**Определение *GIT***

*GIT* – это система контроля версий, которая позволяет отслеживать изменения в файлах и управлять разработкой проектов. Она помогает сохранять историю изменений, работать с разными версиями файлов и эффективно сотрудничать с другими людьми. Наглядный принцип работы системы *GIT* на Рисунке 1.

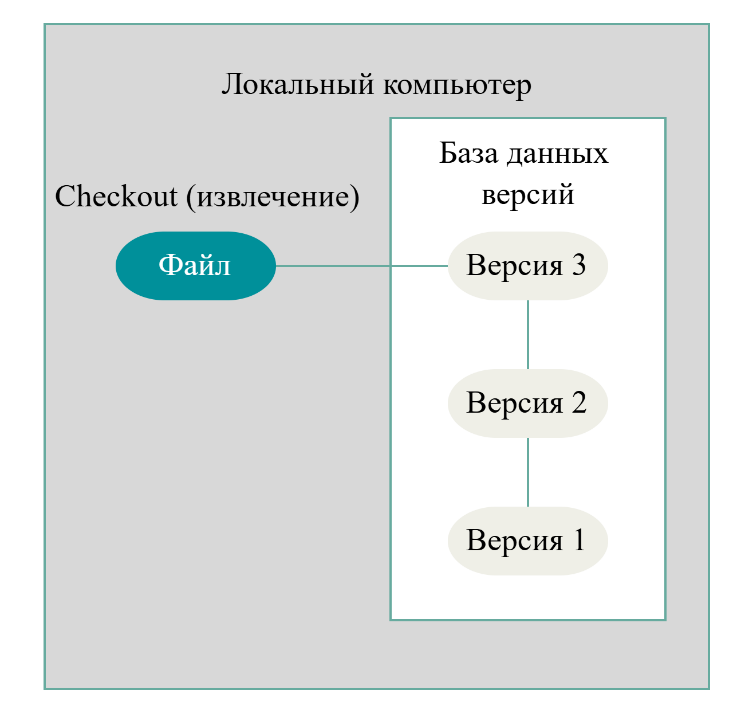


Рисунок 1 – Принцип работы *GIT*

**Что такое локальный репозиторий**

Локальный репозиторий *GIT* – это ваше рабочее пространство для управления проектом. Он позволяет отслеживать изменения, сохранять историю и работать с проектом, не завися от интернета или удалённых серверов.

**Что такое удалённый репозиторий**

Удалённый репозиторий *GIT* – это копия вашего проекта, которая хранится на внешнем сервере или в облачном сервисе (например, *GitHub*, *GitLab*, *Bitbucket*). Удалённый репозиторий используется для совместной работы над проектом, резервного копирования и обмена изменениями между разработчиками. Наглядный принцип работы удалённого репозитория на Рисунке 2.

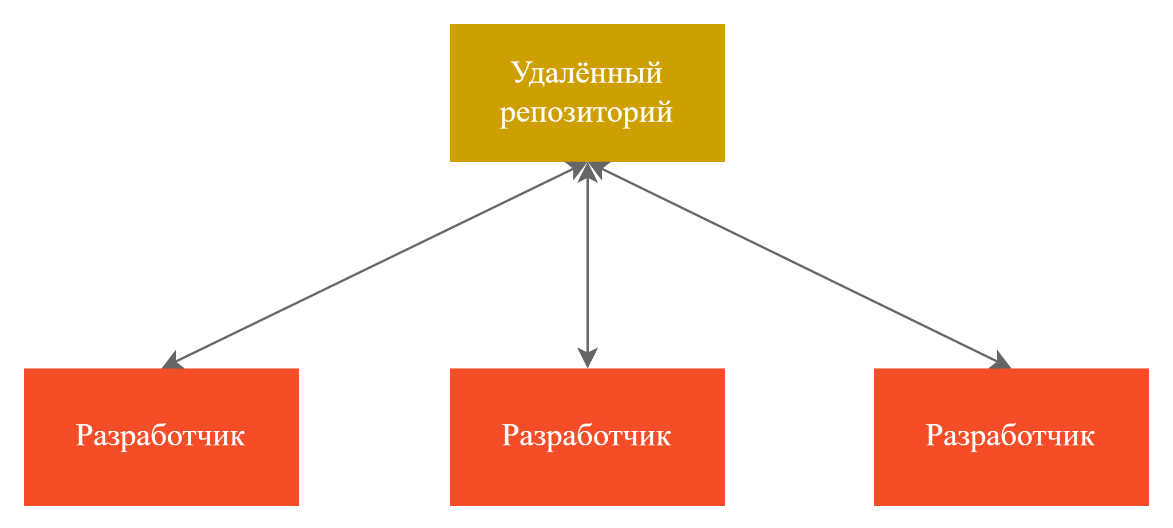


Рисунок 2 – Принцип работы удалённого репозитория

Отношение локальных репозиториев к удалённому репозиторию на Рисунке 3, где показан принцип взаимодействия центрального сервера, который хранит общую базу данных версий файлов и является точкой синхронизации для всех участников, а также локальных рабочих станций, которые имеют полную копию репозитория и локальную базу данных версий, которая отображает актуальную версию файлов.

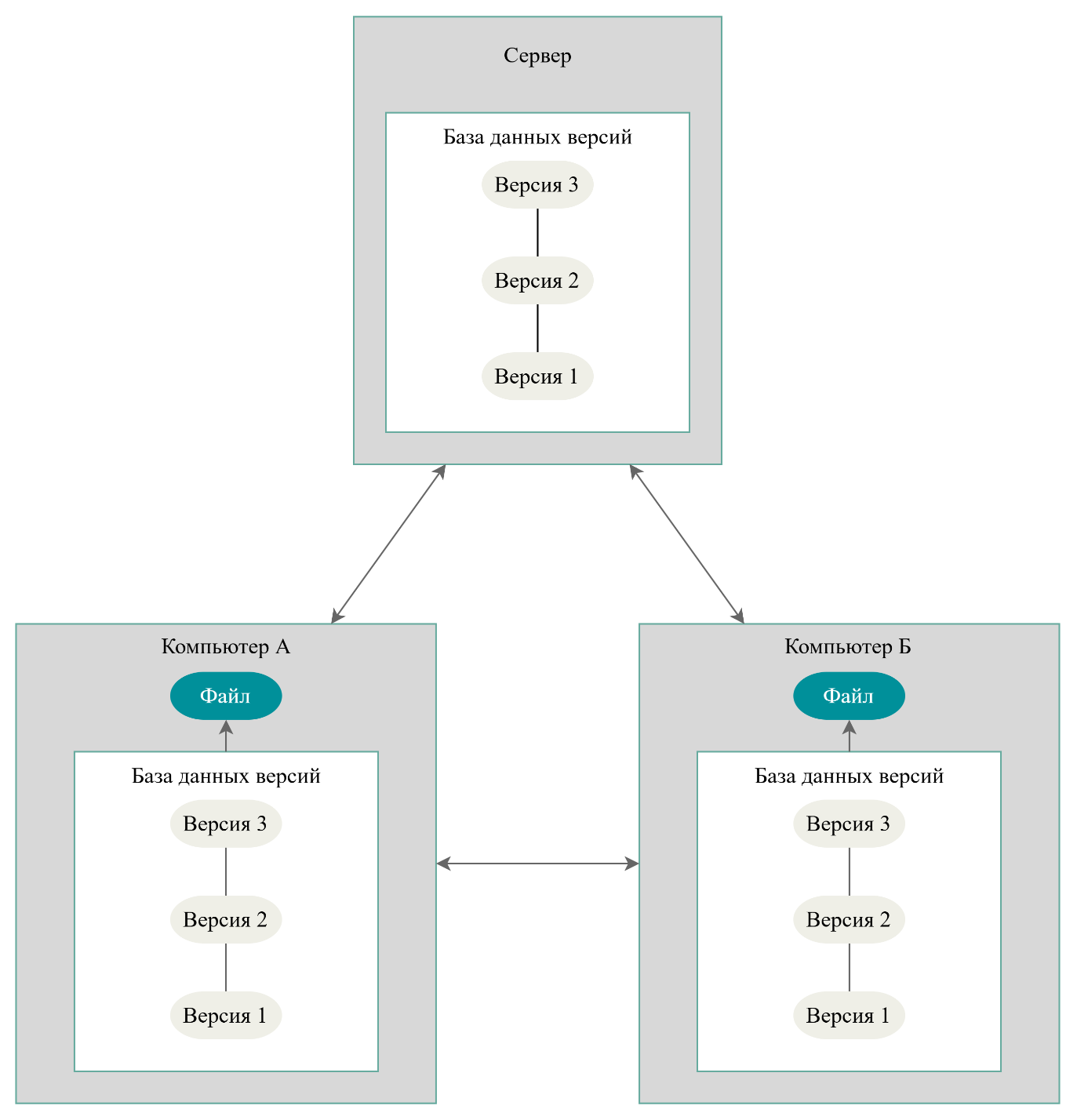


Рисунок 3 – Отношение локальных репозиториев к удалённому репозиторию

Таблица 1 – Сравнение локальных и удалённых репозиториев

|  |  |  |
| --- | --- | --- |
|  | **Локальный репозиторий** | **Удалённый репозиторий** |
| **Место хранения** | На компьютере пользователя | На внешнем сервере |
| **Доступность** | Доступ только у пользователя | Доступ у всех участников проекта |
| **Назначение** | Использование в целях разработки конкретным пользователем | Совместная работа в проекте |

***Commit***

*Commit* (коммит)– это способ хранения изменений в коде. Каждый коммит хранит в себе информацию об изменениях в проекте и об их создателе. Этот инструментарий позволяет разработчику отслеживать свои и чужие изменения в проекте.

Коммит несёт в себе информацию:

1. Комментарий о внесённом изменении.
2. Время создания коммита.
3. Данные об авторе коммита.
4. Ссылка на родительский (предыдущий) коммит или коммиты, если это слияние. Первый коммит в репозитории не имеет родителя.
5. Уникальный хэш-код коммита.
6. Тег, который помечает определенную точку репозитория как важную. У коммита не обязательно должен быть тег.

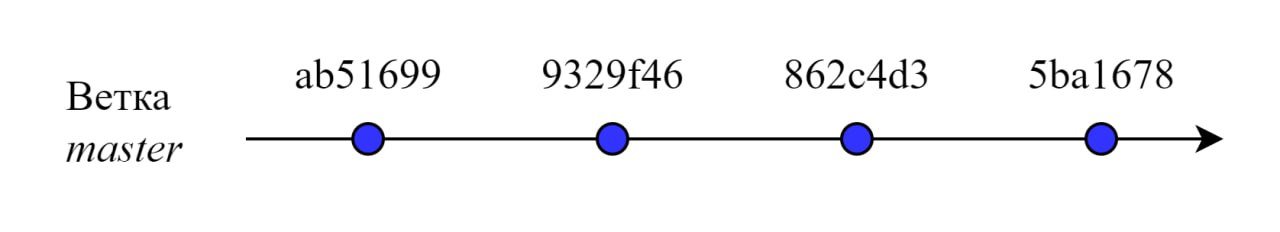
Пример коммитов в тайм-лайне ветки *master* на Рисунке 4:

Рисунок 5 – Коммиты в тайм-лайне ветки *master.*

***Branch***

*Branch* (ветка) – отдельная линия разработки, позволяющая вести работы над изменениями проекта, не затрагивая основную ветку (*master*).

Ветка (*master*) – основная ветка проекта, в которую должны попадать готовые изменения из дочерних веток.

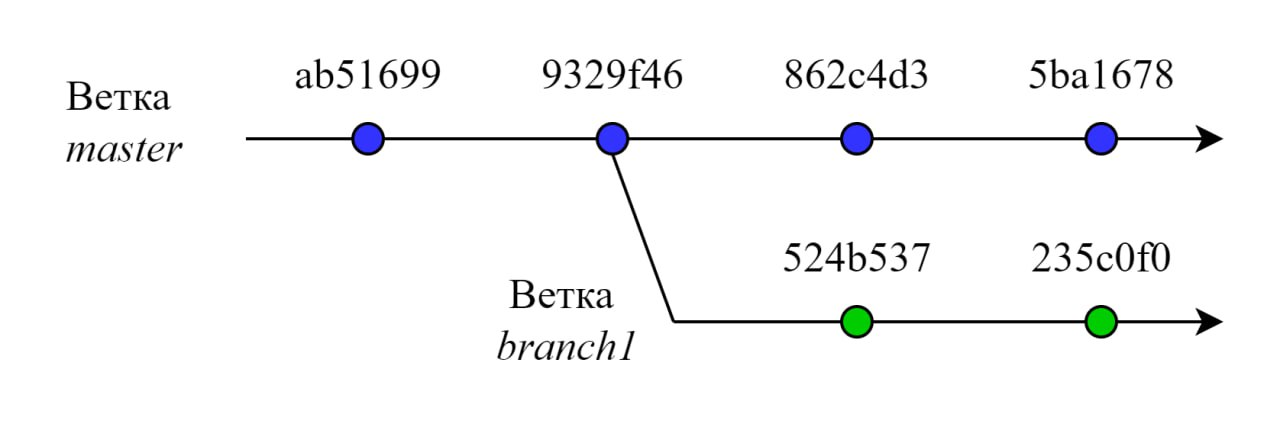
Пример структуры репозитория с разными ветками на Рисунке 6:

Рисунок 6 – Структура репозитория с разными ветками

***Merge***

*Merge* (слияние) – это процесс объединения изменений из одной ветки в другую. Слияние позволяет интегрировать разработку, выполненную в отдельных ветках, обратно в основную ветку разработки.

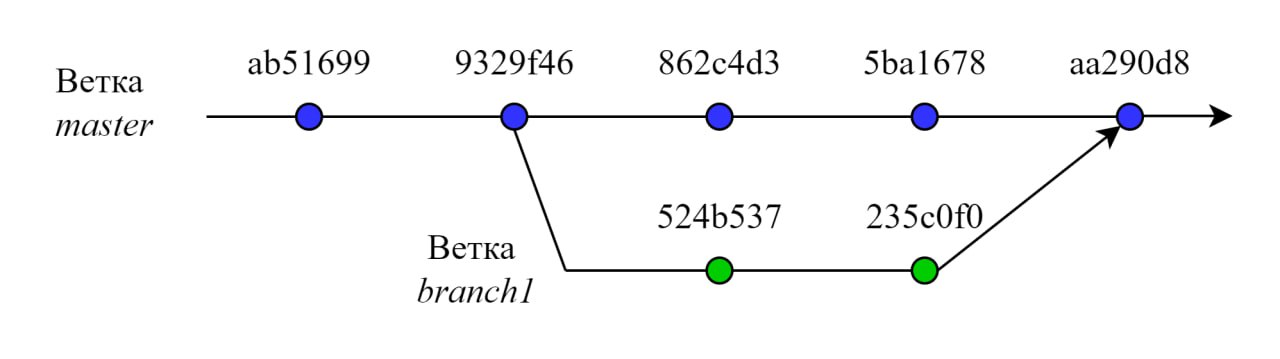
Пример слияния веток в репозитории на Рисунке 7:

Рисунок 7 – Слияние веток в репозитории

***Remote***

*Remote* (ссылка на удалённый репозиторий) - то псевдоним (или короткое имя), которое указывает на адрес удалённого репозитория. По умолчанию *GIT* называет основной удалённый репозиторий *origin*. Каждый удалённый репозиторий имеет уникальный *URL*, по которому можно получить доступ к информации в репозитории