

Министерство науки и высшего образования Российской Федерации
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: 368508

Преподаватель:

Харитоновна Анастасия Евгеньевна

Выполнили:

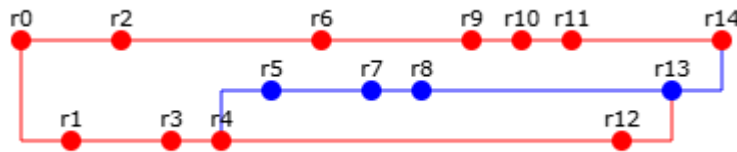
Медведев Владислав Александрович

Михайлов Дмитрий Андреевич

Группа: P3206

Санкт-Петербург, 2025

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.

Git

```
# Инициализация нового Git репозитория
# git init - создаёт новый репозиторий Git в текущей
директории
git init

# Создание и коммит первого файла (Revision 0)
# echo "Revision 0" > file.txt - создает файл "file.txt" с
содержимым "Revision 0"
# git config user.name "red" - устанавливает имя пользователя
"red" для текущих коммитов
# git config user.email "red@example.com" - устанавливает
email пользователя "red" для текущих коммитов
# git add . - добавляет все изменения в индекс для коммита
# git commit -m "Revision 0 (red)" - создает коммит с
сообщением "Revision 0 (red)"
echo "Revision 0" > file.txt
git config user.name "red"
git config user.email "red@example.com"
```

```
git add .
git commit -m "Revision 0 (red)"

# Создание новой ветки red-branch_2 и переключение на неё
# git checkout -b red-branch_2 - создаёт новую ветку
"red-branch_2" и переключается на неё
git checkout -b red-branch_2

# Создание и коммит второго файла в ветке red-branch_2
(Revision 1)
# echo "Revision 1" > file.txt - изменяет содержимое
"file.txt"
# git add . - добавляет изменения в индекс
# git commit -m "Revision 1 (red)" - создаёт коммит с
сообщением "Revision 1 (red)"
echo "Revision 1" > file.txt
git add .
git commit -m "Revision 1 (red)"

# Переключение на ветку master
# git checkout master - переключает на ветку "master"
git checkout master

# Создание коммита на ветке master (Revision 2)
# echo "Revision 2" > file.txt - изменяет содержимое
"file.txt"
# git add . - добавляет изменения в индекс
# git commit -m "Revision 2 (red)" - создаёт новый коммит
echo "Revision 2" > file.txt
git add .
git commit -m "Revision 2 (red)"

# Возврат в ветку red-branch_2 и создание третьего коммита
(Revision 3)
# git checkout red-branch_2 - снова переключается на ветку
"red-branch_2"
git checkout red-branch_2
echo "Revision 3" > file.txt
git add .
git commit -m "Revision 3 (red)"

# Создание четвертого коммита в ветке red-branch_2 (Revision
4)
echo "Revision 4" > file.txt
```

```
git add .
git commit -m "Revision 4 (red)"

# Создание новой ветки blue-branch_1 и переключение на неё
# git checkout -b blue-branch_1 - создаёт ветку
"blue-branch_1" и переключается на неё
git checkout -b blue-branch_1

# Изменение имени и email для пользователя blue
# git config user.name "blue" - меняет имя пользователя для
коммитов в ветке
# git config user.email "blue@example.com" - меняет email
пользователя для коммитов в ветке
git config user.name "blue"
git config user.email "blue@example.com"

# Создание и коммит в ветке blue-branch_1 (Revision 5)
echo "Revision 5" > file.txt
git add .
git commit -m "Revision 5 (blue)"

# Возврат к пользователю red и коммит на master ветке
(Revision 6)
git config user.name "red"
git config user.email "red@example.com"
git checkout master
echo "Revision 6" > file.txt
git add .
git commit -m "Revision 6 (red)"

# Создание коммита в ветке blue-branch_1 (Revision 7)
git checkout blue-branch_1
echo "Revision 7" > file.txt
git add .
git commit -m "Revision 7 (blue)"

# Создание коммита в ветке blue-branch_1 (Revision 8)
echo "Revision 8" > file.txt
git add .
git commit -m "Revision 8 (blue)"

# Возврат к пользователю red и коммит на master ветке
(Revision 9)
git config user.name "red"
```

```
git config user.email "red@example.com"
git checkout master
echo "Revision 9" > file.txt
git add .
git commit -m "Revision 9 (red)"

# Создание и коммит нового файла (Revision 10)
echo "Revision 10" > file.txt
git add .
git commit -m "Revision 10 (red)"

# Коммит с изменением файла (Revision 11)
echo "Revision 11" > file.txt
git add .
git commit -m "Revision 11 (red)"

# Возврат в ветку red-branch_2 и создание коммита (Revision 12)
git checkout red-branch_2
echo "Revision 12" > file.txt
git add .
git commit -m "Revision 12 (red)"

# Слияние ветки red-branch_2 в ветку blue-branch_1 с
комментарием "Revision 13"
# git merge red-branch_2 -m "Revision 13" - сливает изменения
из ветки "red-branch_2" в текущую ветку "blue-branch_1"
git checkout blue-branch_1
git merge red-branch_2 -m "Revision 13"

# Редактирование файла и завершение слияния с коммитом
# vim file.txt - открывает файл для редактирования
# git add file.txt - добавляет измененный файл в индекс
# git commit -m "Revision 13 (red)" - создает коммит с
сообщением "Revision 13 (red)"
vim file.txt
git add file.txt
git commit -m "Revision 13 (red)"

# Переключение на master ветку, изменение имени и email
пользователя blue
git checkout master
git config user.name "blue"
git config user.email "blue@example.com"
```

```
vim file.txt
git add .
git config user.name "red"
git config user.email "red@example.com"
git commit -m "Revision 14 (red)"

# Просмотр истории коммитов с графом
# git log --oneline --graph --all --decorate - показывает
историю коммитов в графическом виде, отображая все ветки и
метки
git log --oneline --graph --all --decorate
```

SVN

```
svnadmin create svnRepo
cd svnRepo/
export PATHF=file:///home/studs/s368530/Lab2_opi/svnRepo
svn mkdir ${PATHF}/trunk ${PATHF}/branches ${PATHF}/tags -m
"feature: create repo"
svn checkout ${PATHF}/trunk workspace
cd workspace
```

```
#r0
cp -f ~/Lab2_opi/commits/commit0/* ./
svn add *
svn commit -m "r0" --username="red"
```

```
#r1
svn copy ${PATHF}/trunk/ ${PATHF}/branches/feature-1 -m
"feature-1" --username="red"
svn switch ${PATHF}/branches/feature-1
cp -f ~/Lab2_opi/commits/commit1/* ./
svn add *
svn commit -m "r1" --username="red"
```

```
#r2
svn switch ${PATHF}/trunk
cp -f ~/Lab2_opi/commits/commit2/* ./
svn add *
svn commit -m "r2" --username="red"
```

```
#r3
svn switch ${PATHF}/branches/feature-1
```

```
cp -f ~/Lab2_opi/commits/commit3/* ./
svn add *
svn commit -m "r3" --username="red"
```

```
#r4
cp -f ~/Lab2_opi/commits/commit4/* ./
svn add *
svn commit -m "r4" --username="red"
```

```
#r5
svn copy ${PATHF}/branches/feature-1
${PATHF}/branches/feature-2 -m "feature-2" --username="blue"
svn switch ${PATHF}/branches/feature-2
cp -f ~/Lab2_opi/commits/commit5/* ./
svn add *
svn commit -m "r5" --username="blue"
```

```
#r6
svn switch ${PATHF}/trunk
cp -f ~/Lab2_opi/commits/commit6/* ./
svn add *
svn commit -m "r6" --username="red"
```

```
#r7
svn switch ${PATHF}/branches/feature-2
cp -f ~/Lab2_opi/commits/commit7/* ./
svn add *
svn commit -m "r7" --username="blue"
```

```
#r8
cp -f ~/Lab2_opi/commits/commit8/* ./
svn add *
svn commit -m "r8" --username="blue"
```

```
#r9
svn switch ${PATHF}/trunk
cp -f ~/Lab2_opi/commits/commit9/* ./
svn add *
svn commit -m "r9" --username="red"
```

```
#r10
cp -f ~/Lab2_opi/commits/commit10/* ./
svn add *
svn commit -m "r10" --username="red"
```

```
#r11
cp -f ~/Lab2_opi/commits/commit11/* ./
svn add *
svn commit -m "r11" --username="red"

#r12
svn switch ${PATHF}/branches/feature-1
cp -f ~/Lab2_opi/commits/commit12/* ./
svn add *
svn commit -m "r12" --username="red"

#r13
svn switch ${PATHF}/branches/feature-2
svn merge ${PATHF}/branches/feature-1
${PATHF}/branches/feature-2 --username="blue"
svn resolved ${PATHF}/workspace/F.java
cp -f ~/Lab2_opi/commits/commit13/* ./
svn add *
svn commit -m "r13" --username="blue"

#r14
svn switch ${PATHF}/trunk
svn merge ${PATHF}/branches/feature-2 ${PATHF}/trunk
--username="red"
cp -f ~/Lab2_opi/commits/commit14/* ./
svn add *
svn commit -m "r14" --username="red"
```

Вывод

В ходе выполнения лабораторной работы мы улучшили свои навыки владения системой контроля версий Git, а также познакомились 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, а также способы разрешения конфликтов. Практическая работа позволила лучше понять принципы работы систем контроля версий и их практическое применение в различных сценариях разработки.