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 «Математическая кибернетика и информационные технологии»

Лабораторная работа №0.2

«Методы ветвления в системе контроля версий»

по дисциплине

«Кроссплатформенные технологии программирования»

Москва 2023

**Содержание**

[1 Цель работы 3](#_Toc95374360)

[2 Задание на лабораторную работу 3](#_Toc95374361)

[3 Ход работы 4](#_Toc95374362)

[3.1 Настроить игнорирование git’ом файлов определенного расширения с использованием glob-шаблонов 4](#_Toc95374363)

[3.2 Вывод лога коммитов в измененном пользовательском формате 5](#_Toc95374364)

[3.3 Переименование ветки 5](#_Toc95374365)

[3.4 Проведение в ветке работы и слияние с веткой master, удаление слитой ветки 6](#_Toc95374366)

[3.5 Настройка удаленного репозитория: получение изменения из репозитория и отправка изменения 6](#_Toc95374367)

[3.6 Организация совместной работы над проектом в соответствии с выбранной идеологией 8](#_Toc95374368)

[4 Вывод 10](#_Toc95374369)

[Список использованных источников 11](#_Toc95374370)

# 1 Цель работы

Изучение подходов к командной разработке в системе контроля версий git.

# 2 Задание на лабораторную работу

В созданном в предыдущей работе либо вновь созданном репозитории:

1. Настроить игнорирование git’ом файлов определенного расширения с использованием glob-шаблонов

2. Вывести лог коммитов в измененном пользовательском формате

3. Переименовать ветку

4. Провести в ветке работу и слить с веткой master, удалить слитую ветку

5. Настроить удалённый репозиторий: получить изменения из репозитория и отправить изменения.

6. Организовать совместную работу над проектом в соответствии с выбранной идеологией

# 3 Ход работы

# 3.1 Настроить игнорирование git’ом файлов определенного расширения с использованием glob-шаблонов

Для того, чтобы игнорировать файлы с расширением html, был создан файл .gitignore c содержимым представленным на рисунке 1.

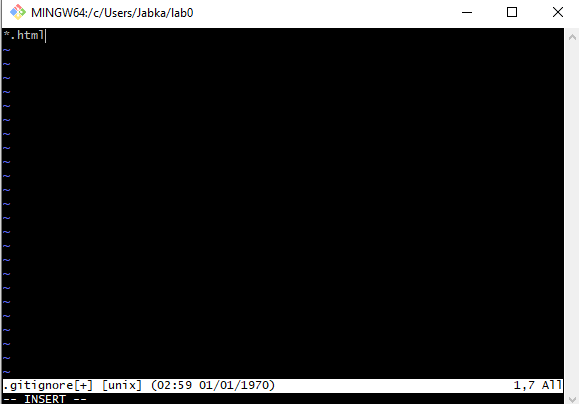


Рисунок 1-Файл .gitignore

Далее по отношению к gitignore был применен commit, что представлено на рисунке 2.

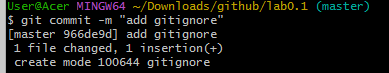


Рисунок 2-git commit

# 3.2 Вывод лога коммитов в измененном пользовательском формате

Для вывода лога коммитов в измененном формате был выполнен git config --global alias.hist "log --pretty=format:'%h %ad | %s%d [%an]' --graph --date=short" и далее введен git hist. Результат работы вывода логов коммита представлен на рисунке 3.

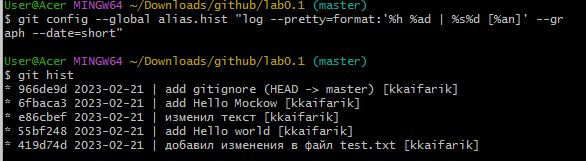


Рисунок 3-Вывод логов коммита

# 3.3 Переименование ветки

Для того, чтобы переименовать ветку new, была введена команда git branch -m new1. Результат работы представлен на рисунке 4.

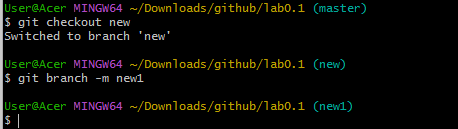


Рисунок 4-Результат работы git branch -m

# 3.4 Проведение в ветке работы и слияние с веткой master, удаление слитой ветки

После проведенной работы в ветке и слияния ее с веткой мастер, ветка new1 была удалена с помощью команды git branch -d new1. Результат работы представлен на рисунке 5.

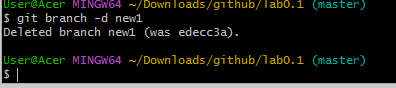


Рисунок 5- Результат работы git branch -d new1

Далее на рисунке 6 мы видим, что осталась только ветка master.

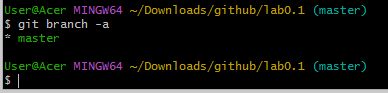


Рисунок 6-Оставшиеся ветки

# 3.5 Настройка удаленного репозитория: получение изменения из репозитория и отправка изменения

Была выполнена команда push, при помощи которой ветка master была скопирована в репозиторий на GitHub. Результат работы представлен на рисунке 7.

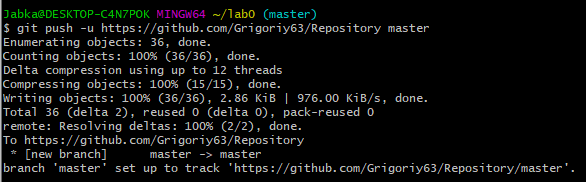


Рисунок 7-Результат работы push

Далее файл в репозитории на GitHub был изменен, что представлено на рисунке 8.

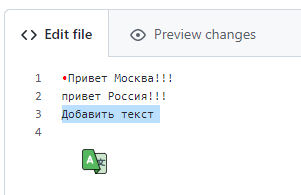


Рисунок 8-Изменение файла в репозитории на GitHub

Для того, чтобы получить эти изменения в локальном репозитории была применена команда pull. Результат ее работы представлен на рисунке 9.

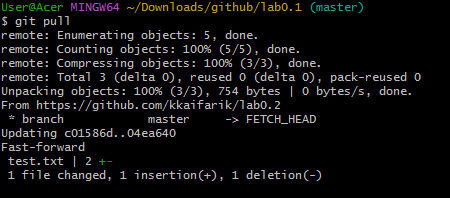


Рисунок 9-Результат работы pull

Далее мы видим, что данные файла были обновлены, в соответствии с изменениями файла в репозитории на GitHub. Результат представлен на рисунке 10.



Рисунок 10-Изменение содержимого файла после исполнения pull

# 3.6 Организация совместной работы над проектом в соответствии с выбранной идеологией

На GitHub был приглашен пользователь, с которым будет проводится совместная работа.

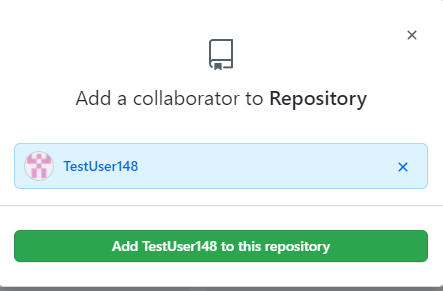


Рисунок 11-Приглашение пользователя для совместной работы

Для того чтобы пользователю работать с репозиторием он его клонирует с GitHub, с помощью git clone. Работа git clone представлена на рисунке 12.

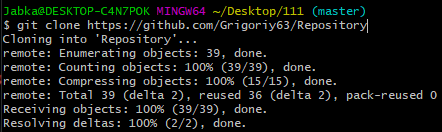


Рисунок 12-Работа clone

Далее в ходе работы пользователь вносит изменения в файл представленные на рисунке 13.

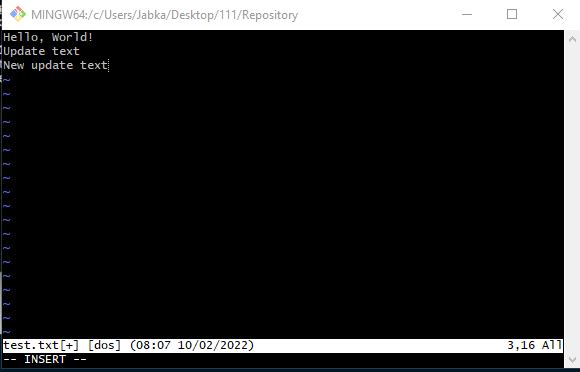


Рисунок 13-Изменения внесенные в файл

На рисунке 14 показано, как с помощью push были отправлены изменения на GitHub.

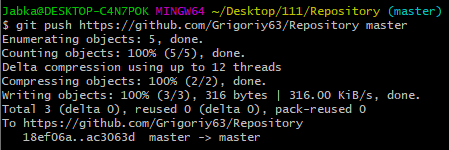


Рисунок 14-Результат работы push

На рисунке 15 мы видим внесенные изменения в репозитории на GitHub.

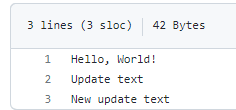


Рисунок 15-Изменения, внесенные пользователем

Далее, чтобы продолжить работу над файлом другому пользователю, необходимо выполнить команду pull представленную на рисунке 16.

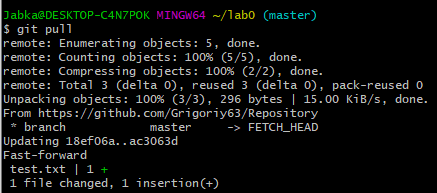


Рисунок 16-Результат работы pull

# 4 Вывод

По итогу проделанной работы изучил подходы к командной разработке в системе контроля версий git.

# Список использованных источников

1) ГОСТ 7.32.2017 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления. – М.: Стандартинформ, 2017, - 27 с.