SourceForge, Slashdot 内部構造詳解

～世界最大のオープンソース開発ネットワークのテクノロジー～

Slashdot 編

VA Linux Systems ジャパン(株) OSDN 事業部
Debian Project Official Developer
安井 卓
<tach@valinux.co.jp>, <tach@debian.org>

December 6, 2001

Agenda

- OSDN について
  - 3 つの "D"
- Slashdot について
  - サイトの説明
  - Slashdot Japan
- Slashcode について
  - 管理者用インターフェース
  - モデレーションシステム
  - コード内部
  - プラグイン構造・作成
- Slashdot Japan の運用から
  - Web クラスタへの移行
  - バックアップ

OSDN とは

O S D N
open source development network

- オープンソースコミュニティのサイト群
- 世界最大規模のネットワーク
- http://osdn.com/
- 2000 年 8 月オープン

- 日本では，OSDN Japan
  - http://osdn.jp/
  - 2001 年 8 月オープン

OSDN の 3 つの "D"

- Discussion（議論）
- Development（開発）
- Distribution（配布）

OSDN の 3 つの "D"

- Discussion（議論）
  - Slashdot (http://slashdot.org/)
  - Linux.Com (http://linux.com/)
  - NewsForge (http://newsforge.com/)
  - Geocrawler (http://geocrawler.com/)
- Development（開発）
  - SourceForge (http://sourceforge.net/)
  - Slashcode (http://slashcode.com/)
- Distribution（配布）
  - Freshmeat (http://freshmeat.net/)
  - Themes.Org (http://themes.org/)
  - ThinkGeek (http://thinkgeek.com/)

Slashdot とは

Slashdot
News for Nerds. Stuff that matters.

- 1997 年に開設されたオタクのための情報サイト
  - Rob "CmdrTaco" Malda 氏と Jeff "Hemos" Bates 氏
- 5000 万 PV / 月，50 万以上の登録ユーザ
- 一般のニュースサイトには載らない変な話題やうわさが多い
- ユーザが自由にコメントをつけることができる
  - コメントを評価するモデレーション機能

オープンソース ジャーナリズム

Slashdot Japan とは

Slashdot
News for Nerds. Stuff that matters.

- Slashdot の日本語版
  - VA Linux Systems ジャパン(株)が 2001 年 5 月に開設
  - コンテンツは VA とは別のスラッシュチームが担当
- 10 月現在で 500 万 PV / 月，5000 以上の登録ユーザ
- 本家 Slashdot の雰囲気をそのまま輸入
  - オープンソース ジャーナリズムの推進

Slashdot Japan はこんなふう

Slashdot Japan のマシン構成

- Web サーバ 5 台
  - PentiumIII 933MHz Dual, 2GB Memory
- データベースサーバ 1 台
  - PentiumIII 850MHz Dual, 1GB Memory
- ファイルサーバ(NFS)・バックエンド 1 台
  - PentiumIII 933MHz Dual, 2GB Memory
- その他，DNS やメールサーバ，ロードバランサ

Web は途中でクラスタリング構成に移行

Slashdot Japan のソフトウエア構成

- Web サーバ
  - Debian GNU/Linux
    - apache (1.3.20)
    - mod_perl (1.25)
    - perl (5.6.1)
- データベースサーバ
  - Debian GNU/Linux
    - MySQL (3.23.xx)
- ファイルサーバ
  - Debian GNU/Linux

そして，Slashcode...

Slashcode とは

Slashcode ... Slashdot Like Storytelling Homepage

- http://slashcode.com/
- Slashdot のようなコミュニティポータルを作成するソフト
- オープンソース(GPL)
- Perl で書かれている
- Apache/mod_perl 上で動作
- もともと slashdot のために書かれた
- 多くのコミュニケーションポータルで採用されている
- テンプレートによるサイトデザイン･カスタマイズ
  - すべて Web 上から操作可能 … Admin インターフェース
- プラグインによる拡張が可能
  - Admin インターフェースや日記システムが用意されている

Slashcode の特徴

- mod_perl で動作
  - apache 上で高速に動作する
- インストールすれば，あとはすべて Web 上で操作可能
  - 各ストーリー･コメントの管理
  - テンプレートによるサイトデザイン
  - 変数によるサイトの動作の調整
- セクションとトピックでストーリーを分類
  - 各ストーリーに対してコメントがつけられる
  - 各コメントの評価をするモデレーションシステム(後述)
- ユーザごとに表示形式やコメントの書き方のカスタマイズ可能
  - スラッシュボックス(後述)の配置

Slashcode と日本語

基本的にそのままで日本語も通るが…

- 一部，Jcode.pm を利用して確実に日本語が通るようにした
  - メールを送信する部分などは iso-2022-jp に変換
  - そのほかは EUC-JP で統一
- メッセージは英語しかないので，日本語に翻訳
  - ほとんどテンプレート内部だが一部コードに埋め込み
  - 多言語化のきっかけは作ってあるがほとんど取り組みはない
- フィルタも英語環境向け
  - 日本語はとりあえずそのまま通す

Slashalikes

Slashcode をもとにしたシステムはたくさんある

- PHP
  - PHPSlash
  - PostNuke
  - PHPNuke
  - thatphpware
  - sips
  - TWIG
  - phpWebLog
- Perl
  - scoop
  - localecho
- Zope (Python)
  - Squishdot

Slashcode が元祖

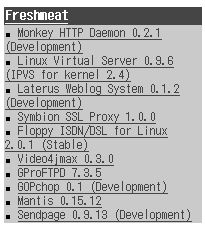
Slashcode の構造・動作

- Perl の Slash ライブラリ
- フロントエンドのスクリプト群
  - index.pl article.pl comments.pl など
- バックエンドデーモン
  - slashd … 常時動作しているデーモン
  - portald … スラッシュボックス生成用
  - moderated … モデレータ権限関連
  - dailyStuff … 毎日動く，統計やデータ整理など

スラッシュボックスとは

各種情報を表示するためのブロック群

- ユーザが表示順やなにを表示するかカスタマイズ可能
- ログインボックス
- 投票機能のボックス
- クイックリンク
- 各サイトのヘッドラインをリンク付きで表示
  - 各サイトから rdf ファイル(XML)をダウンロードして整形



rdf って?

Resource Description Framework
http://www.w3.org/RDF/

ポータルサイトのヘッドラインを配布するための標準フォーマット

- XML で書かれている
  - XML パーサがあれば OK!
- 海外サイトの多くで配布している
  - Slashcode サイトなら自動生成
  - 日本でもごく一部は配布中
  - Slashdot Japan でも配布(http://slashdot.jp/slashdot.rdf)
- サイトを作成したら，ぜひ配布しましょう

rdf の例

http://slashdot.jp/slashdot.rdf

```
<?xml version="1.0" encoding="utf-8"?>

:

<channel rdf:about="http://slashdot.jp/">
<title>スラッシュドット ジャパン: スラッシュドット ジャパン: アレゲなニュースと雑談サイト</title>
<link>http://slashdot.jp/</link>
<description>アレゲなニュースと雑談サイト</description>
:
</channel>

<item rdf:about="http://slashdot.jp/article.pl?sid=01/10/29/0655217">
<title>一本足走行ロボット脚</title>
<link>http://slashdot.jp/article.pl?sid=01/10/29/0655217</link>
</item>

<item rdf:about="http://slashdot.jp/article.pl?sid=01/10/29/0651214">
<title>パロアルトの中学がノートパソコンの購入を強制</title>
<link>http://slashdot.jp/article.pl?sid=01/10/29/0651214</link>
</item>

:
```

rdf のつくりかた

- XML:RSS モジュールを使う

```
use XML::RSS;
:
my $rss = XML::RSS->new(version => '1.0', encoding => 'utf-8');
$rss->channel(
        title => 'title', # サイトのタイトル
        link => 'url',    # サイトの URL
        language => 'ja', # 言語
        description => 'description'); # サイトの概要

$rss->add_item(
        title => 'title', # タイトル
        link => 'url');   # URL

$rss->save('file_name.rdf');
```

rdf の読みとりかた

- XML:RSS モジュールを使う

```
use XML::RSS;
:
my $rss = XML::RSS->new();
$rss->parse($string);

foreach my $item ( @{$rss->{'items'}} ) {
    print "Title: $item->{'title'}\n";
    print "Link: $item->{'link'}\n\n";
}

----------------------------------

Title: 一本足走行ロボット脚
Link: http://slashdot.jp/article.pl?sid=01/10/29/0655217

Title: パロアルトの中学がノートパソコンの購入を強制
Link: http://slashdot.jp/article.pl?sid=01/10/29/0651214
```

Slashdot の裏側

Admin インターフェース

- Web 上でサイトを管理するためのインターフェース
  - ストーリーの管理
  - セクションの管理
  - トピックの管理
  - ユーザの管理
  - コメントフィルタ
  - テンプレートエディタ
  - 変数の管理 など…
- ユーザに権限レベルを設定して，特定の権限だけ与えられる
  - (小)ストーリーの投稿・管理ができる
  - ： セクション・トピックの管理ができる
  - (大)ユーザ・サイトデザイン・変数の管理ができる

テンプレート機能

- デザインはすべてテンプレートで行う
  - 各部分ごとに分かれてテンプレートがある
  - http://www.template-toolkit.org/
    - Perl モジュール
    - 多機能のテンプレートツールキット
    - テンプレート内部で簡単な条件分岐などが可能
- テンプレートの内容もすべてデータベースに入れる
  - データの一括管理
- Web 上で編集可能
  - ホストに shell login する必要なし

テンプレートの例

テンプレート fancybox, misc, default
スラッシュボックスを表示する部分

```
[% width = width || 200 %]

[% IF center %]<center>[% END %]
<table width="[% width %]" border="0" cellpadding="1" cellspacing="0">
<tr valign="top" bgcolor="[% user.bg.3 %]"><td>
      <font size="+1" color="[% user.fg.0 %]"><strong>[% title %]</strong></font>
</td></tr>
<tr><td bgcolor="[% user.bg.4 %]">
      <font size="-1">
      [% contents %]
      </font>
<td></tr>
</table>
[% IF center %]</center>[% END %]
<br>
```

- [% %] で囲まれている部分がテンプレートのコードや変数

掲示板サイトの悩みの種

荒らし投稿対策をどのように行うか

- 管理者による強制削除
  - サイトの規模が小さいうちは有効
  - 規模が大きくなると，特定の管理者に負担がかかる
  - 管理者の意向に沿わないものも消されてしまう
- 投稿時に検閲
  - 強制削除と同様の問題が発生

オープンソース ジャーナリズムに向かないシステム

荒らし対策その 1 … コメントフィルタ

- 正規表現を利用したフィルタリング
  - 特定の文字列を使用した投稿を排除できる
    - First Post...
    - アスキーアート
- 連続投稿のチェック
  - 同じような投稿を連続して行えないようにチェックする
  - 前の投稿から 10 分以上経たないと投稿できない
  - 前のコメントと同じ文字が大量に含まれていないか

荒らし対策その 2 … モデレーション

特定の管理者に依存しない画期的なシステム
荒らし対策というよりも，数多くのコメントの中から，良いコメント(宝石)を見つけだすためのもの

- ユーザがお互いにコメントの評価をしあうシステム
  - コメントに「スコア」をつけて管理をする
  - 決められたポイント内(5 ポイント)でコメントのスコアを上げ下げする
  - 「削除」はしない
    - 読むか読まないかはユーザが判断する
- モデレータの権限がときどきユーザに与えられる

モデレーションの歴史

Slashdot.org で生まれたシステム

- はじめは管理者が削除していた
  - 投稿が増えてきて対処が困難に
- コメントにスコアをつけ，特定の人(25 人くらい)にスコアを上げ下げするだけの権限を与えた
  - やはり，規模が大きくなると困難に
- モデレータを 400 人に増やした
  - 管理者が 400 人を管理できない
  - モデレータの質が落ちた
  - モデレータのパワーを制御するための機構が必要
- モデレータを制限
  - モデレーションできる数を制限
- そして，現在のモデレーションシステムが生まれた

誰がモデレートするのか

- 全ユーザがモデレーションの権利を持つことができる
- ただし，以下の条件に合致すれば
  - ログインしているユーザ(匿名ではない)
  - カルマ(後述)が一定以上あるユーザ
  - 頻繁にアクセスするユーザ
    - トップページだけ Reload するユーザはダメ
- モデレーション権限は 3 日間だけ
  - 特定のユーザが権限を持ち続けるのを防ぐ

カルマについて

カルマとは … そのユーザのスコア(徳?)

- タレコミが採用されたときや，自分のコメントのスコアが上がったときに溜まる
- 逆に，自分のコメントのスコアが下がったときに下がる
  - 変な行動をしてると，あっというまにカルマが下がる
- カルマが高すぎてもあまり意味はなく，一定以上なら同じ
  - カルマを高くするためによい行動をする? → 発想が逆
  - 一定以上のレベルを保つためにカルマを導入しているだけ

どのようにモデレートするのか

- モデレート権限がまわってくると，各コメントごとにプルダウンメニューがつく
  - 普通(+-0)
  - オフトピック･フレームの元･荒らし･よけいなもの(-1)
  - すばらしい洞察･興味深い･参考になる･おもしろおかしい(+1)
  - (カスタマイズ可能)
- モデレートは一回 5 ポイントまで，3 日間のみ
  - (カスタマイズ可能)
- スコアの初期値
  - 登録ユーザのコメント = 1
  - 匿名ユーザのコメント = 0

モデレーションポイントの与え方

- ユーザ数･コメント数からプールポイントを生成
  - ユーザ数やコメント数が多いほど多くなる
- カルマが一定以上のユーザに対して，ランダムにプールポイントを割り振る
  - カルマの大小は関係ない
- プールポイントが一定以上になったユーザに対して権限を与える
  - 5 ポイント，3 日間

Slashdot Japan のモデレーション

- 本家 Slashdot のような歴史をたどらず いきなりモデレーションシステムを導入
- ただし，初めはユーザもコメントも少ないので一般には権限を与えず，管理者がモデレーションしていた
- ユーザがシステムに慣れたころにモデレーションを解放
  - 最初はとまどいがあったようだがおおむね正常に機能している
  - モデレーションに関するストーリーで議論が白熱
  - モデレータのモデレーションがおかしいとの反応も
    - → モデレーションにはモデレーションで対抗
    - メタモデレーション（モデレーションのモデレーション）を行う

モデレーション自体は成功したといえる

Slashcode インストール (1)

インストール前に Slashcode が要求するもの

- DB エンジン(MySQL など)へのアクセス権
  - データベースはあらかじめ作っておく
    - mysqladmin create slash
  - そのデータベースに対して全権限を与える
    - echo 'GRANT ALL PRIVILEGES ON slash.* TO foo@localhost IDENTIFIED BY "bar";' | mysql -u root
- apache / mod_perl
- 以下の Perl モジュール
  - DBI, Msql-Mysql-modules, Data-ShowTable
  - DateManip, TimeDate
  - ApacheDBI, Apache-DBILogConfig, Apache-DBILogger
  - libnet, URI, libwww-perl
  - HTML-Parser, XML-Parser, XML-RSS
  - MIME-Base64, Digest-MD5, File-Spec
  - Mail-Sendmail, Compress-Zlib, Image-Size
  - Storable, Template, DBIx-Password

Slashcode の構造 (2)

インストール その 1

- http://slashcode.com/ からダウンロードして，展開後 make; make install

- 最初に以下のものがインストールされる
  - Perl の Slash モジュール
  - プラグイン
    - Admin インターフェース
    - 日記システム
    - 検索システム
    - など…
  - テーマ
  - デーモン･ツール群

Slashcode の構造 (2)

インストール その 2

- サイトの雛形を作成する
  - どのテーマを採用するか
  - どのプラグインを使うか
  - 管理者ユーザ名・パスワード
  - bin/install-slashsite -u DBIx-Password-pseudo-user
- apache の設定
- slashd のスタート
  - sbin/slashd

これで雛形のサイトが立ち上がる

Slashcode の内部構造

- モジュール化されており，利用しやすいようになっている
  - use Slash; 総合モジュール
  - use Slash::Display; 表示のためのモジュール
  - use Slash::Utility; その他機能のモジュール
- 基本として 4 つのオブジェクトを使う
  - $slashdb = getCurrentDB(); データベース関連
    - データベースへのアクセス一般
  - $user = getCurrentUser(); ユーザ関連
    - ユーザ情報の hashref
  - $form = getCurrentForm(); フォーム関連
    - Web フォームの hashref
  - $constants = getCurrentStatic(); 定数関連
    - 定数の hashref

データはほとんどデータベースに

- クラスタ構成での運用を考えてある
  - データがファイルだと，各マシンでの同期が大変
- データを一カ所に集めて管理する
  - テンプレートなどもすべて DB に入っている
  - dump して持っていけば，サイトのコピーができる

データベースへのアクセス

DBIx::Password を利用

- 疑似ユーザ名を準備して，ホスト･ユーザ名･パスワードを埋め込む
  - 便利だが，セキュリティ的にはダメ(誰でもアクセスできるようになってしまう)
  - インストールするときは，信頼できるユーザしかいないところに限る
    - あるいは，other の read permission を消しておく

$slashdb オブジェクトを通してアクセスする

- $slashdb = getCurrentDB(); データベース関連
  - $slashdb->sqlSelectAll()
  - $slashdb->sqlSelectArrayRef()
  - $slashdb->sqlSelectHashRef()
  - $slashdb->sqlSelect()
  - $slashdb->sqlInsert()
  - $slashdb->sqlDo()

プラグインの書き方 (1)

といっても，プラグインだけでなくすべてのページはこのように書かれている…

- まずモジュールをロード

```
use Slash;
use Slash::Display;
use Slash::Utility;
```

- オブジェクトをつくる (main 関数)

```
sub main {

    my $slashdb = getCurrentDB();
    my $constants = getCurrentStatic();
    my $user = getCurrentUser();
    my $form = getCurrentForm();
```

プラグインの書き方 (2)

ほんの一例

```
    my %ops = (
        list          => \&listArticle,
        edit          => \&editArticle,
        display       => \&displayArticle,
        default       => \&displayDefault,
    );

    my $op = $form->{'op'};
    $op = 'default' unless $ops{$op};

    header("$constants->{sitename} Sample Plugin");
    titlebar("100%","Sample Plugin");

    print createMenu('journal');
    $ops{$op}->($form, $slashdb, $constants);

    footer();
}
( 各サブルーチンを書く )
main();
1;
```

大規模運用のために...

- 高速化
  - CGI で運用するのは無謀
  - mod_perl などを利用する
  - DB へのアクセスを減らすキャッシュ
- クラスタ構成を想定したシステム設計
  - コードとデータの分離
  - データの一括管理

高速化

CGI は実行ごとにプロセスを生成するので遅い

- プロセス生成は結構重い処理
- 頻繁にアクセスのあるページを CGI で書くのは自殺行為

プロセスを生成しない → デーモン化?

- httpd がデーモンで動いている
- httpd (apache) に組み込んじゃえ!

↓

mod_perl mod_ruby

php

mod_perl と mod_ruby

http://perl.apache.org/ mod_perl
http://www.modruby.net/ mod_ruby

- Apache のモジュール
- Apache のプロセスとして動作するので高速
  - 10 倍以上のスピード?
- CGI と同様のインターフェースを備えているため，CGI からの移行が楽

- 逆にそれが命取りにもなる
  - プログラムが変だと Apache まで影響を受ける

データのキャッシュ (1)

Perl DBI のキャッシュ

```
$sth = $dbh->prepare_cached($sql ...)
$sth->execute;
```

- 同じステートメントで二度以上呼び出されたとき，データベースに直接アクセスせずに，答えを返す

- 高速化のためには役立つが，クリティカルな用途の場合は問題が発生する場合が多いので要注意

データのキャッシュ (2)

- mod_perl のスクリプトは基本的に再ロードしない
- 変数も初期化しない
- それを逆手にとって，グローバル変数内にデータを保持
  - データベースへのアクセスが減る
    - 高速に表示ができる
    - データベースサーバへのトラフィックが減る
  - fork したプロセスが死ぬまで有効
  - ときどき内容が消える?
    - 内容をチェックして抜けていたら取りに行く

なにをキャッシングしておくか

あまり内容がかわらないもの
頻繁にアクセスするもの
あまりアクセスがなかったり，内容が変わったりするものは NG

- Slashcode では，ストーリーやテンプレート

キャッシュの有無でサーバの負荷がだいぶ違う(5 倍以上?)

キャッシュを有効に活用するために

- サーバにたくさんメモリを搭載する
  - 各プロセスごとにキャッシュを保持するので，プロセスのサイズが大きくなる
  - 高速に動作するので同時に複数のプロセスが稼働する
  - そのぶんだけメモリが必要
- Apache の設定で最適化
  - MaxClients … 最大同時接続クライアント数
    - 上げすぎて失敗 デフォルト値(150)のまま
  - MaxRequestPerChild … fork したプロセスがこの数だけ処理して死ぬ
    - キャッシュを生かすために上げる(500-1000)
    - メモリリークなどの心配があるのであまり上げすぎない
  - KeepAliveTimeout … キープアライブのタイムアウト時間
    - 短く(3 秒)して，次の要求をすぐに受けられるようにする

Slashdot Japan の運用から…

- Slashdot Japan は 2001 年 5 月にオープン
  - このときはデータベースサーバと Web サーバの 2 台体制
  - オープン時から多少重めだったが，徐々にアクセスが増えていった
  - ピーク時はロードアベレージが 30 を越えた
- 緊急にロードバランサを準備
  - 8 月にクラスタ体制に移行
  - 負荷が負荷を呼ぶ状態を脱して，軽快動作
  - サーバ全体を private network に入れたため，安全

なぜ MySQL を利用するか

それは，高速化のため

- 確かに，Slashcode + MySQL の方が実績がある というのもある
- MySQL に適した場所は?
  - 単純な問い合わせが多い
  - とにかく高速で動作することが必要
  - データの更新が頻繁ではない
  - 多くの機能を必要としない

Web は上記の条件をみたしており
MySQL に最適な環境

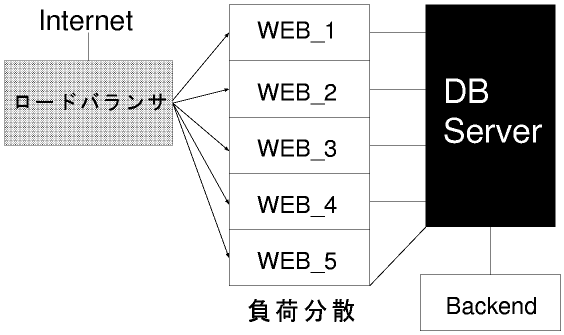
Web クラスタへの移行

サーバが dualhome(eth ポートを複数持っている)だと楽

- もう片方の eth を生かして，そちらを内側に入れる
- apache でそちらの IP でも listen するように設定
- ロードバランサを設定してサーバの外向き eth を殺す
  - この時点でロードバランサ経由のアクセスになる
  - DB サーバも同様に内側を向ける
- その後，追加の Web サーバをセットアップ
  - セットアップ終了したらロードバランサの設定

- 実際のダウンタイムは数分〜数十分になる
  - はずだったが…

ロードバランスの状態



- 5 台の Web サーバで負荷分散
  - 混雑時は 10 倍以上の速度差

データのバックアップ体制

- データセンターにあるため，自分でテープバックアップするのは困難
  - ハードディスクにバックアップを取る
- データはすべてデータベースに入っているので，とりあえず mysqldump で一発
- それ以外(コードや設定など)は rsync や vbackup でバックアップ

mysqldump でバックアップ

- mysqldump は MySQL のデータを SQL 文で書き出す
- その SQL を mysql コマンドに通せば元に戻る

- 使い方:
  - 実行するユーザの ~/.my.cnf に以下のように書いておく

```
[mysqldump]
user = username
password = pass
host = host_of_mysqld
```

  - mysqldump を実行

```
# mysqldump database_name > sqlfile.sql
```

  - cron で定期的に実行する

vbackup で差分バックアップ

ファイルシステムをそのまま HDD → HDD でコピーするバックアップツール

- http://www.rommelwood.de/~martin/vbackup/

- ハードリンクを利用した差分バックアップツール
- 普通のファイルシステムなので，ls cp などのツールがそのまま利用できる
- ハードリンクを利用するため，容量の節約になる
  - ハードリンクが利用できないファイルシステムでは利用不可

- つかいかた
  - vbackup [もとディレクトリ] [バックアップ先]

vbackup で差分バックアップ (2)

例:

```
% vbackup /var /backup/var
% ls /backup/var
2001-10-27@03:17:44  2001-10-28@03:16:28 2001-10-29@03:17:09
```

各日付･時間のディレクトリを作り，その中にファイルが格納される

```
% vbackup arege:/var /backup/var
% ls /backup/var
2001-10-27@03:17:44  2001-10-28@03:16:28 2001-10-29@03:17:09
```

rsh/ssh を利用して，リモートのバックアップもとれる

E N D

つぎは SOURCEFORGE デスヨ...