Федеральное государственное автономное образовательное учреждение высшего образования Национальный исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

Основы программной инженерии

Лабораторная работа №2

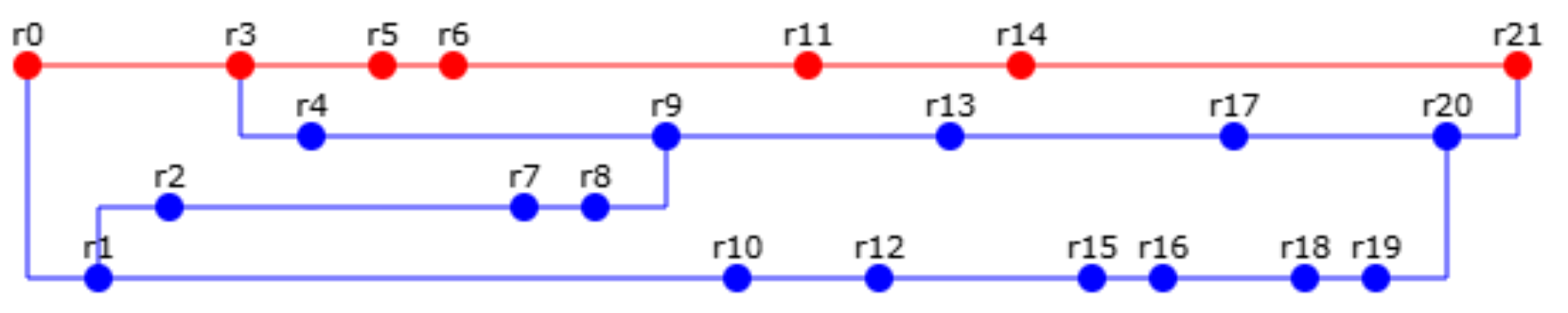
Вариант № 210455

Выполнил: студент группы P3208, Васильев Н. А.

Преподаватель: Воронина Д. С.

Санкт-Петербург 2025

# Текст задания



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, использованных при создании и конфигурации репозиториев в домашнем каталоге пользователя.

Для Git:

# Инициируем репозиторий

git init

# Выбираем красного пользователя

git config user.name "red"

git config user.email "red@domain.com"

# Коммит (r0)

cp -r /home/studs/s366389/opi/commits/commit0/\* .

git add .

git commit -m "commit0"

# Меняем пользователя на синего

git config user.name "blue"

git config user.email "blue@domain.com"

# Создаем новую ветку b1 и делаем в ней коммит (r1)

git checkout -b "b1"

cp -r /home/studs/s366389/opi/commits/commit1/\* .

git add .

git commit -m "commit1"

# Создаем новую ветку b2 и делаем в ней коммит (r2)

git checkout -b "b2"

cp -r /home/studs/s366389/opi/commits/commit2/\* .

git add .

git commit -m "commit2"

# Меняем пользователя на красного

git config user.name "red"

git config user.email "red@domain.com"

# Переключаемся на ветку master и делаем в ней коммит (r3)

git checkout "master"

cp -r /home/studs/s366389/opi/commits/commit3/\* .

git add .

git commit -m "commit3"

# Меняем пользователя на синего

git config user.name "blue"

git config user.email "blue@domain.com"

# Создаем новую ветку b3 и делаем в ней коммит (r4)

git checkout -b "b3"

cp -r /home/studs/s366389/opi/commits/commit4/\* .

git add .

git commit -m "commit4"

# Меняем пользователя на красного

git config user.name "red"

git config user.email "red@domain.com"

# Переключаемся на ветку master и делаем в ней коммит (r5)

git checkout "master"

cp -r /home/studs/s366389/opi/commits/commit5/\* .

git add .

git commit -m "commit5"

# Делаем коммит (r6)

cp -r /home/studs/s366389/opi/commits/commit6/\* .

git add .

git commit -m "commit6"

# Меняем пользователя на синего

git config user.name "blue"

git config user.email "blue@domain.com"

# Переключаемся на ветку b2 и делаем в ней коммит (r7)

git checkout "b2"

cp -r /home/studs/s366389/opi/commits/commit7/\* .

git add .

git commit -m "commit7"

# Делаем коммит (r8)

cp -r /home/studs/s366389/opi/commits/commit8/\* .

git add .

git commit -m "commit8"

# Делаем слияние ветки b2 в b3

git checkout "b3"

git merge "b2"

# Делаем коммит (r9)

cp -r /home/studs/s366389/opi/commits/commit9/\* .

git add .

git commit -m "commit9"

# Переключаемся на ветку b1 и делаем в ней коммит (r10)

git checkout "b1"

cp -r /home/studs/s366389/opi/commits/commit10/\* .

git add .

git commit -m "commit10"

# Меняем пользователя на красного

git config user.name "red"

git config user.email "red@domain.com"

# Переключаемся на ветку master и делаем в ней коммит (r11)

git checkout "master"

cp -r /home/studs/s366389/opi/commits/commit11/\* .

git add .

git commit -m "commit11"

# Меняем пользователя на синего

git config user.name "blue"

git config user.email "blue@domain.com"

# Переключаемся на ветку b1 и делаем в ней коммит (r12)

git checkout "b1"

cp -r /home/studs/s366389/opi/commits/commit12/\* .

git add .

git commit -m "commit12"

# Переключаемся на ветку b3 и делаем в ней коммит (r13)

git checkout "b3"

cp -r /home/studs/s366389/opi/commits/commit13/\* .

git add .

git commit -m "commit13"

# Меняем пользователя на красного

git config user.name "red"

git config user.email "red@domain.com"

# Переключаемся на ветку master и делаем в ней коммит (r14)

git checkout "master"

cp -r /home/studs/s366389/opi/commits/commit14/\* .

git add .

git commit -m "commit14"

# Меняем пользователя на синего

git config user.name "blue"

git config user.email "blue@domain.com"

# Переключаемся на ветку b1 и делаем в ней коммит (r15)

git checkout "b1"

cp -r /home/studs/s366389/opi/commits/commit15/\* .

git add .

git commit -m "commit15"

# Делаем коммит (r16)

cp -r /home/studs/s366389/opi/commits/commit16/\* .

git add .

git commit -m "commit16"

# Переключаемся на ветку b3 и делаем в ней коммит (r17)

git checkout "b3"

cp -r /home/studs/s366389/opi/commits/commit17/\* .

git add .

git commit -m "commit17"

# Переключаемся на ветку b1 и делаем в ней коммит (r18)

git checkout "b1"

cp -r /home/studs/s366389/opi/commits/commit18/\* .

git add .

git commit -m "commit18"

# Делаем коммит (r19)

cp -r /home/studs/s366389/opi/commits/commit19/\* .

git add .

git commit -m "commit19"

# Делаем слияние ветки b1 в b3

git checkout "b3"

git merge "b1"

# Выбираем нужную версию для текстовых файлов, для бинарного используем checkout --theirs

git checkout --theirs \_

git add C.java F.java I.java \_

git commit

# Делаем коммит (r20)

cp -r /home/studs/s366389/opi/commits/commit20/\* .

git add .

git commit -m "commit20"

2

# Меняем пользователя на красного

git config user.name "red"

git config user.email "red@domain.com"

# Делаем слияние ветки b3 в master

git checkout "master"

git merge "b3"

# Выбираем нужную версию для текстовых файлов, для бинарного используем checkout --theirs

git checkout --theirs \_

git add C.java F.java I.java \_

git commit

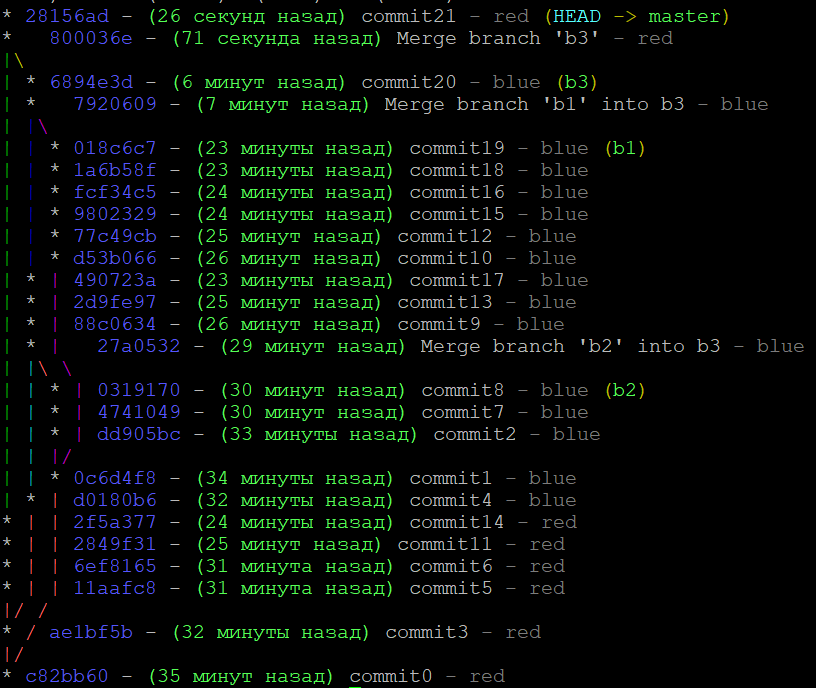
# Делаем коммит (r21)

cp -r /home/studs/s366389/opi/commits/commit21/\* .

git add .

git commit -m "commit21"

В результате получаем:



Разрешение конфликтов:

*(то, что выделено красным, удаляем)*

git merge "b1":

C.java:

<<<<<<< HEAD

public Object gg() {

return new java.util.Random();

}

public void aa() {

return;

=======

public void bb() {

System.out.println(42);

}

public Object pp() {

return this;

}

public long dd() {

return 100500;

}

public int hh() {

return new java.util.Random(10).nextInt(10);

}

public double ad() {

return 12.12;

>>>>>>> b1

F.java:

<<<<<<< HEAD

public java.util.List<String> jj() {

return new java.util.ArrayList<String>();

}

public String nn() {

return "++++++++++[>+++++++>++++++++++>+++>+<<<<-]>++";

}

=======

public String nn() {

return "++++++++++[>+++++++>++++++++++>+++>+<<<<-]>++";

}

public Object pp() {

return this;

}

public long dd() {

return 33;

}

public int af() {

return -1;

}

public int[] ii() {

return new int[]{0, 1, 2, 3, 4};

}

>>>>>>> b1

I.java:

<<<<<<< HEAD

public java.lang.Class qq() {

return getClass();

=======

public int hh() {

return new java.util.Random(10).nextInt(10);

>>>>>>> b1

}

public Object pp() {

return this;

}

<<<<<<< HEAD

=======

public String nn() {

return "++++++++++[>+++++++>++++++++++>+++>+<<<<-]>++";

}

public int cc() {

return 39;

}

public String kk() {

return "Hello world";

}

>>>>>>> b1

git merge "b3":

C.java:

<<<<<<< HEAD

public float ff() {

return 0;

=======

public void aa() {

return;

}

public int hh() {

return new java.util.Random(10).nextInt(10);

}

public String nn() {

"".>+.+++++++..+++.>++.<<+++++++++++++++.>.+++.;

>>>>>>> b3

}

public void bb() {

System.out.println(42);

}

<<<<<<< HEAD

public Object pp() {

return this;

}

public long dd() {

return 100500;

=======

public int ae() {

return java.lang.Math.abs(-7);

}

public int af() {

return -1;

>>>>>>> b3

F.java:

<<<<<<< HEAD

=======

public java.util.List<String> jj() {

return new java.util.ArrayList<String>();

}

>>>>>>> b3

public String nn() {

return "++++++++++[>+++++++>++++++++++>+++>+<<<<-]>++";

}

<<<<<<< HEAD

=======

public int af() {

return -1;

}

>>>>>>> b3

public Object pp() {

return this;

}

<<<<<<< HEAD

public long dd() {

return 33;

=======

public double ee() {

return 100.500;

}

public int[] ii() {

return new int[]{4, 3, 2, 1};

}

public long dd() {

return 99999;

}

public Object rr() {

return null;

}

public String kk() {

return "No";

>>>>>>> b3

I.java:

<<<<<<< HEAD

public java.lang.Class qq() {

return getClass();

=======

public int hh() {

return new java.util.Random(10).nextInt(10);

>>>>>>> b1

}

public Object pp() {

return this;

}

<<<<<<< HEAD

=======

public String nn() {

return "++++++++++[>+++++++>++++++++++>+++>+<<<<-]>++";

}

public int cc() {

return 39;

}

public String kk() {

return "Hello world";

}

>>>>>>> b1

Для SVN:

*# Инициируем репозиторий  
svnadmin create repo  
cd repo  
svn mkdir -m "project structure" file:///home/studs/s366389/opi/svn/repo/trunk file:///home/studs/s366389/opi/svn/repo/branches  
cd ..  
svn checkout file://$(pwd)/repo/trunk/ wc  
cd wc  
   
# Коммит (r0)  
cp -r /home/studs/s366389/opi/commits/commit0/\* .  
svn add --force .  
svn commit -m "commit0" --username=red  
   
# Создаем новую ветку b1 и делаем в ней коммит (r1)  
svn copy file:///home/studs/s366389/opi/svn/repo/trunk file:///home/studs/s366389/opi/svn/repo/branches/b1 -m "Create branch b1"  
svn switch file:///home/studs/s366389/opi/svn/repo/branches/b1  
cp -r /home/studs/s366389/opi/commits/commit1/\* .  
svn add --force .  
svn commit -m "commit1" --username=blue  
   
# Создаем новую ветку b2 и делаем в ней коммит (r2)  
svn copy file:///home/studs/s366389/opi/svn/repo/branches/b1 file:///home/studs/s366389/opi/svn/repo/branches/b2 -m "Create branch b2"  
svn switch file:///home/studs/s366389/opi/svn/repo/branches/b2  
cp -r /home/studs/s366389/opi/commits/commit2/\* .  
svn add --force .  
svn commit -m "commit2" --username=blue  
   
# Переключаемся на ветку trunk и делаем в ней коммит (r3)  
svn switch file:///home/studs/s366389/opi/svn/repo/trunk  
cp -r /home/studs/s366389/opi/commits/commit3/\* .  
svn add --force .  
svn commit -m "commit3" --username=red  
   
# Создаем новую ветку b3 и делаем в ней коммит (r4)  
svn copy file:///home/studs/s366389/opi/svn/repo/trunk file:///home/studs/s366389/opi/svn/repo/branches/b3 -m "Create branch b3"  
svn switch file:///home/studs/s366389/opi/svn/repo/branches/b3  
cp -r /home/studs/s366389/opi/commits/commit4/\* .  
svn add --force .  
svn commit -m "commit4" --username=blue  
   
# Переключаемся на ветку trunk и делаем в ней коммит (r5)  
svn switch file:///home/studs/s366389/opi/svn/repo/trunk  
cp -r /home/studs/s366389/opi/commits/commit5/\* .  
svn add --force .  
svn commit -m "commit5" --username=red  
   
# Делаем коммит (r6)  
cp -r /home/studs/s366389/opi/commits/commit6/\* .  
svn add --force .  
svn commit -m "commit6" --username=red  
   
# Переключаемся на ветку b2 и делаем в ней коммит (r7)  
svn switch file:///home/studs/s366389/opi/svn/repo/branches/b2  
cp -r /home/studs/s366389/opi/commits/commit7/\* .  
svn add --force .  
svn commit -m "commit7" --username=blue  
   
# Делаем коммит (r8)  
cp -r /home/studs/s366389/opi/commits/commit8/\* .  
svn add --force .  
svn commit -m "commit8" --username=blue  
   
# Делаем слияние ветки b2 в b3  
svn update  
svn switch file:///home/studs/s366389/opi/svn/repo/branches/b3  
svn merge file:///home/studs/s366389/opi/svn/repo/branches/b2  
svn status  
   
# Делаем коммит (r9)  
cp -r /home/studs/s366389/opi/commits/commit9/\* .  
svn add --force .  
svn commit -m "commit9" --username=blue  
   
# Переключаемся на ветку b1 и делаем в ней коммит (r10)  
svn switch file:///home/studs/s366389/opi/svn/repo/branches/b1  
cp -r /home/studs/s366389/opi/commits/commit10/\* .  
svn add --force .  
svn commit -m "commit10" --username=blue  
   
# Переключаемся на ветку trunk и делаем в ней коммит (r11)  
svn switch file:///home/studs/s366389/opi/svn/repo/trunk  
cp -r /home/studs/s366389/opi/commits/commit11/\* .  
svn add --force .  
svn commit -m "commit11" --username=red  
   
# Переключаемся на ветку b1 и делаем в ней коммит (r12)  
svn switch file:///home/studs/s366389/opi/svn/repo/branches/b1  
cp -r /home/studs/s366389/opi/commits/commit12/\* .  
svn add --force .  
svn commit -m "commit12" --username=blue  
   
# Переключаемся на ветку b3 и делаем в ней коммит (r13)  
svn switch file:///home/studs/s366389/opi/svn/repo/branches/b3  
cp -r /home/studs/s366389/opi/commits/commit13/\* .  
svn add --force .  
svn commit -m "commit13" --username=blue  
   
# Переключаемся на ветку trunk и делаем в ней коммит (r14)  
svn switch file:///home/studs/s366389/opi/svn/repo/trunk  
cp -r /home/studs/s366389/opi/commits/commit14/\* .  
svn add --force .  
svn commit -m "commit14" --username=red  
   
# Переключаемся на ветку b1 и делаем в ней коммит (r15)  
svn switch file:///home/studs/s366389/opi/svn/repo/branches/b1  
cp -r /home/studs/s366389/opi/commits/commit15/\* .  
svn add --force .  
svn commit -m "commit15" --username=blue  
   
# Делаем коммит (r16)  
cp -r /home/studs/s366389/opi/commits/commit16/\* .  
svn add --force .  
svn commit -m "commit16" --username=blue  
   
# Переключаемся на ветку b3 и делаем в ней коммит (r17)  
svn switch file:///home/studs/s366389/opi/svn/repo/branches/b3  
cp -r /home/studs/s366389/opi/commits/commit17/\* .  
svn add --force .  
svn commit -m "commit17" --username=blue  
   
# Переключаемся на ветку b1 и делаем в ней коммит (r18)  
svn switch file:///home/studs/s366389/opi/svn/repo/branches/b1  
cp -r /home/studs/s366389/opi/commits/commit18/\* .  
svn add --force .  
svn commit -m "commit18" --username=blue  
   
# Делаем коммит (r19)  
cp -r /home/studs/s366389/opi/commits/commit19/\* .  
svn add --force .  
svn commit -m "commit19" --username=blue  
   
# Делаем слияние ветки b1 в b3  
svn update  
svn switch file:///home/studs/s366389/opi/svn/repo/branches/b3  
svn merge file:///home/studs/s366389/opi/svn/repo/branches/b1  
svn status  
svn resolve --accept=working /home/studs/s366389/opi/svn/wc/\*  
   
# Делаем коммит (r20)  
cp -r /home/studs/s366389/opi/commits/commit20/\* .  
svn add --force .  
svn commit -m "commit20" --username=blue  
   
# Делаем слияние ветки b3 в trunk  
svn update  
svn switch file:///home/studs/s366389/opi/svn/repo/trunk  
svn merge file:///home/studs/s366389/opi/svn/repo/branches/b3  
svn status  
svn resolve --accept=working /home/studs/s366389/opi/svn/wc/\*  
   
# Делаем коммит (r21)  
cp -r /home/studs/s366389/opi/commits/commit21/\* .  
svn add --force .  
svn commit -m "commit21" --username=red*

Разрешение конфликтов:

*(то, что выделено красным, удаляем)*

svn merge "b1":

C.java:

<<<<<<< .working

public Object gg() {

return new java.util.Random();

}

public void aa() {

return;

||||||| .merge-left.r4

=======

public void bb() {

System.out.println(42);

}

public Object pp() {

return this;

}

public long dd() {

return 100500;

}

public int hh() {

return new java.util.Random(10).nextInt(10);

}

public double ad() {

return 12.12;

>>>>>>> .merge-right.r24

F.java:

<<<<<<< .working

public java.util.List<String> jj() {

return new java.util.ArrayList<String>();

}

public String nn() {

return "++++++++++[>+++++++>++++++++++>+++>+<<<<-]>++";

}

||||||| .merge-left.r4

=======

public String nn() {

return "++++++++++[>+++++++>++++++++++>+++>+<<<<-]>++";

}

public Object pp() {

return this;

}

public long dd() {

return 33;

}

public int af() {

return -1;

}

public int[] ii() {

return new int[]{0, 1, 2, 3, 4};

}

>>>>>>> .merge-right.r24

I.java:

<<<<<<< .mine

public java.lang.Class qq() {

return getClass();

||||||| .merge-left.r4

=======

public int hh() {

return new java.util.Random(10).nextInt(10);

>>>>>>> .merge-right.r24

}

public Object pp() {

return this;

}

<<<<<<< .working

||||||| .merge-left.r4

=======

public String nn() {

return "++++++++++[>+++++++>++++++++++>+++>+<<<<-]>++";

}

public int cc() {

return 39;

}

public String kk() {

return "Hello world";

}

>>>>>>> .merge-right.r24

svn merge "b3":

C.java:

<<<<<<< .working

public float ff() {

return 0;

||||||| .merge-left.r4

=======

public void aa() {

return;

}

public int hh() {

return new java.util.Random(10).nextInt(10);

}

public String nn() {

"".>+.+++++++..+++.>++.<<+++++++++++++++.>.+++.;

>>>>>>> .merge-right.r26

}

public void bb() {

System.out.println(42);

}

<<<<<<< .mine

public Object pp() {

return this;

}

public long dd() {

return 100500;

||||||| .merge-left.r4

=======

public int ae() {

return java.lang.Math.abs(-7);

}

public int af() {

return -1;

>>>>>>> .merge-right.r26

F.java:

<<<<<<< .working

||||||| .merge-left.r4

=======

public java.util.List<String> jj() {

return new java.util.ArrayList<String>();

}

>>>>>>> .rX

public String nn() {

return "++++++++++[>+++++++>++++++++++>+++>+<<<<-]>++";

}

<<<<<<< .working

||||||| .merge-left.r4

=======

public int af() {

return -1;

}

>>>>>>> .merge-right.r26

public Object pp() {

return this;

}

<<<<<<< .working

public long dd() {

return 33;

||||||| .merge-left.r4

=======

public double ee() {

return 100.500;

}

public int[] ii() {

return new int[]{4, 3, 2, 1};

}

public long dd() {

return 99999;

}

public Object rr() {

return null;

}

public String kk() {

return "No";

>>>>>>> .merge-right.r26

I.java:

<<<<<<< .working

public java.lang.Class qq() {

return getClass();

||||||| .merge-left.r4

=======

public int hh() {

return new java.util.Random(10).nextInt(10);

>>>>>>> .merge-right.r26

}

public Object pp() {

return this;

}

<<<<<<< .working

||||||| .merge-left.r4

=======

public String nn() {

return "++++++++++[>+++++++>++++++++++>+++>+<<<<-]>++";

}

public int cc() {

return 39;

}

public String kk() {

return "Hello world";

}

>>>>>>> .merge-right.r26

Вывод

В ходе выполнения лабораторной работы я улучшил свои знания в области систем контроля версий, таких как Git и Subversion. Были изучены основные команды SVN и Git, а также методы разрешения конфликтов.