Лабораторная работа No2

1. Система контроля версий (СКВ) -- программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение.
2. Хранилище — это содержимое скрытой папки .git. В этой папке хранятся все версии рабочей области и служебная информация.
   1. COMMIT (с англ. — «совершать, фиксировать») — оператор управления транзакциями языка SQL для успешного завершения транзакции
   2. Одним из основных и наиболее мощных инструментов для этого является команда git log (история коммитов)
   3. Рабочая копия- это созданная клиентской программой Subversion локальная копия части данных из хранилища, содержащая помимо собственно данных некоторую служебную информацию (скрытые директории с именем .svn)
3. 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, конечно, администрировать ЦСКВ гораздо легче, чем локальные базы 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 новые версии. 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 управления версиями подвержены той же проблеме: если вся история проекта хранится в одном месте, вы рискуете потерять всё.

1. По умолчанию Git не кеширует учётные данные совсем. Каждое подключение будет запрашивать у вас логин и пароль.

В режиме “cache” учётные данные сохраняются в памяти в течение определённого периода времени. Ни один из паролей никогда не сохраняется на диск и все они удаляются из кеша через 15 минут.

В режиме “store” учётные данные сохраняются на неограниченное время в открытом виде в файле на диске. Это значит что, до тех пор пока вы не измените пароль к Git-серверу, вам не потребуется больше вводить ваши учётные данные. Недостатком такого подхода является то, что ваш пароль хранится в открытом виде в файле в вашем домашнем каталоге.

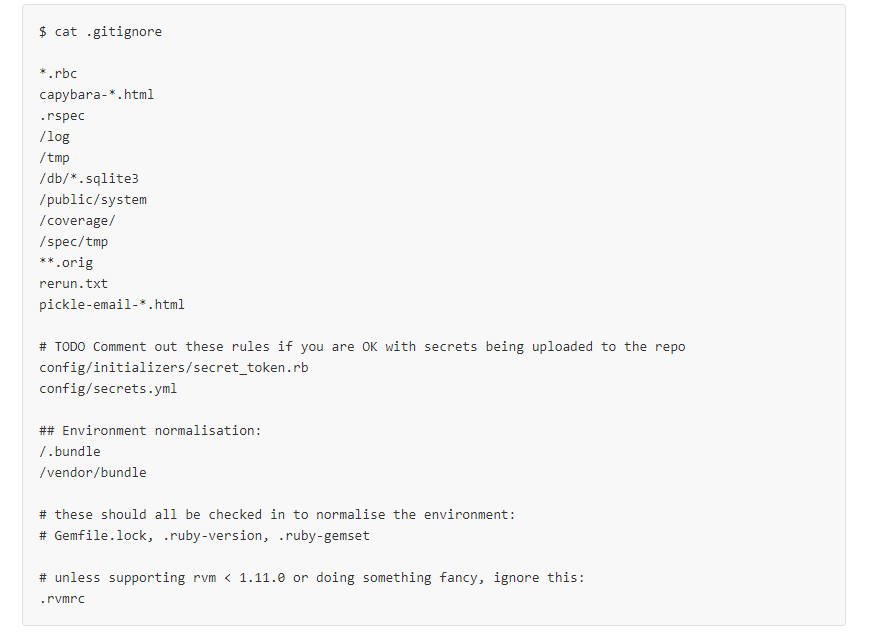
На случай если вы используете Mac, в Git есть режим “osxkeychain”, при использовании которого учётные данные хранятся в защищённом хранилище, привязанному к вашему системному аккаунту. В этом режиме учётные данные сохраняются на диск на неограниченное время, но они шифруются с использованием той же системы, с помощью которой сохраняются HTTPS-сертификаты и автозаполнения для Safari.

В случае если вы используете Windows, вы можете установить помощник, называемый “winstore”. Он похож на “osxkeychain”, описанный выше, но для управления секретной информацией использует Windows Credential Store.

1. Для получения возможности редактирования объекта необходимо произвести его захват в хранилище. При выполнении захвата из хранилища будет получена (в случае наличия) новая версия объекта разработки. После внесения изменений объект необходимо поместить в хранилище, после этого обновленная версия объекта становится доступной для других пользователей, участвующих в процессе разработки конфигурации. Если же изменения не должны быть помещены в хранилище, то предусмотрена процедура отмены захвата, при которой из хранилища получается старая версия объекта разработки. Таким образом, можно выделить четыре простейшие операции работы с хранилищем:

* захват объекта в хранилище;
* помещение объекта в хранилище;
* получение объекта из хранилища;
* отмена захвата в хранилище.

1. diff — утилита сравнения файлов, выводящая разницу между двумя файлами. Эта программа выводит построчно изменения, сделанные в файле (для текстовых файлов). Современные реализации поддерживают также двоичные файлы. Вывод утилиты называется «diff», или, что более распространено, патч, так как он может быть применён с программой patch. Вывод других утилит сравнения файлов также часто называется «diff».
2. Merge объединяет свою копию с изменениями в новой ветке
3. …
4. Объединение результатов работы с хранилизами различных версий файлов
5. Ветвь в Git — это просто легковесный подвижный указатель на один из этих коммитов. Имя ветки по умолчанию в Git — master. Когда вы вначале создаёте коммиты, вам даётся ветвь master, указывающая на последний сделанный коммит.
6. Rebase — один из двух способов объединить изменения, сделанные в одной ветке, с другой веткой. Начинающие и даже опытные пользователи git иногда испытывают нежелание пользоваться ей, так как не видят смысла осваивать еще один способ объединять изменения, когда уже и так прекрасно владеют операцией merge.
7. Зачастую, у вас имеется группа файлов, которые вы не только не хотите автоматически добавлять в репозиторий, но и видеть в списках неотслеживаемых (а чем взрослее проект, тем больше таких файлов может быть). К таким файлам обычно относятся автоматически генерируемые файлы (различные логи, результаты сборки программ и т.п.)



К шаблонам в файле .gitignore применяются следующие правила:

* Пустые строки, а также строки, начинающиеся с #, игнорируются.
* Можно использовать стандартные glob шаблоны.
* Можно заканчивать шаблон символом слэша (/) для указания каталога.
* Можно инвертировать шаблон, использовав восклицательный знак (!) в качестве первого символа.