



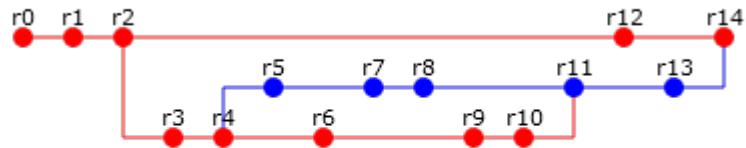
УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники
Методы и средства программной инженерии

Лабораторная работа №2
Вариант №1167

Преподаватели: Письмак Алексей Евгеньевич
Выполнили: Кульбако Артемий Юрьевич, Исаков Антонов Сергеевич
Р3212

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.

Git

Коммит	Команды
init	git init gRepo #создаем репозиторий cd gRepo
r0	git config --global user.name "red" git config --global user.email "red@m.com" cp ../src/com0/* . git add . git commit -m "r0"
r1	# делаем изменения в локальном репозитории git add . git commit -m "r1"
r2	# делаем изменения в локальном репозитории git add . git commit -m "r2"
r3	git checkout -b branch2 # делаем изменения в локальном репозитории git add . git commit -m "r3"
r4	# делаем изменения в локальном репозитории git add . git commit -m "r4"
r5	git config --global user.name "blue" git config --global user.email "blue@m.com" git checkout -b branch3 # делаем изменения в локальном репозитории git add . git commit -m "r5"
r6	git config --global user.name "red" git config --global user.email "red@m.com" git checkout branch2 # делаем изменения в локальном репозитории git add . git commit -m "r6"
r7	git config --global user.name "blue" git config --global user.email "blue@m.com" git checkout branch3 # делаем изменения в локальном репозитории

	git add . git commit -m "r7"
r8	# делаем изменения в локальном репозитории git add . git commit -m "r8"
r9	git config --global user.name "red" git config --global user.email "red@m.com" git checkout branch2 # делаем изменения в локальном репозитории git add . git commit -m "r9"
r10	# делаем изменения в локальном репозитории git add . git commit -m "r10"
r11	git config --global user.name "blue" git config --global user.email "blue@m.com" git checkout branch3 git merge --no-commit branch2 # делаем изменения в локальном репозитории git add . git commit -m "r11"
r12	git config --global user.name "red" git config --global user.email "red@m.com" git checkout master # делаем изменения в локальном репозитории git add . git commit -m "r12"
r13	git config --global user.name "blue" git config --global user.email "blue@m.com" git checkout branch3 # делаем изменения в локальном репозитории git add . git commit -m "r13"
r14	git config --global user.name "red" git config --global user.email "red@m.com" git checkout master git merge --no-commit branch3 # делаем изменения в локальном репозитории git add . git commit -m "r14"

Subversion

Коммит	Команды
Подготовка	svnadmin create repo #создаём репозиторий cd repo svn mkdir -m "project structure" file:///home/s265570/SEB/repo/trunk file:///home/s265570/SEB/repo/branches #создаём структуру проекта cd .. svn checkout file:///home/s265570/SEB/repo/trunk/ workDir #создаём рабочую копию cd workDir
r0	cp ~/SEB/src/com0/* . svn add * svn commit -m "r0" --username=Pupa
r1	svn rm * #удаляем файлы из рабочей директории (если это не сделать, то нельзя будет сделать коммит, т.к. новые файлы ничем не отличаются от старых)

	<pre>cp ~/SEB/src/com1/* . svn add * svn commit -m "r1" --username=Pupa</pre>
r2	<pre>svn rm * cp ~/SEB/src/com2/* . svn add * svn commit -m "r2" --username=Pupa</pre>
r3	<pre>svn copy file:///home/s265570/SEB/repo/trunk file:///home/s265570/SEB/repo/branches/branch2 -m "created branch2" svn switch file:///home/s265570/SEB/repo/branches/branch2 svn rm * cp ~/SEB/src/com3/* . svn add * svn commit -m "r3" --username=Pupa</pre>
r4	<pre>svn rm * cp ~/SEB/src/com4/* . svn add * svn commit -m "r4" --username=Pupa</pre>
r5	<pre>svn copy file:///home/s265570/SEB/repo/branches/branch2 file:///home/s265570/SEB/repo/branches/branch3 -m "created branch3" --username=Lupa svn switch file:///home/s265570/SEB/repo/branches/branch3 svn rm * cp ~/SEB/src/com5/* . svn add * svn commit -m "r5" --username=Lupa</pre>
r6	<pre>svn switch file:///home/s265570/SEB/repo/branches/branch2 svn rm * cp ~/SEB/src/com6/* . svn add * svn commit -m "r6" --username=Pupa</pre>
r7	<pre>svn switch file:///home/s265570/SEB/repo/branches/branch3 svn rm * cp ~/SEB/src/com7/* . svn add * svn commit -m "r7" --username=Lupa</pre>
r8	<pre>svn rm * cp ~/SEB/src/com8/* . svn add * svn commit -m "r8" --username=Lupa</pre>
r9	<pre>svn switch file:///home/s265570/SEB/repo/branches/branch2 svn rm * cp ~/SEB/src/com9/* . svn add * svn commit -m "r9" --username=Pupa</pre>
r10	<pre>svn rm * cp ~/SEB/src/com10/* . svn add * svn commit -m "r10" --username=Pupa</pre>
r11	<pre>svn switch file:///home/s265570/SEB/repo/branches/branch3 svn merge file:///home/s265570/SEB/repo/branches/branch2 svn rm * --force cp ~/SEB/src/com11/* . svn add * svn commit -m "r11" --username=Lupa</pre>
r12	<pre>svn switch file:///home/s265570/SEB/repo/trunk svn rm * cp ~/SEB/src/com12/* . svn add * svn commit -m "r12" --username=Pupa</pre>
r13	<pre>svn switch file:///home/s265570/SEB/repo/branches/branch3</pre>

	<pre>svn rm * cp ~/SEB/src/com13/* . svn add * svn commit -m "r13" --username=Lupa</pre>
r14	<pre>svn switch file:///home/s265570/SEB/repo/trunk svn merge file:///home/s265570/SEB/repo/branches/branch3 svn rm * --force cp ~/SEB/src/com14/* . svn add * svn commit -m "r14" --username=Pupa</pre>

Вывод

Лабораторная №2 познакомила нас с двумя популярными системами контроля версий – Git и Subversion. Системы концептуально различны: git – распределённая, svn – централизованная.

1. При работе с svn возникают исключительно ужасные чувства: неинформативный вывод (commit и status ничего не выводят, если файлы не изменялись, что вводит в ступор).
2. Неудобный add * (необходимость использовать --force для прохода по вложенным директориям)
3. revert без аргументов ничего не делает, как говорят разработчики «*If you provide no targets to svn revert, it will do nothing—to protect you from accidentally losing changes in your working copy, svn revert requires you to provide at least one target*», но, это неудобно: ожидаешь отката с последнему коммиту, а получаешь в ответ снова пустую строку.
4. Структура проекта в репозитории непрозрачная: созданные в ней директории существуют, но явно не отображаются в файловой системе: нельзя манипулировать ими команды операционной системы и использовать автодополнение по нажатию Tab.

В связи с вышеизложенным, нет ощущения, что ты пользуешься надёжным инструментом, и, я бы предпочёл вручную копировать директории и файлы, чем пользоваться svn.

С другой стороны git. Никаких лишних действий — одной командой создал репозиторий и можно сразу приступить к проекту. Информативный вывод — даже если ни один файл не изменился status скажет об этом и покажет имя текущей ветки. Более информативный help — если в svn это небольшая справка, то в git полноценная man страница.

Системы также отличаются техническим устройством веток: svn создаёт копию необходимых файлов, т. е. как таковой «ветки» не существует, мы лишь воспринимаем эту копию как ветку (копирование работает аналогично жёстким ссылкам в Unix-системах), git же вычисляет хэш-сумму объектов и создаёт указатель на новые внутренний объект коммита, что позволяет избавиться от отображения в файловой системе.