Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2**

**дисциплины «Программирование на Python»**

**Вариант**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Говоров Егор Юрьевич  2 курс, группа ИВТ-б-о-22-1,  09.03.02 «Информатика и вычислительная техника», направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем», очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики:  Воронкин Р А., канд. технических наук, доцент кафедры инфокоммуникаций  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2023 г.

Тема: Исследование возможностей Git для работы с локальными репозиториями

Цель работы: исследовать базовые возможности системы контроля версий Git для работы с локальными репозиториями.

Ход работы:

1. Ввел команду gitlog

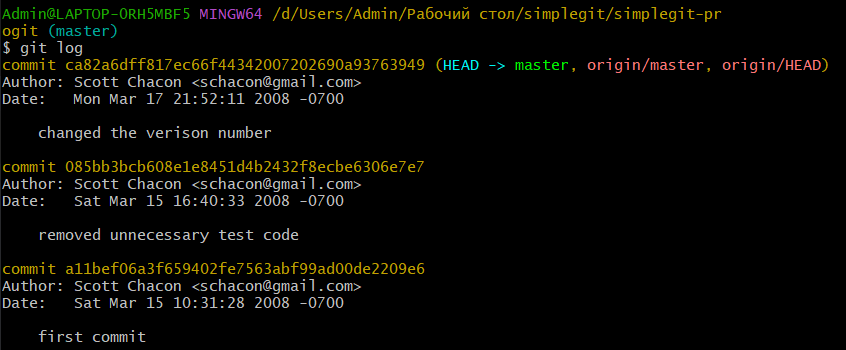


Рис 1.Вывод команды gitlog

1. Ввел дополнительные параметры команды gitlog

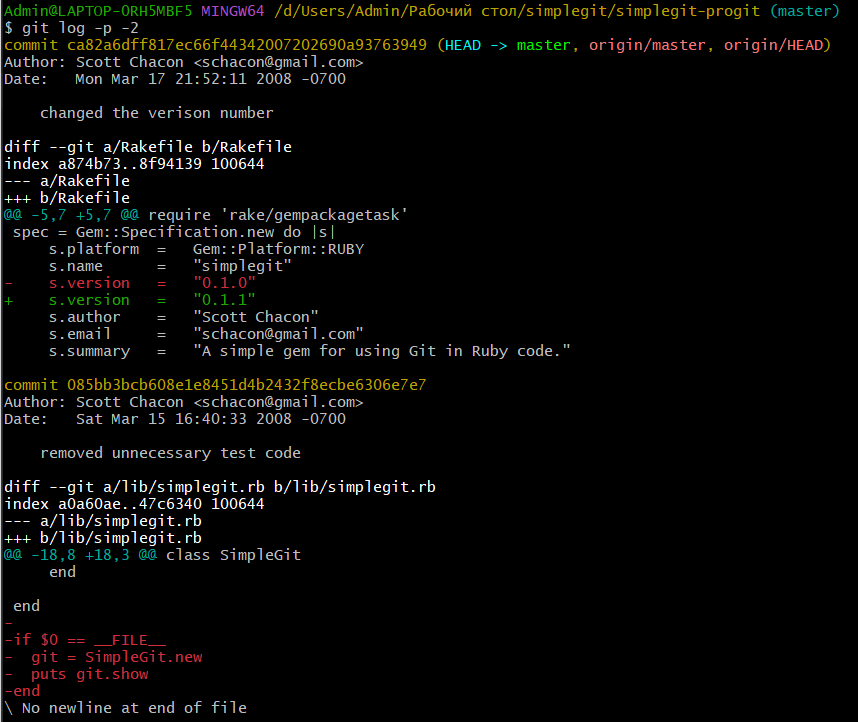


Рис 2. Вывод команды gitlog –p -2

1. Сокращенная история всех коммитов

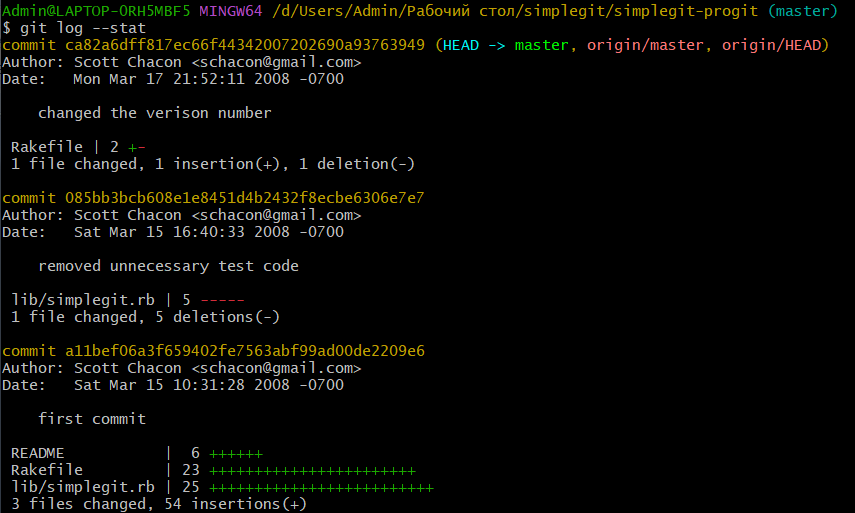


Рис 3. Вывод статистики

1. Изменение формата вывода

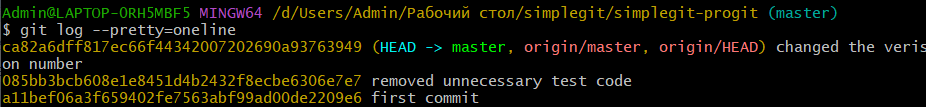


Рис 4. Строчный формат

1. Указание формата вывода

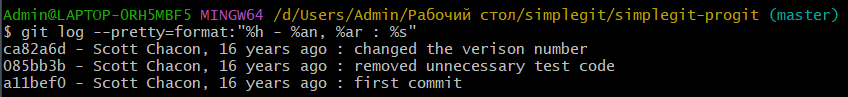


Рис 5.Формат вывода

1. Просмотр удаленных репозиториев

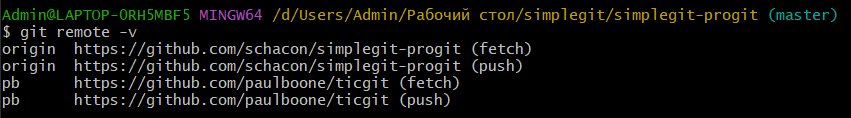


Рис 6. Просмотр адреса для чтения и записи, привязанные к репозиторию

1. Удаленное получение изменений, которых нет у вас

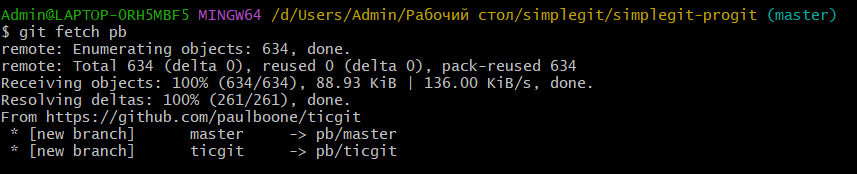


Рис 7. Применение команды git fetch

1. Просмотр удаленного репозитория

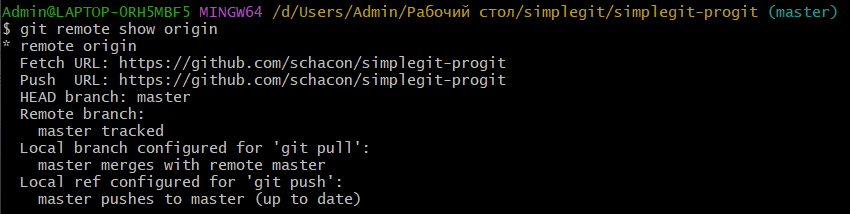


Рис 8. URL удаленного репозитория и его ветки

1. Переименование удаленного репозитория



Рис 9. Переименование репозитория pb в paul

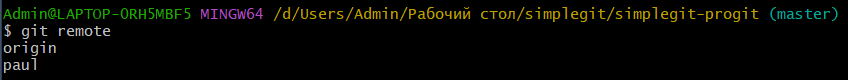


Рис 10. Удаление удаленного репозитория

1. Создание тэгов

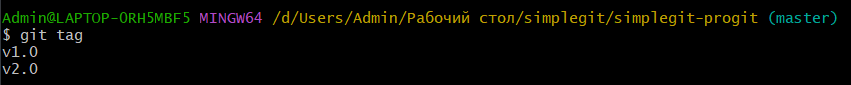


Рис 11. Просмотр созданных тегов

1. Удаленные тэги



Рис 12. Удалил тэг v1.0

1. Переход на тэг

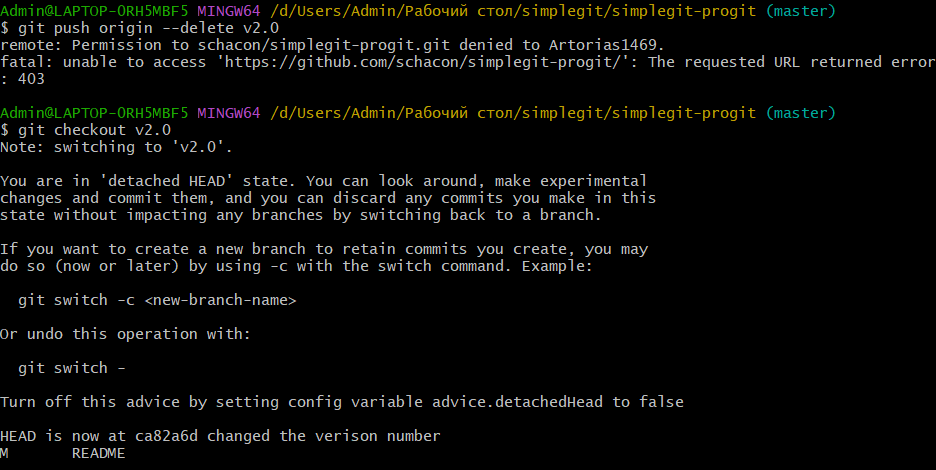


Рис 13. Получение версии файла с указанием его тэга

1. Создал 3 тэга, которыми отметила 3 новых коммита

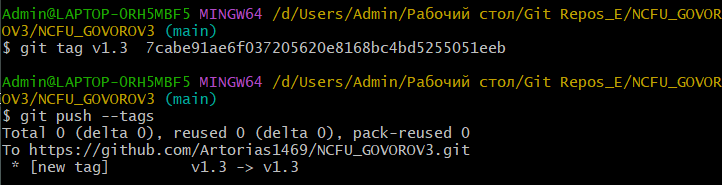


Рис 14. Применил тэг к коммитам, по их хэшу

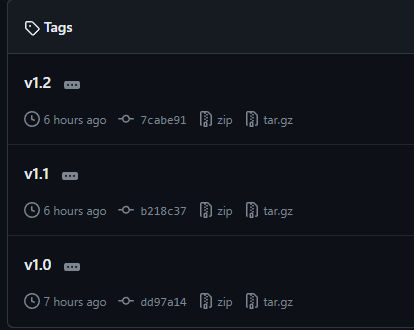


Рис 15. Созданные тэги

1. История создания коммитов

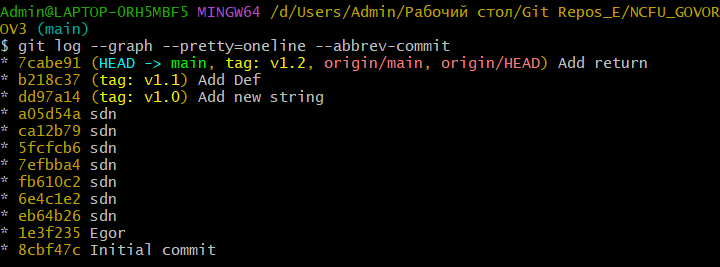


Рис 16.

Применение команды “git log --graph --pretty=oneline --abbrev-commit”

1. Просмотр содержания коммитов

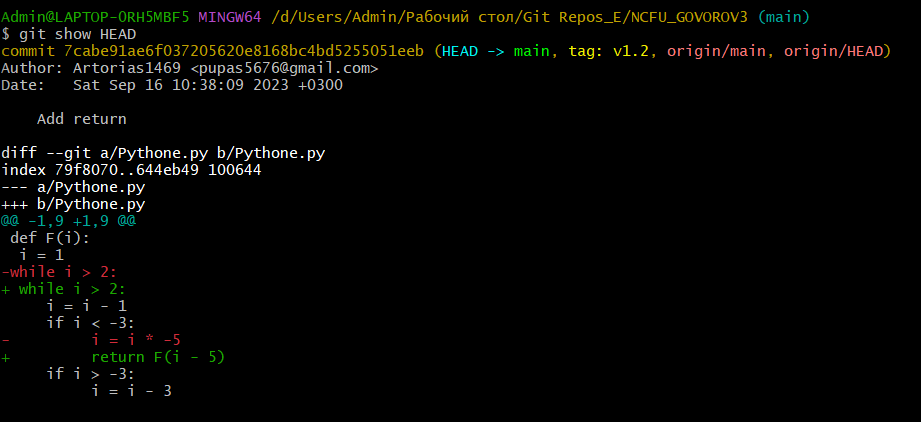


Рис 17. Просмотр последнего коммита

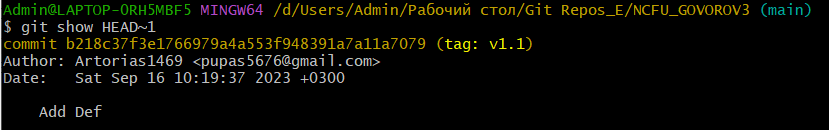


Рис18. Просмотр предпоследнего коммита

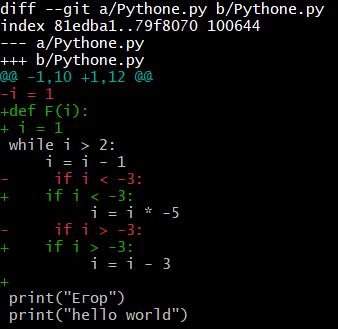


Рис 19. Просмотр предпоследнего коммита

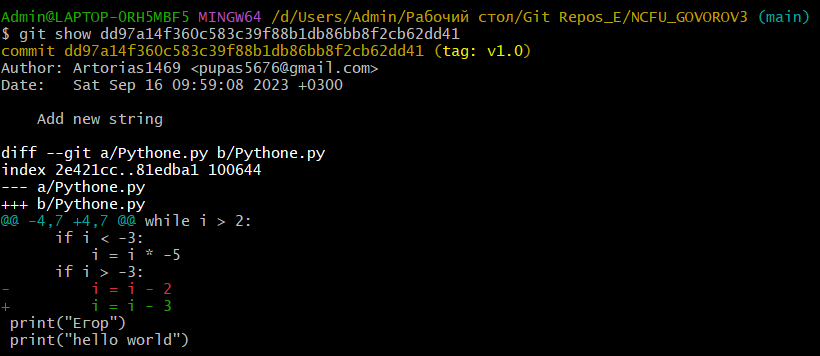


Рис 20. Просмотр коммита с указанным хэшэм

1. Откат к заданной версии

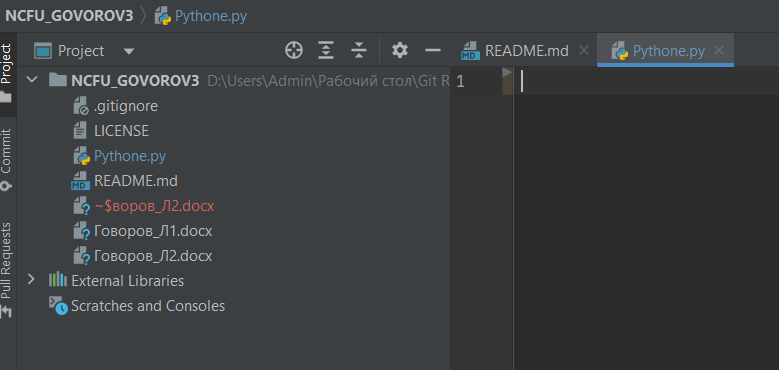


Рис 21. Удалил и сохранил код

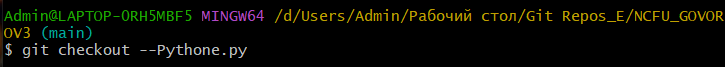


Рис 22. Удаление всех несохраненных изменений

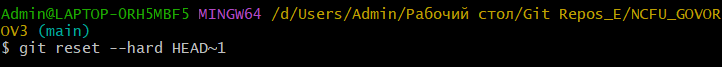


Рис 23. Откат к пред. версии

1. Создания зеркала (форк) на BitBucket

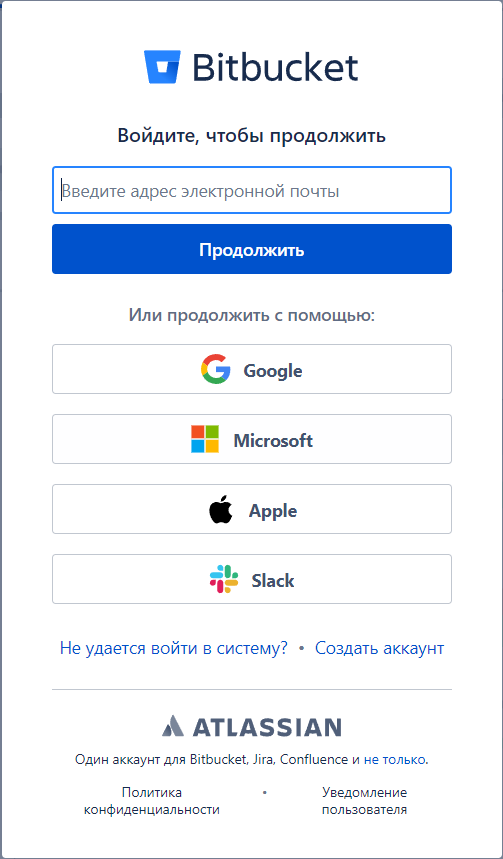


Рис 24. Создал аккаунт на Bitbucket

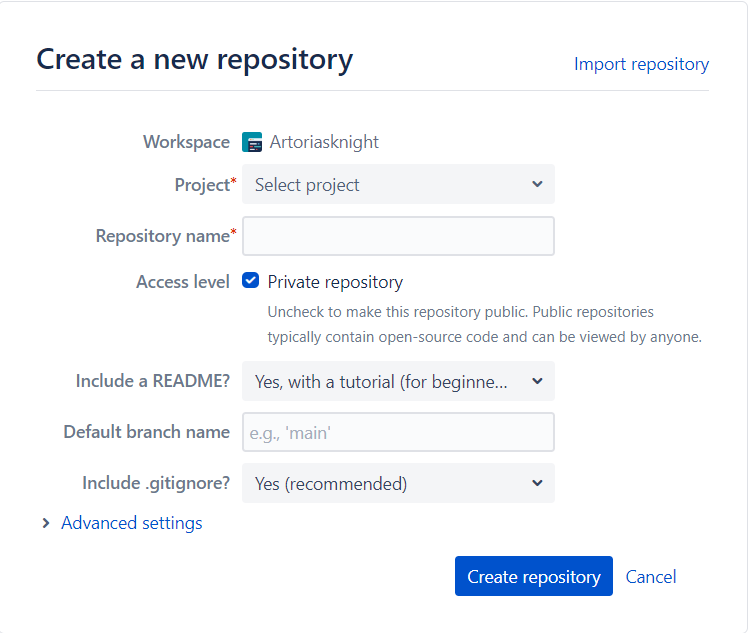
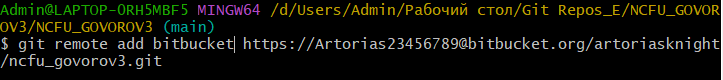


Рис 25. Создал репозиторий



Рис 26. Создал еще один клон репозитория с GitHub

Рис 27. Добавил BitBucket как удаленный репозиторий

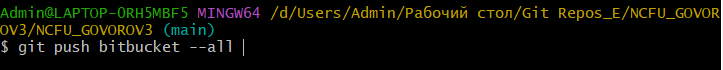


Рис 28. Отправил изменения в репозитоий Bitbucket

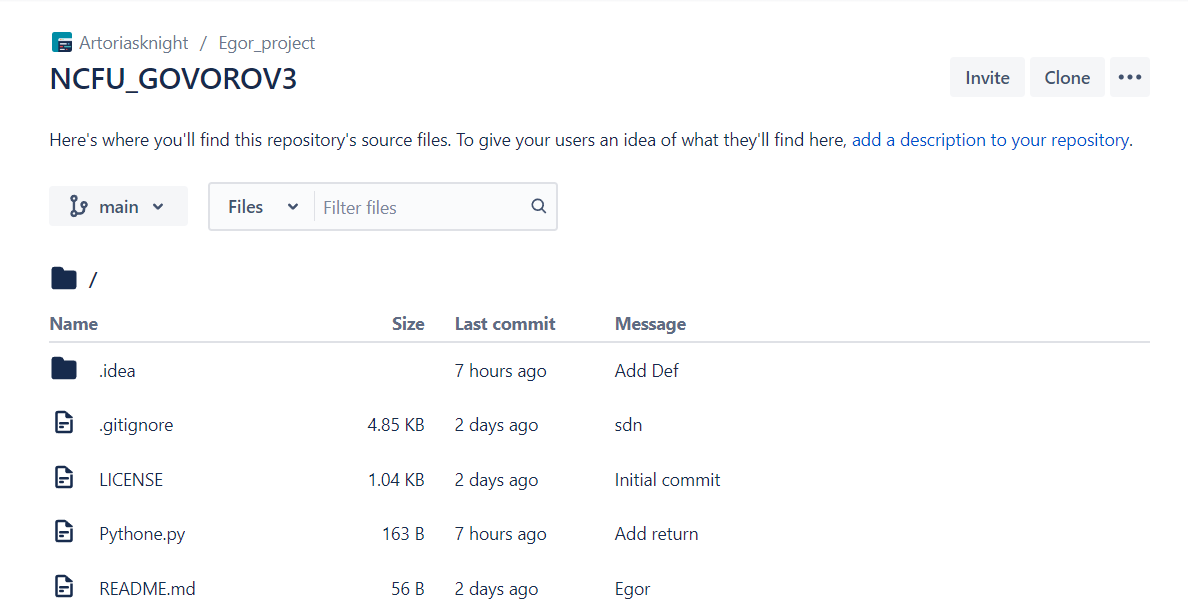


Рис 29. Перенес данные из репозитория

Ответы на вопросы:

1. Простой запрос коммитов git log. Просмотр истории коммитов с ограничением количества выводимых записей: git log -n <количество>. Просмотр истории коммитов в графическом виде: git log –graph. Просмотр истории коммитов с краткой статистикой изменений: git log –stat. Просмотр истории коммитов для определенного файла или каталога: git log <путь к файлу или каталогу>.
2. Например $ git log --since=2.weeks
3. Внесите необходимые изменения в своем рабочем каталоге.

Используйте команду git add для добавления измененных файлов в индекс.

Выполните команду git commit --amend для объединения изменений с последним коммитом. Может открыться текстовый редактор для изменения сообщения коммита.

Сохраните изменения и закройте текстовый редактор.

1. git reset <имя файла>

Эта команда уберет указанный файл из индекса (он будет снова внесен в состояние "неотслеживаемых изменений").

После выполнения этой команды файл больше не будет включен в следующий коммит, но останется в вашем рабочем каталоге с текущими изменениями.

1. git checkout -- <имя файла>

Эта команда отменяет все несохраненные изменения в указанном файле и восстанавливает его к последней зафиксированной версии.

1. Удаленный репозиторий Git (Remote Git Repository) - это репозиторий Git, который расположен на удаленном сервере или в другом месте и используется для совместной работы между разработчиками. Удаленный репозиторий содержит полную или частичную копию проекта и обеспечивает доступ к исходному коду, так что несколько разработчиков могут совместно работать над проектом, синхронизировать изменения и делиться ими.
2. git remote -v

Эта команда отобразит список удаленных репозиториев (их имен и URL-адреса), связанных с вашим локальным репозиторием, а также покажет, какие операции (fetch и push) могут выполняться с каждым из них.

1. git remote add <имя> <URL-адрес>

* Отправка изменений (push):

git push <имя удаленного репозитория> <имя вашей ветки>

git push origin main

* Получение изменений (fetch):

git fetch <имя удаленного репозитория>

git fetch origin

* Получение и объединение изменений (pull):

git pull <имя удаленного репозитория> <имя вашей ветки>

git pull origin main

1. git remote show <имя удаленного репозитория>

Здесь <имя удаленного репозитория> - это имя удаленного репозитория, который вы хотите просмотреть. Например, если удаленный репозиторий называется "origin", выполните:

git remote show origin

1. Обозначение версий: Тэги могут использоваться для указания версий вашего проекта, что облегчает отслеживание, какие версии были выпущены и когда.

Закрепление важных коммитов: Вы можете создавать тэги для фиксации важных моментов в разработке, таких как релизы или стабильные версии.

Улучшение навигации: Тэги помогают легко перемещаться между конкретными коммитами, особенно когда нужно быстро вернуться к определенной версии кода.

Облегчение совместной работы: Тэги позволяют разработчикам быстро идентифицировать и использовать конкретные моменты в истории проекта, что упрощает совместную работу.

1. git tag просмотр всех тэгов

git tag -a <имя тэга> -m "Сообщение" Создание аннотированного тэга

git tag <имя тэга> Создание легковесного тэга

1. Обновление информации о удаленных ветках: Когда вы выполняете git fetch, Git извлекает изменения из удаленного репозитория, но он также сохраняет информацию о удаленных ветках в локальном репозитории. Однако, если в удаленном репозитории удалили какую-то ветку, то она всё равно будет видна в локальном репозитории после git fetch. Команда git fetch --prune удаляет информацию о удаленных ветках из локального репозитория, чтобы он соответствовал состоянию удаленного репозитория.

Удаление локальных веток, которых больше нет на удаленном сервере: Если вы работаете с отслеживаемыми ветками (ветками, которые отслеживаются от удаленных), и ветка была удалена на удаленном сервере, то при использовании git fetch --prune также будет удалена и соответствующая локальная ветка.

Отправка изменений на удаленный сервер: Эта команда отправляет все изменения из вашего локального репозитория на удаленный сервер. Она обновляет удаленный репозиторий так, чтобы он соответствовал вашему локальному состоянию.

Удаление удаленных веток на удаленном сервере: Опция --prune (или -p в сокращенной форме) указывает Git на удаление удаленных веток на удаленном сервере. Это означает, что если какие-то ветки были удалены в вашем локальном репозитории (и, следовательно, больше не существуют), то эти удаленные ветки также будут удалены на удаленном сервере, чтобы поддерживать обе стороны синхронизированными.