

**Университет ИТМО**

**Факультет программной инженерии и компьютерных технологий**

**Лабораторная работа №2 по Методам и Средствам Программной Инженерии**

**Выполнили: Богатов Александр Сергеевич**

**Ерехинский Андрей Владимирович**

**Группа: Р3233**

**Вариант: 1424**

**Преподаватель: Цопа Евгений Алексеевич**

**Санкт-Петербург**

**2022**

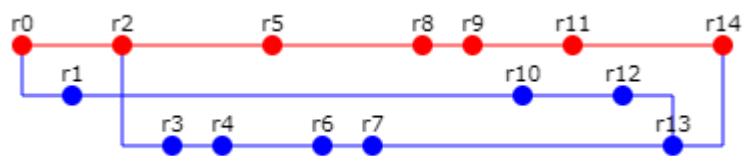
**Задание:**

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.



**Список команд, использованных при создании и конфигурации репозитория в домашнем каталоге пользователя:**

- Svn:
  - svnadmin create
  - svn mkdir
  - svn checkout
  - svn add
  - svn commit
  - svn copy
  - svn switch
  - svn rm
  - svn merge
  - svn resolve
- Git:
  - git init
  - git config
  - git add
  - git commit
  - git checkout
  - git rm
  - git merge
  - git mergetool

**Последовательности команд:**

Ревизия	Subversion	Комментарии
---------	------------	-------------

Конфигурация репозитория	<pre> svnadmin create repo  svn mkdir -m "structure" file:///home/s312418/MSP/Lab2SVN/repo/trunk file:///home/s312418/MSP/Lab2SVN/repo/branches  svn checkout file:///home/s312418/MSP/Lab2SVN/repo/trunk mainDir  cd mainDir </pre>	Создаем репозиторий и стандартную структуру svn репозитория.
r0	<pre> cp ~/MSP/projectFiles/commit0/* .  svn add *  svn commit -m "r0" --username redUser </pre>	<p>Файлы для каждой ревизии содержатся в projectFiles.</p> <p>Добавляем все перемещенные файлы и делаем коммит с указанием пользователя опцией --username</p>
r1	<pre> svn copy file:///home/s312418/MSP/Lab2SVN/repo/trunk file:///home/s312418/MSP/Lab2SVN/repo/branches/second -m "Creating second branch" --username blueUser  svn switch file:///home/s312418/MSP/Lab2SVN/repo/branches/second  cp ~/MSP/projectFiles/commit1/* .  svn commit -m "r1" --username blueUser </pre>	Создаем новую ветку репозитория, в ней проводим аналогичные действия.
r2	<pre> svn switch file:///home/s312418/MSP/Lab2SVN/repo/trunk  cp ~/MSP/projectFiles/commit2/* .  svn commit -m "r2" --username redUser </pre>	

r3	<pre>svn copy file:///home/s312418/MSP/Lab2SVN/repo/trunk file:///home/s312418/MSP/Lab2SVN/repo/branches/third - m "Creating third branch" --username blueUser  svn switch file:///home/s312418/MSP/Lab2SVN/repo/branches/third  cp ~/MSP/projectFiles/commit3/* .  svn add _  svn commit -m "r3" --username blueUser</pre>	Добавляем новые файлы в отслеживаемые svn с помощью svn add.
r4	<pre>cp ~/MSP/projectFiles/commit4/* .  svn rm _  svn commit -m "r4" --username blueUser</pre>	
r5	<pre>svn switch file:///home/s312418/MSP/Lab2SVN/repo/trunk  cp ~/MSP/projectFiles/commit5/* .  svn commit -m "r5" --username redUser</pre>	
r6	<pre>svn switch file:///home/s312418/MSP/Lab2SVN/repo/branches/third  cp ~/MSP/projectFiles/commit6/* .  svn commit -m "r6" --username blueUser</pre>	
r7	<pre>cp ~/MSP/projectFiles/commit7/* .  svn add CH6dS59igv.0Fh  svn commit -m "r7" --username blueUser</pre>	
r8	<pre>svn switch file:///home/s312418/MSP/Lab2SVN/repo/trunk  cp ~/MSP/projectFiles/commit8/* .  svn add CH6dS59igv.0Fh  svn commit -m "r8" --username redUser</pre>	

r9	<pre>cp ~/MSP/projectFiles/commit9/* . svn rm CH6dS59igv.0Fh svn add 4MTUEE020F.0tZ svn commit -m "r9" --username redUser</pre>	Файлы, не присутствующие в новой ревизии, убираем с помощью <code>svn rm</code>
r10	<pre>svn switch file:///home/s312418/MSP/Lab2SVN/repo/branches/second cp ~/MSP/projectFiles/commit10/* . svn commit -m "r10" --username blueUser</pre>	
r11	<pre>svn switch file:///home/s312418/MSP/Lab2SVN/repo/trunk cp ~/MSP/projectFiles/commit11/* . svn rm 4MTUEE020F.0tZ svn add tZ4MTUEE02.Lc0 svn commit -m "r11" --username redUser</pre>	
r12	<pre>svn switch file:///home/s312418/MSP/Lab2SVN/repo/branches/second cp ~/MSP/projectFiles/commit12/* . svn add CH6dS59igv.0Fh svn commit -m "r12" --username blueUser</pre>	
r13	<pre>svn switch file:///home/s312418/MSP/Lab2SVN/repo/branches/third svn merge file:///home/s312418/MSP/Lab2SVN/repo/branches/second svn resolve --accept=working CH6dS59igv.0Fh cp ~/MSP/projectFiles/commit13/* . svn add WMLc0tZ4MT.FI3 svn add _ svn rm CH6dS59igv.0Fh svn commit -m "r13" --username blueUser</pre>	При появлении конфликта состояния файла вручную устраняем его и указываем, что конфликт разрешен командой <code>svn resolve – accept=working</code> .

r14	<pre> svn switch file:///home/s312418/MSP/Lab2SVN/repo/trunk svn merge file:///home/s312418/MSP/Lab2SVN/repo/branches/third tc tc cp ~/MSP/projectFiles/commit14/* . svn rm tZ4MTUEE02.Lc0 svn rm _ --force svn rm WMLc0tZ4MT.FI3 --force svn commit -m "r14" --username redUser </pre>	При конфликте слияния разных версий файла выбираем принять входящие версии файлов в обоих конфликтах.
-----	---	---

Ревизия	Git	Комментарии
Конфигурация репозитория	<pre> git init repo cd repo git config --global user.name "redUser" git config --global user.email "redUser@itmo.ru" </pre>	Создаем репозиторий git, задаем пользователя на ближайший коммит
r0	<pre> cp ~/MSP/projectFiles/commit0/* . git add . git commit -m "r0" </pre>	Добавляем все перемещенные файлы
r1	<pre> git config --global user.name "blueUser" git config --global user.email "blueUser@itmo.ru" git checkout -b second cp ~/MSP/projectFiles/commit1/* . git add . git commit -m "r1" </pre>	git checkout с флагом -b создает новую ветку в репозитории.

r2	<pre>git config --global user.name "redUser" git config --global user.email "redUser@itmo.ru" git checkout master cp ~/MSP/projectFiles/commit2/* . git add . git commit -m "r2"</pre>	Использование git add . добавляет все файлы в список отслеживаемых в репозитории, но в отличие от svn add * не выводит ошибок, при попытке добавить уже отслеживаемый файл.
r3	<pre>git config --global user.name "blueUser" git config --global user.email "blueUser@itmo.ru" git checkout -b third cp ~/MSP/projectFiles/commit3/* . git add . git commit -m "r3"</pre>	
r4	<pre>cp ~/MSP/projectFiles/commit4/* . git add . git rm _ git commit -m "r4"</pre>	
r5	<pre>git config --global user.name "redUser" git config --global user.email "redUser@itmo.ru" git checkout master cp ~/MSP/projectFiles/commit5/* . git add . git commit -m "r5"</pre>	

r6	git config --global user.name "blueUser" git config --global user.email "blueUser@itmo.ru" git checkout third cp ~/MSP/projectFiles/commit6/* . git add . git commit -m "r6"	
r7	cp ~/MSP/projectFiles/commit7/* . git add . git commit -m "r7"	
r8	git config --global user.name "redUser" git config --global user.email "redUser@itmo.ru" git checkout master cp ~/MSP/projectFiles/commit8/* . git add . git commit -m "r8"	
r9	cp ~/MSP/projectFiles/commit9/* . git add . git rm CH6dS59igv.0Fh git commit -m "r9"	
r10	git config --global user.name "blueUser" git config --global user.email "blueUser@itmo.ru" git checkout second cp ~/MSP/projectFiles/commit10/* . git add . git commit -m "r10"	



r11	<pre>git config --global user.name "redUser"  git config --global user.email "redUser@itmo.ru"  git checkout master  cp ~/MSP/projectFiles/commit11/* .  git add .  git rm 4MTUEE020F.0tZ  git commit -m "r11"</pre>	
r12	<pre>git config --global user.name "blueUser"  git config --global user.email "blueUser@itmo.ru"  git checkout second  cp ~/MSP/projectFiles/commit12/* .  git add .  git commit -m "r12"</pre>	
r13	<pre>git checkout third  git merge --no-commit second  cp ~/MSP/projectFiles/commit13/* .  git rm CH6dS59igv.0Fh  git add .  git commit -m "r13"</pre>	В отличии от svn, здесь не возникло конфликта, git автоматически слил два одинаковых файла в 1.
r14	<pre>git config --global user.name "redUser"  git config --global user.email "redUser@itmo.ru"  git checkout master  git merge --no-commit third  git mergetool  ...  cp ~/MSP/projectFiles/commit14/* .  git add .</pre>	Здесь при возникновении конфликта вызываем git mergetool и решаем конфликт через него – можно отредактировать файл или принять одну из существующих версий. Слияние завершится при коммите.

	<pre>git rm _ git rm tZ4MTUEE02.Lc0 git rm WMLc0tZ4MT.FI3 -f git commit -m "r14"</pre>	
--	--	--

**Вывод:**

При выполнении данной лабораторной работы были рассмотрены системы контроля версий Subversion и Git. В связи с простотой структуры создаваемого репозитория, особых трудностей не возникло и порядок работы с разными СКВ получился примерно одинаковым. Было замечено, что автоматическое слияние в SVN не работает с идентичными файлами, созданными в разных ветках, СКВ попросит ручного разрешения конфликта. Также, система веток в SVN реализована созданием отображения ветки в файловой системе, в Git же создается новый указатель на текущий коммит.