

Subversion による共同管理の Web ページの構築

大浦 真@ Debian.org

<http://www.netfort.gr.jp/~ohura/>
ohura@debian.org

2005 年 7 月 24 日

目次

- ① はじめに
- ② 必要なもの
- ③ バージョン管理システムとは
 - CVS の問題点
- ④ Subversion とは
 - Subversion の特徴
 - Subversion のファイルシステム
- ⑤ 構築例
 - クライアント側の準備
 - Web サーバ側の準備
 - 「post-commit 問題」
 - DocumentRoot のパーミッション
 - post-commit の用意

はじめに

- 学校や企業などにおいて、複数の人で一つの Web ページを管理する場合がある。
- 編集の競合などが起きないようにするためにはどうすればいいのか。
→ バージョン管理システムを使う。

必要なもの

- 複数の人で一緒に管理するので、それぞれの人のコンピュータに Web ページのコピーが必要。
→ ネットワークに対応している必要あり。
- FTP などだと、
 - ちゃんと作業の分担を決めておかないと、別の人と同じ作業をしてしまう可能性がある。
 - そのコピーが最新のものでない可能性がある。
→ ネットワーク対応のバージョン管理システムを使ってみよう。

バージョン管理システムとは

- UNIX では昔からバージョン管理システムが使われてきた。
- 区切り毎の作業の内容を保存し、履歴を管理するためのシステム
- プログラミングでも、Web ページでも、ちょっとした文書でもバージョン管理システムを使える。
- 最近のバージョン管理システムは、ほとんどネットワークに対応している。

CVS の問題点

- 現在、最もよく使われているバージョン管理システム。
- 一応、ネットワーク対応しているが、昔の RCS というシステムをそのまま引き継いでいるので、システムがあまりスマートではない。
 - ファイルの移動、削除が上手く管理できない。
 - ディレクトリを上手く扱えない。
 - “post commit” 問題 (後述)

Subversion の特徴

- CVS の欠点を補うために作られた。
 - ファイルやディレクトリの移動、削除がうまく管理できる。
 - 複数のファイルやディレクトリを一度に commit できる。
 - ブランチ、タグの扱いが簡単。
 - ネイティブのクライアント/サーバシステム

Subversion のファイルシステム

- Subversion は OS のファイルシステムの上に特殊なファイルシステムを使ってリポジトリを構成している。
 - BerkeleyDB 1.1 まで標準で使われていた形式。
 - FSFS 1.1 から採用。1.2 では default に。

クライアント側の準備

- クライアント側にも Subversion をインストール。
- サーバとは、SSH を使って通信。
- クライアントからリポジトリにアクセスする時は、URL 表記を使うが、SSH を使う場合は、`svn+ssh://` という URL を使う。

```
svn ls svn+ssh://user@host/var/lib/svn/webpage/
```

Web サーバ側の準備

- ディレクトリ

Subversion のリポジトリ /var/lib/svn/webpage

Apache のドキュメントルート /var/www/webpage

- ユーザ、グループ

ユーザ A、ユーザ B、ともに、グループ webmasters に所属しているとする。

Web サーバ側の準備

- リポジトリの作成

```
svnadmin create --fs-type fsfs \  
/var/lib/svn/webpage
```

- BerkeleyDB ではなく FSFS を使う。

Web サーバ側の準備

- パーミッションとグループ所有権の変更

```
chmod -R g+w /var/lib/svn/webpage  
chgrp -R webmasters /var/lib/svn/webpage
```

- グループ webmasters に所属しているユーザがリポジトリに書き込みができるようにする。
- 実際には、「実行時のグループ ID」(後述)を設定する必要があるが、svnadmin create を実行する時に自動的に設定される。

Web サーバ側の準備

- ドキュメントルートの設定

```
svn checkout file:///var/lib/svn/webpage/trunk\  
/var/www/webpage
```

- Apache の公開領域 (DocumentRoot) にリポジトリの内容を展開する。公開領域をワーキングコピーにする。
- 後は、リポジトリに commit する毎に公開領域の内容を update する。

「post-commit 問題」

- CVS では、サーバ上の CVSR00T/loginfo というファイルを使うことにより、コミットした後に任意のコマンドを実行することができる。
- この機能を使って、コミット後に公開領域を update したり、特定のメールアドレスに log メールを送ることが可能。
- 複数のファイルを同時に commit するとコマンドが複数回、同時に実行されてしまう。
→ 公開領域のファイルの不整合が生じる場合がある。

「post-commit 問題」

- Subversion では、hooks/post-commit というファイルで同様のことができる。
- Subversion では、複数のファイルを同時に commit しても一回の commit として扱われるので、CVS のような問題は生じない。

DocumentRoot のパーミッション

- post-commit は commit したユーザの権限で実行される。

```
chmod -R g+w /var/www/webpage  
chgrp -R webmasters /var/www/webpage
```

- DocumentRoot に対して webmasters グループに所属しているユーザに書き込み権限を与える。

```
find /var/www/webpage -type d \  
-exec chmod g+s {} \;
```

- ディレクトリに「実行時のグループ ID」を設定する。
→ そのディレクトリの中の新規ファイルのグループは webmasters に設定される。

post-commit の用意

- hooks/ に post-commit.tmp1 という post-commit のテンプレートが用意されている。

```
umask 002  
cd /var/www/webpage  
svn update
```

- グループ ID を設定しただけでは、グループが webmasters になっても、グループ書き込み権限がないので、umask 002 としてから、svn update を実行。