Федеральное государственное бюджетное образовательное учреждение высшего образования "Национальный исследовательский университет "МЭИ"

ОТЧЕТ

По лабораторной работе № 2

# «Система контроля версий Git»

ФИО студента Зобов Павел Александрович

Номер группы А-02-18

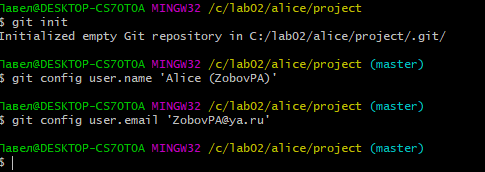
Имя преподавателя Мохов Андрей Сергеевич

Козлюк Дмитрий Александрович

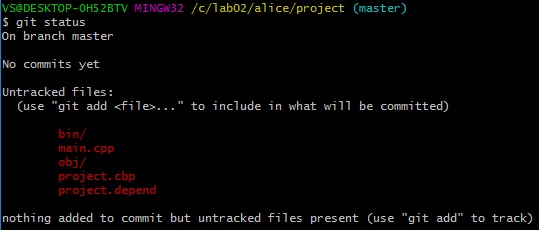
Москва, «НИУ «МЭИ», 2019

Ход работы

1. Инициализируем репозитарий, и настраиваем имя и email разработчика для подписи commit-ов:



1. После создания и сборки проекта в среде, просмотрим состояние рабочей копии:



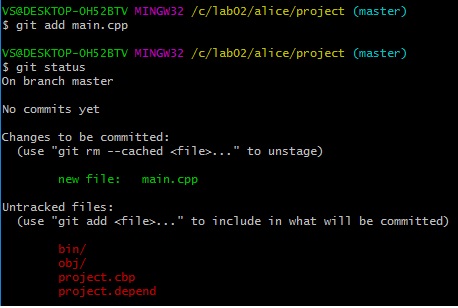
Первая строка – текущая ветвь;

Вторая строка – говорит о том, что коммитов еще не было;

Третья строка – файлы, не отслеживаемые Git;

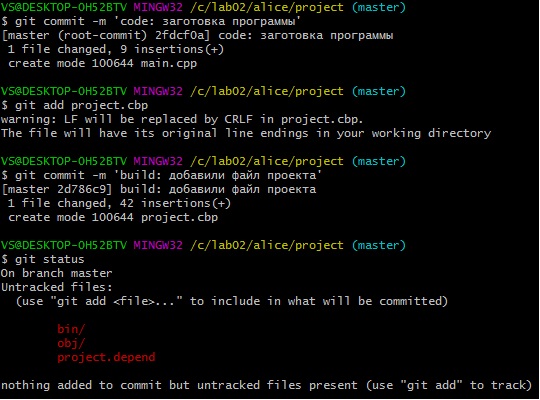
Четвертая строка – ничего не добавлено в коммит, но присутствуют не отслеживаемые файлы;

1. Добавим файл **main.cpp** в отслеживаемые (в индекс) и еще раз проверим состояние рабочей копии:

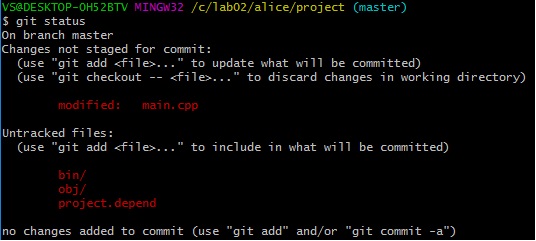


Видим, что файл **main.cpp** стал отслеживаться.

1. Добавим в коммит файлы **main.cpp** и **project.cbp**, приписав к ним короткие сообщения:

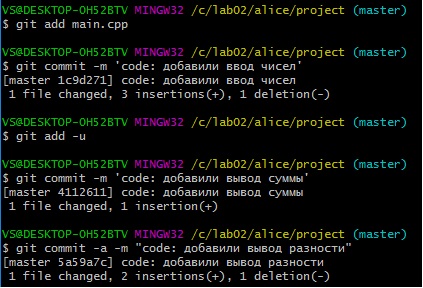
****

1. После изменения кода в файле **main.cpp** видим, что Git отследил изменение файла:

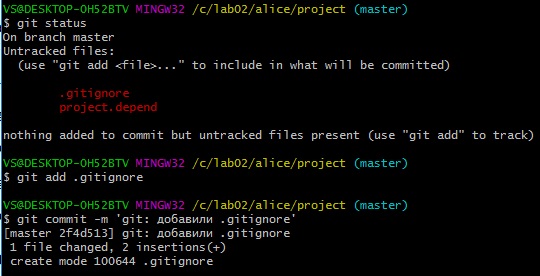


Файл **main.cpp** стал иметь статус ***modified*** (изменен), а не ***new file*** (новый файл в индексе)

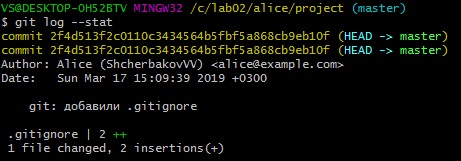
1. Закоммитим файл **main.cpp** уже использовавшимся способом, затем допишем в файл вывод суммы чисел и добавим его в индекс командой для всех измененных файлов, закоммитим. После этого также добавим вывод разности чисел в файл, и сделаем коммит командой в один шаг:



1. Для игнорирования Git каталогов ***bin*** и ***obj***, не нужных для индексирования, в папке проекта создадим файл **.gitignore** и добавим в него эти директории. Видим, что они больше не отслеживаются. Закоммитим **.gitignore**:

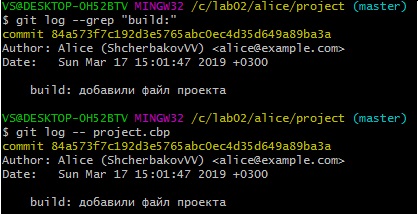


1. Отобразим журнал репозитария для последнего коммита:



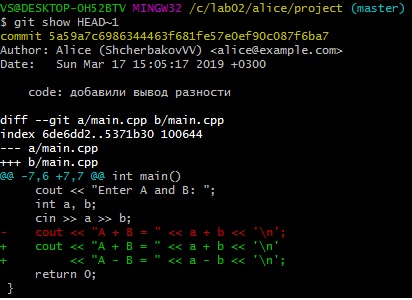
Отображаются хеш данного коммита, маркер ***HEAD*** –«верхушка» ветки ***master***, дата коммита, информация о разработчике и об изменениях в коммите.

1. Применим фильтрацию коммитов:

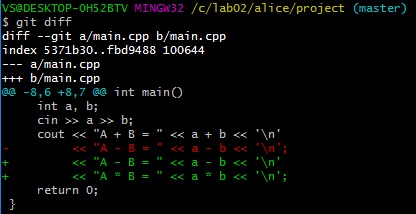


Коммиты можно фильтровать как по сообщениям, так и по именам файлов.

1. Просмотрим предпоследний коммит (отступив от последнего на 1, ***HEAD~1***)

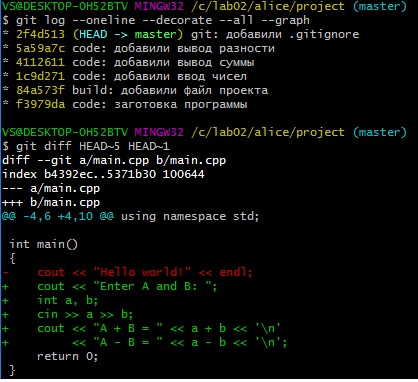


1. Добавим в main.cpp вывод произведения чисел, но не станем коммитить изменения. Просмотрим изменения в рабочей копии:

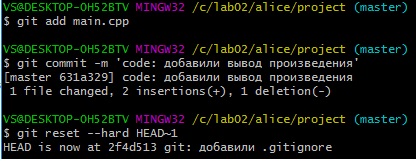


Удаленные строки печатаются красным, добавленные – зеленым.

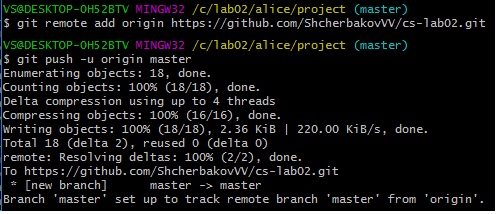
1. Просмотрим изменения между самым первым коммитом и коммитом, добавляющим вывод разности:



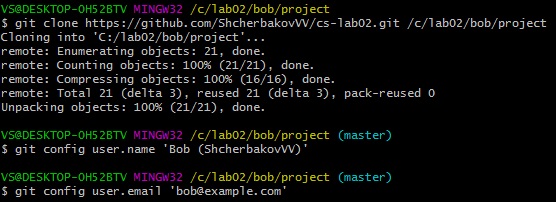
1. Добавим в файл вывод произведения чисел, закоммитим. Откатимся на предыдущий коммит:



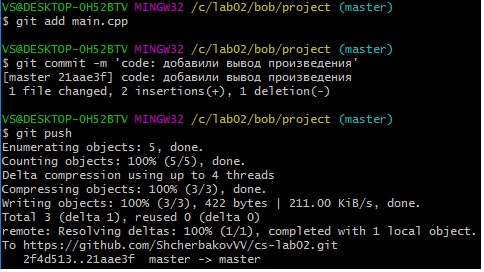
1. После регистрации на GitHub загружаем репозитарий:



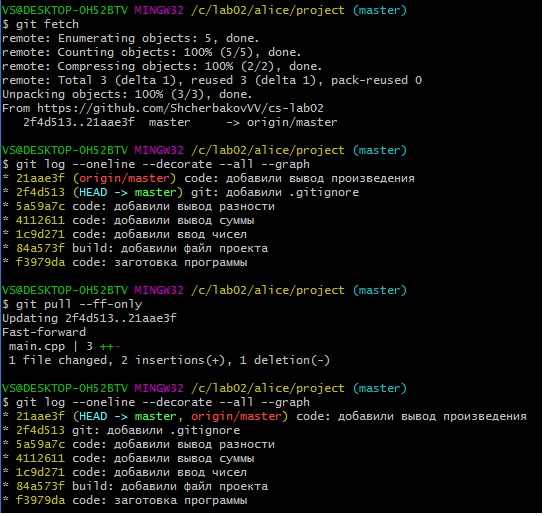
1. Клонируем проект на «машину Боба» и указываем его данные:



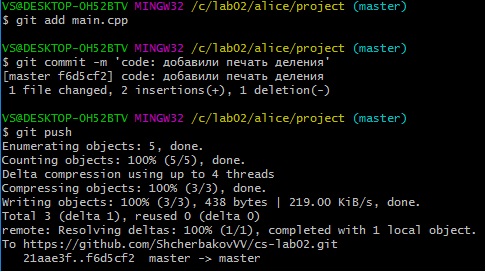
1. Добавляем в **main.cpp** вывод произведения чисел, коммитим изменения, отправляем их на GitHub:

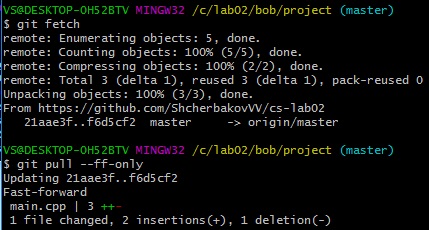


1. На машине Алисы получаем измененную копию, синхронизируем ветки:

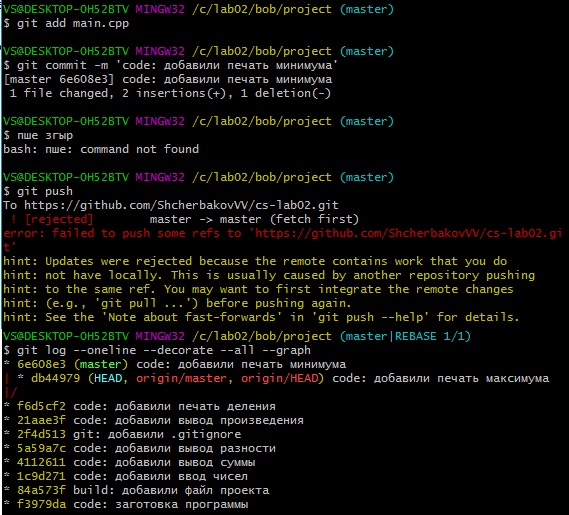


1. С машины Алисы изменяем **main.cpp**, добавив в него печать деления, коммитим, отправляем на GitHub, получаем измененную рабочую копию с машины Боба, синхронизируем ветки:

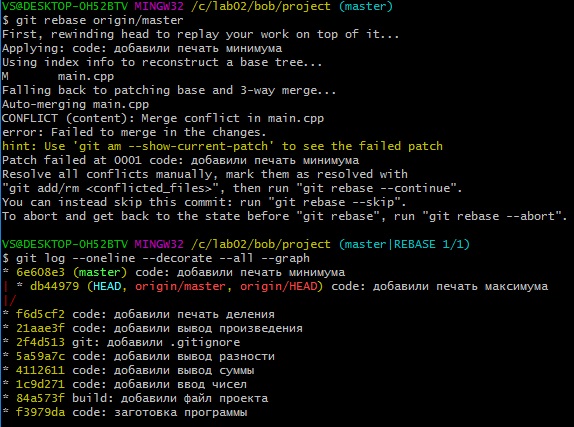




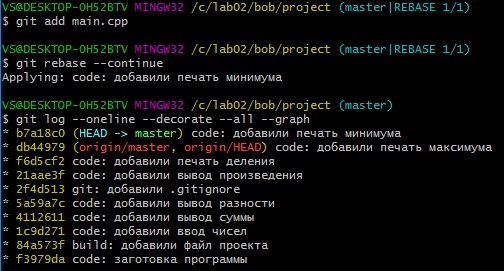
1. После того, как мы на машине Алисы добавили печать максимума, закоммитили изменения и отправили на GitHub, пытаемся сделать то же самое на машине Боба, но с печатью минимума. Что мы получаем:



1. Пробуем исправить конфликт, используя ***git rebase***:

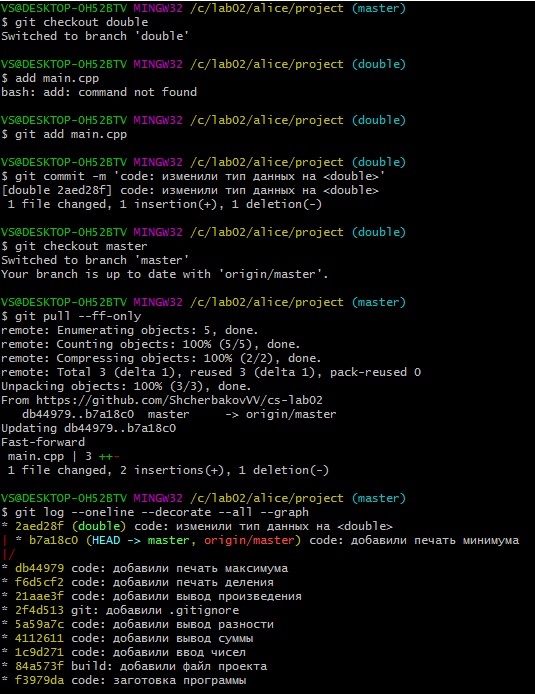


На машине Боба корректируем код, продолжаем прерванную операцию **rebase**:

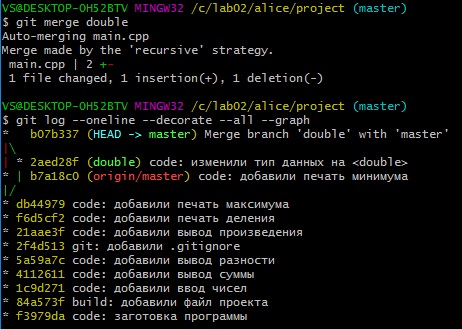


Отправляем на GitHub.

1. На машине Алисы создаем новую ветку ***double***, изменяем тип чисел на “double”, коммитим. Синхронизируем ветвь ***master*** с GitHub:



1. Сливаем ветки:



Отправляем изменения на GitHub.

Ссылка на репозитарий:

<https://github.com/ZobovPA/cs-lab02>