Настройка хранилища CVS - подход FreeBSD

Stijn Hoop <stijn@win.tue.nl >
Издание: 45050

Авторские права © 2001, 2002, 2003 Stijn Hoop

FreeBSD это зарегистрированная торговая марка FreeBSD Foundation.

Многие из обозначений, используемые производителями и продавцами для обозначения своих продуктов, заявляются в качестве торговых марок. Когда такие обозначения появляются в этом документе, и Проекту FreeBSD известно о торговой марке, к обозначению добавляется знак «TM» или «(R)».

2014-06-13 taras.

Аннотация

В этой статье описаны шаги, которые я предпринял для настройки хранилища CVS, использующего те же самые скрипты, что используются в проекте FreeBSD в их настройке. Это имеет некоторые преимущества перед стандартной настройкой CVS, в том числе более точный контроль доступа к дереву исходных текстов и посылку содержательных сообщений электронной почты при каждом коммите.

Содержание

1. Введение 1
2. Первоначальная настройка 2
3. Специфичная для FreeBSD настройка 9

1. Введение

Большинство программных проектов с открытым кодом используют CVS в качестве системы управления исходным кодом. Хотя CVS весьма хороша в этом качестве, у неё есть свои неудобства и недостатки. Одним из них является то, что совместное использование дерева исходных текстов с другими разработчиками может быстро

привести к кошмарным проблемам при администрировании, особенно если кто-то захочет защитить части дерева от общедоступности.

FreeBSD является одним из проектов, использующим CVS. Здесь также имеет большое количество разработчиков, разбросанных по всему миру. Они разработали некоторые скрипты, облегчающие управление хранилищем. Недавно Josef Karthauser <joe@FreeBSD.org > пересмотрел и привел в порядок эти скрипты в целях облегчения их использования в других проектах. В этой статье описан один из методов использования новых скриптов.

Чтобы извлечь максимум информации из этой статьи, вы должны владеть основными методами работы с CVS.

2. Первоначальная настройка

Предупреждение

Наверное, лучше сначала выполнить эту процедуру с пустым тестовым хранилищем, чтобы понять все последствия ваших действий. Как обычно в таких случаях, у вас должны иметься свежие читаемые резервные копии!

2.1. Инициализация хранилища

Первым делом при настройке нового хранилища необходимо его инициализировать, для чего выдать CVS такую команду:

```
% cvs -d path-to-repository  init
```

Результатом ее выполнения будет созданный CVS служебный каталог `CVSROOT`, в котором выполняется вся настройка.

2.2. Группа пользователей хранилища

Теперь мы создадим группу, которая будет владеть хранилищем. В этой группе должны присутствовать все коммиттеры, для того, чтобы они могли писать в хранилище. Для этой группы мы примем стандартное для FreeBSD название `ncvs`.

```
# pw groupadd ncvs
```

Затем вы должны при помощи команды chown(8) сменить владельца и группу для только что добавленного каталога:

```
# chown -R :ncvs path-to-your-repository
```

Это нужно для того, чтобы никто не мог записывать в хранилище, не являясь членом группы.

2.3. Получение исходных текстов

Теперь вам нужно получить каталог `CVSROOT` из хранилища FreeBSD. Проще всего это делается извлечением с анонимного зеркала CVS FreeBSD. Обратитесь к соответствующей главе Руководства для получения дополнительной информации. Мы будем полагать, что исходные тексты хранятся в подкаталоге `CVSROOT-freebsd` текущего каталога.

2.4. Копирование скриптов FreeBSD

Теперь мы скопируем исходные тексты FreeBSD из `CVSROOT` в наше хранилище. Если вы знакомы с CVS, то для вас может иметь смысл попытаться импортировать скрипты, чтобы облегчить синхронизацию с последующими версиями. Однако при этом оказывается, что CVS имеет в этой области недостаток: при импортировании исходных текстов в каталог `CVSROOT` она не будет обновлять необходимые административные файлы. Чтобы в этом убедиться, вам нужно проверить каждый файл после импортирования, при этом смысл `cvs import` теряется. Поэтому рекомендуемым методом является простое копирование скриптов.

Не имеет значения, как вы относитесь к предыдущему параграфу-результат один и тот же. Просто поместите ваш `CVSROOT` и скопируйте файлы FreeBSD поверх ваших локальных (неизмененных) копий:

```
% cvs -d path-to-your-repository  checkout CVSROOT
% cd CVSROOT
% cp ../CVSROOT-freebsd/* .
% cvs add *
```

Заметим, что вы, скорее всего, получите несколько предупреждений о том, что некоторые каталоги не были скопированы; это нормально, вам они не нужны.

2.5. Скрипты

Теперь у вас есть рабочий каталог и точная копия скриптов, которые используются в проекте FreeBSD для работы с хранилищем. Далее следует краткое описание назначения каждого файла.

- `access` - по умолчанию при стандартной настройке этот файл не используется. Он применяется в специфичных для проекта FreeBSD настройках, где он управляет доступом к хранилищу. Вы можете удалить этот файл, если вы не собираетесь использовать такую настройку.

- `avail` - этот файл управляет доступом к хранилищу. В нем вы можете указать группы людей, которым разрешен доступ к хранилищу, а также запретить ком-

миты на уровне каталогов. Вы должны поднастроить его так, чтобы он содержал группы и каталоги, имеющиеся в вашем хранилище.

- `cfg.pm` - этот файл анализирует вашу конфигурацию и содержит настройки по умолчанию. Вы *не* должны изменять этот файл. Вместо этого размещайте ваши изменения в конфигурации в файле `cfg_local.pm` .

- `cfg_local.pm` - этот файл содержит все настраиваемые параметры системы. Вы должны настраивать все параметры здесь, например, куда посылается почта при коммите, с каких хостов можно выполнять коммиты, и прочее. Ниже дается более полная информация об этом.

- `checkoutlist` - эти файлы перечисляют все файлы, управляемые CVS в этом каталоге. Вы должны отредактировать его для удаления некоторых специфичных для FreeBSD файлов.

- `commit_prep.pl` - этот скрипт выполняет различные проверки перед выполнением коммита, в зависимости от того, включили ли вы их в `cfg_local.pm` . Вам не нужно его трогать.

- `commitcheck` - этот скрипт вызывается непосредственно из CVS. Сначала он проверяет, с использованием `cvs_acls.pl` , имеет ли коммитер доступ к указанной части дерева, а затем запускает `commit_prep.pl` для выполнения различных проверок перед коммитом. Если они выполнились нормально, то CVS позволит выполнить коммит. Вам не нужно трогать этот файл.

- `commitinfo` - этот файл используется в CVS для определения того, какой скрипт запускать перед коммитом-в данном случае `commitcheckl` . Вам не нужно трогать этот файл.

- `config` - конфигурационный файл для этого хранилища. Вы должны изменять его при необходимости, но большинство администраторов могут оставить все настройки по умолчанию. Дополнительную информацию о параметрах, которые могут быть здесь заданы, можно найти в руководстве по CVS.

- `cvs_acls.pl` - этот скрипт идентифицирует пользователя и то, имеет ли он доступ к дереву. Это делается на основе информации в файле `avail`. Вам не нужно трогать этот файл.

- `cvsignore` - этот файл перечисляет файлы, которые CVS не должна помещать в хранилище. Вы можете отредактировать его по своему усмотрению. Более полная информация об этом файле находится в руководстве по CVS.

- `cvswrappers` - этот файл используется в CVS для включения или выключения расширения ключевых слов, или должен ли файл считаться бинарным. Вы можете редактировать его по своему усмотрению. Более полная информация об этом файле находится в руководстве по CVS. Имейте ввиду, что опции CVS `-t` и `-f` некорректно работают в режиме клиент/сервер.

- `edithook` - этот файл больше не используется, но оставлен по историческим причинам. Вы можете спокойно удалить этот файл.

- `editinfo` - CVS использует этот файл для настройки редактора. FreeBSD не использует эту функциональность, так как обработка сообщений для журнала выполняется в файлах `verifymsg` и `logcheck`. Это происходит по той причине, что `editinfo` некорректно работает в режиме клиент/сервер, или в случаях когда используются опции `-m` или `-F`. Вам не нужно трогать этот файл.

- `exclude` - в этом файле перечислены регулярные выражения, используемые `commit_prepl.pl` для выделения файлов, в которых могут не содержаться заголовки с номером версии. В настройке FreeBSD все файлы содержащиеся в хранилище должны иметь заголовок с версией (типа $FreeBSD$). Все файлы с именами, которые соответствуют одной из строк этого файла, исключаются из проверки. Вы должны добавить выражения в этот файл, если вы помещаете в хранилище файлы, которые не могут иметь заголовки с версиями. Для целей установки скриптов лучшим решением может оказаться исключение `CVSROOT/` из проверки заголовков.

- `log_accum.pl` - это скрипт, который принимает журнальное сообщение в виде, данном скриптом `logcheck`, и добавляет его к файлу журнала в хранилище для хранения резервной копии. Он также отрабатывает посылку сообщения по электронной почте на адрес, который вы зададите (в файле `cfg_local.pm`). Он подключается к CVS через `loginfo`. Вам не нужно трогать этот файл.

- `logcheck` - при коммите этот файл анализирует сообщение для журнала, которое составляют коммиттеры, и пытается его некоторым образом улучшить. Он подключается к CVS через `logcheck`. Вам не нужно трогать этот файл.

Примечание

Этот скрипт зависит от ряда локальных модификаций CVS, сделанный во FreeBSD: эта версия читает журнальное сообщение повторно после того, как этот скрипт его модифицирует. Стандартная версия CVS этого не делает, что делает этот скрипт бесполезным, так как он не может модифицировать журнальное сообщение, хотя может проверить его на предмет правильности синтаксиса. CVS версии 1.11.2 и выше может быть настроен, чтоб иметь поведение подобное FreeBSD, путем установки опции `ReadLogAfterVerify=always` в файле `config`.

- `loginfo` - этот файл используется CVS для управления того, куда посылается протокольная информация. С помощью этого файла подключается `log_accum.pl` . Вам не нужно трогать этот файл.

- `modules` - этот файл сохраняет своё традиционное назначение в CVS. Вы должны удалить модули FreeBSD из стандартной версии. Вы можете редактировать этот файл по своему усмотрению. Более полная информация об этом файле находится в руководстве по CVS.

- `notify` - этот файл используется в CVS в том случае, если кто-то задаст отслеживание файла. Это не используется в хранилище FreeBSD. Вы можете редактировать его по своему усмотрению. Более полная информация об этом файле находится в руководстве по CVS.

- `options` - этот файл специфичен для версии CVS от FreeBSD, но также поддерживается версией для Debian. Он содержит ключевое слово для расширения в заголовках версий. Вы должны заменить это на ключевое слово, которое вы задали в файле `cfg_local.pm` (если вы используете эту возможность, которая в настоящее время специфична для FreeBSD).

- `rcsinfo` - этот файл отображает каталоги в хранилище на файлы шаблонов, как например `rcstemplate` . По умолчанию FreeBSD использует один шаблон для всего хранилища. Вы можете добавлять другие к этому файлу по своему усмотрению.

- `rcstemplate` - этот файл является актуальным файлом шаблона, который видят коммиттеры, когда помещают что-то в хранилище. Вы должны отредактировать его для описания различных дополнительных параметров, которые вы определили в `cfg_local.pm` .

- `tagcheck` - эти файлы управляют доступом к созданию меток в хранилище. Стандартная для FreeBSD версия не позволяет создавать метки с именами типа RELENG* из-за пересечения с процессом создания релизов. Вы должны отредактировать этот файл по вашему усмотрению.

- `taginfo` - этот файл ставит в соответствие операции с метками над каталогами хранилища скриптам управления доступом, например `tagcheck` . Вам не нужно трогать этот файл.

- `unwrap` - этот скрипт нужен для автоматической обратной обработки («unwrap») двоичных файлов (посмотрите `cvswrappers`) при извлечении. Это не используется в текущей настройке FreeBSD по причине того, что не работает с конфигурацией клиент/сервер. Вам не нужно трогать этот файл.

- `verifymsg` - этот файл ставит в соответствие каталогам хранилища скрипты вторичной обработки журнальных сообщений, например `logcheck` . Вам не нужно трогать этот файл.

- `wrap` - этот скрипт может быть использован для автоматической обработки («wrap») двоичных файлов (посмотрите `cvswrappers`) при помещении в хранилище. Это не используется в текущей настройке FreeBSD, по причине того, что не работает с конфигурацией клиент/сервер. Вам не нужно трогать этот файл.

2.6. Настройка скриптов

Следующим шагом является настройка скриптов так, чтобы они работали в ваших условиях. Вы должны просмотреть все файлы в каталоге и выполнить ваши настройки. В частности, вы может потребоваться отредактировать следующие файлы:

1. Если вы не хотите использовать специфичные для FreeBSD возможности скриптов, то вы можете без последствий удалить файл `access`:

   ```
   % cvs rm -f access
   ```

2. Отредактируйте `avail` так, чтобы он содержал различные каталоги хранилища, доступом к которым вы хотите управлять. Обязательно сохраните строчку `avail||CVSROOT` , иначе вы заблокируете сами себя на следующем шаге.

   Другими параметрами, которые вы можете добавить в этот файл, являются группы коммиттеров. По умолчанию FreeBSD использует файл `access` для перечисления всех коммиттеров, но вы можете использовать любой файл. Вы можете также добавить группы, если хотите (синтаксис описан в начале файла `cvs_acls.pl`).

3. Отредактируйте файл `cfg_local.pm` так, чтобы он содержал параметры, которые вы хотите. В частности, вы должны взглянуть на такие пункты настройки:

   - `%TEMPLATE_HEADERS` - они обрабатываются скриптами ведения протокола, и вставляются ниже почтового сообщения, если они присутствуют и не являются пустыми в сообщении при коммите. Вы можете, наверное, удалить строки `PR` и `MFC after`. И, конечно, вы можете добавить свои собственные.

   - `$MAIL_BRANCH_HDR` - если вы хотите в каждое сообщение при коммите вставлять заголовок, описывающий ветку, в которую был выполнен коммит, задайте это в соответствии с вашим окружением. Или оставьте это пустым, если не хотите иметь такой заголовок.

   - `@COMMIT_HOSTS` - задайте здесь список хостов, с которых можно выполнять коммиты.

   - `$MAILADDRS` - задайте здесь адрес администратора или списка, в который должны направляться почтовые сообщения при коммите.

   - `@LOG_FILE_MAP` - измените этот массив по своему усмотрению - каждое регулярное выражение сравнивается с каталогом коммита, и протокольное со-

общение при коммите сохраняется в подкаталоге `commitlogs` в указанном файле.

- `$COMMITCHECK_EXTRA` - если вы не хотите использовать специфичные для FreeBSD проверки доступа, то вы должны удалить определения `$COMMITCHECK_EXTRA` из этого файла.

Примечание

Изменение параметра `$IDHEADER` гарантированно работает только на платформах FreeBSD; это зависит от специфичных для FreeBSD модификаций в CVS.

Вы можете проверить `cfg.pm` на предмет того, какие другие параметры могут быть изменены, но перечисленное выше является достаточным подмножеством.

4. Отредактируйте `exclude` для удаления специфичных для FreeBSD записей (например, всех строк, которые начинаются с `^ports/` и так далее). Более того, закомментируйте строки, начинающиеся с `^CVSROOT/` , и добавьте одну строку только с `^CVSROOT/` . После установки обработчика (wrapper) вы можете добавить свои заголовки к файлам в каталоге `CVSROOT` и восстановить эти строки, но теперь они будут иметь смысл, только когда вы попытаетесь выполнить коммит позже.

5. Отредактируйте файл `modules` и удалите всё, что относится к FreeBSD. Добавьте собственные модули, если хотите.

6.

Примечание

Этот шаг необходим, если только вы задали значение для `$IDHEADER` в `cfg_local.pm` (что работает только при использовании модифицированной во FreeBSD версии CVS).

Отредактируйте файл `options` так, чтобы он соответствовал метке, которую вы задали в `cfg_local.pm` . Глобальный поиск и замена `FreeBSD` на вашу метку должны сработать.

7. Отредактируйте файл `rcstemplate` так, чтобы он содержал те же самые ключевые слова, что заданы в `cfg_local.pm` .

8. Опционально удалите проверки FreeBSD из `tagcheck`. Вы можете просто добавить `exit 0` в начало файла, чтобы запретить все проверки при установке метки.

9. Последним действием, которое нужно сделать перед тем, как закончить работу, является проверка того, что протоколы коммитов могут сохраняться. По умолчанию они сохраняются в хранилище, в подкаталоге `commitlogs` каталога `CVSROOT`. Этот каталог должен быть создан, так что выполните следующее:

```
% mkdir commitlogs
% cvs add commitlogs
```

А теперь, после тщательной проверки, вы должны выполнить коммит ваших изменений. Убедитесь, что вы дали сами себе доступ к каталогу `CVSROOT` в вашем `avail` до того, как его делать, так как в противном случае вы заблокируете сами себя. Так что убедитесь, что всё именно так, как вы и предполагали, а затем выполните следующее:

```
% cvs commit -m ' - Initial FreeBSD scripts commit '
```

2.7. Тестирование настройки

Вы готовы к первому тестированию: принудительному коммиту в файл `avail`, чтобы убедиться, что всё работает так, как ожидалось.

```
% cvs commit -f -m ' Forced commit to test the new CVSROOT scripts ' ↺
avail
```

Если всё работает, поздравляем! Теперь у вас имеется работающая настройка скриптов FreeBSD для вашего хранилища. Если CVS всё ещё о чём-то сообщает, вернитесь и проверьте, все ли вышеупомянутые шаги были выполнены правильно.

3. Специфичная для FreeBSD настройка

Проект FreeBSD сам по себе использует несколько другую настройку, в которой также используются файлы из подкаталога `freebsd` каталога FreeBSD `CVSROOT`. Проект использует их из-за большого количества коммиттеров, которые все должны быть в одной и той же группе. Поэтому был написан простой обработчик, проверяющий, что люди имеют правильные права на выполнение коммита, а затем устанавливающий идентификатор группы, соответствующий идентификатору хранилища.

Если вашему хранилищу это тоже нужно, то шаги для выполнения этого описаны ниже. Но сначала обзор связанных с этим файлов.

3.1. Файлы, используемые в настройке FreeBSD

- `access` - этот файл управляет информацией о доступе. Вы должны отредактировать этот файл для включения всех участников вашего проекта.

- `freebsd/commitmail.pl` - этот файл больше не используется, но оставлен по историческим причинам. Вам не нужно трогать этот файл.

- `freebsd/cvswrap.c` - это исходный текст обработчика CVS, который вам нужно установить, чтобы проверки доступа реально заработали. Дополнительная информация об этом ниже. Вы должны отредактировать маршруты в макросах `ACCESS` и `REALCVS` так, чтобы они соответствовали вашей настройке.

- `freebsd/mailsend.c` - этот файл нужен в настройке FreeBSD для списков рассылки. Вам не нужно трогать этот файл.

3.2. Процедура

1. Отредактируйте файл `access` так, чтобы он содержал только ваше имя пользователя.

2. Отредактируйте `cvswrap.c` так, чтобы он содержал правильный маршрут для вашей настройки. Это определено в макросе по имени `ACCESS`. Вы должны также изменить расположение реального выполнимого файла `cvs`, если оно не подходит к вашей ситуации. Для стандартного `cvswrap.c` предполагается, что он заменит общесистемную команду cvs, которая будет перемещена в `/usr/bin/ncvs`.

   В моём экземпляре `cvswrap.c` помещено следующее:

   ```
   #define ACCESS "/local/cvsroot/CVSROOT/access"
   #define REALCVS "/usr/bin/ncvs"
   ```

3. Следующим шагом является установка обработчика для того, чтобы проверить правильность установки группы при выполнении коммита. Исходные тексты для этого размещены в `cvswrap.c` из вашего `CVSROOT`.

   Откомпилируйте исходные тексты, которые вы редактировали для включения правильных путей:

   ```
   % cc -o cvs cvswrap.c
   ```

   А затем установите их (для этого вы должны быть пользователем root):

   ```
   # mv /usr/bin/cvs /usr/bin/ncvs
   # mv cvs /usr/bin/cvs
   # chown root:ncvs /usr/bin/cvs /usr/bin/ncvs
   # chmod o-rx /usr/bin/ncvs
   ```

```
# chmod u-w,g+s /usr/bin/cvs
```

При этом обработчик будет установлен по умолчанию как команда `cvs`, что гарантирует всеми, использующими хранилище, получение правильных уровней доступа.

4. Теперь вы можете убрать всех из вашей группы хранилища. Всё управление доступом выполняется вашим обработчиком, и он будет устанавливать правильную группу для доступа.

3.3. Тестирование настройки

Теперь ваш обработчик должен быть установлен. Конечно, вы можете протестировать его, выполнив принудительный коммит в файл `access`:

```
% cvs commit -f -m 'Forced commit to test the new CVSROOT scripts ' ↺
access
```

И снова, если это не сработает, проверьте, правильно ли были выполнены все вышеперечисленные шаги.