ОТЧЕТ

по лабораторной работе № 7

«Исследование распределенной системы контроля версий Mercurial при коллективной разработке программных продуктов»

по дисциплине

«ТЕХНОЛОГИИ СОЗДАНИЯ ПРОГРАММНЫХ ПРОДУКТОВ»

Выполнил студент группы ИС/б-22о

Горбенко К.Н.

Проверил:

Строганов В.А.

* 1. ЦЕЛЬ РАБОТЫ

Исследовать основные подходы к организации взаимодействия команды разработчиков с использованием распределенной системы контроля версий (DVCS). Приобрести практические навыки установки и настойки DVCS Git, организации ветвей разработки и осуществление слияния.

* 1. ЗАДАНИЕ НА РАБОТУ

Разработать модель командной работы согласно варианту, полученному у преподавателя.

. Создать необходимое количество репозиториев, разработать соглашение по предназначению репозиториев.

Создать изменения в одном локальном репозитории, сохранить их в удаленном.

Получить набор изменений из удаленного репозитория в репозиторий отличный от описанного в п.3.3, внести дополнительные изменения и сохранить их в удаленном репозитории.

Внести одновременно разные изменения в локальные репозитории сохранить их все в удаленном, продемонстрировать процесс слияния.

Внести изменения в разных локальных репозиториях в одинаковые файлы, в одинаковых строках. Продемонстрировать процесс слияния при наличии конфликтов.

Продемонстрировать создание именованных веток в локальном репозитории.

Дать краткое описание команд распределенной системы контроля версий используемых при выполнении работы.

Проанализировать результаты работы, сделать выводы.

* 1. ХОД РАБОТЫ
     1. Создание и заполнение удаленного репозитория

Для работы необходим удаленный репозиторий. Создадим репозиторий на Bitbucket. Теперь используем локальный репозиторий, полученный в л.р. № 6 и отправим файлы в удаленный репозиторий на Bitbucket. Используем .hgignore файл.

cd Source\Repos\OfflineElectronicsStore

hg init

hg add .hgignore

hg commit -m "Added .hgignore"

hg add

hg commit -m "Added project existing files"

hg push <https://bitbucket.org/astro6703/offlineelectronicsstore/src>

Результат выполнения команды изображен на рисунке 1:

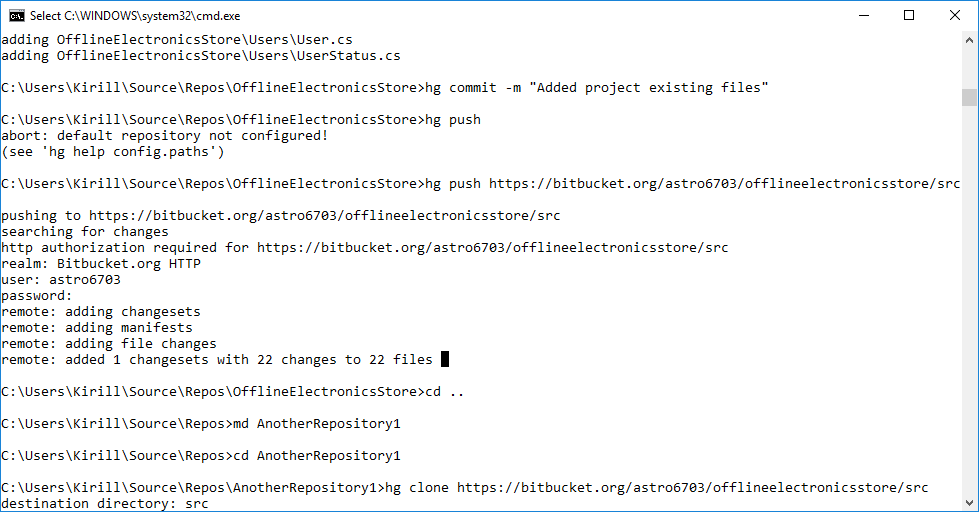


Рисунок 1 – Результат выполнения команды hg push

* + 1. Работа в другом локальном репозитории

Получим данные из удаленного репозитория в другом локальном репозитории. Затем внесем некоторые изменения и отправим их в удаленный репозиторий:

hg clone https://bitbucket.org/astro6703/offlineelectronicsstore

cd offlineelectronicsstore

hg add NewFile.cs

hg commit -m "Added NewFile.cs"

hg push <https://bitbucket.org/astro6703/offlineelectronicsstore/src>

Результат выполнения команды изображен на рисунке 2:

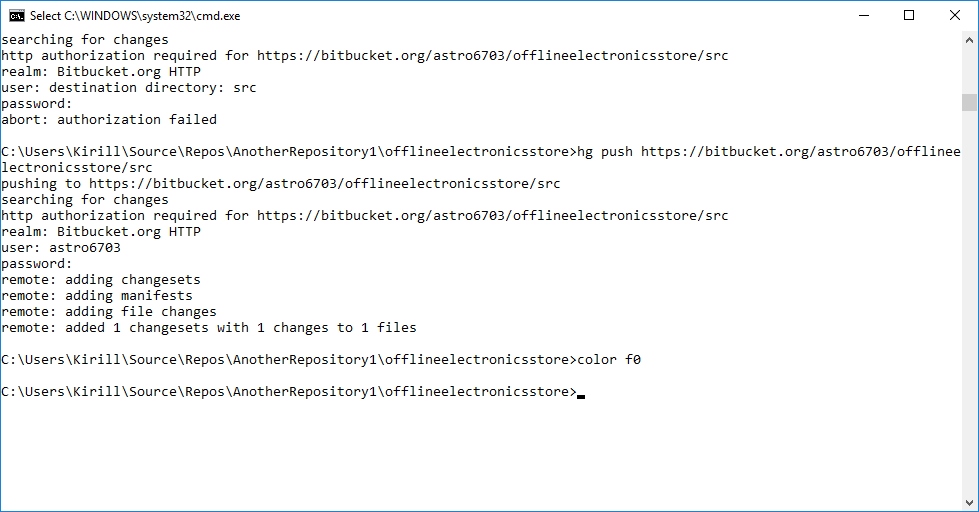


Рисунок 2 – Результат выполнения команды hg push

* + 1. Внесение изменений из различных репозиториев

Внесем разные изменения из двух локальных репозиториев в удаленный репозиторий:

echo "Some file content here" > NotMergingFile.txt

hg add NotMergingFile.txt

hg commit -m "Added not merging file from repository 1"

hg push https://bitbucket.org/astro6703/offlineelectronicsstore/src

Результат:

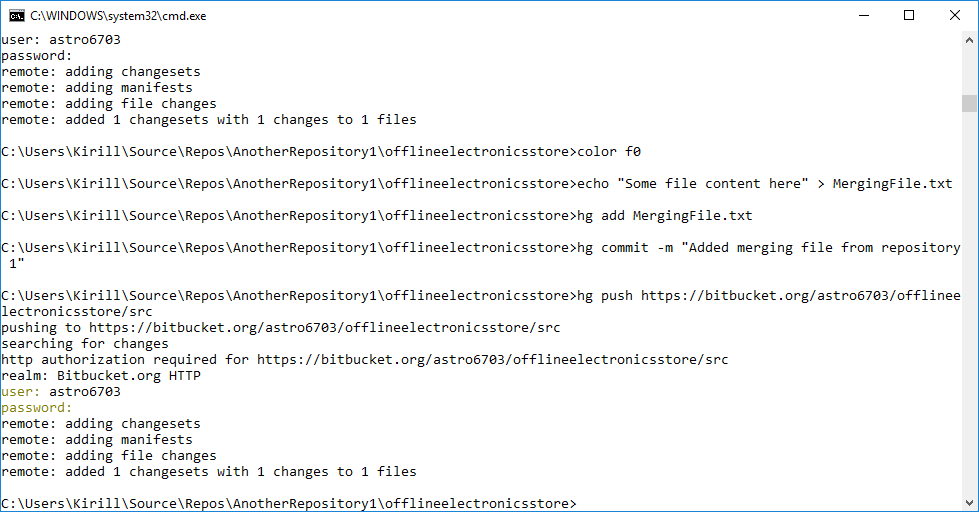


Рисунок 3 – Сохранение изменений первого репозитория в удаленном репозитории

Теперь попробуем внести изменения из другого репозитория:

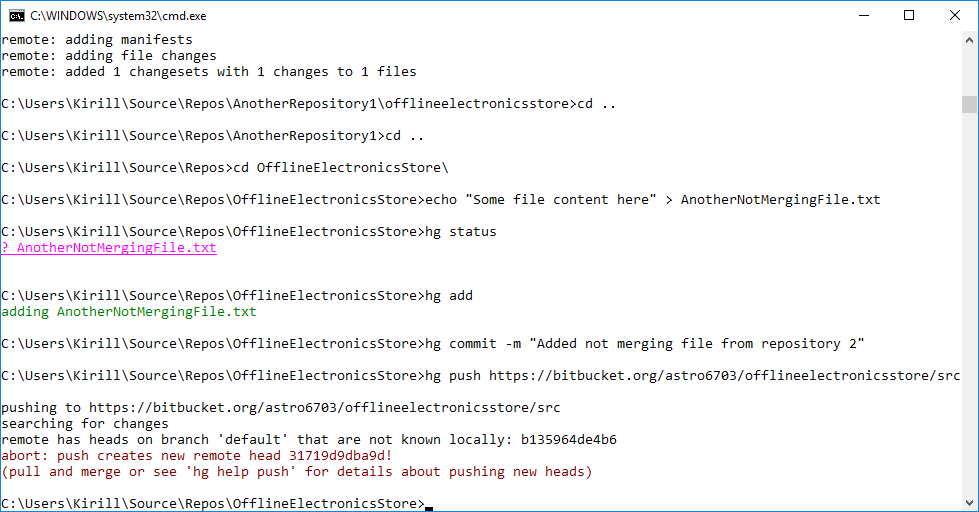


Рисунок 4 – Ошибка при попытке сохранения изменений

Для устранения ошибки необходимо вытянуть последнюю версию удаленного репозитория и слить ее с текущей версией этого локального репозитория:

hg pull <https://bitbucket.org/astro6703/offlineelectronicsstore/src>

hg merge

hg commit -m "Merged heads"

hg push <https://bitbucket.org/astro6703/offlineelectronicsstore/src>

Результат сохранения изменений в удаленном репозитории:



Рисунок 5 – Результат выполнения команды сохранения изменений в удаленном репозитории

* + 1. Внесение конфликтных изменений из различных репозиториев

В текущем локальном репозитории создадим файл и сохраним его в удаленном репозитории:

echo "Some file content" > ConflictFile.txt

hg add

hg commit -m "Added conflict file from repository 2"

hg push <https://bitbucket.org/astro6703/offlineelectronicsstore/src>

Внесем изменения во из второго репозитория:

cd ..\AnotherRepository1\offlineelectronicsstore

echo "Another file content" > ConflictFile.txt

hg add

hg commit -m "Added conflict file from repository 1"

hg push <https://bitbucket.org/astro6703/offlineelectronicsstore/src>

При попытке сохранить изменения в удаленный репозиторий, снова возникает ошибка. Необходимо загрузить последнюю версию из удаленного репозитория:

hg pull <https://bitbucket.org/astro6703/offlineelectronicsstore/src>

hg merge

При выполнении команды возникла ошибка из-за конфликтующего файла. Прежде чем сливать ветви необходимо вручную разрешить конфликт.

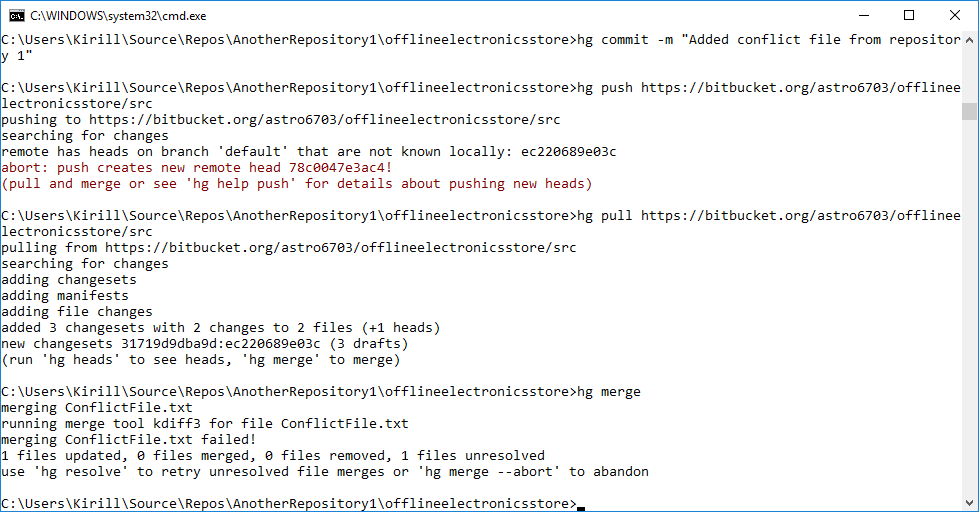


Рисунок 6 – Ошибка при слиянии конфликтующих ветвей

После устранения конфликта проведем слияние:

hg commit -m "Removed conflict file"

hg merge

hg push <https://bitbucket.org/astro6703/offlineelectronicsstore/src>

* + 1. Работа с ветвями

Создадим в одном из локальных репозиториев ветвь:

hg branch implementing-some-feature

hg commit -m "Created new branch"

hg branches

Результат:

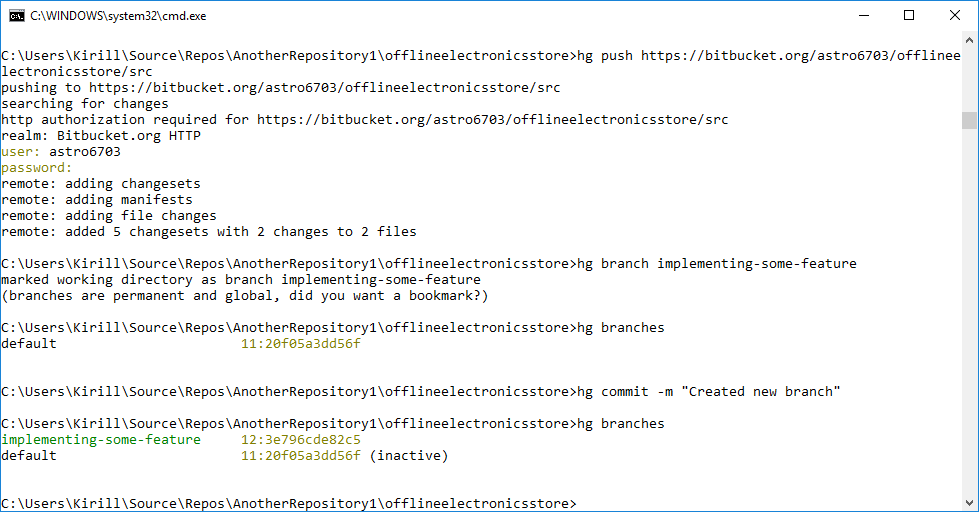


Рисунок 6 – Создание ветви

Теперь добавим коммит внутри ветви:

echo "Some another file content" > SomeNewExcitingFeature.txt

hg add

hg commit -m "Implemented some new exciting feature"

hg push --new-branch https://bitbucket.org/astro6703/offlineelectronicsstore/src

Создадим новый пулл-реквест:

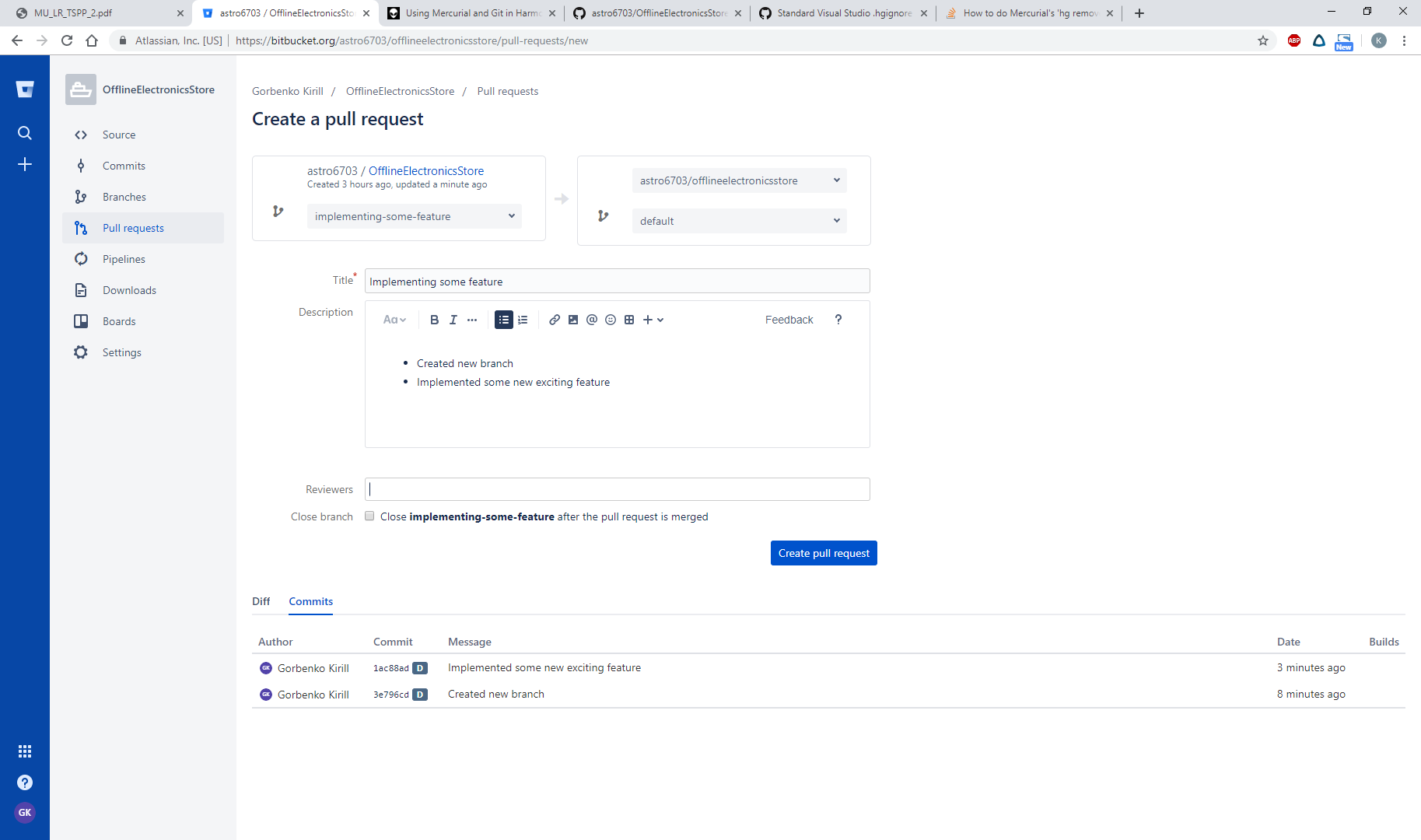


Рисунок 7 – Процесс создания пулл-реквеста

* 1. ВЫВОД

В ходе лабораторной работы была изучена распределенная система контроля версий Mercurial. Для работы был создан удаленный репозиторий на Bitbucket.org. В ходе работы возникали проблемы при попытке сохранить локальные изменения в удаленный репозиторий. В случае, когда конфликтов между локальным и удаленным репозиториями нет, достаточно лишь загрузить последнюю версию удаленного репозитория и слить ее с локальным репозиторием, после чего сохранять. Если после загрузки последней версии репозитория обнаружились конфликты, их придется устранить вручную.

Кроме того, при попытке сохранить локальную ветвь на удаленном репозитории возникла необходимость использовать ключ –new-branch команды push.