СПб НИУ ИТМО

кафедра ИПМ

Основы программной инженерии

Лабораторная работа № 2

Работа с системами управления версиями GIT и SVN

Вариант 3345

Работу выполнил:

Студент II курса

Группы № 2120

Журавлев Виталий

Санкт-Петербург

2013 г.

**Цель работы:**

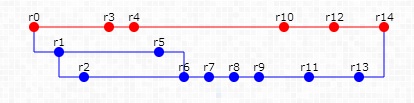
Сконфигурировать в своём домашнем каталоге репозитории svn и git и загрузить в них начальную ревизию файлов с исходными кодами (в соответствии с выданным вариантом).

Воспроизвести последовательность команд для систем контроля версий svn и git, осуществляющих операции над исходным кодом, приведённые на блок-схеме.

При составлении последовательности команд необходимо учитывать следующие условия:

* Цвет элементов схемы указывает на пользователя, совершившего действие (красный - первый, синий - второй).
* Цифры над узлами - номер ревизии. Ревизии создаются последовательно.
* Необходимо разрешать конфликты между версиями, если они возникают.

Вариант 3345:



**Ход работы:**

**SVN**

$ mkdir svn //создаем папку для хранения данных

$ cd svn //и переходим в нее

$ svnadmin create --fs-type fsfs /home/device/svn/project //инициализируем svn

$ svn import --username=Vitalik /home/device/commits/commit0 file:///home/device/svn/project -m "revision1" //добавляем файлы

$ svn copy file:///home/device/svn/project/trunk file:///home/device/svn/project/branches/serega -m="Creating new branch serega" //создаем новую ветку

$ svn switch [file:///home/device/svn/project/branches/serega](file:///home/svn/project/branches/serega) //переходим в нее

$ svn add \* //добавляем след. файлы

$ svn commit --username=Serega -m "revision1" //сохраняем новую ветку

$ svn copy file:///home/device/svn/project/brunches/serega file:///home/device/svn/project/branches/serega2 -m="Creating new branch serega2"

$ svn switch file:///home/device/svn/project/branches/serega2

$ svn add \*

$ svn commit --username=Serega -m "revision2"

$ svn switch file:///home/device/svn/project/trunk

$ svn add \*

$ svn commit --username=Vitalik -m "revision3"

$ svn add \*

$ svn commit --username=Vitalik -m "revision4"

$ svn switch file:///home/device/svn/project/branches/serega

$ svn add \*

$ svn commit --username=Serega -m "revision5"

$ svn switch file:///home/device/svn/project/branches/serega2

$ svn add \*

$ svn commit --username=Serega -m "revision6"

$ svn merge file:///home/device/svn/project/branches/serega //слияние веток

$ svn add \*

$ svn commit --username=Serega -m "revision7"

$ svn add \*

$ svn commit --username=Serega -m "revision8"

$ svn add \*

$ svn commit --username=Serega -m "revision9"

$ svn switch file:///home/device/svn/project/trunk

$ svn add \*

$ svn commit --username=Serega -m "revision10"

$ svn switch file:///home/device/svn/project/branches/serega2

$ svn add \*

$ svn commit --username=Serega -m "revision11"

$ svn switch file:///home/device/svn/project/branches/serega2

$ svn add \*

$ svn commit --username=Serega -m "revision12"

$ svn switch file:///home/device/svn/project/branches/serega2

$ svn add \*

$ svn commit --username=Serega -m "revision13"

$ svn switch file:///home/device/svn/project/trunk

$ svn add \*

$ svn commit --username=Serega -m "revision14"

$ svn merge file:///home/device/svn/project/branches/serega2 //слияние веток

$ svn resolved \* //сообщение о том, что все конфликты решены

**Git:**

$ mkdir git //создаем каталог git

$ cd git //и переходим в него

$ git init //инициализируем репозиторий git

$ git add \* //добавляем в репозиторий файлы первой версии

$ git config --global user.name ‘’Vitalik” //описываем имя пользователя

$ git config --global user.email tiger116km@gmail.com //и его e-mail

$ git commit –m “revision0” //сохраняем состояние с меткой “revision0”

$ git branch Serega //создаем новую ветку

$ git checkout Serega //и переходим в нее

$ git add \* //добавляем файлы

$ git config --global user.name ‘’Serega” //задаем имя другого пользователя

$ git config --global user.email tiger116km@yandex.ru //и его e-mail

$ git commit –m “revision1” //сохраняем состояние

$ git branch trunk //создаем ветку trunk от ветки Serega (см. блок-схему)

$ git checkout trunk //и переходим в нее

$ git add \* --All //добавляем файлы (с заменой и удалением предыдущих)

$ git config --global user.name ‘’Serega” //задаем имя пользователя

$ git config --global user.email tiger116km@yandex.ru //и его e-mail

$ git commit –m “revision2” //сохраняем изменения

$ git merge Serega -m “merge branchs” //сливаем ветку Serega с trunk

…

далее идет процесс добавления и сохранения изменений по тому же принципу, что и в svn

//при каждом новом сохранении (commit) вся информация об изменении в данных записывается в файл, что упрощает возврат на предыдущую версию.

//при попытке слияния веток trunk и master происходила ошибка из-за несовместимости строк кода.

$ git merge trunk -m “merge branchs”

//Вручную были найдены дублирующиеся блоки кода:

B B.java:

public long ac() { return 333; }

public float ff() { return 3.14; }

public void ab() { System.out.println(); }

B D.java:

public Object rr() { return null; }

//После их удаления в рабочих копиях и повторном добавлении в репозиторий ошибка при слиянии устранилась, и впоследствие слияние выполнилось успешно

**Вывод:**

В ходе лабораторной работы я изучил основные аспекты работы с системами контроля версии: SVN и GIT.

Более удобным в работе оказался GIT благодаря более простой системе работы с branch\merge. Так же он быстрее работает, поскольку весь репозиторий расположен локально (в git только папка .git в корне, а у SVN — в каждой папке).

Однако SVN более мощная и универсальная. Так же ей пользуются большинство крупных компаний, что говорит о ее популярности.