

Задание № 2

Групповая работа с репозиторием

Цель работы: познакомиться операциями над файлами в репозитории и с приемами групповой работы.

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 пользователя user1 и user2.
2. Войти в систему под своим логином.
3. Запустить программу Терминал и найти в домашнем каталоге вашего пользователя рабочую (локальную) копию репозитория (далее этот каталог будем называть name). Если копия отсутствует, получить ее с помощью команды `svn checkout`.
4. Обновить содержимое каталога name с помощью команды `svn update`. Проверить, что в каталоге name есть последние версии файлов `first.html` и `second.html`.
5. Пользователь user1 редактирует файл `first.html`, а user2 — `second.html`, при этом в текст документов добавляется новый осмысленный текст. Рекомендуется в текст добавлять информацию об авторе изменений.
6. Просмотреть статус репозитория.
7. Просмотреть отредактированные файлы в браузере, если они содержат html-код, а файлы, не содержащие html-код, можно прочитать простейшим текстовым редактором или консольной утилитой для просмотра текстового файла:
`more имя_файла`
8. Отправить отредактированные файлы в репозиторий и обновить рабочие копии.
9. Просмотреть статус репозитория. Сравнить полученную информацию и информацию п. 6. Сделать выводы.
10. Просмотреть коммиты вашего репозитория (каталога name), каждого из редактируемых файлов и найти коммиты, сделанные вами и вашим напарником.
11. Добавить в «свой» файл абзац. Просмотреть локальные изменения командой `svn status -u`. Какая дополнительная информация о файлах в рабочей копии была получена? Как ее можно использовать?
12. Отменить локальные изменения в файле, не сохраненные в репозитории:
`svn revert ./имя_файла`
 Следует помнить, что команда, выполненная без указания имени файла, отменит все локальные изменения в файлах данного каталога!
13. Каждому пользователю отредактировать «свой» документ — добавить по три строки.
14. Отправить отредактированные файлы в репозиторий и запомнить номер текущей ревизии.
15. Каждому пользователю создать средствами `svn` копию «своего» файла в репозитории, при этом один из пользователей создает копию файла прямо в репозитории, второй — в рабочей копии.
16. Просмотреть статус репозитория, содержимое репозитория и рабочей копии,

сделать выводы, все ли копии файлов имеются у каждого пользователя.

17. Обновить рабочие копии репозитория. Добиться появления копий обоих файлов в каждой рабочей копии.
18. Одному из пользователей удалить средствами svn из локального каталога (рабочей копии) копию «своего» файла, просмотреть статус репозитория и зафиксировать изменение в репозитории. Был ли удален файл из рабочей копии, из репозитория? Завершить удаление.
19. Второму пользователю удалить средствами svn копию «своего» файла из репозитория, просмотреть содержимое и статус репозитория. Как завершить удаление файла?
20. Каждому пользователю переименовать средствами svn «свой» файл в рабочей копии репозитория:

```
svn rename ./имя_файла ./новое_имя_файла
```

Проверить, изменились ли имена файлов в репозитории. Что надо сделать для завершения переименования?

21. Добавить в файлы first.html и second.html абзац, при этом редактирование обоих файлов в своих рабочих копиях репозитория выполняют одновременно оба участника группы. Зафиксировать изменения в репозитории. При возникновении конфликта, (сообщение Commit failed...) выполнить команду svn update, которая предложит варианты разрешения конфликта (см. справочный материал).
22. Обоим пользователям в своих рабочих копиях выполнить одновременно средствами svn переименование одного и того же файла. Зафиксировать изменения. При возникновении конфликта выполнить действия по его разрешению. Сделайте вывод, как сказался этот конфликт на содержимом репозитория и его рабочих копий.
23. Пользователям в своих рабочих копиях выполнить последовательно следующие действия: user1 - удалить командой svn один из файлов и сохранить изменение в репозитории, затем user2 — не обновляя рабочую копию, внести изменения в текст того же файла и сохранить изменения. Объясните причину появления конфликта. Предложите способ разрешения данного конфликта.
24. User1: создать текстовый файл для локального хранения данных и, не добавляя его в репозиторий, проверить статус репозитория. Как будет отображен новый файл?
25. Исключить этот файл из статуса:

```
svn propedit svn:ignore .
```

При выполнении команды будет открыт новый файл, в который следует ввести имя игнорируемого файла, а файл сохранить.

26. User1: просмотреть статус репозитория и сравнить с предыдущим статусом. Объяснить текущий статус. Сохранить изменения в репозитории, проверить содержимое репозитория. Сделайте выводы.
27. User2: после завершения предыдущего действия обновить свою рабочую копию.
28. User2: создать текстовый файл с новым именем для локального хранения данных и, не добавляя его в репозиторий, проверить статус репозитория. Как будет отображен новый файл?
29. Исключить этот файл из статуса:

```
svn propedit svn:ignore .
```

При выполнении команды будет открыт файл, уже содержащий текст, в который следует

ввести имя второго игнорируемого файла, а файл сохранить.

30. User2: просмотреть статус репозитория, сохранить изменения.
31. User1: создать текстовый файл для локального хранения с тем же именем, что и файл, созданный User2, и просмотреть статус репозитория. Будет ли этот файл показан в статусе репозитория?
32. Создать в Qt Creator приложение с графическим интерфейсом пользователя, при этом поместить каталог проекта в каталог name, управляемый svn, и указать в настройках проекта, что он будет управляться системой Subversion (Инструменты/Параметры/Контроль версий/Subversion, команда usr/bin/svn).
33. Установить контроль Subversion над проектом через терминал, если не удалось сделать в Creator.
34. Редактировать исходный код проекта любым способом, чтобы получить не менее четырех рабочих версий.
35. Обновить локальные копии файлов и продемонстрировать результат преподавателю.

Справочный материал

Команды SVN

svn commit -message Поясняющий текст — сохранение репозитории файлов и комментариев к текущей ревизии

svn rename ./старое_имя_файла ./новое_имя_файла — переименование файла в репозитории: сперва переименовывается локальная копия файла, после выполнения коммита переименовывается оригинал в репозитории

svn remove ./путь/имя_файла — удаление файла из репозитория.

svn copy источник приемник — копирование файлов (каталогов), источник — путь и имя копируемого файла (каталога), приемник — путь и имя результата копирования. Команда выполняет копирование в пределах данного репозитория и его рабочих копий. Путь может быть записан в виде локального пути или URL-адреса (например, svn://eniac/sandbox/).

svn status имя файла или каталога — показывает состояние указанного объекта:

A — объект запланирован для добавления в репозиторий

C — объект находится в состоянии конфликта с другими версиями

M — объект был изменен

D — объект был удален

R — объект будет заменен (перемещен)

X — объект не имеет версий

U — объект обновлен

I — объект игнорируется системой svn

G — объект совмещен (например, командами merge, update)

! - объект удален сторонними средствами без участия svn

? - объект не находится под контролем svn

svn propedit - редактирование свойств каталога или файла в репозитории.

svn propedit svn:ignore . — определяет маску для каталога, по которой будут игнорироваться файлы, не находящиеся под контролем svn: в данном случае при выполнении коммита файл не сохраняется в репозитории (игнорируется), а существует только в локальном каталоге, при этом данное несоответствие репозитория и его

локальной копии не будет проявляться при просмотре статуса репозитория. Замечание: если при выполнении коммита появилось сообщение, что File or directory '! is out of date; try updating - выполните update, затем commit.

svn log -v -r номер_ревизии — просмотр детальной информации о ревизиях.

svn revert ./имя_файла — отменяет локальные изменения в файле, не внесенные в репозиторий.

Разрешение конфликта

Выполнить команду `svn update`, которая предложит варианты своего завершения:

(e) edit — редактировать файл с изменениями;

(df) diff-full — показать все конфликтующие изменения;

(r) resolved — принять файл, содержащий маркеры слияния конфликтующих изменений;

(dc) display-conflict — показать все конфликты;

(mf) mine-full — принять свою версию и игнорировать полученные изменения;

(tf) theirs-full — принять полностью полученную версию файла и отказаться от своих изменений;

(p) postpone — отметить конфликт и разрешить его позднее.

При выборе варианта `postpone` в рабочем каталоге появятся конфликтующие версии файла. Следует их просмотреть, внести исправления в основной файл (без дополнительного расширения), затем удалить лишние файлы (`rm first.html.*`) и сохранить основной файл в репозитории.

URL-адрес репозитория: `svn://eniac/имя_группы`

Вопросы к защите

1. Какие действия над файлами можно выполнять в репозитории?
2. Как выполняется добавление нового файла в репозиторий?
3. Как происходит удаление файла из репозитория?
4. Выполнена команда переименования файла в репозитории, в рабочей копии. Появится ли в репозитории файл с новым именем сразу после выполнения команды?
5. Как проверить, есть ли локальные изменения в рабочей копии репозитория?
6. Как отменить локальные изменения в рабочей копии репозитория?
7. Как проверить, нужно ли обновление репозитория?
8. Какая команда выполняет копирование файлов в репозитории?
9. В каком случае наблюдается конфликт локальных изменений?
10. Как уменьшить вероятность конфликтов?
11. Какие способы разрешения конфликта возможны?
12. Как отфильтровать показ в статусе рабочей копии репозитория файлов, не управляемых Subversion?