

Perl & CGI によるインタラクティブ・ホームページ

玉 置 一 雄
工学研究科技術部 計測・制御技術系

はじめに

最近の技術革新や情報化の急速な進展に対応するために、能力開発セミナーが2001年10月31日-11月2日の3日間、雇用・能力開発機構中部職業能力開発促進センター(ポリテクセンター中部)で開講された。この短期集中セミナーは、きめ細かなフォローを行う人材高度化支援事業で、幅広い知識と高度な技術・技能を習得することにある。機械系、電気・電子系、情報・通信系および居住系の4つの系に分けて開講されており、今回は情報・通信系のネットワーク技術の中の「Perl (Practical Extraction and Report Language) & CGI (Common Gateway Interface)によるインタラクティブ・ホームページ」コースに参加したので報告する。

1. 動的なホームページ

動的なホームページには、JAVA 言語で書かれた JAVA アプレット、HTML (Hypertext Markup Language) 内にプログラムを書き込む JavaScript や VBScript、UNIX 上で動く Perl やシェルスクリプト(sh, csh)で書かれた CGI、Microsoft の IIS (Internet Information Server) 稼動NTサーバ上で動く IDC (Internet Database Connector : DB アクセス機能を持つ IIS 特有のもの) 等がある。

前者の2つは、図1に示すように Web クライアント側で実行して結果を Web クライアントで表示するのに対して、後者の2つは、図2に示すように Web サーバ上で処理をし、動作結果を Web クライアントで表示する。このようにプログラムを作ってホームページを動かすには、クライアント側(ブラウザ上)で動かすものと、サーバ側(WWW サーバ上)で動かすものと大きく2つに分類できる。

JavaScript や VBScript はダウンロードしたブラウザで動作しており、このブラウザでファイル进行操作するとダウンロードした一般のユーザのハードディスクを使用することになる。クラッカーが進入しユーザのハードディスクを破壊することも簡単に行なえるのでとても危険である。したがってファイル処理は、CGI が有効である。

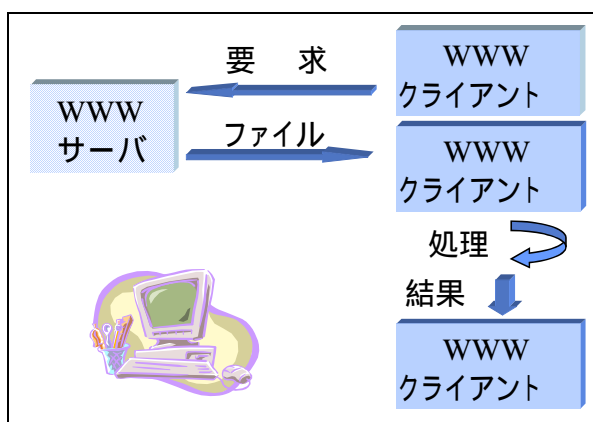


図1. JavaScript, VBScript の動作

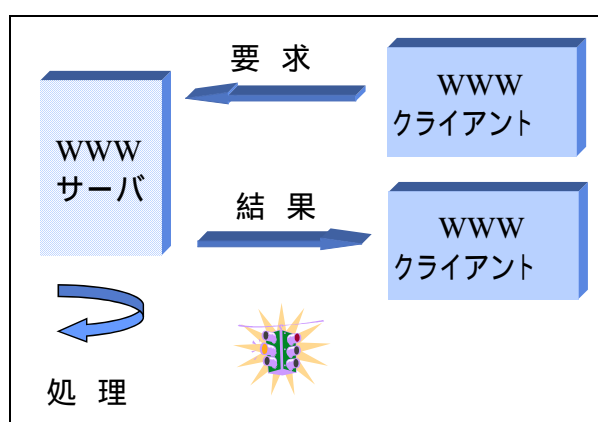


図2. Common Gateway Interface の動作

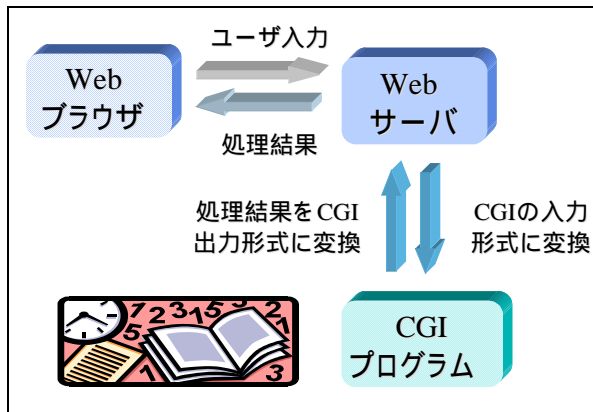


図3.CGIプログラムの動作手順

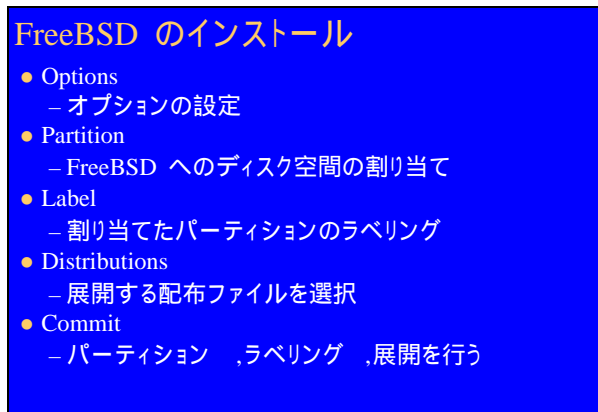


図4.FreeBSDのインストール

2. CGIプログラムの動作手順

CGIプログラムの動作手順を図3に示す.詳しく見るとブラウザがPerlで書かれたCGIプログラムファイルを要求すれば,CGIファイルがブラウザに送信されるのではなく,サーバ側でCGIプログラムが動き,そのCGIプログラムがブラウザへHTML文書内の文字を送信する.

CGI作成のメリットは,次のような点にある.

1. ユーザの入力情報によって出力させる情報を変更でき,さまざまなことをWeb上で行うことが可能になる.
2. クライアントは処理結果を表示するだけなので,非力なクライアントでも大丈夫である.
3. 携帯電話などの端末からCGIを呼び出し,結果を携帯電話で表示することが可能である.
4. ユーザに対し動作内容,情報の隠蔽が行える.

しかし非常に簡単な処理まで行うことは,Webサーバに負荷が集中することになり避けなければならない.

3. FreeBSDのインストール (4.3RELEASE)

ブータブルCD-ROMを起動してKeymapを設定し,Customによるインストールをする.「Options」の設定から実際の展開作業「Commit」までを図4に示す.インストール後,システム管理者のパスワードの設定,ユーザおよびグループの追加,タイムゾーンの設定,ネットワークサービス等の環境設定および確認を行う.

3.1 リアル環境(ホスト環境)とバーチャル環境(jail環境)

はじめに図5に示すようにインストールしたFreeBSD環境(ホスト環境)をリアル環境と呼び,Webが動作している環境(jail環境)をバーチャル環境と呼ぶことにする.

3.2 セキュアサーバの構築 (Ver2.2)

セキュアサーバの構築は,図6のような手順で行う.まずjail用ディレクトリツリーの作成をする.リアル環境に一般ユーザでログインして,SUコマンドで管理者(root)になる./cdrom/seminarからインストーラ(jailsetup.sh)を/rootにコピーして実行する.ここでインストーラはシェルスクリプトになっているので次のように入力する.

```
Freebsd# mount/cdrom
Freebsd# cp /cdrom/seminar/jailsetup.sh ~/
Freebsd# sh ./jailsetup.sh
```

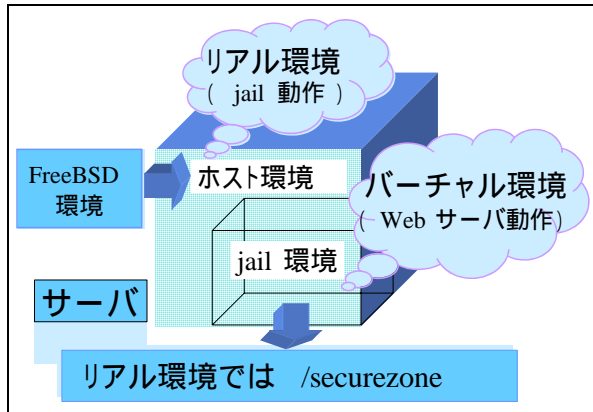


図 5.リアル環境とバーチャル環境

セキュアサーバの構築 (Ver2.2)

- リアル環境 (ホスト環境) の設定
 - インストーラを使用した jail 環境の構築
 - LANカード (インターフェース) 名
 - IP アドレス, ホスト名, ドメイン名
- バーチャル環境 (jail 環境) の設定
 - シングルユーザモードで jail 環境へ移る
- バーチャル環境の動作確認
 - バーチャル環境へ telnet する
 - バーチャル環境へ ftp する

図 6.セキュアサーバの構築

jail 環境設定画面が表示されるので各項目に jail 環境用の情報を設定する。設定が確認できれば,GO->jail Configure set it を選択し,jail 環境のセットアップに進む。

次に作業ディレクトリ/root から jail 環境になるために(下記参照),シェルスクリプト(single.sh)を実行し,hostname コマンドで jail 環境のホスト名であることを確認する。jail 環境の作業ディレクトリ/root の setup.sh というシェルスクリプトを実行し,jail 環境へ Web サーバを構築する。jail 環境の管理者(root)と webmaster のパスワードを入力し,Web サーバアプリケーションのセットアップをする。Setting OK! into jail environmental というメッセージが表示されたら,バーチャル環境からリアル環境へ戻り,システムを再起動する。

(リアル環境)	Freebsd# sh ./single.sh	
(バーチャル環境)	#hostname	
(バーチャル環境)	jail14.mech.nagoya-u.ac.jp	jail 環境のホスト名
(バーチャル環境)	#cd /root	
(バーチャル環境)	#sh ./setup.sh	
(バーチャル環境)	#exit	
(リアル環境)	Freebsd# shutdown -r now	

最後にマルチユーザモードで起動し,jail を開始する。バーチャル環境へ telnet, FTP(File Transfer Protocol)して動作確認をする。

4. CGI の作成手順

スクリプトの作成から Web サーバへファイルをアップロードしスクリプトを実行するまでを図7に示す。

まずスクリプトの作成を行う。オリジナルの CGI を作成するには perl の知識が必要であるが,インターネット上にはさまざまなフリーの CGI スクリプトが公開されている。そのスクリプトをダウンロードして,設置すれば比較的簡単に作成することができる。つぎにエラーが無くなるまで端末で十分テストを行う。ここでデバッグされていないスクリプトを実行させるとか,あいまいな知識で作成したスクリプトを実行させるとサーバがハングアップすることもある。デバッグを繰り返しエラーが無くなったらサーバへスクリプトをアップロードする。環境設定を行い,最後に正常に動作しているか確認する。

4.1 COUNTDOWN スクリプト (Perl)

指定した数値からのカウントダウンを表示するスクリプトを図 8 に示す。ある期待する状態まで同じ処理を繰り返し実行することをループと呼ぶ。ループ処理を行うための While ステートメントの例を示す。これは式の値が真の間ステートメントを繰り返し、偽になるとループを抜けて次の命令に移る。しかし最初から偽であった場合にはステートメントは1回も実行されず次の命令に移る。ここで\$numberの値が0以上のときにはループのWhileステートメントを実行する。このスクリプトを実行すると、「初期値を入力して下さい!」と表示され、たとえば100を入力して[ENTER]キーを押すと100から0までカウントダウンを行う。

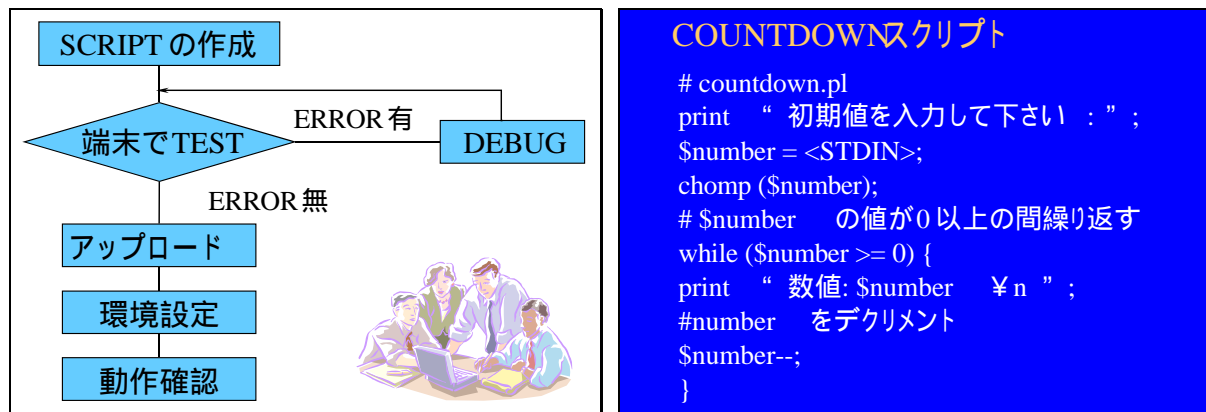


図 7. CGI 作成手順

図 8. Countdown スクリプト

4.2 サーバへの CGI の登録(転送)

まずテキストコンバートツールでテキスト変換する。漢字コードの設定と読み込んだファイルの漢字コードが一致しない場合、ホームページやメールの文字化けが起こる。また改行コードが異なると、改行が全くなるとか、が表示されることがある。

Windows では改行コードが CR (Carriage Return) + LF (Line Feed) で漢字コードが Shift-JIS, Macintosh では改行コードが CR で漢字コードが Shift-JIS が使われており, UNIX サーバで動かす Perl スクリプトでは改行コードを LF, 漢字コードを EUC にする必要がある。

つぎにサーバへのアップロードには FTP クライアントソフトを用いる。Macintosh では Fetch, Windows では FFFTP や FTP Explorer が有名である。TEXT (ASCII) モードは改行コードを UNIX 用に交換して転送する。一方 BINARY (IMAGE) モードはファイルのデータを変更せずにそのまま転送する。したがって TEXT ファイルは前者の転送モード (パソコン上で予め改行コードを UNIX の LF に交換すれば BINARY モードで転送が可能である。), 画像ファイルは後者の転送モードを使用する。

最後に UNIX 系の OS では Owner (所有者のユーザ), Group (同じユーザグループに属するユーザ), Other (すべてのユーザ) に分類してそれぞれファイルにアクセスする権限を設定する。権限の種類には Read (読み込み可能), Write (書き込み可能), Execute (実行可能) がある。パーミッションは CGI スクリプトでは 755, 読み書きファイルでは 666 というように数字で表され, 左から Owner, Group, Other に対応している。数字は Read が 4, Write が 2, Execute が 1 を表し, 権限が複数ある時はその足し算で表す。

ファイルのパーミッションの指定の誤り, 構文エラー等のスクリプトに何らかの問題があってスクリプトが実行で

きないときに Internal Server Error が起こる。これはサーバに非常に大きな負荷をかけることになり、最悪の場合サーバをダウンさせデータの復旧が不可能になることもあるので注意が必要である。

5. セキュリティ

前にも述べたように、CGI は Web サーバ上で実行されるため、スクリプト上ではシステムをクラッシュすることも可能である。スクリプトが永ループに入り、サーバがダウンすることもある。スクリプトを繰り返しテストしてからスクリプトをサーバへアップロードすることが大切である。特にアンケート、掲示板等を受け取りそれに対して処理を行う CGI を作成した場合、open の引数として最後に「|」が付いた文字列を渡すと、コマンドとして実行してブラウザ上に表示されパスワードを盗まれることもある。このように入力データに不正な文字が入っていないかチェックすることも重要である。なお以下のことにも細心の注意を払わなければならない。

プログラムやデータを守るために、ファイルの属性を必要以上に読み書き実行可能にしない。

ブラウザ経由でサブディレクトリの一覧を取られないようにサブディレクトリには index.html をおく。

CGI プログラムを非表示にするように、.htaccess の設定をする。

<META>タグを使って別の画像を張り付けられないようにタグを無効にする。

このように CGI は、クラッカーの標的になりやすく危険であり、なにか新しい防御の開発が望まれる。

6. アクセスカウンタ

図9にアクセスカウンタの作成の手順を、図10にアクセスカウンタの CGI プログラムの基本形を示す。

- 1.まずカウンタファイル(これまでにアクセスされた回数の記録ファイル)に記録されている数値を読み込む。
- 2.ファイルをロック(ファイルを操作しているとき、他の人がそのファイルを操作できない仕組み)する。
- 3.数値を1増加する。
- 4.ファイルのロックを解除する。
- 5.更新した数値をタグを使って表現する。
- 6.更新した数値をカウンタファイルに記述する。

7. CGI プログラム(カウンタの基本形)

CGI は、Shell や C 言語でも作成できるが、ここではコンパイルする必要もなく手軽な Perl を用いる。

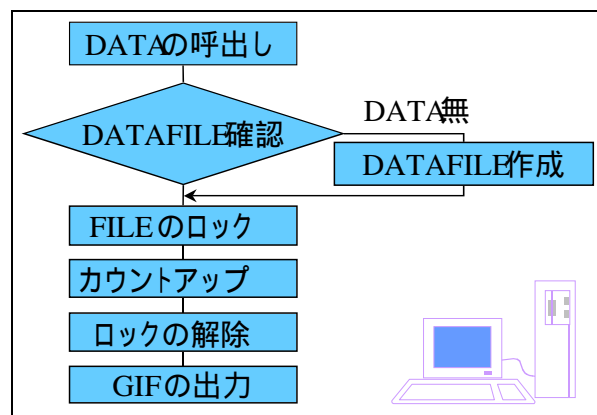


図9.アクセスカウンタ作成手順

```
CGIプログラム (カウンタの基本形)

#!/usr/local/bin/perl
$I=1;
$countfile = " your directory/tamaki.txt ";
open(COUNTER, " countfile ");
$count = <COUNTER>;
close COUNTER;
++$count;
open(COUNTER, " >countfile ");
print COUNTER $count;
close COUNTER;
print " Content-type:text/html \n\n ";
print $count;
END
```

図10.カウンタの CGI プログラム

図 10 に示すように Perl で作成するスクリプトの先頭行には、必ず Perl が実行できるサーバのパスを入力する。CGI のほとんどの結果が、HTML 文書、GIF 画像、ファイルに記録するかである。GIF 画像を返すスクリプトの代表的なものがグラフィックアクセスカウンタである。それ以外のほとんどのスクリプトは、ファイル操作をし、HTML 文書を作成する。

まず、CGI の結果がどのような種類の結果になるか記述する。Web サーバは Web クライアントへその情報を伝え、Web クライアントは受信出来るデータかどうかを判断する。それがヘッダーの役目であり、ヘッダーの最後には必ず空白行が必要である。空白行がないとサーバは次に来るデータの特定ができないので Server Error となる。ブラウザ側ではどこまでがヘッダーでどこからがデータ部かを区別するために何も記述していない空白行を識別子として用いる。

つぎに HTML 文書作成を宣言するヘッダーは `print "Content-type:text/html\r\n\r\n";` と記述する。たとえば無編集のテキストのヘッダーは `Content-type: text/plain`、GIF(Graphics Interchange Format) のヘッダーは `Content-type: image/gif`、JPEG(Joint Photographic Experts Group) のヘッダーは `Content-type: image/jpeg` となる。このヘッダーより下に通常の HTML タグで文書を作成する。ヘッダーには `Content-type` 以外にも多数あり複数のヘッダーを指定することができる。ヘッダーを連続して記述するときは最後のヘッダーの後に空白行を用いる。変数は「数値」「文字」どちらでも代入でき、他の多くの言語のように使用する変数名を宣言する必要がなく、どこからでも使用することができる。

おわりに

CGI を作成する際に注意する点として、各プラットフォームの文字コードの違いがある。掲示板のような入力を受け付けるスクリプトを作成する場合は、入力データをそのまま保存してしまうと一つのファイルに複数の文字コードが存在することになる。こういった問題はプログラマ側で対応する必要がある。また、立派なホームページを作成してもアクセスに長い時間がかかったのでは役に立たない。読者の身になってホームページ作成をすることが大切である。

参考文献

1. ポリテクセンター中部「FreeBSD ネットワークサーバ構築」能力開発セミナー資料 (2001).
2. 武藤健志/トップスタジオ編著「独習 Perl - Teach Yourself Perl」翔泳社 (2000).
3. 結城浩著「Perl で作る CGI 入門(基礎編)」ソフトバンク パブリッシング社 (2001).
4. 三島俊司著「CGI のための実践入門 Perl」技術評論社 (1998).
5. 紙谷歌寿彦著「はじめての人のためのかんたん Perl/CGI 入門」秀和システム社 (2001).
6. Knave 著「CGI 作成講座,Perl プログラミング講座」Knave Web 講座 (2001).
7. Johansen 著「CGI 設置法」Johansen's Computer & Internet Information (2001).
8. 秋本祥一、古川剛共著「CGI プログラミング入門」翔泳社 (1997).
9. 秋本祥一、古川剛共著「CGI 基礎講座」翔泳社 (1999).

