

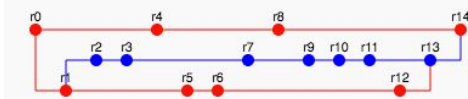
ИТМО
Факультет ПИИКТ

«Методы и средства
Программной инженерии»

Лабораторная работа №2
Вариант 21525

Санкт-Петербург
2020г

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.

Код:

GIT	SVN
<pre>{ git init git status # commit 0 git config --local user.name "Boba"; git config --local user.email "boba@gmail.com" yes cp -rf ../commits/commit0/* ./ git add -A git commit -m r0 # commit 1 git checkout -b r1 yes cp -rf ../commits/commit1/* ./ git add -A git commit -m r1 # commit 2 git config --local user.name 'Biba'; git config --local user.email 'biba@gmail.com' git checkout -b r2 yes cp -rf ../commits/commit2/* ./ git add -A git commit -m r2 # commit 3 yes cp -rf ../commits/commit3/* ./ git add -A git commit -m r3 # commit 4 git config --local user.name "Boba"; git config --local user.email "boba@gmail.com"</pre>	<pre>#!/bin/bash workDir=`pwd`; sourcePath="\$workDir/commits"; repositoryName="repo"; projectName="proj"; svnRepoDir="/tmp"; repositoryFullPath=\$svnRepoDir/\$repositoryName; projectWCPath=\$repositoryName/\$projectName; initStructurePath=\$svnRepoDir/\$projectName clearWorkDir() { echo "[INFO] clear dirs"; rm -rf \$repositoryFullPath \$projectWCPath \$initStructurePath 2>/dev/null; } loadAndCommit() { echo "[INFO] load r\$1 [\$2]"; set -f; conflictedFiles=`svn status egrep '^(.+C)(^C.*)' tr -d 'C'`; while [[\$conflictedFiles]]; do echo "[INFO] try to auto resolve conflicts"; for file in \$conflictedFiles; do</pre>

```

git checkout master
yes | cp -rf ../commits/commit4/* ./
git add -A
git commit -m r4
# commit 5
git checkout r1
yes | cp -rf ../commits/commit5/* ./
git add -A
git commit -m r5
# commit 6
git checkout r1
yes | cp -rf ../commits/commit6/* ./
git add -A
git commit -m r6
# commit 7
git config --local user.name 'Biba'; git
config --local user.email 'biba@gmail.com'
git checkout r2
yes | cp -rf ../commits/commit7/* ./
git add -A
git commit -m r7
# commit 8
git config --local user.name "Boba"; git
config --local user.email "boba@gmail.com"
git checkout master
yes | cp -rf ../commits/commit8/* ./
git add -A
git commit -m r8
# commit 9
git config --local user.name 'Biba'; git
config --local user.email 'biba@gmail.com'
git checkout r2
yes | cp -rf ../commits/commit9/* ./
git add -A
git commit -m r9
# commit 10
yes | cp -rf ../commits/commit10/* ./
git add -A
git commit -m r10
# commit 11
yes | cp -rf ../commits/commit11/* ./
git add -A
git commit -m r11
# commit 12
git config --local user.name "Boba"; git
config --local user.email "boba@gmail.com"
git checkout r1

```

```

svn resolve $file
--accept=working;
done;
conflictedFiles=`svn status | egrep
'(^.+C)|(^C.*)' | tr -d ' C';
done;

set +f;
cp $sourcePath/r$1/* .;
set -f;

for file in `ls`; do
    if ! [[ -f $sourcePath/r$1/$file ]]; then
        rm $file;
    fi
done;

newFiles=`svn status | egrep '^\.?.$' | cut
-c 2- | tr -d ' '`;
for newFile in $newFiles; do svn add
"$newFile"; done;
set +f;

svn commit -m "$2 [commit$1]"
--username $3;
}

init() {
    echo "[INFO] check paths";
    if [ -d $repositoryFullPath ]; then
        echo "[ERROR] "
        $repositoryFullPath "already exists";
        return -1;
    fi
    if [ -d $projectWCPATH ]; then
        echo "[ERROR] " $projectWCPATH
        "already exists";
        return -1;
    fi

    echo "[INFO] start init svn repo";
    svnadmin create $repositoryFullPath;

    echo "[INFO] create init structure dirs";
    mkdir -p $initStructurePath/trunk
    $initStructurePath/branches
    $initStructurePath/tags;

    echo "[INFO] import init structure to
repo";
    svn import $initStructurePath

```

```

yes | cp -rf ../commits/commit12/* ./
git add -A
git commit -m r12
# commit 13
git config --local user.name 'Biba'; git
config --local user.email 'biba@gmail.com'
git checkout r2
git merge --no-ff r1
cp -rf ../commits/commit13/* ./
git checkout --ours -- *
git checkout --ours -- J.java
git checkout --ours -- I.java
git add -A
git commit -m r13
# commit 14
git config --local user.name "Boba"; git
config --local user.email "boba@gmail.com"
git checkout master
git merge --no-ff r2
rm -f "*"
rm -f "ABNQBatpyO.0W1"
cp -rf ../commits/commit14/* ./
git checkout --ours -- ./
git add -A
git commit -m r14
} &> log.txt

git log --all --decorate --oneline --graph

```

```

file://$repositoryFullPath/$projectName -m
"init";

    echo "[INFO] checkout project";
    svn checkout
file://$repositoryFullPath/$projectName/trunk $projectWCPath
}

createBranch() {
    echo "[INFO] create branch $2 from $1";
    svn copy
file://$repositoryFullPath/$projectName/$1 \

file://$repositoryFullPath/$projectName/branches/$2 -m $3 --username $4;
}

switch() {
    echo "[INFO] switch to $1";
    svn switch
file://$repositoryFullPath/$projectName/$1;
}

createBranchAndSwitch() {
    createBranch $1 $2 $3 $4;
    switch branches/$2;
}

merge() {
    svn merge
file:/// $repositoryFullPath/$projectName/$1
--accept p --username $2;
}

main() {
    if ! [[ -d commits ]]; then
        echo "[ERROR] cd to dir with
commits/";
        return -1;
    fi
    clearWorkDir;
    init;
    cd $projectWCPath;
    loadAndCommit 0 "commit-by-red-user"
red;
    createBranchAndSwitch trunk
some-feature-by-red-user
"some-feature-by-red-user" red;
    loadAndCommit 1 "commit-by-red-user"
red;
    createBranchAndSwitch

```

	<pre> branches/some-feature-by-red-user \ some-feature-by-blue-user "some-featur-by-blue-user" blue; loadAndCommit 2 "commit-by-blue-user" blue; loadAndCommit 3 "commit-by-blue-user" blue; switch trunk; loadAndCommit 4 "commit-by-red-user" red; switch branches/some-feature-by-red-user; loadAndCommit 5 "commit-by-red-user" red; loadAndCommit 6 "commit-by-red-user" red; switch branches/some-feature-by-blue-user; loadAndCommit 7 "commit-by-blue-user" blue; switch trunk; loadAndCommit 8 "commit-by-red-user" red; switch branches/some-feature-by-blue-user; loadAndCommit 9 "commit-by-blue-user" blue; loadAndCommit 10 "commit-by-blue-user" blue; loadAndCommit 11 "commit-by-blue-user" blue; switch branches/some-feature-by-red-user; loadAndCommit 12 "commit-by-red-user" red; switch branches/some-feature-by-blue-user; merge branches/some-feature-by-red-user blue; loadAndCommit 13 "commit-by-blue-user" blue; switch trunk; merge branches/some-feature-by-blue-user red; loadAndCommit 14 "commit-by-blue-user" blue; cd - 1>/dev/null } main; </pre>
--	--

Вывод:

Интересная лаба, конечно, я ожидал, что будет юзаться там `remote`-ы всякие, а тут просто локальный контроль версий, в целом понравилось, узнал, как разрешать конфликты, но решать в дальнейшем их, конечно, не буду. Буду продолжать придерживаться старой схемы: вкидывать рандомные команды со стакаверфлоу пока не запушится на гитхаб. Касательно `svn`, интуитивно он вероятно понятнее чем `git` на первых порах погружения в этот дивный мир дЕвЕлОпИнГа, но за три дня использования сложилось какое-то, возможно даже предвзятое, мнение о том, что `svn` морально устарел. Никогда его раньше не использовал и надеюсь не придется в будущем, но в резюме +1 пункт есть, спасибо и на этом.

