Министерство образования и науки Украины

Харьковский национальный университет «Харьковский политехнический институт»

Лабораторная работа №4

Основы работы с ветвями в GIT. Основы слияния. Разрешение конфликтов.

Выполнил:

ст. гр. КИТ-14П

Парфёнов Б.И.

Проверил:

Почебут М. В.

Харьков 2015

**Задание на лабораторную работу №4.**

Все действия выполняются индивидуально и направлены на получение навыков работы с GIT.

Отчет оформляется в виде скриншотов с комментариями действий при необходимости.

Последовательность выполнения работы:

Часть 6. Создание ветвей и слияние.

1. Создайте новую ветвь greet\_name (где name – Ваша фамилия), используя команду

git checkout -b <branchname>

1. Добавьте новый каталог (пример lib\_name) и файл *greeter\_name.rb* c содержимым:

class Greeter

def initialize(who)

@who = who

end

def greet

"Hello, #{@who}"

end

end

1. Сделайте 3 commits в ветку greet\_name
2. Продемонстрируйте переключение на ветку master.
3. Создайте файл README и произведите commit в master.
4. Произведите слияние ветвей greet и master – продемонстрируйте историю commits. (используйте опцию Fast forward).
5. Изобразите в удобном для Вас графическом редакторе граф коммитов, где вершинами графа будут коды коммитов.

Часть 7.

Смоделируйте искусственно конфликт, при слиянии описанном в предыдущей части.

Часть 8.

Используя команду reset верните указатель на состояние ветвей до их слияния.

Часть 9. Проделайте шаги 3-5 в части 6. Выполните слияние изменений командой rebase. Изобразите в удобном для Вас графическом редакторе граф коммитов, где вершинами графа будут коды коммитов.

Объясните разницу между merge и rebase. На примере Ваших графов.

Для начала работы создайте/зарегистрируйтесь (в случае отсутствия на текущий сервисе (типа момент) на удаленном *github*) *,* т.е. инициализируйте удаленный репозиторий.

Часть 10.

1. Разбейтесь на пары.
2. Выберите в качестве общего репозитория один из двух удаленных.
3. Student1 заливает все содержимое своих папок в удаленный репозиторий.

Student 2 «забирает» проект к себе в локальный репозиторий.

1. Выполните команду git status.

Проведите каждый по три коммита в локальный репозиторий и «залейте изменения» в удаленный. (разрешается использование и merge и rebase).

**Выполнение**

1. Выполнено создание ветки greet\_Parfenov.

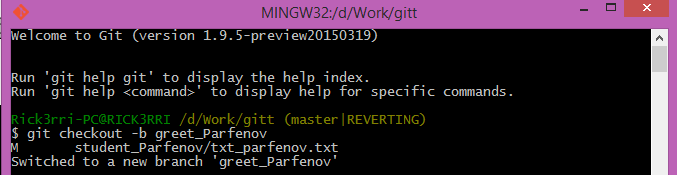


Рисунок 1 – Создание ветки

1. Создан файл greeter\_Batulin.rb и сделан коммит.

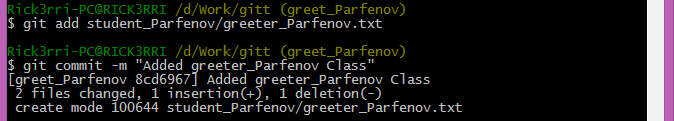
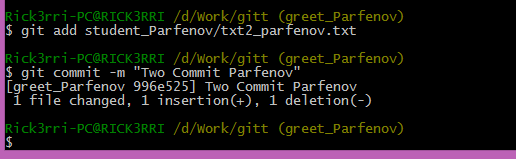


Рисунок 2 – Создание файл и его коммит

1. Созданы еще 2 файла и сделаны 2 коммита.



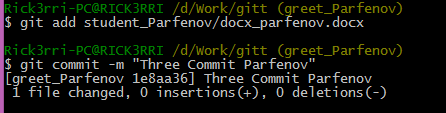


Рисунок 3 – Создание трех файлов и три коммита

1. Сделано переключение на ветку master.



Рисунок 4 – Переключение на ветку master

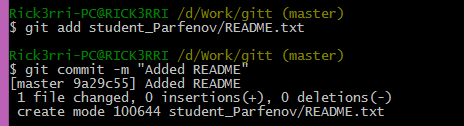
1. Создан файл README и сделан коммит в ветку master

Рисунок 5 – Создание файла README и его коммит

1. Выполено слияние веток greeter\_Parfenov и master. А также показана лог проекта

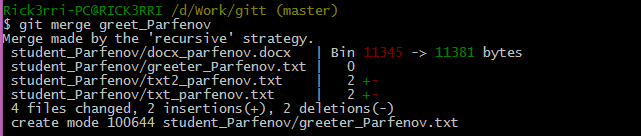


Рисунок 6 – Слияние веток

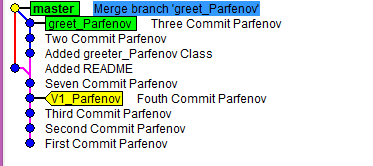


Рисунок 7 – отображение графа коммитов

1. Создан искусственный конфликт.

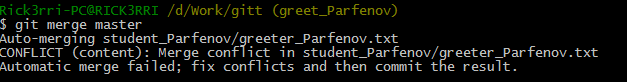


Рисунок 8 – Конфликт

1. Командой reset был произведен возврат до слияния.

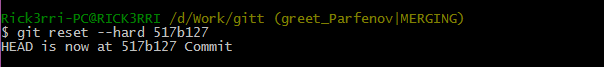


Рисунок 9 – Возврат до слияния

1. Выполнен rebase.

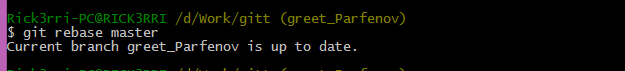


Рисунок 10 – Rebase

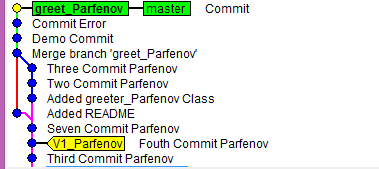


Рисунок 11 – Граф после rebase

При использовании команды rebase стирается история ветки, в отличии от merge. Команду rebase предпочтительно использовать для коротких непубличных веток, когда не важна ее история. В ином случае нужно использовать merge, но данная команда может создать сложности в читабельности и структурировании истории коммитов.