

# **Особенности использования систем контроля версий.**

## **Цель работы.**

Изучить особенности различных систем контроля версий.

## **Теоретические сведения.**

Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.

### **Локальные системы контроля**

Самая простая система контроля версий представляет собой простое копирование файлов в другой каталог (как правило добавляя текущую дату к названию каталога). Такой подход очень распространён, потому что прост, но он и чаще даёт сбои. Велика вероятность ошибиться каталогом, а также сложно найти среди сохранённых версий конкретную.

Чтобы решить эту проблему, были разработаны локальные СКВ с простой базой данных, в которой хранятся все изменения нужных файлов (рисунок 1).

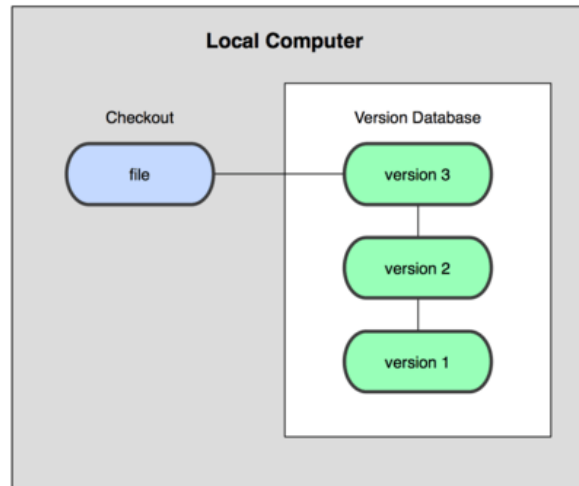


Рисунок 1 - Схема локальной СКВ

Одной из наиболее популярных СКВ такого типа является rcs.. Эта утилита основана на работе с наборами патчей между парами версий (патч — файл, описывающий различие между файлами), которые хранятся в специальном формате на диске. Это позволяет пересоздать любой файл на любой момент времени, последовательно накладывая патчи. Одним из важных достоинств данной утилиты является значительное уменьшение занимаемого старыми версиями места на диске. Однако в подобных системах крайне ограничены возможности по совместной работе нескольких разработчиков над одним проектом.

### **Централизованные системы контроля версий**

Для обеспечения совместной работы нескольких разработчиков над одним проектом были созданы централизованные системы контроля версий (ЦСКВ). В таких системах (например CVS, Subversion и Perforce) есть центральный сервер, на котором хранятся все файлы под версионным контролем, и ряд клиентов, которые получают копии файлов из него.(рисунок 2).

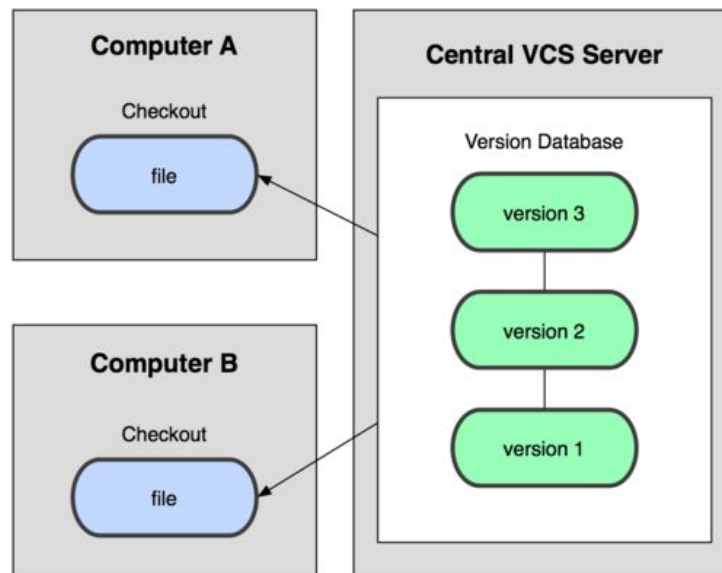


Рисунок 2 - Схема централизованного контроля версий

Такой подход имеет множество преимуществ, особенно над локальными СКВ. К примеру, все знают, кто и чем занимается в проекте. У администраторов есть чёткий контроль над тем, кто и что может делать, и возможности по администрированию ЦСКВ намного богаче чем в случае использования локальных баз на каждом клиенте.

Однако при таком подходе есть и несколько серьёзных недостатков. Наиболее очевидный — централизованный сервер является уязвимым местом всей системы. Если сервер выключается на час, то в течение часа разработчики не могут взаимодействовать, и никто не может сохранить новой версии своей работы. Если же повреждается диск с центральной базой данных и нет резервной копии, вы теряете абсолютно всё — всю историю проекта, разве что за исключением нескольких рабочих версий, сохранившихся на рабочих машинах пользователей. Локальные системы контроля версий подвержены той же проблеме: если вся история проекта хранится в одном месте, вы рискуете потерять всё.

### **Распределённые системы контроля версий**

Для устранения уязвимого звена в виде центрального сервера были созданы распределённые системы контроля версий (РСКВ). В таких системах как Git, Mercurial, Bazaar или Darcs клиенты не просто выгружают последние версии файлов, а полностью копируют весь репозиторий. Поэтому в случае, когда "умирает" сервер, через который шла работа, любой клиентский репозиторий может быть скопирован обратно на сервер, чтобы восстановить базу данных. Каждый раз, когда клиент забирает свежую версию файлов, он создаёт себе полную копию всех данных (рисунок 3).

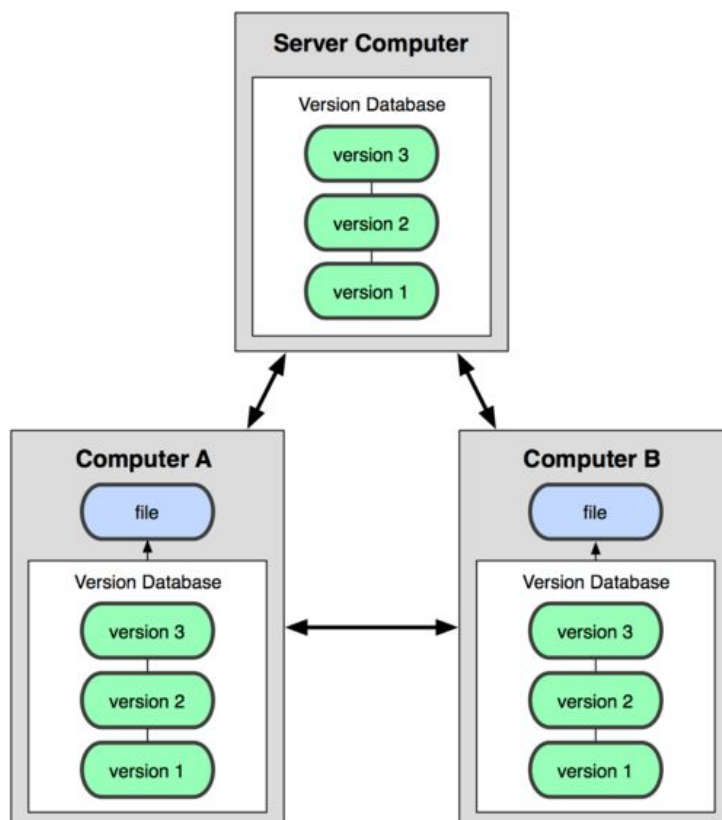


Рисунок 3 - Схема распределённой системы контроля версий

Кроме того, в большей части этих систем можно работать с несколькими удалёнными репозиториями, таким образом, можно одновременно работать по-разному с разными группами людей в рамках одного проекта. Так, в одном проекте можно одновременно вести несколько типов рабочих процессов, что невозможно в централизованных системах. Однако при этом повышается сложность синхронизации состояний репозитория на различных клиентах, а

также общая уязвимость системы к утечке данных. Также повышается нагрузка на администратора проекта при разрешении конфликтов версий в ситуациях когда несколько разработчиков предлагают разные модификации одних и тех же файлов.

### **Задания к лабораторной работе.**

#### **Часть 1. Работа с локальным репозиторием.**

1. Используя систему контроля версий git, создать локальный репозиторий для программы, написанной в третьей работе;
2. Используя команду commit, поместить текущее состояние файлов с исходными текстами программы и библиотек в созданный репозиторий.
3. Создать две ветви программы в репозитории - в одной убрать из программы поддержку ОС семейства Windows, в другой - ОС семейства Linux
4. Сравнить ветви и показать найденные конфликты.
5. Просмотреть историю изменений
6. Восстановить изначальную версию программы из репозитория

В отчете привести использованные команды и результаты их исполнения.

#### **Часть 2. Работа с глобальным репозиторием.**

1. Самостоятельно изучить различия основных систем контроля версий - CVS, SVN, Git.
2. Рассмотреть доступные глобальные репозитории для любой из систем контроля версий
3. Описать процесс регистрации и работы с одним из рассмотренных глобальных репозиториев.

В отчете привести сравнение систем контроля версий, обзор различных глобальных репозиториев для одной из них и описание процесса работы с одним из глобальных репозиториев.

### **Контрольные вопросы.**

- 1) Что такое система контроля версий?
- 2) Что представляет собой репозиторий в системе контроля версий и что конкретно хранит?
- 3) Что такое ветви версий в репозитории, что происходит при слиянии ветвей и что такое конфликты ветвей?
- 4) Основные команды Git.
- 5) Основные команды SVN
- 6) Отличия Git и SVN

### **Использованные и рекомендуемые источники:**

1. <https://biz30.timedoctor.com/ru/c%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0-%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8F-%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B9/>
2. <https://stackoverflow.com/questions/802573/difference-between-git-and-cvs>
3. <https://git-scm.com/book/ru/v1/Введение-О-контроле-версий>
4. <https://habr.com/ru/post/440816/>
5. <https://habr.com/ru/company/playrix/blog/345732/>
6. <http://svnbook.red-bean.com/nightly/ru/svn-book.html>