

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ТРАНСПОРТНЫХ КОММУНИКАЦИЙ
КАФЕДРА «ГЕОДЕЗИЯ И АЭРОКОСМИЧЕСКИЕ ГЕОТЕХНОЛОГИИ»

ОТЧЕТ

по лабораторной работе №5 «Командный проект с использованием Git»

Выполнил:
студ. гр. 11405117
Бурак А.А.

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.

Одна из самых популярных систем 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 репозитория, 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 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.

1. Установка git

В Linux установка выполняется из терминала через следующие команды:

```
sudo apt-get install git
```

Для вывода текущей версии git в Linux следует ввести:

```
git -version
```

В случае успешной установки в консоль выведется версия git.

2. Настройка git

Первое, что вам следует сделать после установки Git — указать имя и адрес электронной почты, использованные при регистрации на GitHub. Это важно, потому что каждый коммит в Git содержит эту информацию, и она включена в коммиты, передаваемые вами, и не может быть далее изменена:

```
git config --global user.name
```

```
git config --global user.email myEmail@example.com
```

Для проверки используемой конфигурации можно использовать команду

```
git config --list
```

Эти настройки достаточно сделать только один раз, поскольку в этом случае Git будет использовать эти данные для всех проектов в этой системе.

3. Копирование существующего репозитория

Клонирование репозитория осуществляется командой

```
git clone <url>
```

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

4. Поделиться изменениями на GitHub

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 дополняют содержимое удаленного репозитория.

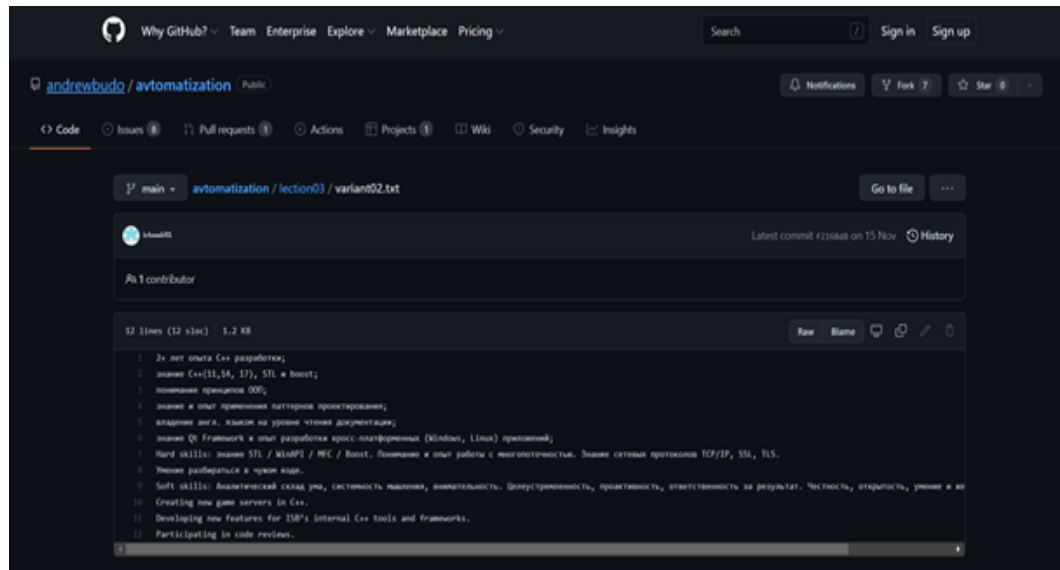


Рисунок 1. Отправленный файл в общий проект