

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
“Санкт-Петербургский национальный исследовательский
университет
информационных технологий, механики и оптики”

Лабораторная работа № 2
по дисциплине
“МЕТОДЫ И СРЕДСТВА ПРОГРАММНОЙ ИНЖЕНЕРИИ”

Университет ИТМО
Санкт-Петербург
3.04.2023

Задание:

Вариант: 25774



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.

Отчёт по работе должен содержать:

1. Задание и блок-схему в соответствии с вариантом.
2. Список команд, использованных при создании и конфигурации репозитория в домашнем каталоге пользователя.
3. Номера ревизий и соответствующие им последовательности команд с комментариями (для svn и git).
4. Выводы по работе.

Вопросы к защите лабораторной работы:

1. Системы контроля версий - назначение, примеры решений.
2. Ревизии и ветки.
3. Основные операции над данными в системах контроля версий. Основные команды svn и git.
4. Виды конфликтов и способы их решения.

Реализация Git:

```
cd Desktop
cd lab2-mispi
git init
git config --local user.name "evaarakchieva"
git config --local user.email "evaa13@yandex.ru"
```

```
git checkout -b branch1
mkdir src
cp -f ~/Desktop/commits/commit0/* src/
git add .
git commit --author="Eva <evaa13@yandex.ru >" -m "Revision0 (r0)"
```

```
git checkout -b branch2
cp -f ~/Desktop/commits/commit1/* src/
git add .
git commit --author="Liza <lizoksure@gmail.com >" -m "Revision1 (r1)"
```

```
git checkout -b branch3
cp -f ~/Desktop/commits/commit2/* src/
git add .
git commit --author="Eva <evaa13@yandex.ru >" -m "Revision2 (r2)"
```

```
git checkout branch2
cp -f ~/Desktop/commits/commit3/* src/
git add .
git commit --author="Liza <lizoksure@gmail.com >" -m "Revision3 (r3)"
cp -f ~/Desktop/commits/commit4/* src/
git add .
git commit --author="Liza <lizoksure@gmail.com >" -m "Revision4 (r4)"
cp -f ~/Desktop/commits/commit5/* src/
git add .
git commit --author="Liza <lizoksure@gmail.com >" -m "Revision5 (r5)"
```

```
git checkout branch3
cp -f ~/Desktop/commits/commit6/* src/
git add .
git commit --author="Eva <evaa13@yandex.ru >" -m "Revision6 (r6)"
cp -f ~/Desktop/commits/commit7/* src/
git add .
git commit --author="Eva <evaa13@yandex.ru >" -m "Revision7 (r7)"
```

```
git checkout branch1
cp -f ~/Desktop/commits/commit8/* src/
git add .
git commit --author="Eva <evaa13@yandex.ru >" -m "Revision8 (r8)"
cp -f ~/Desktop/commits/commit9/* src/
git add .
git commit --author="Eva <evaa13@yandex.ru >" -m "Revision9 (r9)"
```

```
git checkout branch2
cp -f ~/Desktop/commits/commit10/* src/
git add .
git commit --author="Liza <lizoksure@gmail.com >" -m "Revision10 (r10)"
```

```
git checkout branch3
git merge branch2 -m "Merging r10 with r7 (r11)"
```

Auto-merging src/K.java
CONFLICT (content): Merge conflict in src/K.java

```
Auto-merging src/I.java
CONFLICT (content): Merge conflict in src/I.java
Auto-merging src/G.java
CONFLICT (content): Merge conflict in src/G.java
Auto-merging src/E.java
CONFLICT (content): Merge conflict in src/E.java
Auto-merging src/*
CONFLICT (add/add): Merge conflict in src/*
Automatic merge failed; fix conflicts and then commit the result.
```

```
#resolving conflicts
git add .
git commit -m "merged and resolved the conflicts"
[branch3 4603ebb] merged and resolved the conflicts
```

```
git checkout branch1
git merge branch3 -m "Merging r11 with r9 (r12)"
```

```
Auto-merging src/K.java
CONFLICT (content): Merge conflict in src/K.java
Auto-merging src/I.java
CONFLICT (content): Merge conflict in src/I.java
Auto-merging src/G.java
CONFLICT (content): Merge conflict in src/G.java
Auto-merging src/E.java
CONFLICT (content): Merge conflict in src/E.java
Auto-merging src/*
CONFLICT (add/add): Merge conflict in src/*
Automatic merge failed; fix conflicts and then commit the result.
```

```
#resolving conflicts
git add .
git commit -m "merged and resolved the conflicts"
[branch1 d2ddcac] merged and resolved the conflicts
```

```
cp -f ~/Desktop/commits/commit13/* src/
git add .
git commit --author="Eva <evaa13@yandex.ru >" -m "Revision13 (r13)"
cp -f ~/Desktop/commits/commit14/* src/
git add .
git commit --author="Eva <evaa13@yandex.ru >" -m "Revision14 (r14)"
```

git rebase -ip 159eb69970e60f9618561b55cdaacc9c42931ece
в окне rebase изменили pick на reword
сохранили изменения

Реализация SVN:

```
#!/bin/bash
svn co <URL-адрес репозитория> --username <имя пользователя> --password <пароль>
```

```
svn mkdir <URL-адрес директории> -m "Creating trunk"
```

```
cd trunk
```

```
svn mkdir branches -m "Creating branches directory"
```

```
svn mkdir tags -m "Creating tags directory"
```

```
touch file.txt
```

```
svn add file.txt
```

```
svn copy <URL-адрес trunk> <URL-адрес branches/branch1> -m "Creating branch1"
```

```
svn switch <URL-адрес branches/branch1>
```

```
svn commit -m "Initial commit (r0)" --with-revprop svn:author=User1
```

```
svn copy <URL-адрес branches/branch1> <URL-адрес branches/branch2> -m "Creating branch2"
```

```
echo "Change1" > file.txt
```

```
svn commit -m "Revision 1 (r1)" --with-revprop svn:author=User2
```

```
svn copy <URL-адрес branches/branch2> <URL-адрес branches/branch3> -m "Creating branch3"
```

```
echo "Change2" > file.txt
```

```
svn commit -m "Revision 2 (r2)" --with-revprop svn:author=User1
```

```
svn switch <URL-адрес branches/branch2>
```

```
echo "Change3" > file.txt
```

```
svn commit -m "Revision 3 (r3)" --with-revprop svn:author=User2
```

```
echo "Change4" > file.txt
```

```
svn commit -m "Revision 4 (r4)" --with-revprop svn:author=User2
```

```
echo "Change5" > file.txt
```

```
svn commit -m "Revision 5 (r5)" --with-revprop svn:author=User2
```

```
svn switch <URL-адрес branches/branch3>
```

```
echo "Change6" > file.txt
```

```
svn commit -m "Revision 6 (r6)" --with-revprop svn:author=User1
```

```
echo "Change7" > file.txt
```

```
svn commit -m "Revision 7 (r7)" --with-revprop svn:author=User1
```

```
svn switch <URL-адрес branches/branch1>
```

```
echo "Change8" > file.txt
```

```
svn commit -m "Revision 8 (r8)" --with-revprop svn:author=User1
```

```
echo "Change9" > file.txt
```

```
svn commit -m "Revision 9 (r9)" --with-revprop svn:author=User1
```

```
svn switch <URL-адрес branches/branch2>
```

```
echo "Change10" > file.txt
```

```
svn commit -m "Revision 10 (r10)" --with-revprop svn:author=User2
```

```
svn switch <URL-адрес branches/branch3>
```

```
svn merge <URL-адрес branches/branch2> -c <номер ревизии> --accept=mine-full -m "Merging r7 with r10 (r11)"  
--with-revprop svn:author=User1
```

```
svn switch <URL-адрес branches/branch1>
```

```
svn merge <URL-адрес branches/branch3> -c <номер ревизии> --accept=mine-full -m "Merging r11 with r19 (r12)"  
--with-revprop svn:author=User1
```

```
echo "Change13" > file.txt
```

```
svn commit -m "Revision 13 (r13)" --with-revprop svn:author=User1
```

```
echo "Change14" > file.txt
```

```
svn commit -m "Revision 14 (r14)" --with-revprop svn:author=User1
```

Вывод:

В процессе выполнения лабораторной 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.