**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ»**

Кафедра математической кибернетики и информационных технологий

Лабораторная работа №1

по дисциплине «Кроссплатформенные технологии программирования»

на тему «Основы работы с системой контроля версий»

Выполнил студент

группы БСТ2003

Чечеткин.С.А.

Проверила:

Полянцева К.А.

Москва 2022

**Оглавление**

[Цель и задание: 3](#_Toc95336418)

[Ход выполнения лабораторной работы: 3](#_Toc95336419)

[1.Создание репозитория или клонирование существующего 3](#_Toc95336420)

[2. Внесение ряда изменений в файл(ы) и выполнение коммитов (кол-во не меньше 2) 4](#_Toc95336421)

[3. Создание веток и выполнение ряда изменений в проекте в созданных ветках 7](#_Toc95336422)

[4. Слияние ветки с веткой мастер без конфликтов 8](#_Toc95336423)

[5. Слияние ветки с веткой мастер с решением возникающих конфликтов 8](#_Toc95336424)

[6. Откат изменений до одного из предыдущих коммитов 10](#_Toc95336425)

[Вывод 12](#_Toc95336426)

[Список использованных источников: 13](#_Toc95336427)

# Цель и задание:

**Цель:** Изучение базовых функций системы контроля версий на примере Git.

**Задание:** Определить проект (состоящий из нескольких файлов), который будет добавлен под версионный контроль, и произвести ряд операций в СКВ, включающих:

1. Создание репозитория или клонирование существующего
2. Внесение ряда изменений в файл(ы) и выполнение коммитов (кол-во не меньше 2)
3. Создание веток и выполнение ряда изменений в проекте в созданных ветках
4. Слияние ветки с веткой мастер без конфликтов
5. Слияние ветки с веткой мастер с решением возникающих конфликтов
6. Откат изменений до одного из предыдущих коммитов

# Ход выполнения лабораторной работы:

## Создание репозитория или клонирование существующего

Для создания репозитория необходимо создать папку, в которой будет храниться проект. После создания папки, в ней открываем Git Bash, после чего открывается консоль с путём в эту папку. Необходимо ввести команду: git init – проинициализируем папку. После чего в ней создастся папка: .git. Таким образом был создан репозиторий. Ниже приведен скриншот создания репозитория.

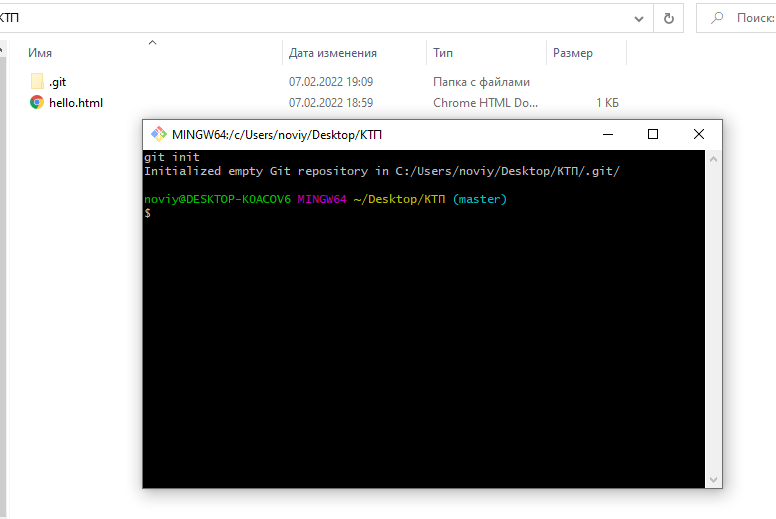


Рисунок 1 – Создание репозитория

## 2. Внесение ряда изменений в файл(ы) и выполнение коммитов (кол-во не меньше 2)

Переходим в Git и прописываем по очереди следующие команды:

git status – проверка состояние репозитория

git add <название файла> – добавление файла

git status – проверка состояние репозитория

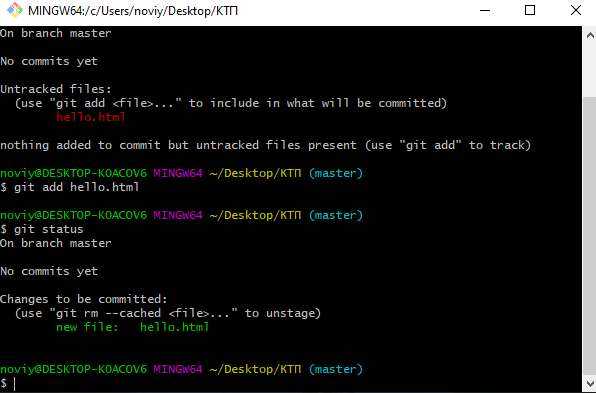


Рисунок 2 – Ввод команд

Выполнив эти команды, мы убедились, что изначально файл не отслеживался Git’ом, после чего мы добавили этот файл в контролируемую зону Git, после чего убедились, что файл был добавлен

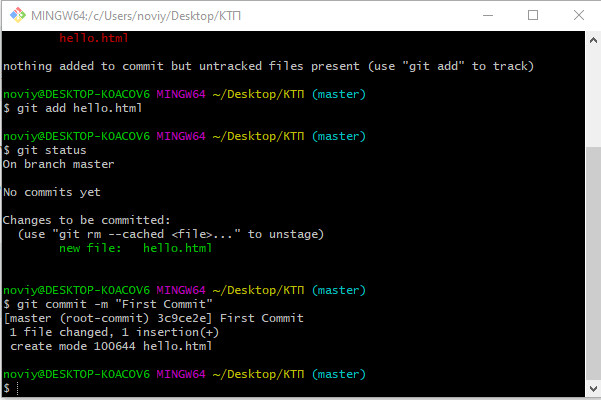


Рисунок 3 – Первый commit

Прописываем команду git commit –m «First Commit». Этой командой мы зафиксируем текущую версию файлов.

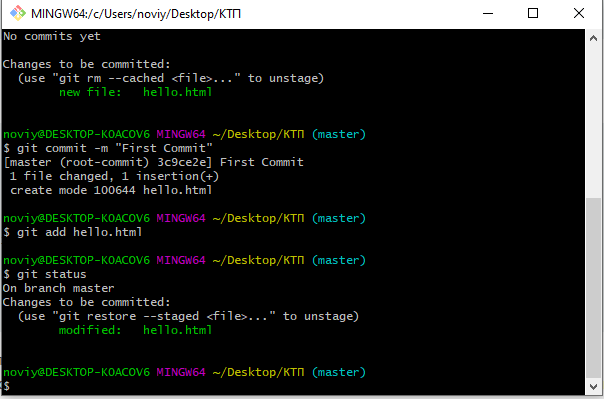


Рисунок 4 – Изменение файла

Прописываем команды:

git add <название файла> - Добавляем измененный файл в git

git status – Проверяем индексацию изменений git, но изменение пока не перманентно записано в репозиторий

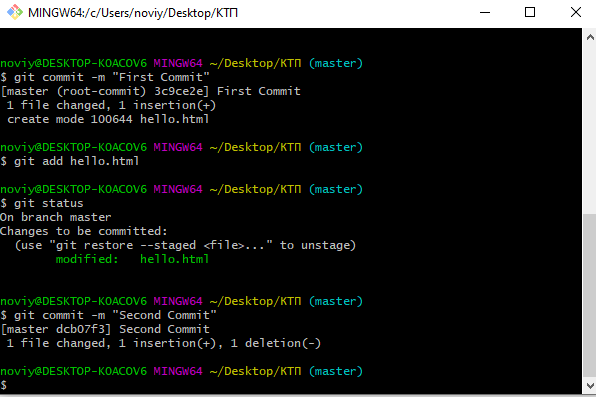


Рисунок 5 – Второй commit

Прописываем команду git commit –m «Second Commit». Этой командой мы зафиксируем текущую версию файлов.

## 3. Создание веток и выполнение ряда изменений в проекте в созданных ветках

git branch firstBranch – команда, создающая новую ветку

git branch – команда, выводящая список веток

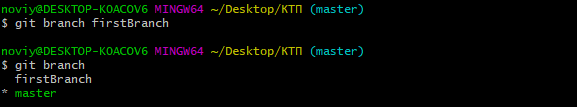


Рисунок 6 – Создание ветки

git checkout firstBranch – переход на ветку firstBranch

Создаём новый файл в папке, при этом мы должны находиться в ветке firstBranch, после чего добавляем новый файл в систему контроля Git. После чего делаем коммит.

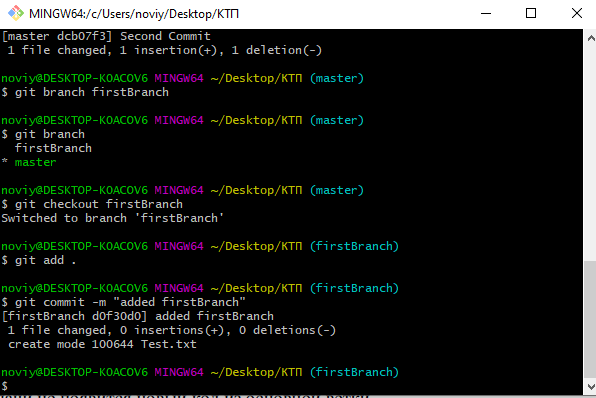


Рисунок 7 – Выполнение коммитов в ветке firstBranch

## 4. Слияние ветки с веткой мастер без конфликтов

git merge firstBranch – команда, сливающая ветки

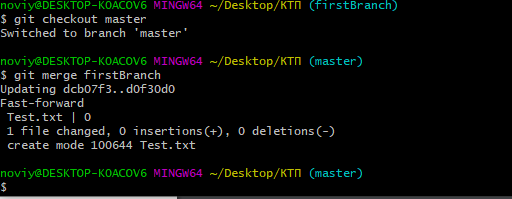
****

Рисунок 8 – Слияние веток

## 5. Слияние ветки с веткой мастер с решением возникающих конфликтов

Рассмотрим сценарий слияния с конфликтом. Для создания конфликта были проделаны следующие действия:

* Были закоммитены изменения файла Test.txt в ветке firstBranch
* Далее, переключив ветку на master, были снова сделаны изменения в файле Test.txt и сделан коммит
* Произведено слияние

После выполнения слияния, получаем ошибку следующего формата

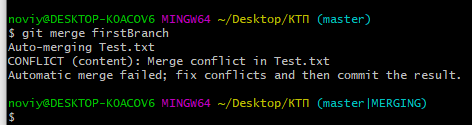


Рисунок 9 – Ошибка слияния

Для просмотра содержимого файла Test.txt воспользуемся командой cat

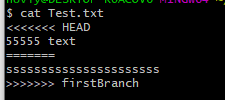


Рисунок 10 – Содержимое файла Test.txt

Как видим, появились новые строки, их можно рассматривать как «разделители конфликта». Строка ======= является «центром» конфликта. Все содержимое между этим центром и строкой <<<<<<< HEAD находится в текущей ветке master, на которую ссылается указатель HEAD. А все содержимое между центром и строкой >>>>>>> firstBranch является содержимым ветки для слияния.

Самый простой способ разрешить конфликт — отредактировать конфликтующий файл. Открываем файл Test.txt в привычном редакторе. И просто удалим все разделители конфликта. Измененное содержимое файла Test.txt будет выглядеть следующим образом:

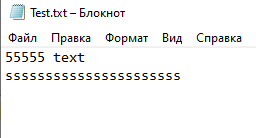


Рисунок 11 – Измененный файл Test.txt

После редактирования файла выполняем команду git add . , чтобы добавить новое объединенное содержимое в раздел проиндексированных файлов. Для завершения слияния создаём новый коммит, выполнив следующую команду:

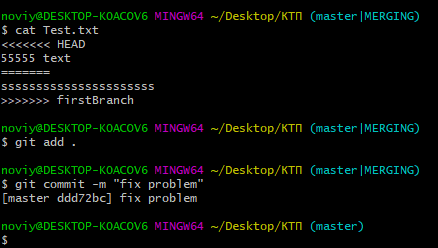


Рисунок 12 – Commit «fix problem»

Git обнаружит, что конфликт разрешен, и создаст новый коммит слияния для завершения процедуры слияния.

## 6. Откат изменений до одного из предыдущих коммитов

git reset HEAD^ - команда, откатывающая на один шаг назад

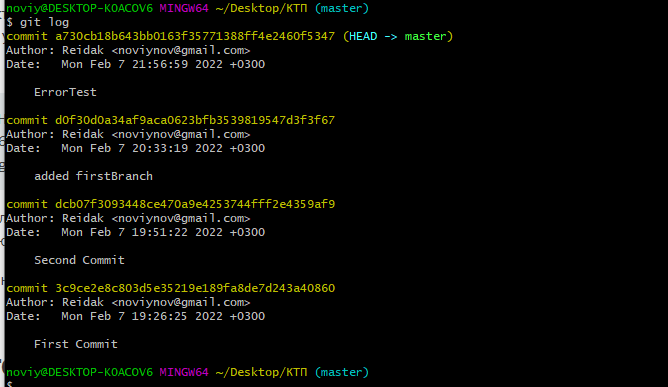


Рисунок 13 – До отката

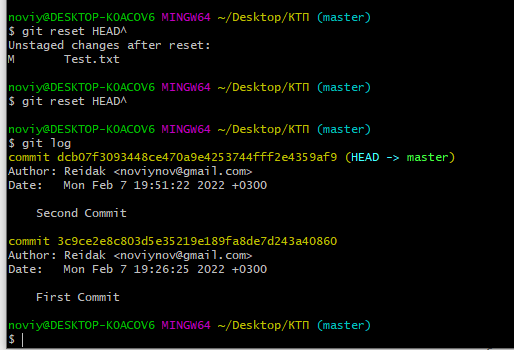


Рисунок 14 – После отката

# Вывод

Выполнив данные задачи, были изучены базовые функции системы контроля версий на примере Git.

# Список использованных источников:

1. Камаев В.А., Костерин В.В. Технологии программирования. М.: Высшая школа,

2006.

1. Жоголев Е.А.Технология программирования. – М.: Научный мир, 2004.

Git:

1. Scott Chacon, Ben Straub «Pro Git»
2. [git-scm.com](https://git-scm.com/)