Харьковский национальный университет им. В.Н. Каразина

Факультет компьютерных наук

# Лабораторная работа №1

По учебной дисциплине

**Математические методы и технологии тестирования и верификации программного обеспечения**

«Изучение системы контроля версии на примере Git»

|  |  |
| --- | --- |
|  | Выполнил:  Студент группы КС-22  Филатов Виталий Витальевич  Проверил:  Доц. Малахов Сергей Витальевич |

**Тема:** Изучение системы контроля версии на примере Git.

**Цель работы:** Цель – получить базовые знания о том, что такое Git и чем он отличается от централизованных систем контроля версий. Также получить рабочую версию Git на компьютере, настроенную и персонализированную.

**1. Установка Git**

Установочный файл можно скачать с официального сайта <https://git-scm.com>. После запуска открывается окно установки Git (Рисунок 1.1).



Рисунок 1.1 – окно установки Git

Нажимаем next, до полной установки. После установки запуститься консоль (Рисунок 1.2)



Рисунок 1.2 – окно Git Bash

**2. Первоначальная настройка Git**

Настройку нужно сделать только один раз. При обновлении версии настройки сохраняться. Но при необходимости, вы можете поменять настройки в любой момент (Рисунок 2.1, Рисунок 2.2).

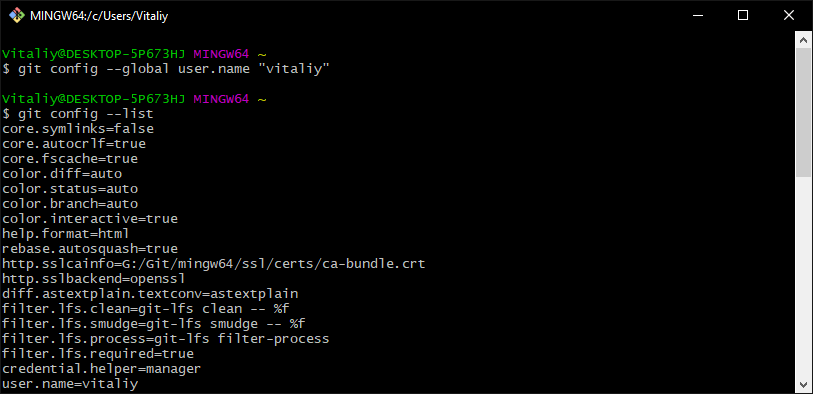


Рисунок 2.1 – установка имени пользователя

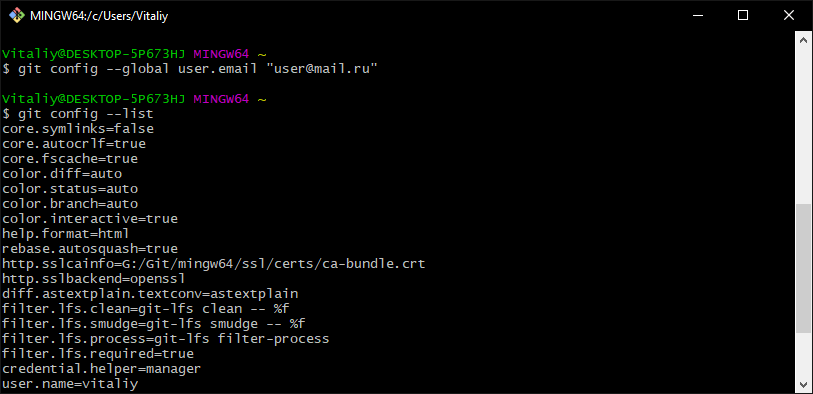


Рисунок 2.2 – установка почты пользователя

**3. Создание коммитов**

Изначально надо создать локальный Git репозиторий с помощью команды «*git init*» (Рисунок 3.1).

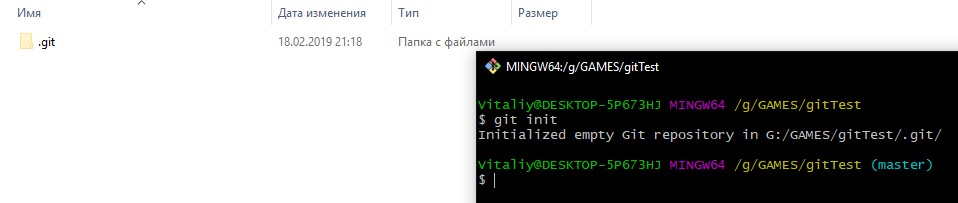


Рисунок 3.1 – инициализация локального Git репозитория

Далее необходимо добавить файл в наш Git репозиторий (Рисунок 3.2).

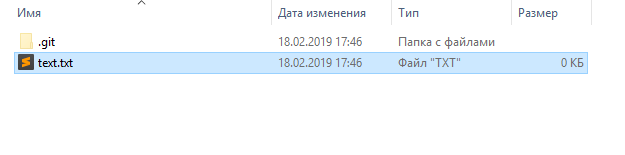


Рисунок 3.2 – добавление файла в репозиторий

Далее необходимо зарегистрировать наш файл в репозитории с помощью команды «*git add .* » (Рисунок 3.3).

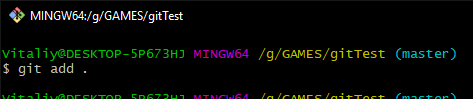


Рисунок 3.3 – добавление файлов в репозиторий

Теперь остается только сделать коммит командой «*git commit –m “Message”*» (Рисунок 3.4).

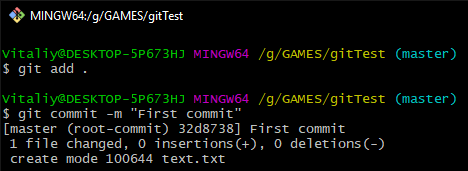


Рисунок 3.4 – создание коммита

**4. Просмотр истории изменений в файлах**

Для просмотра истории коммитов используется команда «*git log*»

(Рисунок 4.1).

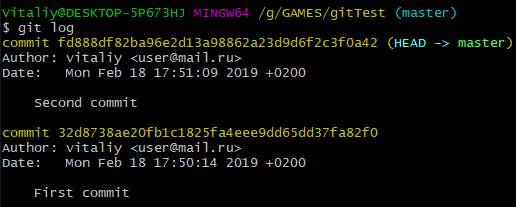


Рисунок 4.1 – история коммитов, выведенная командой *git commit*

**5. Создание новых веток и переключение между ними**

Для создания новых веток используется команда «*git branch название\_ветки*». Для переключения между ветками используется команда «*git checkout имя\_ветки*» (Рисунок 5.1).

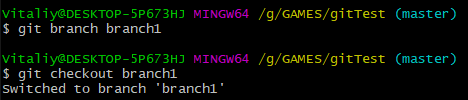


Рисунок 5.1 – создание новой ветки и переключение на неё

**6. Слияние веток**

Для слияния веток необходимо переключиться на ветку, которую надо обновить и выполнить команду «*git merge название\_ветки*». (Рисунок 6.1)

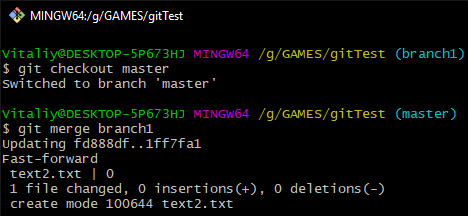


Рисунок 6.1 – слияние ветки branch1 с веткой master

**7. Отправка данных на удаленный репозиторий**

Удаленному репозиторию можно присвоить локальное имя (Рисунок 7.1).



Рисунок 7.1 – присвоение имени удаленному репозиторию

Для отправки данных на удаленный репозиторий используется команда «*git push –u имя\_репозитория имя\_нашей\_ветки*» (Рисунок 7.2).

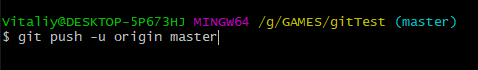


Рисунок 7.2 – отправки данных с ветки на удаленный репозиторий

После выполнения команды файлы будут находиться в удаленном репозитории, откуда их потом можно будет в дальнейшем скачать и изменить (Рисунок 7.3).

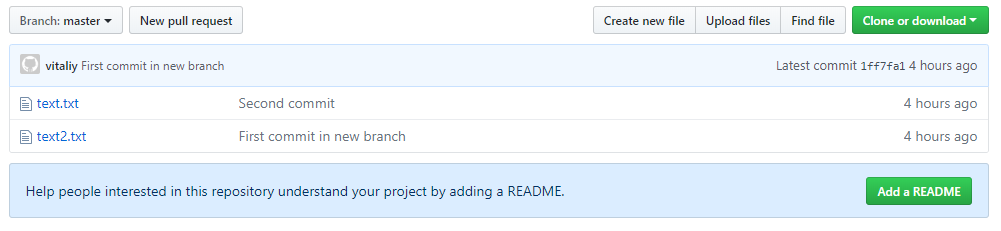


Рисунок 7.3 – данный находятся в удаленном Git репозитории

**ВЫВОДЫ**

Входе лабораторной работы была установлена система контроля версий Git. Я научился добавлять файлы в репозиторий, создавать новые ветки, переключаться между ними, выполнять слияние веток, а также отправлять данные на удаленный Git репозиторий.