

Лабораторная работа №1. Работа с git

Выполнил:

Серенко Данил Сергеевич

НФИБд-03-19, 1032192859, РУДН, 2022

Тема

Знакомство с основными возможностями git

Выполнение

Подготовка

- Прежде всего нужно настроить окружение, для удобной и корректной работы с git
 - `git config --global user.name "Your Name"`
 - `git config --global user.email "your_email@whatever.com"`
 - `git config --global core.autocrlf input`
 - `git config --global core.safecrlf true`
 - `git config --global core.quotepath off`

Создание репозитория и сохранение изменений

- Чтобы создать git репозиторий из пустого каталога, применим команду `git init`.
- Использовали команду `git status`, чтобы проверить текущее состояние репозитория.
- Выполнили команду `git add`, чтобы проиндексировать изменения.
- Чтобы сделать коммит воспользовались командой `git commit -m`.
- Получили список произведенных изменений командой `git log`.
- Для возвращения назад в историю использовали команду `checkout`, которая копирует любой снимок из репозитория в рабочий каталог.
- Создал тег первой версии командой `git tag v1`, для проверки доступных тегов использовал команду `git tag`.

Отмена нежелательных изменений и коммитов

- Для отмены проиндексированных изменений (перед коммитом) команда `git reset`, которая сбрасывает буферную зону к HEAD. Это очищает буферную зону от изменений, которые мы только что проиндексировали.
- Для отмены коммита, можно создать новый коммит, который удаляет изменения, сохраненные нежелательным коммитом. Воспользовались командой `git revert HEAD`.

- Для сброса коммита используем команду `git reset --hard v1`. Параметр `--hard` указывает, что рабочий каталог должен быть обновлен в соответствии с новым head ветки.
- Чтобы удалить тег и коммиты, на которые он ссылался, воспользуемся сборщиком мусора `git tag -d oops`.

Изменение коммита

- Чтобы изменить предыдущий коммит воспользуемся командой `git commit --amend`.

Структура репозитория и работа с ней

- Для перемещения файлов есть команда `git mv`. Она удаляет файл из начального каталога, создает этот файл в новом каталоге и эти факты сразу проиндексированы и готовы к коммиту.
- Вся информация git находится в каталоге `.git`. Просмотреть можно командой `ls -C .git`.
- С помощью команды `git cat-file` можно выводить определенные коммиты или каталоги по их хэшу.

Ветки

- Для создания новой ветки используем команду `git checkout -b name_branch`
- Для переключения между ветками `git checkout name_branch`
- Для слияния веток используем команду `git merge`.
- При возникновении конфликтов в изменениях, разрешали конфликты вручную.
- Чтобы посмотреть ветки в репозитории воспользовались командой `git branch`. Для отображения удаленных веток добавили метку `-a`.

Клонирование репозитория

- Для создания клонированного репозитория воспользовались командой `git clone`.

Работа с удаленными репозиториями

- Для извлечения новых коммитов из удаленного репозитория используем команду `git fetch`. Но она не сливает их с наработками в локальных ветках.
- Для извлечения новых коммитов из удаленного репозитория и сливания изменения, воспользовались командой `git pull`.
- Для отслеживания удаленной ветки, можем добавить локальную ветку командой `git branch --track`.
- Для создания чистого репозитория, то есть репозитория, в котором нет рабочих каталогов, используем команду `git clone --bare`.
- Чтобы добавить репозиторий к оригинальному репозиторию, воспользовались командой `git remote add shared`.
- Для отправки изменений в общий репозиторий использовали команду `git push shared master`.

Вывод

Ознакомились с основными возможностями git