

1歩進んだサーバープログラミング

柔軟なファイル処理を
実現する
File::モジュール

CPAN(Comprehensive Perl Archive Network)は、Perlプログラムとドキュメントの膨大なコレクションである。プログラミングに便利な機能を寄せ集めたもので、さしずめ知恵の宝庫といえよう。本連載では、CPANで4,000個以上提供されているPerlモジュールから便利なものを厳選し、その機能を自分のPerlプログラムから使い、効率よくプログラミングする方法を解説する。

CPAN

Powered by Sun

<http://www.cpan.org/>

実用的なCGIに必要なファイル処理

第2回では、CGI.pmを使ってフォーム入力画面の生成を行い、CGI.pmとJcode.pmモジュールを使ってフォーム入力された結果表示をさせるという、フォーム処理の基本を実際に行った。しかし、CGIプログラムを作成するときには、フォームの作成やパラメータの処理だけでなく、各種のファイル操作なども必要になってくる。そこで、今回はCGI.pmとファイル操作などのモジュールを組み合わせて簡易掲示板を作ることで、実用的なCGIプログラミングについて解説していく。

サンプルプログラム: 簡易掲示板

それでは早速、これまで学んできたCGI.pmやJcode.pmを使った簡易掲示板のサンプルコードを見てみよう(コード1)。

「えっ、これだけ?」と思う読者もいるかもしれない。しかし、実際にこれだけのコード量で掲示板ができる。そして、これまで説明してきたとおり、モジュールをうまく使うことで、HTMLタグが入り乱れて非常に見づらいコードになることが避けられている。さらに今回の目玉として、ファイルのパスの指定が異なるウィンドウズ/Unix/Mac OSといった違いも吸収できるようになっている。

実際にサンプルコードの動作例を見よう。初めてこのindex.cgiを実行すると、新規メッセージの入力フォームと「現在のところメッセージはありません」というメッセージが表示される。ここで、フ

ォームにメッセージを入力して「書き込み」ボタンをクリックすると、新規メッセージの入力フォームとその下に書き込まれたメッセージが表示される。メッセージは新しい順に表示される。メッセージを削除したり、スレッドを立ててツリー表示をさせたりするといったことはできないが、サンプルプログラムを拡張すればもちろん対応できる。

サンプルプログラムの構造

ここで、サンプルプログラムの動作の概略を説明しておく。このサンプルプログラムは、大きく分けると、新規メッセージ入力フォームの表示部分、保存されているメッセージ一覧の取得部分、新規メッセージが書き込まれたときの処理部分、メッセージ表示処理部分となる。書き込まれたメッセージはメッセージ番号をファイル名とするファイルとして、「\$msgdir」で指定されるディレクトリ内に保存される。4つの処理部分の処理内容を具体的に説明しよう。

・新規メッセージ入力フォームの表示部分

前回説明したCGI.pmモジュールによるフォーム表示そのものである。

・メッセージ一覧の取得部分

メッセージを保存する\$msgdirディレクトリが存在しない場合には新規に作成する。\$msgdirディレクトリの中にあるファイルの一覧を取得し、それを番号の大きい順にリストとして読み込む。

・新規メッセージの入力処理部分

フォーム変数Messageの内容がある場合には、メッセージ書き込みの処理を行う。書き込み処理としては、各変数の文字コードをJcode.pmモジュールを使って統一(ここではEUCコード)HTML形式で番号の一番大きいファイル名にメッセージを保存する。

・メッセージ表示処理部分

メッセージを表示する場合は、番号の大きい順に\$msgdirの各メッセージを読み出して表示する。メッセージが書き込まれるときにあらかじめHTMLのタグを付けてファイルに保存してあるので、表示するときにはファイルから読み込むだけでいい。

簡易掲示板のサンプルプログラム

お名前

題名

E-Mail

メッセージ
最新のメッセージから順番に表示されます。

[No.1]はじめて書き込みます! 投稿者:情報技研 投稿日:2003/03/05 22:53:37
メッセージをここに書きます。HTMLタグをそのまま書き込みますので、改行したいときは自分でタグを入れる必要があります。
これでここに改行が入ることになります。

図1 サンプルコードの実行結果

コード1 簡易掲示板のサンプルコード (index.cgi)

```
#!/usr/local/bin/perl
use CGI qw(:standard);
use CGI::Pretty;
use Jcode;
use File::Spec;
use File::Path;
use Date::Format;

## 変数設定
## /home/www/cgi-bin/messagesを使う
$msgsdir =
    File::Spec->rootdir.File::Spec->catdir(
        'home','www','cgi-bin','msgs');

## 入力フォームの表示
## このサンプルプログラムではシンプルな投稿のみを扱う
print
    header(-charset=>'x-euc-jp'),
    start_html,
    h2('簡易掲示板のサンプルプログラム'),
    start_form(-action=>'index.cgi'),
    table(
        {border=>0},
        Tr([
            td(['お名前:', textfield('Name', '', 30)]),
            td(['題 名:', textfield('Title', '', 30)]),
            td(['E-Mail:', textfield('EMail', '', 30)]),
        ]),
    ),
    br('メッセージ:'),
    textarea('Message', '', 5, 60),
    p,
    submit('書き込み'),
    reset('リセット'),
    end_form,
    hr;

## メッセージ一覧の取得
if (!( -e $msgsdir )) { mkpath($msgsdir, 0, 0755) }
opendir (DIR, $msgsdir);
@filelist =
    sort {$a<=>$b} grep (/^d+/, readdir(DIR));
$msgs = scalar(@filelist);

## 書き込み処理
## (フォーム変数 Message が存在するときに処理)
$msg = jcode(param('Message'))->euc;
if ($msg){
    $msgs++;
    $name = jcode(param('Name'))->euc;
```

```
$title = jcode(param('Title'))->euc;
$email = jcode(param('EMail'))->euc;

$filename =
    File::Spec->catfile(($msgsdir), $msgs);
open (FILE, ">$filename");

print FILE
    table(
        {border=>1, width=>'90%',
         bgcolor=>"#F0F0F0"},
        Tr([
            td([
                table(
                    {border=>0},
                    Tr([
                        td([
                            "[No.$msgs]",
                            $title,
                            '投稿者:',
                            .a({-href=>"mailto:$email"},
                                "$name"),
                            '投稿日:',
                            .time2str('%Y/%m/%d %T',time)
                        ]) # /td
                    ]) # /tr
                ], # /table
            ]), # /td
            td(["$msg"]),
        ]), # /tr
    ), # /table
    "\n";
}

## メッセージ表示処理
## メッセージがある場合には単純にファイルを読み込んで表示する
if (!$msgs){
    print h3('現在のところメッセージはありません。');
}else{
    for ($i = $msgs; $i > 0; $i--){
        $filename =
            File::Spec->catfile(($msgsdir), $i);
        open (FILE, $filename);
        print p,<FILE>;
    }
}

print end_html;
```

以上の動作を行うことで簡易掲示板を実現している。HTMLの出力とフォーム変数の処理を楽にするCGI.pm、出力するHTMLを見やすくするCGI::Pretty.pm、文字コードの変換を行うJcode.pmについてはこれまで説明したが、今回新たに、OSに依

存しないパスの指定を可能にするFile::Spec.pm、ファイル操作を行うFile::Path.pm、日付文字列の取得を簡単に行えるDate::Format.pmというモジュールが登場している。それでは、まずそれぞれのモジュールについて説明していこう。

File::Spec.pm モジュール

OS環境に依存しないファイル名操作を行う

【カテゴリー】 Perlの標準モジュール
【バージョン】 0.82
【作者】 Barrie Slaymaker氏
【URL】 <http://search.cpan.org/author/RBS/File-Spec-0.82/Spec.pm>

File::Spec.pm のココがスゴイ!

File::Spec.pmは、OSなどの環境に依存せずにパス名の処理を行うモジュールである。たとえば、Unixではディレクトリー指定は「/」で行うが、ウィンドウズでは「¥」となる。こういったOSによるパスの指定方法の違いを吸収するのがFile::Specモジュールだ。異なるOS間での移植性を高めるときに重宝する。

たとえば、サンプルプログラムでは次のようにしている。

```
File::Spec->catdir('home','www','cgi-bin','msgs')  
これは、Unixでは「home/www/cgi-bin/msgs」に、ウィンドウズ  
では「home¥www¥cgi-bin¥msgs」になる。
```

File::Spec.pm のメソッド

canonpath メソッド

パス名を整形する。たとえばUnix上の「/root///Mail」を「/root/Mail」に整形できる。

【構文】

```
File::Spec->canonpath(path)
```

catdir メソッド

2つ以上のディレクトリー名をディレクトリー区切り文字列で連結してパスを生成する。

【構文】

```
File::Spec->catdir(dir[,dir...])
```

catfile メソッド

1つ以上のディレクトリー名と1つのファイル名をディレクトリー区切り文字列で連結してファイルのパスを生成する。

【構文】

```
File::Spec->catfile(dir[,dir...], file)
```

rootdir メソッド

ルートディレクトリーを表す文字列を返す。

【構文】

```
File::Spec->rootdir
```

curdir メソッド

カレントディレクトリーを表す文字列を返す。

【構文】

```
File::Spec->curdir
```

updir メソッド

親ディレクトリーを表す文字列を返す。

【構文】

```
File::Spec->updir
```

abs2rel メソッド

あるパス(ベースパス)から見た、別のパス(絶対パス)への相対パス名を返す。

【構文】

```
File::Spec->abs2rel(ベースパス,絶対パス)
```

rel2abs メソッド

あるパス(ベースパス)から相対パス名で示される別のパスを絶対パス名にして返す。

【構文】

```
File::Spec->rel2abs(相対パス,ベースパス)
```

no_upwards メソッド

ファイル名のリストから、親ディレクトリーとカレントディレクトリーを表すファイル名を取り除く。Unixの場合「..」と「.」が取り除かれる。

【構文】

```
File::Spec->no_upwards(files)
```

file_name_is_absolute メソッド

パス名が絶対パスの場合には真、相対パスの場合には偽を返す。

【構文】

```
File::Spec->file_name_is_absolute(path)
```

devnull メソッド

NULLデバイスを表す文字列を表示する。

【構文】

```
File::Spec->devnull
```

tmpdir メソッド

一時ディレクトリー名を返す。

【構文】

```
File::Spec->tmpdir
```

File::Path.pm モジュール

ディレクトリーの作成と削除を行う

【カテゴリー】Perlの標準モジュール
【バージョン】1.05
【作者】Tim Bunce氏、Charles Bailey氏
【URL】<http://search.cpan.org/author/JHI/perl-5.8.0/lib/File/Path.pm>

File::Path.pm のメソッド

mkpath メソッド

ディレクトリーを作成する。

【構文】

```
mkpath(path,prt,perm)
```

【引数】

- ・path
作成するパス名
- ・prt
ブール値を指定する。真の場合、ディレクトリーを作成するたびにその名前を出力する。デフォルトは偽。
- ・perm
ディレクトリー作成時に使用するパーミッション。デフォルトは0777。

rmtree メソッド

ディレクトリーとその中のファイル/ディレクトリーを再帰的にすべて削除する。

【構文】

```
rmtree(root,prt,skip)
```

【引数】

- ・root
削除するディレクトリー。その中のファイルとディレクトリーも対象となる。
- ・prt
ブール値を指定する。真の場合、削除する個々のファイル名が表示される。デフォルトは偽。
- ・skip
ブール値を指定する。真の場合、ユーザーが削除権限のあるディレクトリーのみを削除する。デフォルトは偽。

Date::Format.pm モジュール

time や localtime で得られるデータを任意の文字列形式にフォーマットする

【カテゴリー】データタイプユーティリティー
【バージョン】1.14
【作者】Graham Barr氏
【URL】<http://search.cpan.org/author/GBARR/TimeDate-1.14/lib/Date/Format.pm>

Date::Format.pm モジュールは、time や localtime などの関数から得られた時間情報を任意の書式の文字列に変換できる便利なモジュールである。Perlの標準モジュールでないため、自分でインストールする必要がある。

Date::Format.pm のメソッド

ctime メソッド

time 関数の結果を"%a %b %e %T %Y%n"にフォーマットする。このフォーマットの表示例は「Thu Mar 6 04:00:00 2003」。

【構文】

```
ctime(time,timezone)
```

asctime メソッド

localtime(time)関数の結果を"%a %b %e %T %Y%n"にフォーマットする。

【構文】

```
asctime(localtime(time),timezone)
```

time2str メソッド

テンプレートに合わせてtime関数の結果をフォーマットする。

【構文】

```
time2str(template,time[,timezone])
```

【引数】

- ・template
フォーマットを指定する。フォーマットの詳細についてはこのモジュールの解説ページを参照。
- ・time
時間を取得するtime関数を指定する。
- ・timezone
タイムゾーンの指定。デフォルトはシステムのタイムゾーン。

strftime メソッド

テンプレートに合わせてlocaltime(time)関数の結果をフォーマットする。書式はtime2str とほぼ同じ。

【構文】

```
strftime(template,localtime(time),timezone)
```

File::Copy.pm モジュール

Unixのcpコマンドやmvコマンドと同様の処理を行う

【カテゴリ】Perlの標準モジュール
【バージョン】2.05
【作者】Aaron Sherman氏
【URL】<http://search.cpan.org/author/JHI/perl-5.8.0/lib/File/Copy.pm>

File::Copy.pmモジュールはサンプルプログラムでは利用していないが、使い慣れた便利なファイル操作の機能を提供するものなのでここで紹介しておこう。

File::Copy モジュールのメソッド

copy メソッド

source ファイルをdestにコピーする。

【構文】

```
copy(source, dest[, buffersize])
```

【引数】

- ・source
コピー元のファイルを指定する。

・dest

コピー先のファイルを指定する。

・buffersize

コピーに使用するバッファサイズを指定する。

move メソッド

source ファイルをdestに移動する。

【構文】

```
move(source, dest)
```

【引数】

- ・source
移動元のファイルを指定する。
- ・dest
移動先のファイルを指定する。

その他の代表的なファイル関連モジュール

これまでに紹介したものの以外にも、ファイル操作関連のモジュールとして便利なものがいくつかあるので紹介しておこう。どれもPerlの標準モジュールなので、ファイル関連の操作をする場合には必要に応じてこれらのモジュールを検討するべきだろう。

File::Basename モジュール

ファイル名を解析する。

【メソッド】

```
·fileparse($fullname, @suffixlist)
```

File::Compare モジュール

2つのファイルの内容を比較する。

【メソッド】

```
·compare_text($file1, $file2)
```

File::Find モジュール

指定されたディレクトリーからファイルを検索する。

【メソッド】

```
·find(\&wanted, @directories);  
  find(\%options, @directories);  
·finddepth(\&wanted, @directories);  
  finddepth(\%options, @directories);
```

File::Temp モジュール

一時ファイルや一時ディレクトリーを作る。一時ファイルを作るときにはファイル名とファイルハンドルが同時に返されるのが特徴だ。一時ファイル名をテンプレートから作ることもできる。

各メソッドの具体的な動作の違いについては、モジュールの解説ページを参照。

【メソッド】

```
·tmpdir([template[, option => value]])  
一時ディレクトリーの作成  
·tmpfile([template[, option => value]])  
一時ファイルの作成  
·mkstemp(template);  
一時ファイルの作成  
·mktemp(template[, $suffix]);  
一時ファイルの作成  
·mkdtemp(template[, $suffix]);  
一時ディレクトリーの作成  
·mktemp(template);  
一時ファイルの作成  
·tmpnam()  
一時ファイルの作成  
·tmpfile()  
一時ファイルの作成
```

Perlモジュールのユーザー環境へのインストール

CPANなどからダウンロードしたPerlモジュールを利用したいホストが自分の管理下でない場合(管理者権限を持っていない場合)があるだろう。サービスプロバイダーのウェブホスティングを利用している場合など、特にウェブ関係の場合には必ずしも管理者権限でモジュールのインストールや管理ができるとは限らない。その場合には、アクセス権限のあるディレクトリ(ユーザーのホームディレクトリなど)内にモジュールをインストールすることになる。ここでは、その方法とその場合のモジュールの使用方法について触れておく。

ポイントは、CPANで「perl Makefile.pl」が実行されるときにユーザーのディレクトリを使うようにパラメーターを設定しておくことだ。

第1回では、CPAN.pmモジュールを使ってモジュールのインストールを行う方法を説明した。これをユーザー環境で行うために、CPAN.pmをユーザー環境で実行して初期設定を行う。

```
user@hostname% perl -MCPAN -e shell
```

第1回で解説した初期設定が始まり、ユーザーのホームディレクトリ(この例では/home/user)にその設定が保存される。第1回の解説のとおり初期化を行うと、システム共通のディレクトリにダウンロードしたモジュールをインストールする設定になってしまうので、インストール先をユーザーのホームディレクトリ(もしくは書き込み権限のあるディレクトリ)に設定する必要がある。CPAN.pmでの初期設定の途中で次に示すようにMakefile.PL実行時のパラメーターを設定するところがある。

```
Parameters for the 'perl Makefile.PL' command? []
```

ここで、次のように入力する。

```
PREFIX=/home/user \
```

```
LIB=/home/user/lib/perl \
```

```
INSTALLMAN1DIR=/home/user/man/man1 \
```

```
INSTALLMAN3DIR=/home/user/man/man3
```

ここでは、ユーザー「user」のホームディレクトリ「/home/user」下にインストールすることを想定している。それぞれのパラメーターの意味は次のとおりである。

- ・PREFIX

インストールの基本ディレクトリ

- ・LIB

モジュールがインストールされるディレクトリ

- ・INSTALLMAN1DIR

セクション1のマニュアルがインストールされるディレクトリ

- ・INSTALLMAN3DIR

セクション3のマニュアルがインストールされるディレクトリ

すでにCPANの初期設定が済んでいる場合でも、CPANのシェルを起動してo conf initと入力することで再度初期設定を実行できる。

また、これらのパラメーターは、CPAN.pmモジュールの初期設定時だけでなく、CPAN.pmモジュールをインタラクティブモードで実行して、次のように入力することで設定できる。具体的には、「o conf makepl_arg」で値を設定し、「o conf commit」でオプションパラメーターの設定を保存するのだ。

実際に入力するのは太字で示した部分だ。

```
user@hostname% perl -MCPAN -e shell
```

```
cpan shell -- CPAN exploration and modules
installation (v1.48)
```

```
ReadLine support available (try ``install
Bundle::CPAN'')
```

```
cpan> o conf makepl_arg "PREFIX=/home/user
```

```
LIB=/home/user/lib/perl \
```

```
INSTALLMAN1DIR=/home/user/man/man1 \
```

```
INSTALLMAN3DIR=/home/user/man/man3"
```

```
makepl_arg PREFIX=/home/user LIB=/
home/user/lib/perl INSTALLMAN1DIR=/home/user
/man/man1 INSTALLMAN3DIR=/home/user/man/man3
```

```
cpan> o conf commit
```

```
commit: wrote makepl_arg
```

これで、ユーザー環境へのPerlモジュールのインストールはできた。しかし、各Perlスクリプト中で利用するモジュールを「use」を使って宣言した場合、モジュールの検索は標準のライブラリパスから行われるため、そのままではユーザー環境にインストールしたモジュールは利用されない。そこで、作成する各Perlスクリプトについて、明示的にライブラリのパスを指定する必要がある。ユーザー環境のパスは次のように行う。ただし、このライブラリのパス指定は利用するモジュールのuse宣言の前に置かなければならないということに注意してほしい。

```
## 上記LIBで指定したパスを指定する
```

```
use lib ('/home/user/lib/perl');
```

```
## ライブラリパスの指定の後でuseを指定する
```

```
use CGI;
```




[インターネットマガジン バックナンバーアーカイブ] ご利用上の注意

このPDFファイルは、株式会社インプレスR&D(株式会社インプレスから分割)が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

<http://i.impressRD.jp/bn>

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容(技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど)は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者(執筆者、写真の撮影者、イラストの作成者、編集部など)が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元(株式会社インプレス R&D)、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D(当時は株式会社インプレス)と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接のおよび間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先

株式会社インプレスR&D

All-in-One INTERNET magazine 編集部

im-info@impress.co.jp