

Лабораторная работа № 3

Управление файлами в Subversion

Цель работы: научиться выполнять восстановление более ранних версий файлов командой `merge` (слияние, объединение), при помощи патч-файла; познакомиться с понятиями «главная ветвь проекта», «ответвление», научиться создавать ответвления проекта и выполнять их слияние.

Задание

1. Объединиться в группы по два человека и определить, с каким каталогом репозитория работает каждая группа. У каждого участника группы должен быть свой логин. Таким образом, с одним каталогом репозитория работают одновременно два студента с условными именами `user1` и `user2`.
2. Войти в систему под своим логином и обновить репозиторий.
3. Распределить роли участников группы: определить, кто из группы будет руководителем, а кто — исполнителем.
4. Прочитать справочный материал о команде `merge` и повторить материал о команде `update`.
5. Просмотреть историю изменений файлов вашего каталога и выбрать любой файл, где вы в последней ревизии выполняли изменение текста. Проверьте, как изменился текст в последней ревизии по сравнению с предпоследней. Проверьте текст этого же файла в рабочей копии.
6. Выполните восстановление более ранней версии выбранного файла при помощи `update: svn update -r N имя_файла`, где `N` — номер выбранной более ранней ревизии. Проверьте, что содержимое файла в рабочей копии после выполнения команды соответствует `N`-ой ревизии.
7. Проверьте статус репозитория и сделайте вывод о необходимости и возможности сохранения изменений в репозитории. Выполните `update` и проверьте содержимое файла.
8. Выполните восстановление предыдущей версии выбранного файла при помощи `merge: svn merge -r Тек_рев:Пред_рев имя_файла`. Проверьте содержимое файла в рабочей копии и статус репозитория.
9. Сохраните изменения в репозитории и выполните обновление рабочей копии. Сделайте вывод о произошедших изменениях.
10. Руководителю создать в репозитории (на сервере) два каталога `trunk` (в переводе с английского — ствол, для хранения главной ветви проекта) и `branches` (для хранения ответвлений проекта). затем скопировать в каталог `trunk` файлы `first.html`, `second.html`. Каждую операцию сохранения в репозитории (коммиты) НЕ ЗАБЫВАЙТЕ КОММЕНТИРОВАТЬ! Файлы в

каталоге trunk образуют главную ветвь проекта.

- 11.Руководителю **скопировать** каталог trunk в каталог branches и назвать его branch1 (создать ответвление). Файлы из каталога trunk будут редактироваться. Файлы из каталога branch1 также могут редактироваться, причем независимо от главной ветви.
- 12.Руководитель группы назначает ответственных за редактирование ветвей проекта — главной ветви и ветви branch1.
- 13.Исполнитель в своем аккаунте получает локальную копию «своей» ветви репозитория и переходит в этот каталог.
- 14.Каждому участнику проекта отредактировать «свою» ветку репозитория — добавить по одному новому html-файлу со своим именем (user1.html, user2.html). Сохранить и прокомментировать изменения в репозитории. Просмотреть историю изменений ветки. Просмотреть историю изменений ветки с включением служебной информации.
- 15.Каждому участнику проекта выполнить редактирование нового файла (user1.html, user2.html) в своей ветви и сохранить в репозитории.
- 16.Каждому участнику проекта создать патч для этого файла. Патч представляет собой файл, содержащий отличия в текстах двух ревизий (последней и предпоследней) редактируемого файла:
`svn diff -r предпослед_ревизия:последняя_ревизия имя_файла > ~/patchfile`
- 17.Просмотреть содержимое патча.
- 18.Откатить содержимое файла к предыдущей версии с помощью патча (при выполнении команды patch в ответ на вопрос ввести латинскую букву «у»):
`svn update — если не сделали ранее`
`patch -p0 < ~/patchfile`
`svn diff`
19. Проверить статус репозитория и сделать вывод.
20. Каждому участнику проекта еще раз отредактировать файл first.html (second.html) и сохранить в репозитории в «своей» ветке. Создать новый файл и добавить под управление репозитория. Оба изменения сохранить в репозитории одним коммитом.
21. Руководителю выполнить слияние главной ветви проекта с ответвлением для того, чтобы изменения, сделанные в ответвлении, были получены в главной ветви (команда svn merge, текущий каталог при выполнении слияния - trunk):

`svn merge -r rtrunk:rbranch1 svn://eniac/sandbox/.../branch1,`

где rtrunk – номер ревизии, в которой была создана ветвь trunk или для нее выполнялось последнее слияние; rbranch1 – номер последней ревизии в branch1.

- 22.Просмотреть и проанализировать статус trunk, branch1, выполнить коммит, который **ОБЯЗАТЕЛЬНО ПРОКОММЕНТИРОВАТЬ**. Проверить, что в trunk появился новый файл и изменен файл index.html.

23.Обновить trunk.

24.Создать ответвление branch2 и повторить пункты 11-23.

25.Просмотреть и проанализировать историю изменений ветви trunk.

26.Отменить последний коммит для trunk через обратное слияние, для этого правильно указать номера ревизий в ветви trunk:

```
svn update
svn merge -r последняя_ревизия:предпол_ревизия ./
svn diff
svn commit --message "Изменения отменены"
```

27.Продемонстрируйте преподавателю результаты слияния веток, отмены коммита, патч-файлы.

Справочный материал

1. Ветви SVN

В репозитории можно хранить одновременно несколько ветвей проекта. В проекте выделяют главную ветвь, которая разрабатывается и завершается готовым продуктом, и ответвления. В качестве ответвления может быть завершённая версия программы, которая не подвергается дальнейшим изменениям. Также ответвление от главной ветви может использоваться для разработки параллельно с разработкой главной ветви. Результатом разработки может быть новый программный продукт. В ответвлении могут быть проверены альтернативные решения задач основного проекта, которые в случае успеха применяются в основной ветви проекта. Файлы проекта, относящиеся к главной ветви, помещают обычно в отдельный каталог trunk, для хранения ответвлений создается каталог branches. Ответвления создают копированием основной ветви средствами Subversion.

Для синхронизации ветвей используется команда **svn merge** — она выполняет слияние ветвей проекта. Слияние (изменение данных) выполняется для текущего каталога. Для примера рассмотрим слияние ветви branch5 с основной ветвью trunk (изменения наблюдаем в branch5). Текущий каталог при выполнении команды — branch5:

```
svn merge -r rev1:rev2 svn://eniac/sandbox/.../trunk
```

где указать в качестве rev1 номер ревизии создания или последнего слияния текущей ветки (в нашем примере branch5), а в качестве rev2 - номер последней ревизии ветви, с которой происходит слияние (trunk).

Также команда **svn merge** используется для восстановления в репозитории более раннего состояния его каталогов или отдельных файлов :

```
svn merge -r rev путь/имя_каталога_файла
```

где rev — номер более ранней ревизии восстанавливаемого каталога или файла.

Для просмотра истории слияний можно воспользоваться командой **svn log** в

следующем виде:

```
svn log --verbose
```

или

```
svn log -v
```

2. Команды Linux

patch - программная утилита, предназначенная для переноса изменений между различными версиями текстовых файлов. Информация об изменениях в файлах сохраняется в специальном текстовом файле, называемом patch-файлом.

```
patch -p0 < путь/patch-файл
```

Знак < означает ввод данных, необходимых для выполнения команды patch, из указанного файла.

В примерах использован URL-адрес репозитория svn://eniac/sandbox.

3. Вопросы к защите

1. Как создать новый каталог в репозитории?
2. Для чего нужны ответвления от главной линии проекта?
3. Какой должна быть структура каталогов репозитория с ответвлениями?
4. Что означают имена каталогов trunk, branches?
5. Как создать новую ветвь проекта?
6. Как откатить ветвь проекта в исходное состояние?
7. Что представляет собой команда patch?
8. Каково назначение команды svn merge?
9. Как выполнить слияние главной ветви с ответвлением?
10. Что такое обратное слияние?