

## Лабораторная работа № 1

### Знакомство с системой управления версиями Subversion

Цель работы: получить начальное представление о возможностях программных продуктов, входящих в группу систем контроля версий, на примере программы Subversion; познакомиться с терминами (репозиторий, ревизия), командами системы, алгоритмом работы в системе, научиться выполнять обмен данными с репозиторием, просматривать содержимое репозитория.

#### Задание

1. Запустить программу Терминал, с помощью команды `svn help` изучить справочную информацию о командах системы `svn` (`ls`, `add`, `commit`) и попытаться понять их назначение:

```
svn help имя_команды
```

2. Просмотреть домашний (текущий) каталог (команда операционной системы `ls`).

3. Просмотреть содержимое репозитория, хранящегося на сервере. В команде `svn ls` следует указать сетевой путь к репозиторию, состоящий из названия протокола доступа в репозиторий — `svn`, имени сервера — `eniac`, имени репозитория — `test`. **Фактическое размещение репозитория, используемого при выполнении лабораторных работ, следует узнать у преподавателя** (для студенческих групп вместо `test` может использоваться каталог с именем группы, например, `ikp171`).

```
svn ls svn://eniac/test
```

4. Получить локальную (рабочую) копию репозитория:

```
svn checkout svn://eniac/test
```

5. Просмотреть домашний каталог, убедиться, что каталог `test` получен.

6. Перейти в рабочий каталог `test` и обновить его содержимое с помощью команды `svn update`. Так как каталог `test` и репозиторий пусты и обновлять нечего, результатом работы будет только информация о номере текущей ревизии.

**Замечание 1:** вся работа с репозиторием должна проводиться в пределах этого каталога. **Замечание 2:** дальнейшая работа с репозиторием (на следующих лабораторных работах) должна начинаться с обновления вашей локальной копии репозитория (команда `svn update`) для получения последних версий файлов.

7. В рабочем каталоге `test` (локальной или рабочей копии репозитория) создать каталог с вашим именем (`name`). Использовать команду операционной системы `mkdir name`.

8. Передать созданный каталог под управление репозитория:

```
svn add name
```

9. Просмотреть содержимое репозитория `svn ls svn://eniac/test`, убедиться, что на данном этапе каталога `name` в репозитории нет.
10. Экспортировать каталог в репозиторий:  
`svn commit name`  
или  
`svn commit`  
или  
`svn commit - -message "Поясняющий текст"`
11. Повторно просмотреть репозиторий и убедиться, что каталог `name` в репозитории появился.
12. Перейти в каталог `name`.
13. Создать текстовый документ `first.html` с произвольным текстом с помощью простейшего текстового редактора (`vim`, `nano`, `gedit`, `mousepad`) и сохранить его в текущем каталоге `name`.
14. Добавить файл `first.html` в репозиторий:  
`svn add ./first.html`
15. Сохранить `first.html` в репозитории с помощью команды `commit` и запомнить номер ревизии и выполненное действие (сделать комментарий к коммиту).
16. Просмотреть содержимое репозитория и его локальной копии . Какие изменения произошли после выполнения предыдущего пункта?
17. Просмотреть список ревизий (`svn log`), найти в нем последнюю выполненную вами ревизию.
18. Выполнить обновление локальной копии репозитория (`svn update`), еще раз просмотреть список ревизий и сделать вывод.
19. Просмотреть текущее состояние репозитория (`svn status`).
20. Отредактировать файл `first.html` таким образом, чтобы в нем было не менее четырех строчек, и сохранить.
21. Просмотреть локальные изменения файла `first.html`  
`svn diff ./first.html`
22. Выполнить обновление файла `first.html` в репозитории  
`svn commit ./first.html`  
и запомнить номер ревизии и выполненное действие.
23. Повторить просмотр локальных изменений файла `first.html`. Сделать вывод о работе команды `diff`.
24. Просмотреть текущее состояние репозитория и сделать вывод.
25. Создать в каталоге `test` документ `second.html` с помощью простейшего текстового редактора и сохранить его в локальном каталоге `name`.
26. Повторить действия из пп.14-24 для файла `second.html`.
27. Отредактировать и сохранить оба файла `first.html` и `second.html`. В процессе редактирования выполнить следующие действия: добавление новой строки, удаление старой строки, изменение одной из старых строк.

28. Отправить оба файла в репозиторий.
29. Просмотреть список ревизий для файла `first.html` и определить номер ревизии, соответствующий последней редакции файла
- ```
svn log ./first.html
```
30. Откатить файл `first.html` к предыдущей ревизии, для этого прочитайте справочный материал о команде `svn update`. Изменился ли статус репозитория?
31. Удалить файл `second.html` из локального каталога `name` средствами операционной системы. Как изменился статус репозитория?
32. Выполнить обновление локальной папки `name` и ее просмотр. Какие версии файлов были загружены? Сделайте вывод.
33. Просмотреть различия между данной версией файла и предыдущей для `second.html`:
- ```
svn diff -r номер_версии_1:номер_версии_2 ./second.html
```
34. Показать содержимое репозитория и отчет преподавателю.

## ***Справочный материал***

### **1. Термины система управления версиями Subversion**

**Subversion** — это бесплатное клиент-серверное приложение с открытым исходным кодом. Subversion позволяет сохранять каталоги и файлы, отслеживать изменения, сделанные в них, дает доступ к более ранним версиям файлов, помогает восстановить удаленные. Хранение данных осуществляется в централизованном хранилище. Subversion может управлять различными типами данных, но наиболее эффективно применение системы для управления текстовыми документами, например, файлами, содержащими программный код.

**Репозиторий** — каталог на сервере, настроенный для хранения данных в системе управления версиями.

**Локальная (рабочая) копия репозитория** — локальный каталог находящийся в домашнем каталоге пользователя.

**Ревизия** — порядковый номер состояния репозитория. Например, при отправке в репозиторий отредактированных файлов, формируется новое состояние репозитория с новым номером.

Команды SVN

**svn help имя\_команды** – получение справочной информации о команде системы svn.

**svn update** — получение обновлений из репозитория. Этой командой также можно получить более раннюю версию файла:

```
svn update -r номер_ревизии имя_файла
```

**svn commit имя\_файла** — отправка в репозиторий (коммит) файла (папки), при этом порядковый номер отправки данных в репозиторий присваивается соответствующей ревизии. Во время выполнения команды будет открыто окно

текстового редактора (vim) для ввода комментария к коммиту. В комментарии рекомендуется кратко описать, какие объекты были изменены/удалены/добавлены, кем, охарактеризовать сделанные изменения.

**svn commit** — отправка в репозиторий (коммит) **всех** измененных файлов и каталогов, аналогично команде **svn commit имя\_файла**.

**svn commit - - message "Поясняющий текст"** - отправка в репозиторий (коммит) файла (папки) и комментария к коммиту.

**svn ls URL-адрес репозитория** — просмотр каталога репозитория. Также команда используется для просмотра каталогов в репозитории. URL-адрес представляет собой строку вида: название\_протокола://имя\_сервера/путь, например, svn://eniac/test. Протокол svn — специальный протокол для доступа к svn-серверу.

**svn log имя\_файла** — команда показывает список ревизий, в которых изменялся данный файл.

**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 имел один тип (файл, каталог, ссылка), но был замещен объектом другого типа,

! - объект удален сторонними средствами без участия svn,

? - объект не находится под контролем svn,

\* - на сервере имеется более новая версия объекта.

**svn diff имя\_файла** — команда показывает локальные изменения файла построчно.

**svn diff -r N:M имя\_файла** – команда показывает различия в текстах файла, соответствующих ревизиям с номерами N и M.

## Текстовый редактор vim

vim имя\_файла — запуск редактора из командной строки,

a – переход в режим ввода текста,

ESC – переход в обычный режим редактора,

: - переход в командный режим редактора

wq — сохранение файла и выход,

q — выход из редактора без сохранения (при этом файл не должен быть изменен),

q! - выход из редактора без сохранения с отбрасыванием сделанных изменений.

## Команды управления каталогами и файлами в Linux

man команда — получение справочной помощи о команде Linux.

mkdir имя\_каталога — создать каталог,

mkdir -p путь/имя\_каталога — создание нового каталога и каталогов, указанных в пути, если они еще не созданы,

ls путь/имя\_каталога — просмотреть каталог,

cd путь/имя\_каталога — перейти в другой каталог,

more путь/имя\_файла — просмотреть содержимое файла,

rm путь/имя\_файла — удалить файл,

rmdir имя\_каталога — удалить пустой каталог,

nano имя\_файла — создание текстового файла с помощью редактора nano.

Обозначения каталогов, используемые для записи пути:

. - текущий каталог,

.. - родительский каталог,

~ - домашний каталог пользователя.

ls -a ./ - просмотр подробной информации о текущем каталоге,

ls mydir — просмотр краткой информации о каталоге mydir, входящем в текущий каталог,

more ./my.txt — просмотр файла, находящегося в текущем каталоге,

rm ../hello.txt — удаление текстового файла из родительского каталога,

cd ../mydir — переход из каталога в каталог того же уровня.

mv исходный\_файл выходной\_файл — переименование файла (каталога),

mv путь/исходный\_файл новый\_путь — перемещение файла (каталога).

## Вопросы к защите

1. Какое назначение имеет программы Subversion?
2. Что такое репозиторий?
3. Что такое ревизия?
4. Как просмотреть содержимое репозитория?
5. Как добавить файл, каталог в репозиторий?
6. Как обновить файлы в рабочей папке?
7. Когда выполняется обновление файлов в рабочей папке?
8. Какая команда используется для отправки измененных файлов в репозиторий?
9. Как просмотреть предыдущие ревизии?
10. Как обнаружить расхождения между репозиторием и его локальной копией?
11. Как откатиться к предыдущей версии файла?
12. Как сравнить содержимое одного и того же файла из разных ревизий?