

# Университет ИТМО

Факультет программной инженерии и компьютерной техники

Дисциплина «Основы программной инженерии»

## Отчет

По лабораторной работе №2

Вариант: 320408

Выполнил:

*Терновский И.Е*

Санкт-Петербург, 2024 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

Для системы контроля git:

```
git config
git add
git checkout
git merge
```

Для системы контроля svn:

```
svn propset
svn rm
svn add
svn commit
svn copy
svn switch
svnadmin create
svn mkdir
```

Последовательность команд:

## Git:

```
changeToFirst() {
#Меняет пользователя на первого(красного)
    git config --local user.name first
    git config --local user.email first@red.com
}

changeToSecond() {
#Меняет пользователя на второго(синего)
    git config --local user.name second
    git config --local user.email second@blue.com
}

commit() {
# копируем с перезаписью представленную ревизию
    cp -f ../commits/commit$1/* .
# добавляем все файлы в гит (-а не сработало)
    git add .
# делаем коммит в репозиторий ревизии
    git commit --allow-empty -m "r$1"
}

branch() {
# создаем и сразу переходим на новую ветку
    git checkout -b "$1"
}

checkout() {
# переходим на новую ветку
    git checkout "$1"
}

merge() {
# объединяем в текущую ветку, другую ветку (без коммита)
    git merge $1 --no-commit
}

mergestrat() {
# используется если нужно разрешить конфликт, стратегия передается 2 аргументом
    git merge -s $2 $1 --no-commit
}

#init
rm -rf git
mkdir git
cd git
git init -b "1"

#r0
changeToFirst
commit 0

#r1
changeToFirst
commit 1

#r2
changeToSecond
branch 3
commit 2

#r3
changeToFirst
branch 4
commit 3

#r4
changeToFirst
commit 4

#r5
changeToSecond
branch 2
commit 5
```

#r6  
changeToSecond  
commit 6

#r7  
changeToSecond  
checkout 3  
commit 7

#r8  
changeToFirst  
checkout 1  
commit 8

#r9  
changeToFirst  
checkout 4  
commit 9

#r10-11  
changeToFirst  
checkout 1  
commit 10  
commit 11

#r12-13  
changeToSecond  
checkout 3  
commit 12  
commit 13

#r14  
changeToFirst  
checkout 4  
commit 14

#r15  
changeToFirst  
checkout 1  
commit 15

#r16  
changeToFirst  
checkout 4  
commit 16

#r17  
changeToFirst  
checkout 1  
commit 17

#r18  
changeToFirst  
checkout 4  
commit 18

#r19  
changeToFirst  
checkout 1  
commit 19

#r20  
changeToSecond  
checkout 3  
commit 20

#r21  
changeToFirst  
checkout 4  
commit 21

#r22  
changeToSecond  
checkout 2  
merge 4  
commit 22

```

#r23
changeToFirst
checkout 1
commit 23

#r24
changeToSecond
checkout 3
commit 24

#r25
changeToFirst
checkout 1
commit 25

#r26
changeToSecond
checkout 2
commit 26

#r27-29
changeToFirst
checkout 1
commit 27
commit 28
commit 29

#r30
changeToSecond
checkout 2
commit 30

#r31
changeToSecond
checkout 3
mergestrat 3 ours
commit 31

#r32
changeToFirst
checkout 1
mergestrat 3 ours
commit 32

```

## svn:

```

changeToFirst() {
#Меняет пользователя на первого(красного)
  svn propset svn:author first --revprop -r HEAD
}

changeToSecond() {
#Меняет пользователя на второго(синего)
  svn propset svn:author second --revprop -r HEAD
}

commit() {
  svn rm * --force
  cp ../../commits/commit"$1"/* .
  svn add * --force # force для каталогов
  svn commit -m revision"$1"
}

branch() {
  echo "Creating branch$1"
  svn copy ^/trunk ^/branches/branch"$1" -m "Creating branch $1"
}

checkout() {
  echo "Checking out branch$1"
  svn switch ^/branches/branch"$1"
}

```

```

#same as checkout branch0 in git
checkoutTrunk() {
    echo "Checking out trunk"
    svn switch ^/trunk
}
#postpone (по умолчанию): оставляет файлы в состоянии конфликта для последующего ручного разрешения.
#base: выбирает базовую версию.
#working: выбирает рабочую версию (т.е. вашу локальную версию).
#mine-conflict: выбирает вашу версию для конфликтных изменений.
#theirs-conflict: выбирает их версию для конфликтных изменений.
#mine-full: выбирает вашу версию для всех изменений.
#theirs-full: выбирает их версию для всех изменений
merge() {
    svn merge --accept working ^/branches/branch"$1"
}

# init
rm -rf svn
mkdir svn
cd svn || exit

REMOTE_URL="file://$(pwd -P)/repo"

svnadmin create repo
svn mkdir "$REMOTE_URL"/trunk "$REMOTE_URL"/branches -m "init"
# создаем скрипт который будет разрешать любые изменения в метаданных репозитория
cd repo/hooks
echo '#!/bin/sh'
exit 0' > pre-revprop-change
chmod +x pre-revprop-change
cd ../..

# Создание рабочей копии
svn checkout "$REMOTE_URL"/trunk working_copy
cd working_copy

#r0
changeToFirst
commit 0

#r1
changeToFirst
commit 1

#r2
changeToSecond
branch 3
checkout 3
commit 2

#r3
changeToFirst
branch 4
checkout 4
commit 3

#r4
changeToFirst
commit 4

#r5
changeToSecond
branch 2
checkout 2
commit 5

#r6
changeToSecond
commit 6

#r7
changeToSecond
checkout 3
commit 7

#r8

```

changeToFirst  
checkoutTrunk  
commit 8

#r9  
changeToFirst  
checkout 4  
commit 9

#r10-11  
changeToFirst  
checkoutTrunk  
commit 10  
changeToFirst  
commit 11

#r12-13  
changeToSecond  
checkout 3  
commit 12  
commit 13

#r14  
changeToFirst  
checkout 4  
commit 14

#r15  
changeToFirst  
checkoutTrunk  
commit 15

#r16  
changeToFirst  
checkout 4  
commit 16

#r17  
changeToFirst  
checkoutTrunk  
commit 17

#r18  
changeToFirst  
checkout 4  
commit 18

#r19  
changeToFirst  
checkoutTrunk  
commit 19

#r20  
changeToSecond  
checkout 3  
commit 20

#r21  
changeToFirst  
checkout 4  
commit 21

#r22  
changeToSecond  
checkout 2  
merge 4  
commit 22

#r23  
changeToFirst  
checkoutTrunk  
commit 23

#r24  
changeToSecond  
checkout 3  
commit 24

```
#r25
changeToFirst
checkoutTrunk
commit 25

#r26
changeToSecond
checkout 2
commit 26

#r27-29
changeToFirst
checkoutTrunk
commit 27
changeToFirst
commit 28
changeToFirst
commit 29

#r30
changeToSecond
checkout 2
commit 30

#r31
changeToSecond
checkout 3
merge 3
commit 31

#r32
changeToFirst
checkoutTrunk
merge 3
changeToFirst
commit 32
```

## Вывод:

В ходе выполнения работы я познакомился с системами контроля версий git и svn, узнал какие есть стратегии разрешения конфликтов, а так же, что самое важное, понял, что svn никогда в жизни использовать не буду.