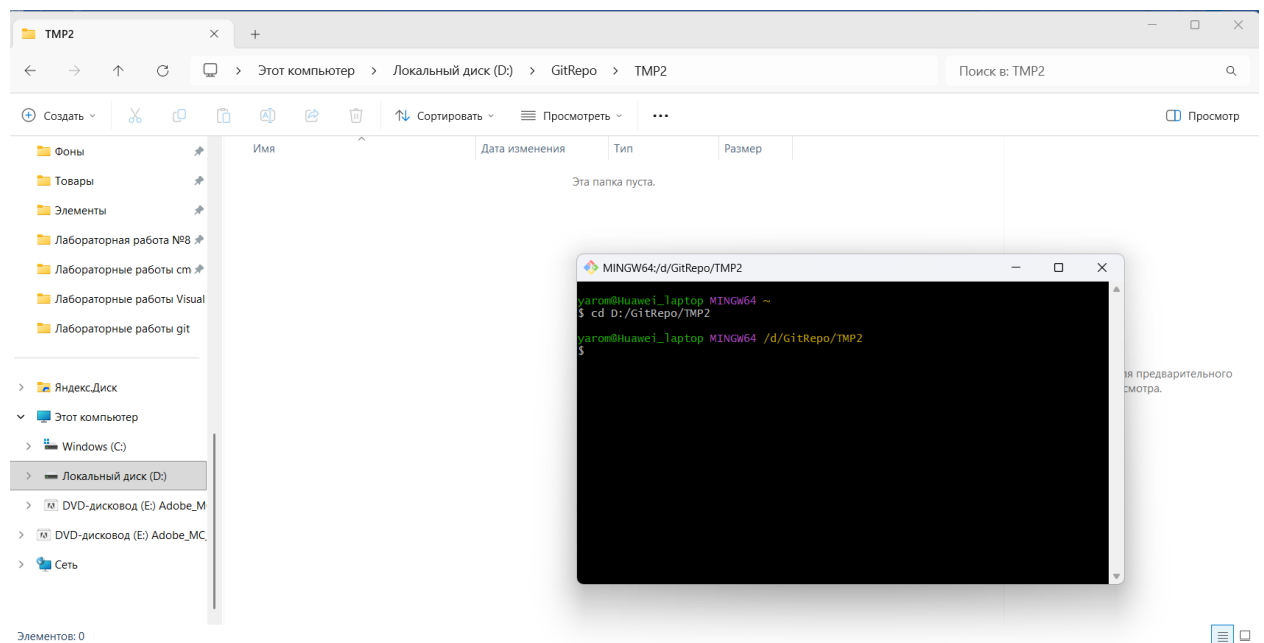


Рис. 1. Создание новой папки и запуска ее в GitBash



```
MINGW64:/d/GitRepo2

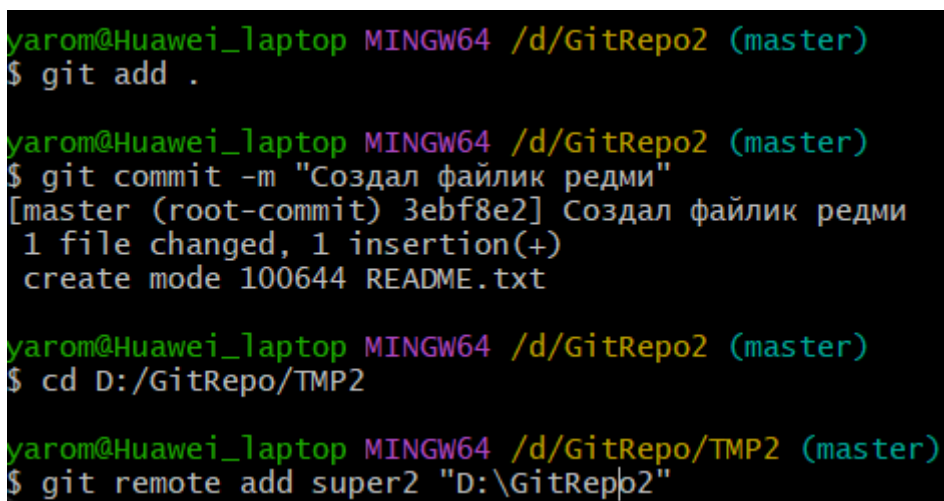
yarom@Huawei_laptop MINGW64 ~
$ cd D:/GitRepo/TMP2

yarom@Huawei_laptop MINGW64 /d/GitRepo/TMP2
$ git init
Initialized empty Git repository in D:/GitRepo/TMP2/.git/

yarom@Huawei_laptop MINGW64 /d/GitRepo/TMP2 (master)
$ cd D:/GitRepo2

yarom@Huawei_laptop MINGW64 /d/GitRepo2
$ git init
Initialized empty Git repository in D:/GitRepo2/.git/
```

Рис. 2. Создание 2 репозитория и их инициализация



```
yarom@Huawei_laptop MINGW64 /d/GitRepo2 (master)
$ git add .

yarom@Huawei_laptop MINGW64 /d/GitRepo2 (master)
$ git commit -m "Создал файл readme"
[master (root-commit) 3ebf8e2] Создал файл readme
1 file changed, 1 insertion(+)
create mode 100644 README.txt

yarom@Huawei_laptop MINGW64 /d/GitRepo2 (master)
$ cd D:/GitRepo/TMP2

yarom@Huawei_laptop MINGW64 /d/GitRepo/TMP2 (master)
$ git remote add super2 "D:\GitRepo2"
```

Рис. 3. Закоммитил изменения, создав файл README и добавил ссылку на удаленный репозиторий в текущем репозитории



```
yarom@Huawei_laptop MINGW64 /d/GitRepo/TMP2 (master)
$ git remote
super2

yarom@Huawei_laptop MINGW64 /d/GitRepo/TMP2 (master)
$ git remote -v
super2 D:\GitRepo2 (fetch)
super2 D:\GitRepo2 (push)

yarom@Huawei_laptop MINGW64 /d/GitRepo/TMP2 (master)
$ git remote show super2
* remote super2
Fetch URL: D:\GitRepo2
Push URL: D:\GitRepo2
HEAD branch: master
Remote branch:
master new (next fetch will store in remotes/super2)
```

Рис. 4. Просмотр удаленных репо, прикрепленных к текущему репо

```

yarom@Huawei_laptop MINGW64 /d/GitRepo/TMP2 (master)
$ git fetch super2
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 243 bytes | 15.00 KiB/s, done.
From D:\GitRepo2
* [new branch]      master      -> super2/master

yarom@Huawei_laptop MINGW64 /d/GitRepo/TMP2 (master)
$ git checkout --track super2/master
branch 'master' set up to track 'super2/master'.
Already on 'master'

```

Рис. 5. Перенес информацию об изменениях и связывание веток между собой

```

yarom@Huawei_laptop MINGW64 /d/GitRepo/TMP2 (master)
$ git pull
Already up to date.

```

Рис. 6. Выкачал все изменения из удаленного репозитория

Контрольные вопросы

1. Это именованная последовательность коммитов, представляющая собой отдельную линию истории изменений в проекте.
2. На локальные (находящиеся на вашем компьютере) и удаленные (находящиеся на сервере или в другой сетевой папке).
3. С помощью команды `git remote add [имя_удаленного_репозитория] [путь_или_ссылка_к_нему]`.
4. С помощью команды `git remote -v`.
5. Отображает детальную информацию о конкретном удаленном репозитории: адреса для получения и отправки данных, ветки, которые отслеживаются, и их текущее состояние.

Вывод

В ходе выполнения работы я на практике изучил принципы взаимодействия между репозиториями. Я понял разницу между командой `git fetch`, которая только обновляет информацию о состоянии удаленных веток, и `git pull`, которая физически вливает изменения в мой проект. Также я научился использовать команду `git checkout --track` для связывания локальной ветки с веткой удаленного репозитория.