



УНИВЕРСИТЕТ ИТМО

Факультет программной
инженерии и компьютерной техники

Лабораторная работа № 2
по дисциплине
«Методы и средства программной инженерии»

Вариант 3300

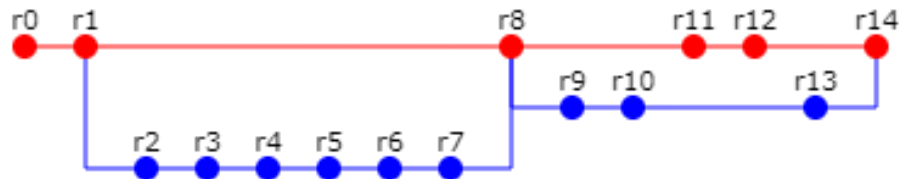
Преподаватель: Покид Александр Владимирович

Выполнили: Геллер Л. А., Блинова М. А.

Группа: Р3230

Санкт-Петербург
2021 г.

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

Git:

init - инициализируем репозиторий
config - использовали для задания имя и email пользователя
add - добавить файл (готовим коммит)
commit - коммит (фиксируем изменения в текущей ветке)
checkout - переключить на ветку, создать ветку (-b)
merge - подгружает изменения из веток в рабочую копию, учитывая структуру

Svn:

svnadmin create - создать репозиторий
import - загружаем в репозиторий
checkout(co) - загрузить рабочую копию из репозитория
add - добавить файл (готовим коммит)
commit (ci) - коммит (фиксируем изменения в текущей ветке)
switch (sw) - обновить рабочую копию другим URL
update - обновить рабочую копию (можно указать версию)
merge - подгружает изменения из веток в рабочую копию, учитывая структуру
copy (cp) - Копируем в рабочую копию или хранилище

Ревизии GIT

	<code>git init</code>	Создаем репозиторий
r0	<code>git config --local user.name user1</code> <code>git config --local user.email user1@mail.com</code> <code>git add .</code> <code>git commit -m r0</code>	Первый пользователь коммитит свой код
r1	<code>git add .</code> <code>git commit -m r1</code>	
r2	<code>git checkout -b branch1</code> <code>git config --local user.name user2</code> <code>git config --local user.email user2@mail.com</code> <code>git add .</code> <code>git commit -m r2</code>	Второй пользователь создает свою ветку и коммитит туда свой код
r3	<code>git add .</code> <code>git commit -m r3</code>	
r4	<code>git add .</code> <code>git commit -m r4</code>	
r5	<code>git add .</code> <code>git commit -m r5</code>	
r6	<code>git add .</code> <code>git commit -m r6</code>	
r7	<code>git add .</code> <code>git commit -m r7</code>	

r8	git checkout master git config --local user.name user1 git config --local user.email user1@mail.com git merge branch1 -m master git add . git commit -m r8	Первый пользователь добавляет к себе изменения с ветки второго и коммитит свой код
r9	git checkout -b branch2 git config --local user.name user2 git config --local user.email user2@mail.com git add . git commit -m r9	Второй пользователь коммитит код в свою ветку
r10	git add . git commit -m r10	
r11	git checkout master git config --local user.name user1 git config --local user.email user1@mail.com git add . git commit -m r11	Первый пользователь коммитит код в свою ветку (master)
r12	git add . git commit -m r12	
r13	git checkout branch2 git config --local user.name user2 git config --local user.email user2@mail.com git add . git commit -m r13	Второй пользователь коммитит код в свою ветку
r14	git checkout master git config --local user.name user1 git config --local user.email user1@mail.com git merge branch2 -m master git add . git commit -m r14	Первый пользователь добавляет к себе изменения с ветки второго и коммитит свой код

Ревизии SVN

(1)	mkdir work svnadmin create repo cd work svn import /home/s282518/MISPI/lab_2/SVN/work file:///home/s282518/MISPI/lab_2/SVN/repo/trunk -m "Init trunk"	Создаем trunk
r0 (2)	svn co file:///home/s282518/MISPI/lab_2/SVN/repo/trunk svn add * svn ci -m r0 --username=user1	Первый пользователь коммитит свой код в trunk
r1 (3)	svn add * --force svn ci -m r1 --username=user1	
r2 (4)	svn co file:///home/s282518/MISPI/lab_2/SVN/repo/trunk svn add * --force svn ci -m r2 --username=user2	
r3 (5)	svn add * --force svn ci -m r3 --username=user2	Второй пользователь коммитит свой код в trunk
r4 (6)	svn add * --force svn ci -m r4 --username=user2	
r5 (7)	svn add * --force svn ci -m r5 --username=user2	
r6 (8)	svn add * --force svn ci -m r6 --username=user2	
r7 (9)	svn add * --force svn ci -m r7 --username=user2	
r8 (10)	svn update svn merge file:///home/s282518/MISPI/lab_2/SVN/repo/trunk svn add * --force svn ci -m r8 --username=user1	Первый пользователь добавляет к себе изменения второго и коммитит свой код в trunk
r9 (12)	svn update svn cp file:///home/s282518/MISPI/lab_2/SVN/repo/trunk file:///home/s282518/MISPI/lab_2/SVN/repo/branch2 -m "Init branch2" --username=user2	Второй пользователь коммитит код в свою ветку

	<pre>svn co file:///home/s282518/MISPI/lab_2/SVN/repo/branch2 svn add * --force svn ci -m r9 --username=user2</pre>	
r10 (13)	<pre>svn add * --force svn ci -m r10 --username=user2</pre>	
r11 (14)	<pre>svn update (mc) svn add * --force svn ci -m r11 --username=user1</pre>	Первый пользователь коммитит код
r12 (15)	<pre>svn add * --force svn ci -m r12 --username=user1</pre>	
r13 (16)	<pre>svn add * --force svn ci -m r13 --username=user2</pre>	Второй пользователь коммитит код в свою ветку
r14 (17)	<pre>svn update svn merge file:///home/s282518/MISPI/lab_2/SVN/repo/branch2 // solve conflict svn add * --force svn ci -m r14 --username=user1</pre>	Первый пользователь добавляет к себе изменения второго и коммитит свой код в trunk

Вывод

При выполнении данной лабораторной работы мы столкнулись с двумя системами контроля версий – svn и git.