

Национальный исследовательский университет ИТМО
Факультет программной инженерии и компьютерной техники

Методы и средства программной инженерии.

Лабораторная работа №2.

Вариант: 1428

Выполнили:

Газизов Анвар Васильевич

Романов Артём Максимович

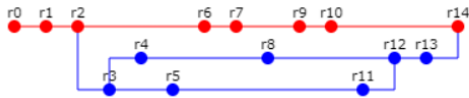
Поток: 1.4

Преподаватель: Цопа Е. А.

Санкт-Петербург

2022

1. 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.

2. Список команд

Git:

```
git init
git config
git add
git commit
git checkout
git merge
```

Svn:

```
svnadmin create
svn import
svn checkout
svn add
svn commit
svn mkdir
svn copy
svn switch
svn merge
svn resolve
```

3. Последовательности команд

Git:

r0:

```
git init
git config --global user.email
"artrom1511@gmail.com"
git config --global user.name "Artyom"
cp ../commits/commit0/* .
git add .
git commit -m "initial commit"
```

r1:

```
cp ../commits/commit1/* .
git add .
git commit -m "r1"
```

r2:

```
cp ../commits/commit2/* .
git add .
git commit -m "r2"
```

r3:

```
git config --global user.email
"anvariphone2016@gmail.com"
git config --global user.name "Anvar"

cp ../commits/commit3/* .
git checkout -b br2
git add .
```

r4:

```
# On branch br1 nothing to commit, working directory clean
git checkout -b br1
cp ../commits/commit4/* .
git add .
git commit -m "r4"
```

r5:

```
git checkout br2
cp ../commits/commit5/* .
```

```
git add .
git commit -m "r5"
r6:
git config --global user.email
"artrom1511@gmail.com"
git config --global user.name "Artyom"
git checkout master
cp ../commits/commit6/* .
git add .
git commit -m "r6"
r7:
cp ../commits/commit7/* .
git add .
git commit -m "r7"
r8:
git config --global user.email
"anvariphone2016@gmail.com"
git config --global user.name "Anvar"
git checkout br1
cp ../commits/commit8/* .
git add .
git commit -m "r8"
r9:
git config --global user.email
"artrom1511@gmail.com"
git config --global user.name "Artyom"
cp ../commits/commit9/* .
git add .
git commit -m "r9"
r10:
cp ../commits/commit10/* .
git add .
git commit -m "r10"
r11:
git config --global user.email
"anvariphone2016@gmail.com"
git config --global user.name "Anvar"
git checkout br2
```

```

    cp ../commits/commit11/* .
    git add .
    git commit -m "r11"
merge br2 -> br1:
    git checkout br1
    git merge br2 --no-commit
    # CONFLICT (content): Merge conflict in F.java
    # Automatic merge failed; fix conflicts and then commit the result.
    git add F.java
    git commit -m "merge with fixes"
r12:
    cp ../commits/commit12/* .
    git add .
    git commit -m "r12"
r13:
    cp ../commits/commit13/* .
    git add .
    git commit -m "r13"
merge br1 -> master:
    git checkout master
    git merge br1 --no-commit
    # CONFLICT (content): Merge conflict in F.java
    # Auto-merging *
    # CONFLICT (add/add): Merge conflict in *
    # Automatic merge failed; fix conflicts and then commit the result.
    git add \*
    git add F.java
    git commit -m "merge br1 -> master"
r14:
    git config --global user.email
"artrom1511@gmail.com"
    git config --global user.name "Artyom"
    git checkout master
    cp ../commits/commit14/* .
    git add .
    git commit -m "r14"

```

Svn:

r0:

```
svnadmin create origin
COMMITTS="/home/s307360/commits/"
REMOTE_URL="file://$(pwd -P)/origin"
svn import $COMMITTS/commit0 $REMOTE_URL/trunk -m
"r0" --username=Artyom
svn checkout $REMOTE_URL/trunk working_copy
cd working_copy
```

r1:

```
cp $COMMITTS/commit1/* .
svn add yL2zIxMVjN.Ve5
svn commit -m "r1" --username=Artyom
```

r2:

```
cp $COMMITTS/commit2/* .
svn add 5yL2zIxMVj.9Ve
svn commit -m "r2" --username=Artyom
```

r3:

```
svn mkdir $REMOTE_URL/branches -m "Add branches"
--username=Anvar
svn copy $REMOTE_URL/trunk
$REMOTE_URL/branches/br2 -m "Add br2" -username=Anvar
svn switch $REMOTE_URL/branches/br2
cp $COMMITTS/commit3/* .
svn commit -m "r3" --username=Anvar
```

r4:

```
svn copy $REMOTE_URL/branches/br2
$REMOTE_URL/branches/br1 -m "Add br1" -username=Anvar
svn switch $REMOTE_URL/branches/br1
cp $COMMITTS/commit4/* .
svn commit -m "r4" --username=Anvar
```

r5:

```
svn switch $REMOTE_URL/branches/br2
cp $COMMITTS/commit5/* .
svn commit -m "r5" --username=Anvar
```

r6:

```
svn switch $REMOTE_URL/trunk
cp $COMMITTS/commit6/* .
```

```
svn commit -m "r6" --username=Artyom
r7:
cp $COMMITITS/commit7/* .
svn commit -m "r7" --username=Artyom
r8:
svn switch $REMOTE_URL/branches/br1
cp $COMMITITS/commit8/* .
svn commit -m "r8" --username=Anvar
r9:
svn switch $REMOTE_URL/trunk
cp $COMMITITS/commit9/* .
svn add \*
svn commit -m "r9" --username=Artyom
r10:
cp $COMMITITS/commit10/* .
svn add RU609Ve5yL.zql
svn commit -m "r10" --username=Artyom
r11:
svn switch $REMOTE_URL/branches/br2
cp $COMMITITS/commit11/* .
svn commit -m "r11" --username=Anvar
merge br2 -> br1:
svn switch $REMOTE_URL/branches/br1
svn merge $REMOTE_URL/branches/br2
tf
svn commit -m "merge br2 -> br1" --username=Anvar

r12:
cp $COMMITITS/commit12/* .
svn commit -m "r12" --username=Anvar
r13:
cp $COMMITITS/commit13/* .
svn add \*
svn add hzY4zqlRU6.qtO
svn commit -m "r13" --username=Anvar
```

```
merge br1 -> trunk:
    svn switch $REMOTE_URL/trunk
    svn merge $REMOTE_URL/branches/br1
    mf
    svn resolve --accept=working \*
    svn commit -m "merge br1 -> trunk"
--username=Anvar
r14:
    cp $COMMITTS/commit14/* .
    svn commit -m "r14" --username=Artyom
```

4. Вывод:

В результате выполнения данной лабораторной работы мы ознакомились с системами контроля версий svn и git и получили базовые навыки работы с этими инструментами.