БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ТРАНСПОРТНЫХ КОММУНИКАЦИЙ

КАФЕДРА «ГЕОДЕЗИЯ И АЭРОКОСМИЧЕСКИЕ ГЕОТЕХНОЛОГИИ»

ОТЧЕТ

по лабораторной работе №4 «Система контроля версий Git»

Выполнил:

студ. гр. 11405117

Таркан А.В.

Проверил:

Будо А.Ю., ст. преп.

Минск, 2021

Часто разработчики трудятся в команде над одним проектом, а значит, сразу несколько человек могут изменять один файл одновременно. Чтобы избежать путаницы, в таких случаях используют систему контроля версий, которая позволяет хранить историю изменений проекта и при необходимости помогает вернуться к предыдущей версии.

Система контроля версий ­ это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.

Простое копирование проекта в другой каталог имеет множество недостатков: избыточность, дублирование, отсутствие истории изменений.

Для решения проблемы взаимодействия разработчиков существуют распределенные системы контроля версий, такие как Git, суть которой заключается в том, что клиент полностью копирует весь репозиторий. Каждый раз клиент создает себе новую версию файла

Одна из самых популярных ­ система Git. Работу Git можно сравнить с процессом сохранения и загрузки.

Git ­ это распределённая и децентрализованная система управления версиями файлов. Децентрализованная система означает, что у каждого разработчика есть личный репозиторий проекта с полным набором всех версий. Репозиторий хранит все рабочие файлы и историю их изменений. Все необходимые для работы файлы находятся на компьютере. При этом постоянное подключение к сети не требуется. При командной разработке нужна синхронизация репозиториев, так как проект ­ один и его состояние должно быть у всех одинаковым.

В итоге проект работает так:

* репозиторий хранит все версии проекта; случае передачи этого проекта другому человеку, он увидит всё, что с ним происходило до этого.
* ничего не теряется и не удаляется бесследно; при удалении файла в новой версии добавляется запись о том, что файл был удалён;
* всегда можно вернуться к любой из версий проекта, загрузив её из хранилища в рабочую область.

Для использования системы контроля версий Git нужно:

1. Установить программу git на системе.
2. Настроить программу.
3. Создать локальный репозиторий или копировать репозиторий существующего проекта.
4. Отправлять изменения на GitHub.

Клонирование репозитория осуществляется командой

git clone <url>

При выполнении git clone с сервера забирается каждая версия каждого файла из истории проекта.

В системе контроля версий Git можно сохранить текущее состояние проекта. Для этого есть специальная команда commit. Она делает так, что новая версия проекта сохраняется и добавляется в хранилище. В файле с сохранением отображаются: все изменения, которые происходили в рабочей области, автор изменений и краткий комментарий, описывающий суть изменений. Каждый коммит хранит полное состояние рабочей области, её папок и файлов проекта.

Создание коммита имеет следующую последовательность:

* добавления файлов в коммит:

git add [file]

* создание коммита из выбранных файлов:

git commit

для добавления комментария к коммиту :

git commit –m “Система контроля версий Git”

Для проверки изменений, внесенных в коммит:

git show

После создания коммита, следует поделиться им в удаленный репозиторий используя git push. Таким образом все изменения будут доступны другим пользователям.

Перед этим следует убедиться, что файлы репозитория соответствуют последнему обновлению командой git pull. Это нужно для предотвращения конфликтов файлов в локальном и удаленном репозитории.

Для отправления коммита в удаленный репозиторий:

git push

Если все сделано верно, то отправленные файлы дополнят содержимое удаленного репозитория.