

平成 14 年度 中部職業能力開発促進センター能力開発セミナー 「PHP + PostgreSQL 連携プログラミング」コース受講報告

岡田 佳浩

名古屋大学工学部・工学研究科技術部 電子・情報技術系

はじめに

中部職業能力開発促進センターでは、近年の技術革新に対応した幅広い知識と高度な技術・技能を習得するため、能力開発セミナーとして短期集中型のセミナーを数多く開講し、計画的にスキルアップを図れる能力開発の支援を行っている。今回、多岐にわたるコースの中から、平成 14 年 11 月 6 日(水)から 8 日(金)にかけて開講された、情報・通信系データベース技術の「PHP + PostgreSQL 連携プログラミング」コースを受講した。以下、その概要を報告する。

1. PHP の概要

PHP は HTML 埋め込み型のスクリプト言語である。PHP の特徴として、①プログラムの記述が容易。②実行速度が高速。③実行環境として対応する OS が多い。④オープンソースである。等があげられる。PHP は通常の CGI としても使用可能であるが、PHP モジュールを Web サーバ (Apache など) に組み込むことにより、CGI と比較して大幅な処理の高速化や、サーバへの負荷低減が実現可能である。図 1 のようにブラウザから CGI を実行要求した場合、サーバは CGI プログラムを実行するためのプロセスを生成するので、サーバに負荷がかかる。しかし、PHP を実行要求した場合、PHP スクリプトを実行するプロセスを Web サーバ (Apache モジュール) の一部に組み込み実行するので、サーバに負荷がかからない。デメリットとして、PHP のインタプリタ機能をモジュールとして Web サーバ自体に組み込むため、Web サーバ自身の実行イメージサイズが大きくなる。この対処法として、Apache には DSO (Dynamic Shared Object) 機能があり、これにより、必要なときにモジュールを動的に読み込む (実行時にリンクさせる) ことができる。

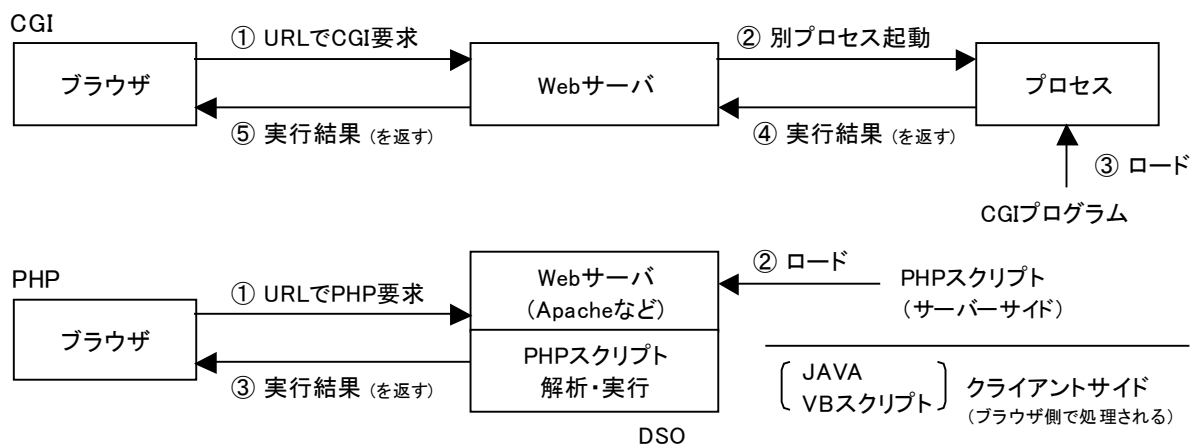


図 1 PHP と CGI の動作比較

2. PHP の導入

はじめに、サーバよりソースファイル PHP-4.2.2.tar.gz を FTP で入手し、これを展開した。

(tar xvzf PHP-4.2.2.tar.gz)。続いて、以下のようなオプションをつけて configure スクリプトを実行し、Makefile を作成した。

```
./configure --with-pgsql=/usr/local/pgsql --with-apxs=/usr/local/apache/bin/apxs
--enable-mbstr-enc-trans --enable-versioning --with-trans-sid --enable-mbstring
```

これらオプションの意味は、PostgreSQL 接続のためのインターフェイス有効。Apache モジュールとしてコンパイル（これで、Apache を起動することで同時に PHP も立ち上がり、Apache のプロセスの一部として動作させることができる）。Post、Cookie 入力の文字コード自動検出。PHP3 のスクリプトも実行可。Cookie を利用できない場合セッション ID を自動的に付加。PHP 国際化機能有効（マルチバイト文字列である日本語を利用可とする）。である。

Makefile が作成できたら、make コマンドでコンパイルを行う。コンパイルが無事終了したら、make install でインストールする。

次に、/usr/local/apache/conf/httpd.conf ファイルの次の部分を変更した。

```
#AddType application/x-httpd-php .php → AddType application/x-httpd-php .php
```

以上の変更（コメントをはずす）により、PHP4 のスクリプトファイルの拡張子は、.php になる。次に、/etc/ld.so.conf ファイル内に次の 1 行を追加してから ldconfig コマンドを実行する。

```
/usr/local/pgsql/lib
```

この後、Apache を再起動すると PHP が利用可能となる。PHP の動作確認は、

```
<? phpinfo() ?>
```

と記述した phpinfo.php ファイルを /usr/local/apache/htdocs 内に作成し、ファイル作成後、ブラウザを起動して、phpinfo.php を URL で指定し、正しく表示されるか確認する。

3. PHP の基本文法

3.1 PHP スクリプト記述タグ

ブラウザに「Hello, PHP!」と表示する PHP スクリプトを右に示す。PHP は HTML に埋め込むことが可能である。ただし、HTML に PHP を埋め込む場合、PHP をどの部分から埋め込んでいるのかわかるように目印を付ける必要がある。PHP スクリプトは、「<?」で始まり、「?>」で終わる（「<?php」で始まり、「?>」で終わる他の例もある）。また、実行する文の終わりには「;」を付ける。また、文字列は「"」または「'」で囲む。両者の違いは、「"」で囲んだ文字列内に変数が含まれている場合、その変数は展開されるが、それに対し「'」で囲んだ場合は、文字列内に変数が含まれていても変数の展開はされない。なお、記述するコマンドには大文字・小文字の区別はないが、変数は大文字、小文字が区別されるので注意が必要。

```
<html>
<head>
  <title>hello</title>
</head>
<body>
  <?
  print("Hello, PHP!");
  ?>
</body>
</html>
```

3.2 デバッグ

通常、PHP スクリプトはブラウザ上で実行する。このとき、コード内にエラーがあるとエラーメッセージを表示する。PHP でのエラーメッセージは以下のような形式で表示される。

```
エラーレベル : エラーメッセージ in ファイル名 on 行番号
```

このメッセージででてきたエラーレベルは、エラーの重大度に応じて、パースエラー、致命的エラー (Fatal error)、警告 (Warning)、注意 (Notice) の 4 つのレベルに分けられる。ただし、

デフォルトでは注意レベルのメッセージは表示されない。

3.3 変数

PHP では、変数の定義を明示的にする必要がない。変数の型（右表参照）は、その変数が利用されるとき自動的に決定されるため、変数の型と格納するデータの型について意識する必要がない。

なお、PHP では、次のような変数名規則があるので注意が必要。

- 変数名の前には「\$」を付ける。
- 変数名の大文字、小文字は区別される。（\$A と \$a は別の変数）
- 通常、定数名はすべて大文字で指定する。

よって、変数名はすべてを大文字にしないようにする。

- 変数名は英文字またはアンダースコア(_)から始まる名前を付けることができる。2文字目以降は、数字を含めることも可。
- 変数名の文字数は、200文字程度のものを使用しても問題ない。
- 変数名や関数名に日本語を使用することもできるが、すすめられない。

表 1 変数の型

型名	内容
Boolean	論理値
integer	整数型
double	実数型
string	文字列型
array	配列
object	オブジェクト
resource	リソース

3.4 演算子

PHP では、算術演算子、代入演算子、比較演算子、論理演算子、ビット演算子、文字列演算子など、C や perl から取り込んだ多くの演算子がサポートされている。式の中に複数の演算子がある場合、優先順位の高い演算子から評価され、同等の演算子の場合は左側にある演算子から順に評価される。演算子の種類にかかわらず、計算順位を上げたいときは、カッコでくくる。

3.5 変数出力

変数のデータを出力するには、print 関数や printf 関数を利用する。書式を指定して変数の値を出力する場合は printf 関数を使用する。構文は次の通り。

```
printf (string format,mixed [args] ...)
```

printf 関数は、フォーマット format に基づき引数 mix を出力する。フォーマット制御用変換指定子は、パディング文字、アライメント指定子、表示幅指定子、精度指定子、型指定子で構成される。

3.6 異なる型同士の演算

PHP では、算術演算子を使って文字列を演算すると、文字列の最初の部分に数値として有効な部分がある場合、文字列は数値に変換される。整数、小数、指数が有効な数値として認識される。数値に変換できない場合、文字列は「0」として評価される。また、整数型の数値と実数型の数値で演算を行った場合、結果は実数型の数値になる。

3.7 文字列操作関数、変数操作関数等

PHP では、多数の組み込み関数が利用できる。この中には、文字列操作に関する関数として、文字列の検索を行う関数、文字列の長さを調べる関数、必要な文字列を取得する関数、任意の文字列を別の文字列に置き換える関数などがある。また、変数操作に関する関数として、変数

の型をチェックする関数、キャスト、型変換を行う関数などがある。さらに、ユーザー定義関数を作成することもできる。

3.8 スコープ

変数は、スコープ（有効範囲）による分類で「グローバル変数」と「ローカル変数」に分けられる。ユーザー定義関数の中で定義された変数はローカル変数となり、ユーザー定義関数の外で定義された変数はグローバル変数となる。PHP では、ユーザー定義関数からグローバル変数を参照できないので、関数内でグローバル変数を定義するときは「global」と明示する必要がある。通常、関数内で使用される変数は、関数の呼び出し元に戻ると、解放されて消滅する。この関数内の変数の値を保持させたいときは、「static」を付けて変数を定義（静的変数）する。

3.9 定数

定数とは、決まった値を保持するための入れ物である。PHP では、定数を利用するときは、

```
define("定数名", 値);
```

のように、define 関数を利用して定義する。値について、文字列が半角数字から始まる場合（後に文字が入る場合）および、文字列が半角の空白を含む場合は、引用符で文字列を括らないとエラーになる。また、定数名は変数と区別するため、慣例的に大文字で定義することが多い。なお、定数には有効範囲は関係ないので、関数内で global を指定する必要がない。

3.10 制御構造

PHP では、ある特定の条件で処理を分岐したり、処理を繰り返したり、処理から脱出するなど、一般のプログラミングに必要な制御構造が用意されている。処理を分岐するための if 文や switch 文、処理を繰り返す for 文、while 文、do while 文などがある。

3.11 配列

配列とは、一連のデータ全体に 1 つの名前を付けてまとめたものである。配列に並んでいる変数それぞれを要素といい、次のように添字（インデックス）を利用して各要素を指定する。

```
$配列型変数名[添字]
```

PHP の配列は宣言の必要がないため、各要素ごとに異なる型のデータを代入することができる。また、動的に要素を増やすことや添字の連続していない番号の要素をいきなり利用することも可能である。PHP では、添字に数値を使う一般的な配列のほかに、添字に任意の文字列を使える連想配列も使用できる。配列の初期化には、値を連続的に代入するか、または array 関数を使用するかいずれかを用いる。

また、PHP では多次元配列を扱うことができる。多次元配列とは、配列に配列を入れたものである。添字が 1 つの配列は一次元配列で 1 列の表、添字が 2 つ続けば二次元配列で複数列の表、3 つ続けば三次元配列で縦×横×高さの表となる。n 次元配列の書式例は次のようになる。

```
$配列型変数名[添字 1][添字 2][添字 3]·····[添字 n]
```

3.12 クラス

PHP ではクラスを利用できる。クラスとは現実のある側面をモデル化し、要約したものであ

る。よって、クラスは、データとそのデータに対して実行可能な操作を持ち合わせる。クラスの定義は、変数とメソッドを定義するだけである。なお、クラスはあくまでも型なので、利用するためにはオブジェクト（実体）が必要である。オブジェクトの生成は new 文を利用する。クラス名と同じ名前のメソッド名を定義するとコンストラクタとなる。コンストラクタは、オブジェクトを生成したときに自動的に実行される。自オブジェクトが持つ変数やメソッドを利用するには、「this」という特別の変数を使う。this は自身のオブジェクトを表す。

クラスには「継承」という概念がある。継承とは、あるクラスを元に変数やメソッドを追加し、拡張したクラスを新たに作成する仕組みである。このとき、基になったクラスを親クラス、新たに作成されたクラスを子クラスという。継承を利用するときは、extends 文を利用する。また、親クラスで定義されているメソッドと同じ名前のメソッドを子クラスで定義することをオーバーライド（再定義）という。

3.13 include 文

include 文は、指定した外部ファイルのコードを読み込み実行する。よって、複数のページで利用できそうな関数やクラス、定数などは外部ファイルとして定義しておくことで集中管理ができて便利である。

3.14 実行演算子

PHP のプログラムからシェルコマンドを実行することができる。「`」（バッククォート）でシェルコマンドを囲んで実行すると、その出力を返す。実行できるコマンドは、Apache に与えられている権限（デフォルトユーザーは nobody）と同じ権限で実行される。そのため、セキュリティ上問題になることもあり、取り扱いには注意が必要。

3.15 ファイル操作

PHP では、ファイル操作に関する関数も多数用意されている。ファイルの存在を確認する関数、ファイルを開く関数や閉じる関数、ファイルの入力や出力を行う関数などがある。

4. PostgreSQL との連携

PHP では、DBMS と連携させるためのインターフェイスを多数用意している。PHP が用意している DBMS 用インターフェイスを利用し、Web と DBMS を連携させたアプリケーションを作成できる。PHP では図 2 のような流れで DBMS との連携をとる。

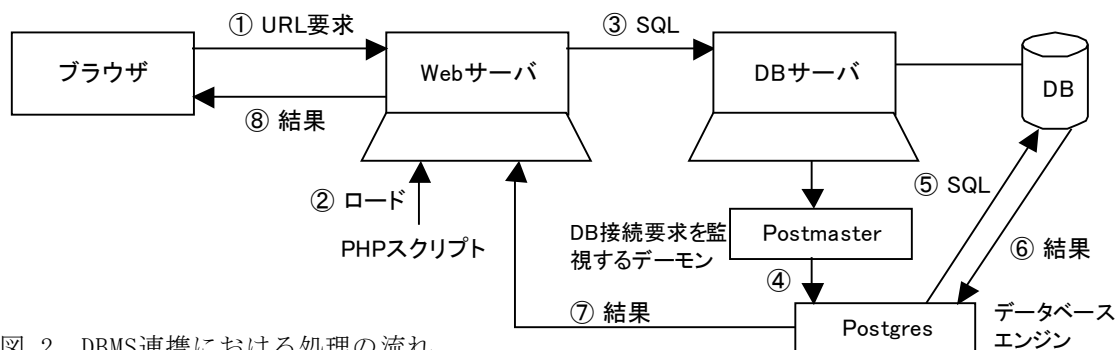


図 2 DBMS連携における処理の流れ

はじめに、PostgreSQL への TCP/IP 接続設定を行った。PHP から PostgreSQL へは TCP/IP プロトコルを利用して接続するため、「postgres.conf」ファイルの内容を次のように変更し、TCP/IP による接続を許可する設定を行った。

```
#tcpip_socket = false → tcpip_socket = true
```

TCP/IP による接続の許可は、「postmaster」コマンドに「-i」オプションを付けても行える。

続いて、データベースを利用可能とするため、次の操作を行った。

- ① initdb コマンドでデータベース初期化
- ② postmaster -i & でポストマスターを起動
- ③ createdb db00 でデータベース作成（データベース名は db00）
- ④ create user nobody with password 'pass'; で nobody ユーザー作成
- ⑤ pg_hba.conf ファイルの設定（外からの接続許可や認証方法などを設定）
- ⑥ pg_ctl stop でポストマスター終了
- ⑦ postmaster -i & でポストマスター(再)起動

データベース用データはあらかじめテキストファイルにて用意されていたので、これを PostgreSQL へデータインポートを行って利用した。

PHP から PostgreSQL への接続は、pg_connect 関数を、接続を閉じるには pg_close 関数を利用する。また、PHP スクリプトから SQL 文を発行するには pg_exec 関数を利用する。検索の結果セットの取得にもいろいろな関数が用意されている。

5. ストアドプロシージャ (PL/pgSQL 関数) ほか

PL/pgSQL は SQL 言語を使用するユーザー定義関数である。PL/pgSQL の特徴は、制御構造を利用できる。組み込み関数、ユーザー定義関数を呼び出すことができる。SQL 文をブロック単位で送信するので、パフォーマンスに優れている。などである。ここでは、PL/pgSQL のインストールを行い、要点について簡単に説明を受けた。講習時間が残りわずかとなったこともあり、データベーストリガー、クッキー、セッション管理、ラージオブジェクトについては要点について簡単な説明を受けて本講習を修了した。

おわりに

本セミナーは、能力開発セミナー受講ステップで Web・DB コースフローの最下段に位置していることと、本コースの受講要件に、「Linux[オペレーション]を受講された方、または同等の知識と技術を有する方」とあり、はじめからこのコースを選択して、はたしてついて行けるだろうか？と内心不安であったが、親切丁寧に教えていただき、特に意識することなく、大変有意義に学習することができた。また、今回のセミナーでは、PHP と PostgreSQL を扱うので、一度に両方の基礎的知識が得られるのではとの期待もあったが、講義の大半は PHP に関するもので、PostgreSQL は実習で使ったといった感じで終わってしまった。3日間という限られた時間では致し方ないのかもしれない。今回のセミナーで受講した内容をもとに、より一層知識を深め、今後の業務に生かしていきたい。