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: **1571**

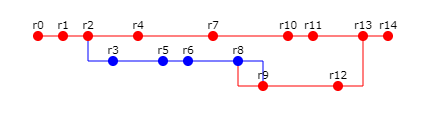
**Преподаватель**:   
Цопа Евгений Алексеевич

**Выполнил**: Барсуков Максим

**Группа**: Р3215

Санкт-Петербург, 2024

# 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.

# Реализация с использованием Git

|  |
| --- |
| 1. #!/bin/bash 2. *# Очистка ненужных данных и backup .git* 3. cp -r .git .rotter 4. cp .gitignore .rotterignore 5. rm -rf .git 6. rm -f .gitignore 7. echo "- Создан бэкап .git" 8. find src -type f ! -name '.keep' -delete 9. echo "- src очищен" 10. *# Создание репозитория и начальная ревизия (r0)* 11. git init 12. echo "- git init" 13. *# Настройка .git/config* 14. echo -e "\n[merge]\n\ttool = nano" >> .git/config 15. *# Настройка пользователей* 16. git config user.name "red" 17. git config user.email "red@example.com" 18. echo "- Пользователь red создан" 19. git checkout -b branch1 20. *# Новый .gitignore {* 21. echo ".resources" > .gitignore 22. echo ".rotter" >> .gitignore 23. echo "commits" >> .gitignore 24. echo "docs" >> .gitignore 25. echo "src/.keep" >> .gitignore 26. echo ".editorconfig" >> .gitignore 27. echo ".rotterignore" >> .gitignore 28. echo "git.sh" >> .gitignore 29. echo "svn.sh" >> .gitignore 30. echo ".gitattributes" >> .gitignore 31. echo "LICENSE" >> .gitignore 32. echo "README.md" >> .gitignore 33. echo "return-my-git-plz.sh" >> .gitignore 34. git add .gitignore 35. echo "- Новый .gitignore создан" 36. *# }* 37. *# Ревизия r0 (пользователь 1) {* 38. unzip -o commits/commit0.zip -d src 39. git add . 40. git commit -m "Initial commit (r0)" 41. echo "- Коммит 0 (red)" 42. *# }* 43. *# Ревизии r1-r2 (пользователь 1) {* 44. unzip -o commits/commit1.zip -d src 45. git add . 46. git commit -m "Revision 1 (r1)" 47. echo "- Коммит 1 (red)" 48. unzip -o commits/commit2.zip -d src 49. git add . 50. git commit -m "Revision 2 (r2)" 51. echo "- Коммит 2 (red)" 52. *# }* 53. *# Ревизия r3 (пользователь 2) {* 54. git checkout -b branch2 55. unzip -o commits/commit3.zip -d src 56. git add . 57. git commit --author="blue <blue@example.com>" -m "Revision 3 (r3)" 58. echo "- Коммит 3 (blue)" 59. *# }* 60. *# Ревизия r4 (пользователь 1) {* 61. git checkout branch1 62. unzip -o commits/commit4.zip -d src 63. git add . 64. git commit -m "Revision 4 (r4)" 65. echo "- Коммит 4 (red)" 66. *# }* 67. *# Ревизии r5-r6 (пользователь 2) {* 68. git checkout branch2 69. unzip -o commits/commit5.zip -d src 70. git add . 71. git commit --author="blue <blue@example.com>" -m "Revision 5 (r5)" 72. echo "- Коммит 5 (blue)" 73. unzip -o commits/commit6.zip -d src 74. git add . 75. git commit --author="blue <blue@example.com>" -m "Revision 6 (r6)" 76. echo "- Коммит 6 (blue)" 77. *# }* 78. *# Ревизия r7 (пользователь 1) {* 79. git checkout branch1 80. unzip -o commits/commit7.zip -d src 81. git add . 82. git commit -m "Revision 7 (r7)" 83. echo "- Коммит 7 (red)" 84. *# }* 85. *# Ревизия r8 (пользователь 2) {* 86. git checkout branch2 87. unzip -o commits/commit8.zip -d src 88. git add . 89. git commit --author="blue <blue@example.com>" -m "Revision 8 (r8)" 90. echo "- Коммит 8 (blue)" 91. *# }* 92. *# Ревизия r9 {* 93. git checkout -b branch3 94. *# Создание файла для ревизии r9 с участием обоих пользователей* 95. unzip -o commits/commit9.zip -d src 96. git add . 97. git commit -m "Revision 9 (r9)" 98. echo "- Коммит 9 (red)" 99. git checkout branch2 100. git merge --ff-only branch3 101. *# }* 102. *# Ревизии r10-r11 (пользователь 1) {* 103. git checkout branch1 104. unzip -o commits/commit10.zip -d src 105. git add . 106. git commit -m "Revision 10 (r10)" 107. echo "- Коммит 10 (red)" 108. unzip -o commits/commit11.zip -d src 109. git add . 110. git commit -m "Revision 11 (r11)" 111. echo "- Коммит 11 (red)" 112. *# }* 113. *# Ревизии r12 {* 114. git checkout branch3 115. unzip -o commits/commit12.zip -d src 116. git add . 117. git commit -m "Revision 12 (r12)" 118. echo "- Коммит 12 (red)" 119. *# }* 120. *# Мердж ревизии r11 с r12 # {* 121. git checkout branch1 122. *# Эта опция сохранит вашу версию файла (ветку, из которой выполняется слияние) в случае конфликта* 123. *# git merge --no-commit branch3 -Xours* 124. *# Эта опция сохранит версию файла из ветки, с которой выполняется слияние, в случае конфликта* 125. *# git merge --no-commit branch3 -Xtheirs* 126. git merge --no-commit branch3 127. *##* 128. *## ИСПРАВЛЕНИЕ КОНФЛИКТА ВРУЧНУЮ* 129. *##* 130. git add . 131. echo "- Слияние r11 и r12" 132. *# Отменяем merge в случае отмены* 133. *# git merge --abort* 134. *# Ревизии r13-r14 (пользователь 1) {* 135. unzip -o commits/commit13.zip -d src 136. git add . 137. git commit -m "Revision 13 (r13)" 138. echo "- Коммит 13 (red)" 139. unzip -o commits/commit14.zip -d src 140. git add . 141. git commit -m "Revision 14 (r14)" 142. echo "- Коммит 14 (red)" 143. *# }* 144. *# Вывод графа* 145. git log --graph --abbrev-commit --decorate --format=format:'%C(bold blue)%h%C(reset) - %C(bold green)(%ar)%C(reset) %C(white)%s%C(reset) %C(dim white)- %an%C(reset)%C(auto)%d%C(reset)' --all |

# Реализация с использованием SVN

|  |
| --- |
| 1. #!/bin/bash 2. *# Создание локального репозитория* 3. svnadmin create repo 4. REPO\_URL="file://$(pwd)/repo" 5. *# Создаём структуру проекта* 6. cd repo 7. svn mkdir -m "project structure" $REPO\_URL/trunk $REPO\_URL/branches 8. cd .. 9. *# Создание рабочей копии* 10. svn checkout $REPO\_URL/trunk/ wc 11. cd wc 12. *# Начальная ревизия (пользователь 1) {* 13. unzip -o ../commits/commit0.zip -d . 14. svn add \* 15. svn commit -m "Initial commit (r0)" --username=red 16. *# }* 17. *# Ревизия r1 (пользователь 1) {* 18. *# Удаляем файлы из рабочей директории* 19. *# (если это не сделать, то `... is already under version control`)* 20. svn rm \* 21. unzip -o ../commits/commit1.zip -d . 22. svn add \* 23. svn commit -m "Revision 1 (r1)" --username=red 24. *# }* 25. *# Ревизия r2 (пользователь 1) {* 26. svn rm \* 27. unzip -o ../commits/commit2.zip -d . 28. svn add \* 29. svn commit -m "Revision 2 (r2)" --username=red 30. *# }* 31. *# Ревизия r3 (пользователь 2) {* 32. svn copy $REPO\_URL/trunk $REPO\_URL/branches/branch2 -m "Creating branch2" 33. svn switch $REPO\_URL/branches/branch2 34. svn rm \* 35. unzip -o ../commits/commit3.zip -d . 36. svn add \* 37. svn commit -m "Revision 3 (r3)" --username=blue 38. *# }* 39. *# Ревизия r4 (пользователь 1) {* 40. svn switch $REPO\_URL/trunk 41. svn rm \* 42. unzip -o ../commits/commit4.zip -d . 43. svn add \* 44. svn commit -m "Revision 4 (r4)" --username=red 45. *# }* 46. *# Ревизии r5-r6 (пользователь 2) {* 47. svn switch $REPO\_URL/branches/branch2 48. svn rm \* 49. unzip -o ../commits/commit5.zip -d . 50. svn add \* 51. svn commit -m "Revision 5 (r5)" --username=blue 52. svn rm \* 53. unzip -o ../commits/commit6.zip -d . 54. svn add \* 55. svn commit -m "Revision 6 (r6)" --username=blue 56. *# }* 57. *# Ревизия r7 (пользователь 1) {* 58. svn switch $REPO\_URL/trunk 59. svn rm \* 60. unzip -o ../commits/commit7.zip -d . 61. svn add \* 62. svn commit -m "Revision 7 (r7)" --username=red 63. *# }* 64. *# Ревизия r8 (пользователь 2) {* 65. svn switch $REPO\_URL/branches/branch2 66. svn rm \* 67. unzip -o ../commits/commit8.zip -d . 68. svn add \* 69. svn commit -m "Revision 8 (r8)" --username=blue 70. *# }* 71. svn update 72. *# Ревизия r9 (пользователь 1) {* 73. svn copy $REPO\_URL/branches/branch2 $REPO\_URL/branches/branch3 -m "Creating branch3" 74. svn switch $REPO\_URL/branches/branch3 75. svn rm \* --force 76. svn merge $REPO\_URL/branches/branch2 77. unzip -o ../commits/commit9.zip -d . 78. svn add \* 79. svn commit -m "Revision 9 (r9)" --username=red 80. *# }* 81. svn update 82. *# Ревизии r10-r11 (пользователь 1) {* 83. svn switch $REPO\_URL/trunk 84. svn rm \* 85. unzip -o ../commits/commit10.zip -d . 86. svn add \* 87. svn commit -m "Revision 10 (r10)" --username=red 88. svn rm \* 89. unzip -o ../commits/commit11.zip -d . 90. svn add \* 91. svn commit -m "Revision 11 (r11)" --username=red 92. *# }* 93. *# Ревизия r12 (пользователь 1) {* 94. svn switch $REPO\_URL/branches/branch3 95. svn rm \* 96. unzip -o ../commits/commit12.zip -d . 97. svn add \* 98. svn commit -m "Revision 12 (r12)" --username=red 99. *# }* 100. svn update 101. *# Мердж ревизии r11 с r12 {* 102. svn switch $REPO\_URL/trunk 103. svn merge $REPO\_URL/branches/branch3 104. nano Lab4.java 105. svn resolved Lab4.java 106. *# }* 107. *# Ревизии r13-r14 (пользователь 1) {* 108. svn rm \* --force 109. unzip -o ../commits/commit13.zip -d . 110. svn add \* 111. svn commit -m "Revision 13 (r13)" --username=red 112. svn rm \* 113. unzip -o ../commits/commit14.zip -d . 114. svn add \* 115. svn commit -m "Revision 14 (r14)" --username=red 116. *# }* 117. svn update |

# Вывод

В ходе выполнения лабораторной работы я улучшил свои навыки владения системой контроля версий Git, а также познакомился 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.