Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
дисциплины
«Основы кроссплатформенного программирования»

Выполнил:
Ляш Денис Александрович
2 курс, группа ИТС-б-о-23-1,
11.03.02 «Инфокоммуникационные технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и сети»,
очная форма обучения

_____
(подпись)

Проверил:
Воронкин Р.А.

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2024 г.

Тема: Исследование основных возможностей Git и GitHub

Цель: исследовать базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.

Порядок выполнения работы:

1. Изучил и законспектировал основные сведения.

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

Когда вы производите какие-либо действия в Git, практически все из них только добавляют новые данные в базу Git. Очень сложно заставить систему удалить данные либо сделать что-то, что нельзя впоследствии отменить. Как и в любой другой СКВ, вы можете потерять или испортить свои изменения, пока они не зафиксированы, но после того, как вы зафиксируете снимок в Git, будет очень сложно что-либо потерять, особенно, если вы регулярно синхронизируете свою базу с другим репозиторием. Всё это превращает использование Git в одно удовольствие, потому что мы знаем, что можем экспериментировать, не боясь серьёзных проблем

У Git есть три основных состояния, в которых могут находиться ваши файлы: зафиксированное (committed), изменённое (modified) и подготовленное (staged).

Зафиксированный значит, что файл уже сохранён в вашей локальной базе.

К изменённым относятся файлы, которые поменялись, но ещё не были зафиксированы.

Подготовленные файлы — это изменённые файлы, отмеченные для включения в следующий коммит.

Базовый подход в работе с Git выглядит так:

1. Вы изменяете файлы в вашей рабочей директории.

2. Вы выборочно добавляете в индекс только те изменения, которые должны попасть в следующий коммит, добавляя тем самым снимки только этих изменений в область подготовленных файлов.

3. Когда вы делаете коммит, используются файлы из индекса как есть, и этот снимок сохраняется в вашу Git-директорию.

2. Начало работы с Git

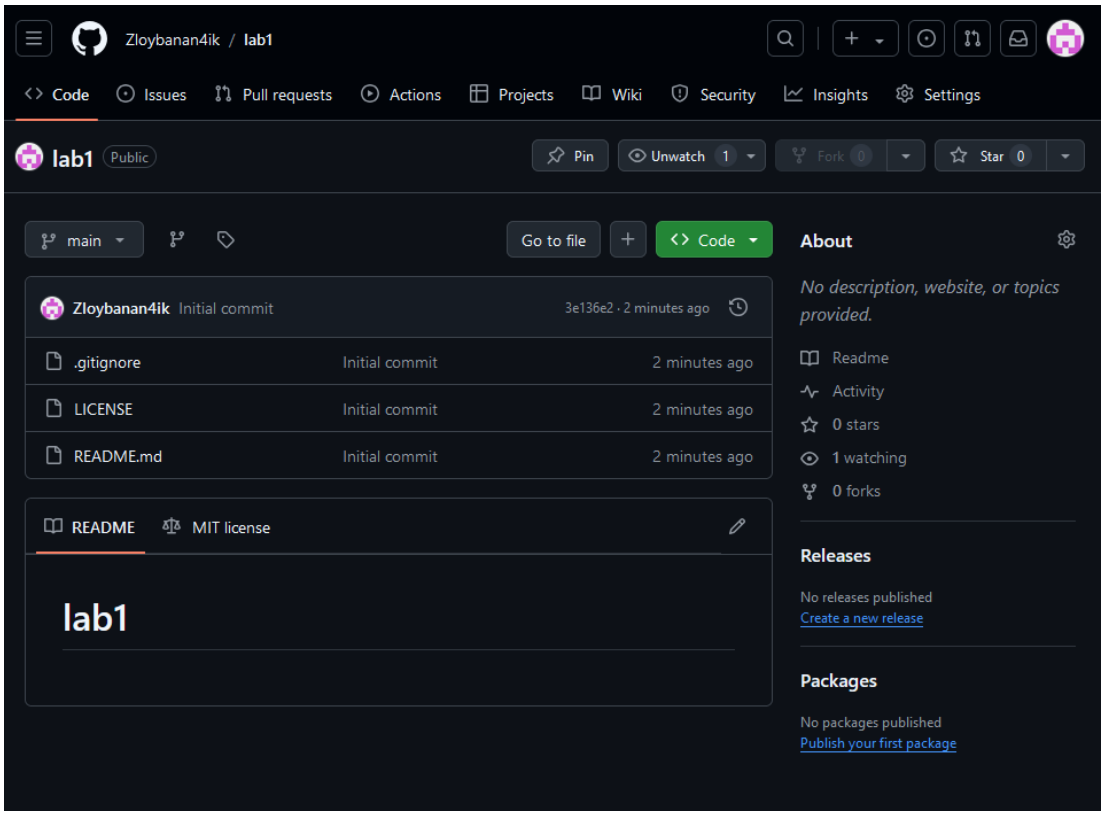

Рис. 1 Создал репозиторий, используя GitHub:

Добавил в настройки git свое имя и адрес электронной почты, связанные с моей учетной записью в GitHub:

```
user@DESKTOP-VI4I94J MINGW64 ~/workspace
$ git config --global user.name Zloybanan4ik

user@DESKTOP-VI4I94J MINGW64 ~/workspace
$ git config --global user.email denik55557777809@gmail.com
```

Рис.2. – Добавление своих данных в настройки git

Перешел в каталог и клонировал в него хранилище, используя адрес:

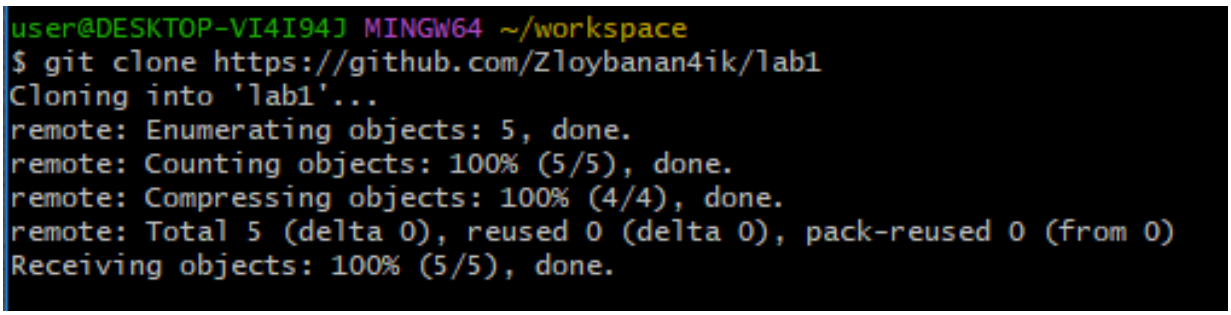

Рис.3 – Клонирование репозитория

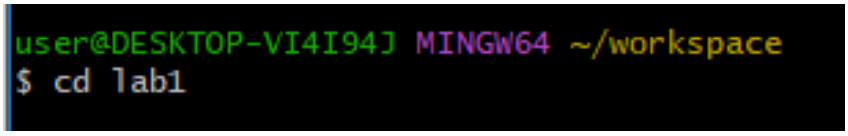

Рис. 4 переход в папку репозитория

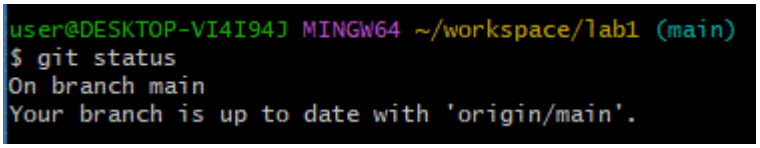

Рис. 5 проверка статуса git

Внес изменения в файл Readme.md

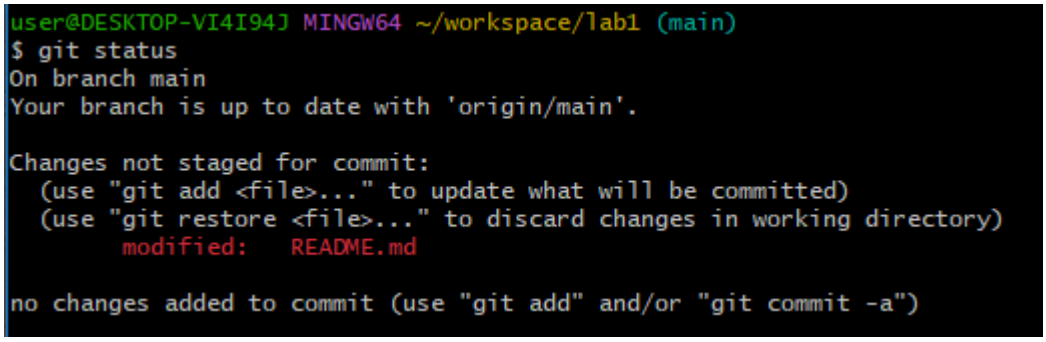

Рис. 6 внесенные изменения в файл README.md

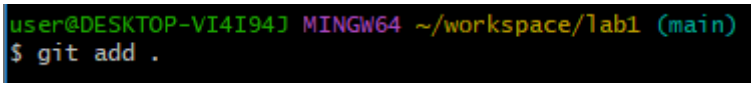

Рис. 7 подключение версионного контроля для всех файлов

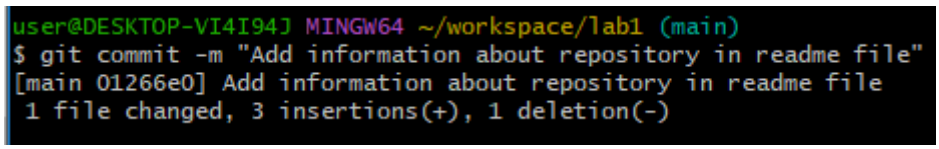

Рис. 8 фиксация изменений в локальном репозитории

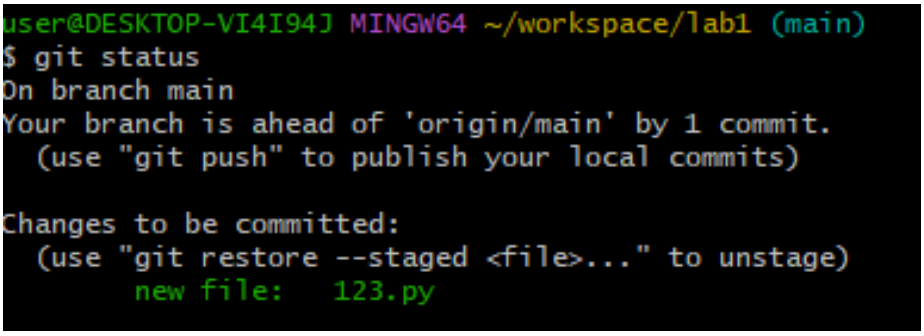

Рис. 9 результат создания файла 123.py

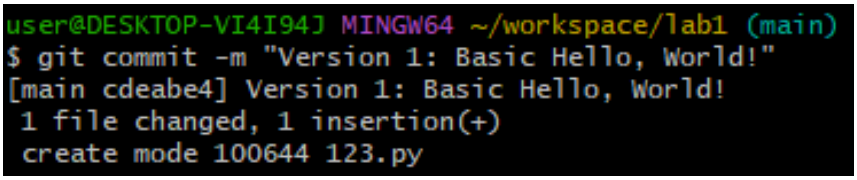

Рис. 10 первый коммит программы

Далее выполнил 7 коммитов программы, после чего загрузил ее на Github

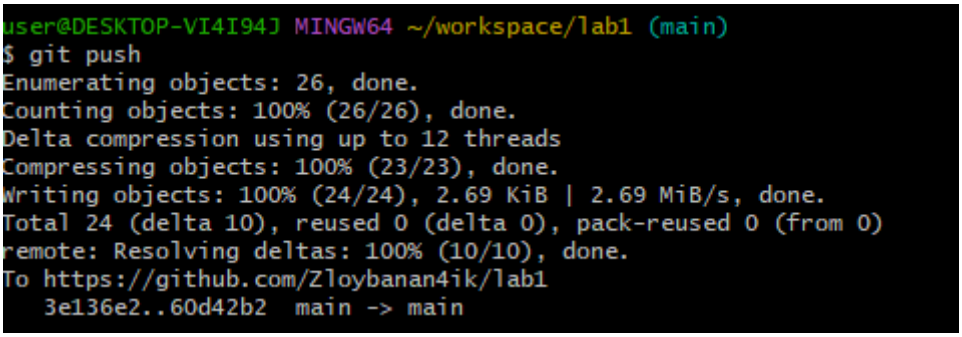

Рис. 11 загрузка локального репозитория на github

3. По завершению отчета написал вывод.

Ответы на контрольные вопросы:

1. Что такое СКВ и каково ее назначение?

Система контроля версий (СКВ) — это программное обеспечение, которое позволяет разработчикам отслеживать изменения в исходном коде, возвращаться к предыдущим версиям файлов, сравнивать их, сливать изменения, а также управлять командной работой над проектом.

2. В чем недостатки локальных и централизованных СКВ?

- Локальные СКВ: изменения отслеживаются только на одном компьютере. Это усложняет командную работу, восстановление данных в случае поломки компьютера.

- Централизованные СКВ: единственный сервер хранит все данные и историю изменений. Если сервер недоступен или поврежден, работа затрудняется.

3. К какой СКВ относится Git?

Git относится к распределенным системам контроля версий (РСКВ), где каждый разработчик имеет полную копию репозитория.

4. В чем концептуальное отличие Git от других СКВ?

Главное отличие Git заключается в его распределенной природе: каждый пользователь имеет локальную копию всего репозитория с полной историей изменений. Это делает систему более гибкой и устойчивой к сбоям, в отличие от централизованных систем.

5. Как обеспечивается целостность хранимых данных в Git?

Git использует хеширование (SHA-1) для всех объектов (файлы, каталоги, коммиты), что гарантирует целостность данных. Если данные изменены, хеш изменяется, и система сразу замечает проблему.

6. В каких состояниях могут находиться файлы в Git? Как связаны эти состояния?

Файлы могут находиться в трех состояниях:

- Измененные (modified): файл был изменен, но не добавлен в индекс.

- Проиндексированные (staged): файл добавлен в индекс (через `git add`), готов к коммиту.

- Зафиксированные (committed): изменения сохранены в истории репозитория.

Связаны эти состояния процессом работы: сначала файл изменяется, затем индексируется, и в конечном итоге фиксируется коммитом.

7. Что такое профиль пользователя в GitHub?

Профиль пользователя в GitHub — это учетная запись, содержащая информацию о пользователе (имя, контактные данные, проекты, активности), которая также позволяет управлять репозиториями и взаимодействовать с другими разработчиками.

8. Какие бывают репозитории в GitHub?

Репозитории могут быть:

- Публичные: доступны всем пользователям GitHub.

- Приватные: видны только владельцам и приглашенным пользователям.

9. Укажите основные этапы модели работы с GitHub.

1. Создание репозитория.
2. Клонирование репозитория на локальный компьютер.
3. Внесение изменений и фиксация (коммит).
4. Отправка изменений на сервер (push).
5. Получение изменений из удаленного репозитория (pull).
6. Управление ветками и слияниями (merge).

10. Как осуществляется первоначальная настройка Git после установки?

После установки необходимо настроить имя пользователя и email:

```bash
git config --global user.name "Ваше Имя"
git config --global user.email "ваш.email@example.com"
```

11. Опишите этапы создания репозитория в GitHub.

1. Войти в GitHub.
2. Нажать "New Repository".
3. Указать имя репозитория.
4. Выбрать тип репозитория (публичный или приватный).
5. Создать репозиторий.

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

GitHub поддерживает лицензии:

- MIT
- Apache 2.0
- GPL (GNU General Public License)
- BSD и др.

13. Как осуществляется клонирование репозитория GitHub? Зачем нужно клонировать репозиторий?

Клонирование осуществляется командой:

```bash
git clone https://github.com/username/repository.git
```

Клонирование необходимо для того, чтобы получить копию удаленного репозитория на локальный компьютер и начать с ним работу.

14. Как проверить состояние локального репозитория Git?

Команда:

```bash
git status
```

Показывает текущее состояние файлов: изменены ли они, проиндексированы ли и нужно ли их зафиксировать.

15. Как изменяется состояние локального репозитория Git после выполнения следующих операций?

- Добавление/изменение файла: файл переходит в состояние "modified".
- git add: файл перемещается в состояние "staged".
- git commit: файл переходит в состояние "committed".
- git push: изменения отправляются на удаленный сервер.

16. Как синхронизировать два локальных репозитория с репозиторием GitHub?

1. На первом компьютере:

```bash
git clone https://github.com/username/repository.git
```

2. На втором компьютере:

```bash
git clone https://github.com/username/repository.git
```

Для синхронизации изменений:

```bash
git pull
git add .
```

```
git commit -m "Сообщение"
git push
```

17. GitHub является не единственным сервисом, работающим с Git. Какие сервисы еще Вам известны?

Среди других сервисов:

- GitLab
- Bitbucket

GitLab, например, предоставляет более мощные функции CI/CD по сравнению с GitHub и имеет возможность установки на собственный сервер.

18. Какие Вам известны программные средства с графическим интерфейсом пользователя для работы с Git?

- GitKraken
- SourceTree
- GitHub Desktop

В GitKraken для выполнения коммита используется простая форма интерфейса, которая позволяет добавлять файлы в индекс и выполнять коммит в несколько кликов.

Вывод:

В ходе лабораторной работы были изучены основные возможности Git и GitHub. Git, как распределенная система контроля версий, обеспечивает эффективное управление изменениями и командную работу над проектами. GitHub предоставляет удобные инструменты для хостинга репозиториев и совместной работы. Были освоены основные команды Git и процессы взаимодействия с удаленными репозиториями. Работа с Git и GitHub позволяет

эффективно управлять версиями проекта, синхронизировать изменения и обеспечить надежное хранение данных.