Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий и управления в технических системах

Кафедра ИС

ОТЧЕТ

по лабораторной работе №6

Исследование распределенных систем контроля версий Mercurial при коллективной разработке программных продуктов

Выполнил:

ст. гр. ИС/б-21-о

Куркчи А. Э.

Проверил:

Тлуховская Н. П.

Севастополь

2016

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 Mercurial, организации ветвей разработки и осуществление слияния.

1. Постановка задачи

Система – Магазин по продаже компьютерной техники.

Система предназначена для автоматизации работы магазина, в котором необходимо предусмотреть работу нескольких подразделений.

В системе должны быть реализованы следующие функции: для руководства магазина: заказ товаров у поставщиков, просмотр статистики по продажам за разные периоды и различные виды товаров; для работников склада: прием товаров на баланс, учет товаров на складе (просмотр наличия товара на складе, его количества), формирование отчета по изменению загруженности склада; для кассиров — принять деньги за товар, выдать деньги в случае возврата товара, сформировать чек для покупателя; для продавцов — формирование накладной для покупателя.

2.1. Разработать модель командной работы согласно полученному у преподавателя.

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

варианту,

разработать

2.3. Создать изменения в одном локальном репозитории, сохранить их

в удаленном.

2.4.Получить набор изменений из удаленного репозитория в

репозиторий отличный от описанного в п.3.3, внести дополнительные изменения и сохранить их в удаленном репозитории.

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

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

2.7.Продемонстрировать создание именованных веток в локальном репозитории.

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

2.9. Проанализировать результаты работы, сделать выводы

1. Ход работы

Создание центрального репозитория с помощью команды hg init.

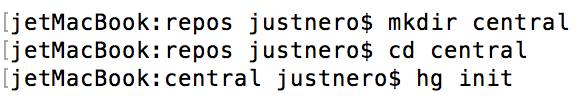


Рисунок 1 – Создание центрального репозитория

Клонирование центрального репозитория в локальный репозиторий первого пользователя, создание файла main.cpp, добавление его в индекс, и проверка статуса рабочего дерева.

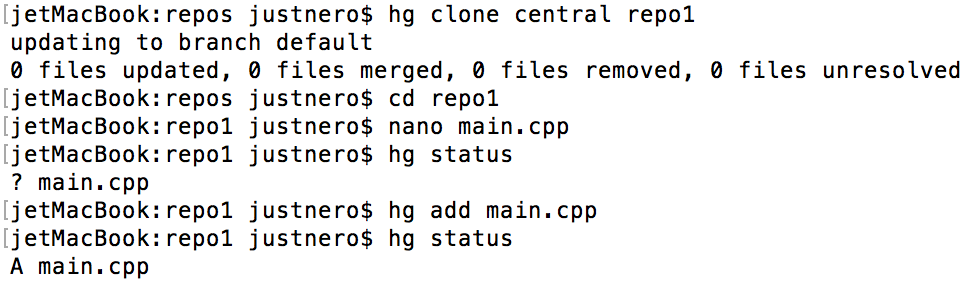


Рисунок 2 – Работа с локальным репозиторием 1

Коммит изменений, проверка исходящих коммитов и пуш в центральный репозиторий

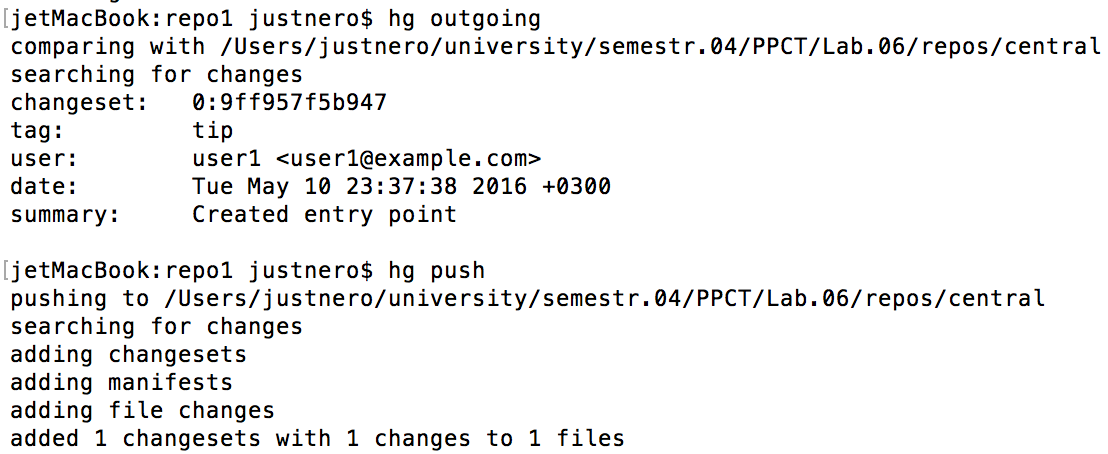


Рисунок 3 – Продолжение работы с репозиторием 1

Клонирование центрального репозитория в локальный репозиторий второго пользователя и обновление данных из него, так как склонирован репозиторий был сразу после создания центрального, до изменений первого. Изменение файла main.cpp и коммит изменений в локальный репозиторий с последующим пушем в центральный. Создание файла data.txt, добавление его в индекс, коммит и пуш в центральный репозиторий.

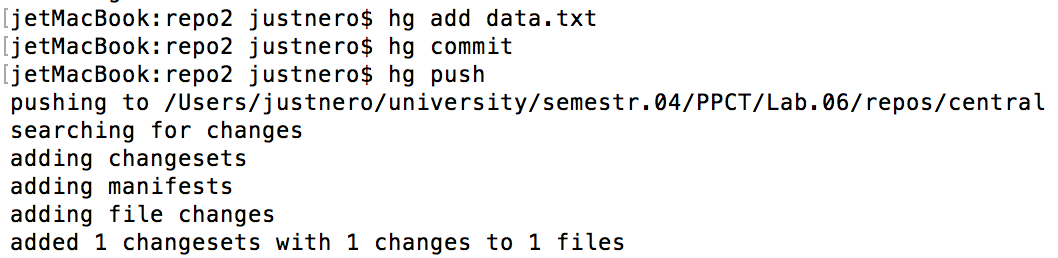
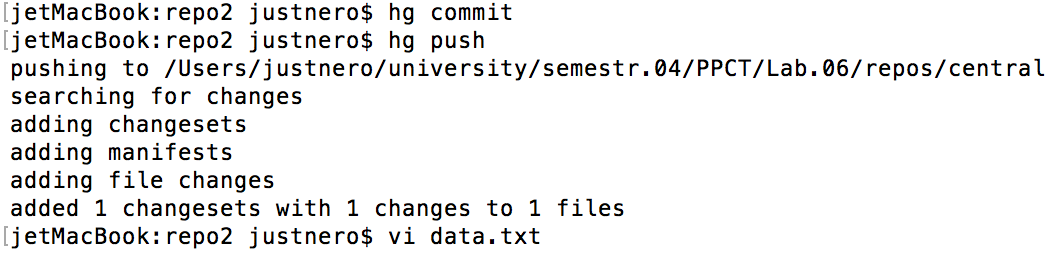
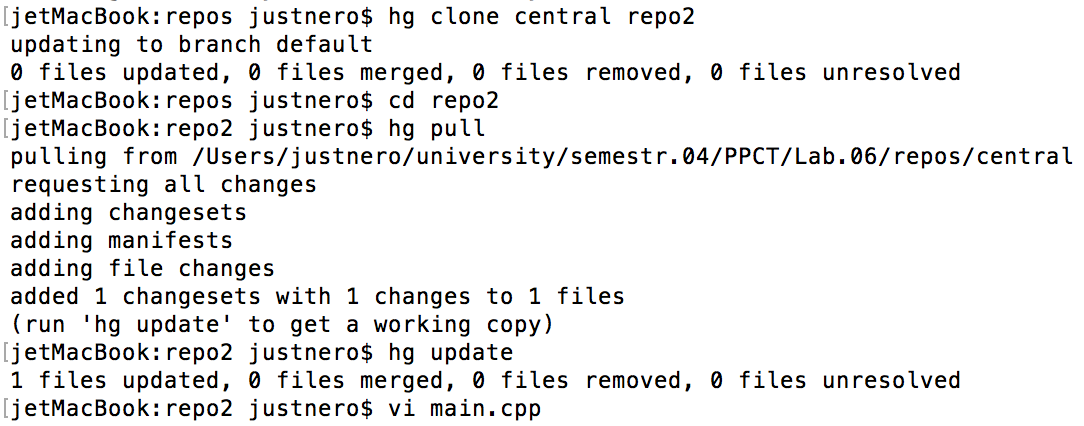


Рисунок 4 – Работа с репозиторием 2

Клонирование на пустого центрального репозитория в локальный репозиторий третьего пользователя. И обновление на этапе второго чейндж сета. Изменение файла main.cpp, его коммит и попытка пуша в центральный репозиторий, где возникает конфликт с 3м чейндж сетом от второго пользователя. Для решения этой проблемы производится пулл из центрального репозитория и мердж, который в последствии коммитится и пушится в центральный репозиторий.

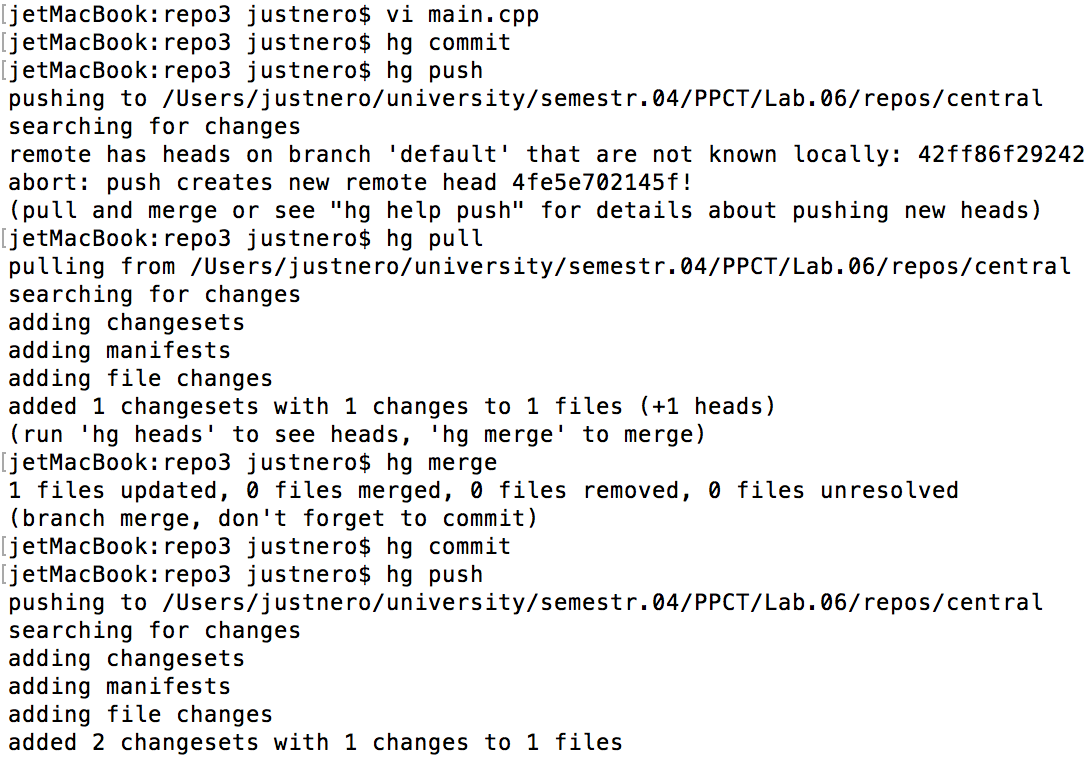
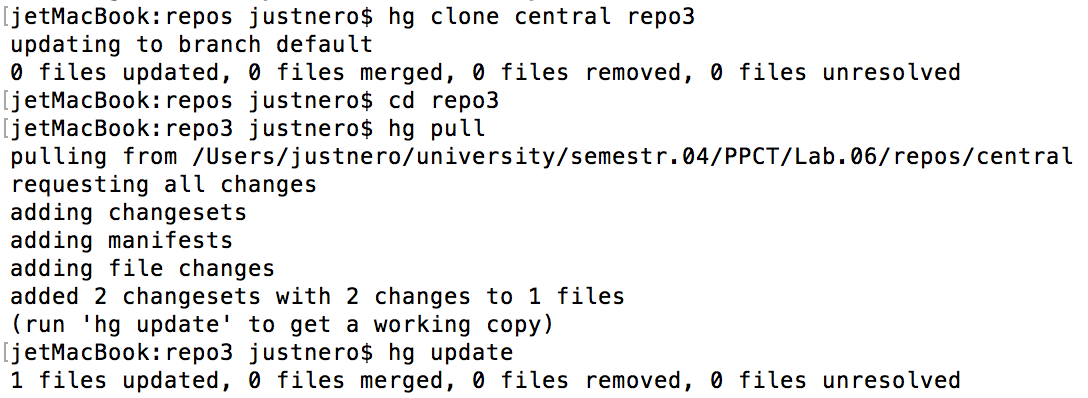


Рисунок 5 – Работа с репозиторием 3

Продолжая работу с 3 репозиторием изменяется файл main.cpp и изменения коммитятся и пушатся в центральный репозиторий.

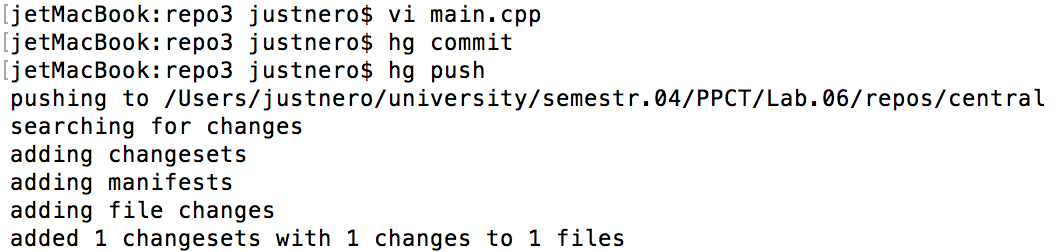


Рисунок 5 – Продолжение работы с репозиторием 3

В это же время в первом репозитории изменяется та же строка файла main.cpp, изменения коммитятся, но при пуше возникает конфликт. Для решения конфликта изменения пулятся из центральнго репозитория и вручную мерджатся, путём выбора изменений из первого репозитория. Решённый конфликт коммитится и пушится в центральный репозиторий.

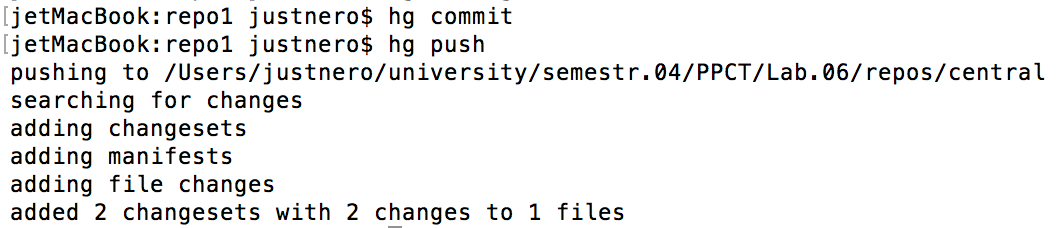
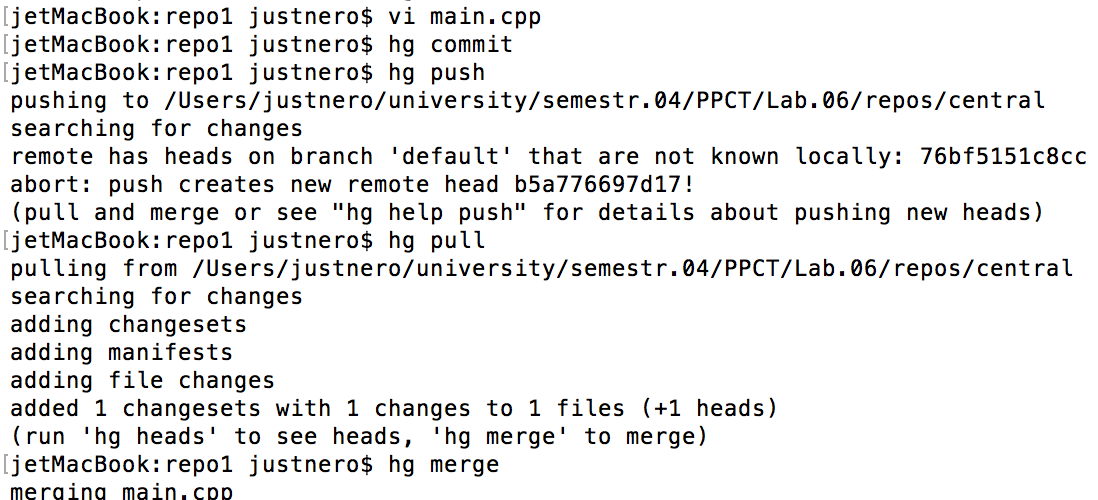


Рисунок 7 – Решение конфликта в репозитории 1

В репозиторий 2 подтягиваются все произведенные изменения и создаётся ветка run, в которой исправляется баг в файле main.cpp и производится его компиляция в файл a.out, после чего все файлы в текущей директории добавляются в индекс и коммитятся. Для пуша с созданием новой ветки в центральном репозитории используется ключ –new-branch.

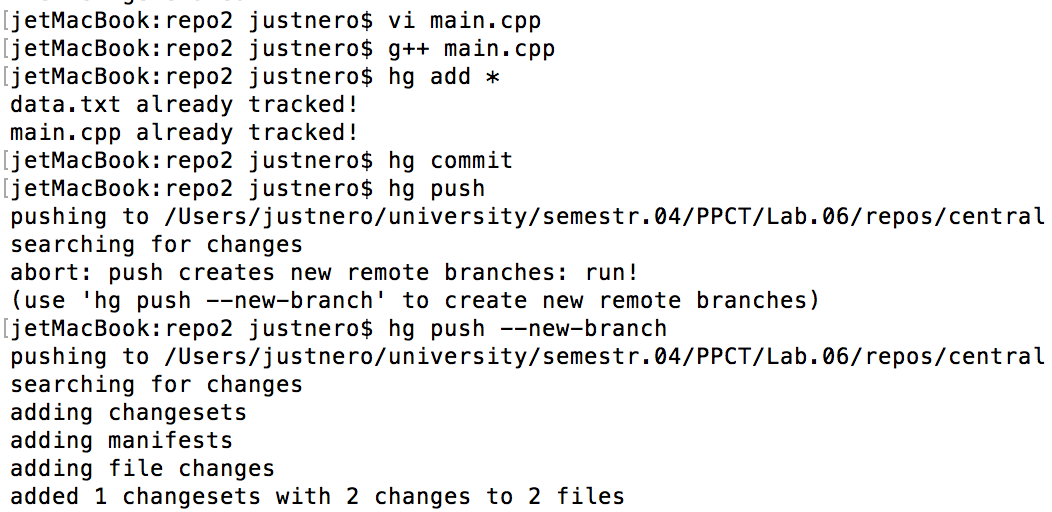
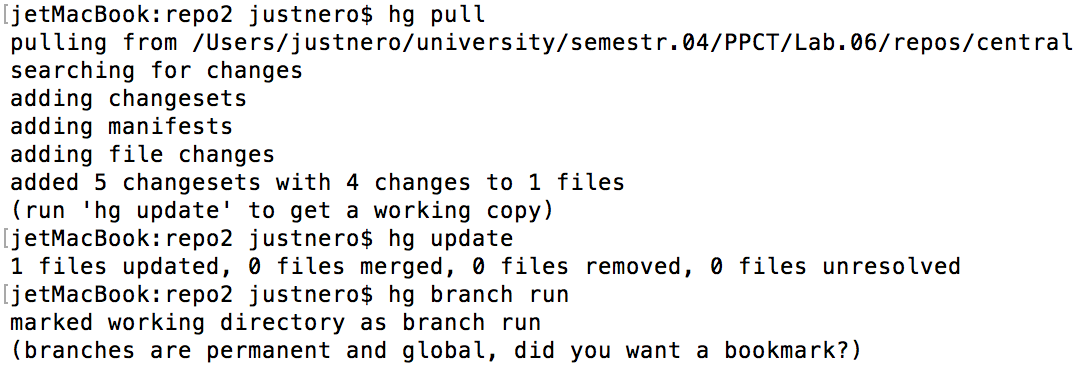


Рисунок 8 – Работа с ветками в репозитории 2

Выводы

В ходе лабораторной работы были исследованы основные подходы к организации взаимодействия команды разработчиков с использованием распределенной системы контроля версий (DVCS). Приобретены практические навыки установки и настройки DVCS Mercurial, организации ветвей разработки и осуществления слияния.