Федеральное государственное автономное образовательное учреждение высшего образования **«Национальный исследовательский университет ИТМО»**

Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа №2**

по дисциплине **«Основы программной инженерии»**

Вариант: **323610**

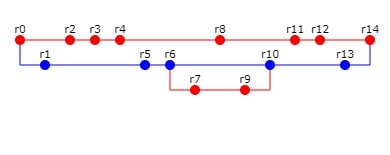
**Преподаватель**:   
Кулинич Ярослав Вадимович

**Выполнил**: Нодири Хисравхон

**Группа**: P3231

Санкт-Петербург, 2024

# 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).

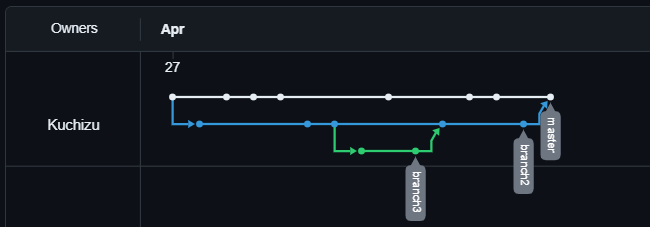
Воспроизвести последовательность команд для систем контроля версий svn и git, осуществляющих операции над исходным кодом, приведённые на блок-схеме.

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.

# Реализация с использованием Git

|  |
| --- |
| #!/bin/bash  # Создание репозитория и начальная ревизия (r0)  git init  git remote add origin https://github.com/Kuchizu/Lab-2\_OPI.git  echo "- git init"  # Настройка .git/config  echo -e "\n[merge]\n\ttool = nano" >> .git/config  # Настройка пользователя  git config user.name "red"  git config user.email "red@example.com"  echo "- Пользователь red создан"  # Ревизия r0 (пользователь 1) {  unzip -o commits/commit0.zip -d src  git add .  git commit -m "Initial commit (r0)"  git push -u origin master  echo "- Коммит 0 (red)"  # }  # Ревизия r1 (пользователь 2) {  git checkout -b branch2  unzip -o commits/commit1.zip -d src  git add .  git commit --author="blue <blue@example.com>" -m "Revision 1 (r1)"  git push -u origin branch2  echo "- Коммит 1 (blue)"  # }  # Ревизии r2-r4 (пользователь 1) {  git checkout master  unzip -o commits/commit2.zip -d src  git add .  git commit -m "Revision 2 (r2)"  git push -u origin master  echo "- Коммит 2 (red)"  unzip -o commits/commit3.zip -d src  git add .  git commit --allow-empty -m "Revision 3 (r3)"  git push -u origin master  echo "- Коммит 3 (red)"  unzip -o commits/commit4.zip -d src  git add .  git commit --allow-empty -m "Revision 4 (r4)"  git push -u origin master  echo "- Коммит 4 (red)"  # }  # Ревизии r5-r6 (пользователь 2) {  git checkout branch2  unzip -o commits/commit5.zip -d src  git add .  git commit --allow-empty --author="blue <blue@example.com>" -m "Revision 5 (r5)"  git push -u origin branch2  echo "- Коммит 5 (blue)"  unzip -o commits/commit6.zip -d src  git add .  git commit --allow-empty --author="blue <blue@example.com>" -m "Revision 6 (r6)"  echo "- Коммит 6 (blue)"  git push -u origin branch2  # }  # Ревизии r7-r9 (пользователь 1) {  git checkout -b branch3  unzip -o commits/commit7.zip -d src  git add .  git commit --allow-empty -m "Revision 7 (r7)"  git push -u origin branch3  git checkout master  unzip -o commits/commit8.zip -d src  git add .  git commit --allow-empty -m "Revision 8 (r8)"  git push -u origin master  git checkout branch3  unzip -o commits/commit9.zip -d src  git add .  git commit --allow-empty -m "Revision 9 (r9)"  git push -u origin branch3  # }  # Ревизия r10 (пользователь 2) {  git checkout branch2  git merge --no-ff branch3 -m "Merging two branches -> r10"  unzip -o commits/commit10.zip -d src  git add .  git commit -m "Revision 10 (r10)"  git push -u origin branch2  # }  # Ревизии r11-r12 (пользователь 1) {  git checkout master  unzip -o commits/commit11.zip -d src  git add .  git commit --allow-empty -m "Revision 11 (r11)"  unzip -o commits/commit12.zip -d src  git add .  git commit --allow-empty -m "Revision 12 (r12)"  git push -u origin master  # }  # Ревизия r13 (пользователь 2) {  git checkout branch2  unzip -o commits/commit13.zip -d src  git add .  git commit --author="blue <blue@example.com>" -m "Revision 13 (r13)"  git push -u origin branch2  # }  # Ревизия r14 (пользователь 1) {  git checkout master  git checkout --ours "src/\*"  git merge branch2 -m "Merging two branches -> r14"  git push -u origin master  # } |



# Реализация с использованием SVN

|  |
| --- |
| #!/bin/bash  # Создание локального репозитория  svnadmin create repo  REPO\_URL="file://$(pwd)/repo"  # Создаём структуру проекта  cd repo  svn mkdir $REPO\_URL/trunk $REPO\_URL/branches -m "Project structure"  cd ..  # Создание рабочей копии  svn checkout $REPO\_URL/trunk wc  cd wc  # Начальная ревизия r0 (пользователь 1) {  echo $RANDOM > commit1.txt  svn add commit1.txt  svn commit -m "Initial commit (r0)" --username red  echo "- Ревизия 0 (red)"  # }  # Создание ветки branch2 и ревизия r1 (пользователь 2) {  svn copy $REPO\_URL/trunk $REPO\_URL/branches/branch2 -m "Creating branch2"  svn switch $REPO\_URL/branches/branch2  echo $RANDOM > commit1.txt  svn add commit1.txt  svn commit -m "Revision 1 (r1)" --username blue  echo "- Ревизия 1 (blue)"  # }  # Возвращение к trunk и ревизии r2-r4 (пользователь 1) {  svn switch $REPO\_URL/trunk  **for** i **in** 2 3 4; **do**  echo $RANDOM > "commit$i.txt"  svn add "commit$i.txt"  svn commit -m "Revision $i (r$i)" --username red  echo "- Ревизия $i (red)"  **done**  # }  # Ревизии r5-r6 (пользователь 2) на ветке branch2:  svn switch $REPO\_URL/branches/branch2  **for** i **in** 5 6; **do**  echo $RANDOM > "commit$i.txt"  svn add "commit$i.txt"  svn commit -m "Revision $i (r$i)" --username blue  echo "- Ревизия $i (blue)"  **done**  # }  # Создание ветки branch3 и ревизии r7-r9 (пользователь 1):  svn copy $REPO\_URL/trunk $REPO\_URL/branches/branch3 -m "Creating branch3"  svn switch $REPO\_URL/branches/branch3  echo $RANDOM > "commit7.txt"  svn add "commit7.txt"  svn commit -m "Revision 7 (r7)" --username red  echo "- Ревизия 7 (red)"  svn switch $REPO\_URL/trunk  echo $RANDOM > "commit8.txt"  svn add "commit8.txt"  svn commit -m "Revision 8 (r8)" --username red  echo "- Ревизия 8 (red)"  svn switch $REPO\_URL/branches/branch3  echo $RANDOM > "commit9.txt"  svn add "commit9.txt"  svn commit -m "Revision 9 (r9)" --username red  echo "- Ревизия 9 (red)"  # }  # Слияние ветки branch3 в branch2 и ревизия r10 (пользователь 2):  svn switch $REPO\_URL/branches/branch2  svn merge --accept postpone $REPO\_URL/branches/branch3  # Решение конфликтов здесь вручную, затем:  svn resolved commit10.txt  echo $RANDOM > commit10.txt  svn add commit10.txt  svn commit -m "Revision 10 (r10)" --username blue  echo "- Ревизия 10 (blue)"  # }  # Ревизии r11-r12 (пользователь 1) в trunk:  svn switch $REPO\_URL/trunk  **for** i **in** 11 12; **do**  echo $RANDOM > "commit$i.txt"  svn add "commit$i.txt"  svn commit -m "Revision $i (r$i)" --username red  echo "- Ревизия $i (red)"  **done**  # }  # Ревизия r13 (пользователь 2) в branch2:  svn switch $REPO\_URL/branches/branch2  echo $RANDOM > commit13.txt  svn add commit13.txt  svn commit -m "Revision 13 (r13)" --username blue  echo "- Ревизия 13 (blue)"  # }  # Слияние ветки branch2 в trunk и ревизия r14 (пользователь 1):  svn switch $REPO\_URL/trunk  svn merge --accept postpone $REPO\_URL/branches/branch2  # Решение конфликтов здесь вручную, затем:  svn resolved commit14.txt  echo $RANDOM > commit14.txt  svn add commit14.txt  svn commit -m "Revision 14 (r14)" --username red  echo "- Ревизия 14 (red)"  # }  # Обновление рабочей копии  svn update  echo "Рабочая копия обновлена"  cd .. |

# Вывод

В ходе выполнения лабораторной работы я улучшил свои навыки владения системой контроля версий Git, а также познакомился с Subversion. Во время выполнения работы были настроены репозитории SVN и Git в домашнем каталоге пользователя, загружены начальные ревизии файлов с исходными кодами, а также выполнены операции над исходным кодом в соответствии с блок-схемой. Были изучены основные команды SVN и Git. Практическая работа позволила лучше понять принципы работы систем контроля версий и их практическое применение в различных сценариях разработки.