

Система контроля версий

Зачем это нужно?

- Хранение версий файлов, причем обычно хранятся только изменения между предыдущей и текущей версией и таким образом хранилище не растет слишком быстро;
- Возможность получить любые предыдущие версии хранимых файлов;
- Просмотр изменений внесенных между заданными в запросе версиями;
- Сохранение и просмотр комментариев и авторов к внесенным изменениям;

RCS

- RCS (Revision Control System, Система контроля ревизий) была разработана в начале 1980-х годов Вальтером Тичи (Walter F. Tichy).
- Система позволяет хранить версии только одного файла.
- Информация о версиях хранится в специальном файле с именем оригинального файла к которому в конце добавлены символы ',v'.

RCS

- Для хранения версий система использует утилиту diff.

Рассмотрим пример сессии с RCS:

1. Когда мы хотим положить файл под контроль RCS мы используем команду ci (от check-in, зарегистрировать):

`ci file.txt`

Создается файл `file.txt,v` ; удаляется `file.txt` (если не сказано этого не делать).

Запрашивается описание для всех хранимых версий.

RCS

`co file.txt`

Эта команда вынимает последнюю версию файла из `file.txt,v`. Теперь можно редактировать файл `file.txt`

`ci file.txt`

При выполнении этой команды система запросит описание изменений и затем сохранит новую версию файла.

RCS

Недостатки:

- Работа только с одним файлом, каждый файл должен контролироваться отдельно;
- Неудобный механизм одновременной работы нескольких пользователей с системой, хранилище просто блокируется пока заблокировавший его пользователь не разблокирует его;

CVS

- CVS (Concurrent Versions System, Система совместных версий) пока остается самой широко используемой системой, но быстро теряет свою популярность из-за недостатков. Дик Грун (Dick Grune) разработал CVS в середине 1980-х.
- Для хранения индивидуальных файлов CVS (также как и RCS) использует файлы в RCS формате, но позволяет управлять группами файлов расположенных в директориях.

CVS

- CVS использует клиент-сервер архитектуру в которой вся информация о версиях хранится на сервере. Это позволяет использовать CVS даже географически распределенным командами пользователей где каждый пользователь имеет свой рабочий директорий с копией проекта.

CVS

- Пользователи могут использовать систему совместно.
- Возможные конфликты при изменении одного и того же файла разрешаются тем, что система позволяет вносить изменения только в самую последнюю версию файла.
- CVS также позволяет вести несколько линий разработки проекта с помощью ветвей (branches) разработки.

CVS

Пример сессии с CVS:

Прежде всего надо импортировать проект в CVS, это делается с помощью команды `import` (импортировать):

- `cd some-project`
- `cvs import -m "New project" path-in-repository none start`

CVS

Далее необходимо создать новый директорий в котором будет находится рабочая копия проекта под контролем CVS и загрузить проект с помощью команды checkout (контроль), или сокращенно co:

- `cd some-working-dir`
- `cvs checkout path-in-repository`

CVS

Внести в проект изменения и залить их в репозиторий можно с помощью команды `commit` (совершить изменения), или сокращенно `ci`:

- `cv commit -m "Some changes"`

CVS

Обновить рабочий директорий новой версией проекта из репозитория можно с помощью команды `update` (обновить), или сокращенно `up`:

- `cv update`

CVS

Недостатки:

- Так как версии хранятся в файлах RCS нет возможности сохранять версии директорий. Стандартный способ обойти это препятствие - это сохранить какой-либо файл (например, README.txt) в директории;
- Перемещение, или переименование файлов не подвержено контролю версий. Стандартный способ сделать это: сначала скопировать файл, удалить старый с помощью команды `cv remove` и затем добавить с его новым именем с помощью команды `cv add`;

SVN

- [Subversion \(SVN\)](#) Subversion (SVN) был разработан в 2000 году по инициативе фирмы [CollabNet](#).
- SVN также как и CVS использует клиент-сервер архитектуру.
- Изменения по сравнению с CVS:
 - A. Атомарное внесение изменений (commit). В случае если обработка коммита была прервана не будет внесено никаких изменений.

SVN

- Б. Переименование, копирование и перемещение файлов сохраняет всю историю изменений.
- В. Директории, символические ссылки и метаданные подвержены контролю версий.
- Г. Эффективное хранение изменений для бинарных файлов.

SVN

Примеры команд (во многом повторяют cvs):

- `cd some-project`
- `svn import -m "New project" path-in-repository`

не нужно указывать метки разработчика и проекта

- `cd some-working-dir`
- `svn checkout path-in-repository`

SVN

- `svn commit -m "Some changes"`
- `svn up`

Темы докладов:

Концепции распределенной системы
контроля версий:

- Git
- Mercurial
- Bazaar
- Codeville
- Darcs
- Monotone

Практические задания:

1. Написать последовательности действий, которые надо совершить, чтобы перезаписать в файл `C:\CurrentRunNumber\data.txt` Номер запуска (какое либо число). При помощи команд командной строки
2. Прodelать то же самое в RCS
3. Изменить файл из CVS
4. Перезаписать файл в SVN