# Webサーバー上の32ビットFORTH

| | |
|---|---|
| 著者名(日) | 貞方 一也 |
| 雑誌名 | 北海道医療大学看護福祉学部紀要 |
| 巻 | 11 |
| ページ | 59-65 |
| 発行年 | 2004 |
| URL | http://id.nii.ac.jp/1145/00006688/ |

＜研究報告＞

# Webサーバー上の32ビットFORTH

貞　方　一　也*

抄　録：今回のプロジェクトはWebサーバー上でFORTHを動かすことを目的とする。プロジェクトの内容は、まずLANに接続するWindowsマシーンの上でWebサーバーを立ち上げること、次に、このサーバー上で、cgiプログラムによってFORTHをとにかく動かすこと、最後に、FORTHを対話的に動かす仕組みを作り上げることである。プロジェクトは成功し、直接起動されたFORTHと同じく、Webサーバー上のFORTHを対話的に動かすことができるようになった。

キーワード：32ビットFORTH、Webサーバー、cgiプログラム、対話的実行

## はじめに

10年以上前から筆者はWindowsのDOS窓で動く32ビットFORTHの開発を続けている。最新の第2.7版のFORTHは、次の特長を持っている[1]。

(1) 小数点付多数桁数というデータ型を持つこと

これは、10万進法で整数部は1桁、小数部は何桁でも持つことができるデータ型である。

(2) 小数点付多数桁数を入出力とする関数を持つこと

このような関数を使って、例えば、eのπ乗を何千桁の精度でも計算することができる。

FORTHなどのプログラムを自分のコンピュータに置く場合はそれを起動し実行することができる。一方、Webサーバーに置かれたプログラムも自分のコンピュータから実行することができる。例えば、我が大学のLANのWebサーバーに置かれたperlが実行可能である。なお、このサーバーはUnixマシーンである。我々のFORTHはWindowsのプログラムであり、Unixマシーンでは実行することはできない。

今回の筆者のプロジェクトは、Webサーバー上のFORTHを別のコンピュータから動かすことを目的としている。ここで、FORTHがWindowsのプログラムであるため、WebサーバーもWindowsマシーン上に設置されねばならない。

我々のプロジェクトは次の3つの部分からなる。

(1) Webサーバーの立ち上げ

Windowsマシーン上にWebサーバーを立ち上げる。そして、Webサーバー上に置かれたperlのプログラムが別のコンピュータから実行可能であることを確かめる。

(2) FORTHをとりあえず動かすための仕組み

perlのcgiプログラムによってFORTHを動かすことができる。しかし、起動したコンピュータがFORTHの出力を受け取るためには、FORTHの起動メッセージと終了メッセージに工夫が必要である。また、この段階では、実行ソースは、あらかじめファイルに書き込まれているものに限られる。

(3) FORTHを対話的に動かすための仕組みの導入

htmlのウィンドウの入力エリアにソースを入力し、それを実行させる仕組みを導入する。実際は、これは入力されたソースを一度作業ファイルに書き出し、作業ファイルからそれをFORTHのソース領域に読み出して実行するという仕組みである。

これらの3つの部分について以下に詳しく述べる。

## 1．Webサーバーの立ち上げ

筆者のWindowsマシーン上にWebサーバーを立ち上げることにする。このサーバーには、サーバー自身であるlocalhostからアクセスできる。もし、このマシーンがLANに接続していれば、そのLANに接続している別のマシーンからWebサーバーにアクセスできる。なお、Webサーバーの制御言語としてはperlを使い、Webサーバーとしては、apacheを使うものとする。

＊人間基礎科学講座

### 1.1 perlのインストール

Active Perlのインストール・パッケージ[2]をダウンロードし、それを実行するとperlがインストールされる。ただし、次の2点の変更を行う。

① perlのインストール・ディレクトリ

c : ¥usrとする。したがって、perlの実行プログラムのディレクトリはc : ¥usr¥binとなる。

② path

もともとのpathにc : ¥usr¥binを加える。

### 1.2 apacheのインストール

Webサーバーapacheのインストール・パッケージ[3]をダウンロードし実行すると、apacheがインストールされる。なお、apacheの設定ファイルhttpd.confの内容を次のように変更する。

① サーバーアドレス

ServerName 127.0.0.1：80

② 文書ルート

DocumentRoot "C : /2ishp/public_html"

③ ディレクトリ

＜Directory "c : /2ishp/public_html">

④ 除外

#AddDefaultCharset ISO-8859-1

⑤ ハンドラーの追加

AddHandler send-as-is asis

AddHandler cgi-script.cgi

⑥ cgiの使用

Options Indexes FollowSymLinks MultiViews
ExecCGI Includes

⑦ 言語

LanguagePriority ja en ca cs da de el eo es
et fr he hr it ko ltz nl nn no pl pt pt-BR
ru sv zh-CN zh-TW

### 1.3 ホーム・ディレクトリへアクセス

簡単なindex.htmlを作成し、それをホーム・ディレクトリへ入れておく。まず、apacheをスタートさせる。その後、コマンドプロンプトから次のように入力する。

http : //localhost : 80/

その結果、apacheに接続してindex.htmlが開かれる。なお、WebサーバーがLANに組み込まれ、それがアドレス（例えば192.168.0.1）を与えられているとき、LANに接続している別のコンピュータから、次のように入力してWebサーバーのホーム・ディレクトリへアクセスすることができる。

http : //192.168.0.1 : 80/

### 1.4 perlのプログラムの実行

ホーム・ディレクトリの中のサブ・ディレクトリkeiji07に次のような内容のプログラムhello.cgiをおく。

```
#!/usr/bin/perl
require "jcode.pl";
print qq(Content-type：text/html；
        charset=$CHARSET¥n¥n)；
print qq(<!DOCTYPE HTML PUBLIC "-//W3C//DTD
        HTML4.01//EN">¥n)；
print qq(<html>¥n)；
print qq(<head><title>メッセージ
        </title></head>¥n)；
print qq(<body>Hello, world!</body>¥n)；
print qq(</html>¥n)；
```

コマンド・プロンプトから次のように入力してhello.cgiを実行する。

http : //localhost : 80/keiji07/hello.cgi

このとき、メッセージというタイトルのhtmlのウィンドウが開かれ、そこにHello, world！と表示される。

hello.cgiは、文書のタイプ、htmlタグ、本体、htmlタグと順番に出力するプログラムであるが、ウィンドウには本体（Hello, world！）のみが表示される。なお、cgiプログラムについては文献[4]が参考になる。

## 2．Webサーバー上でFORTHを実行する仕組み

32ビットFORTHには、次の2つのFORTHがある。

① forth32w.exe

1MバイトのDOSメモリーのうちの空きメモリーを使うFORTHである。自力で（補助プログラムなしで）32ビットのプログラムとなる。第2.7版のメモリの大きさは50000Hバイトである。

② forthex.exp

拡張メモリーを使って動作する。第2.7版のメモリの大きさは1000000Hである（4Gバイトまで拡張可能）。起動には補助プログラムexe32.exeを必要とする。

第2節と第3節では、forth32w.exeを取り上げるが、起動に必要な補助プログラムにも修正を要することを除いて、forthex.expも同様に扱うことができる。

### 2.1 FORTHの起動

FORTHを第1.4節で述べたperlのように実行することはできない。しかし、まず、perlを起動し、perlのコマ

ンドsystemを使ってFORTHを起動することはできる。例えば、次のperlプログラムprogtest.plがあるとする。

```
#!/usr/bin/perl
$str=system(" cmd /Cc : /forth/forth32w.exe
  ");                          ・・・文(A)
exit;
```

コマンドプロンプトから

perl progtest.pl

と入力し実行すると、forth32wが起動される。ここで、例えば、πを1000桁計算して、得られた結果を表示し、その後終了するには、次のように入力し実行する。

1000 bnp-pi bnp. bye　　・・・処理(B)

### 2.2 FORTHの自動実行

Webサーバー上のプログラムはperlのようにソースのプログラムを実行できなければならない(自動実行機能と呼ぶ)。FORTHは本来対話型のプログラムであるが、2通りの自動実行機能を持っている。その1つがコマンドラインのオプション/iを使うものである(もう1つの方法は、次の第3節で取り上げる)。

この方法で前の第2.1節の処理(B)を行わせるには、文(A)の代わりに次のような文とする。

```
$str=system("cmd /Cc : /forth/forth32w.exe /i
    1000 bnp-pi bnp. bye ");  ・・・文(B')
```

### 2.3 FORTHのウィンドウへの出力

FORTHの出力はすべて標準出力向けのものであり、htmlのウィンドウへ向けたものではない。FORTHの出力をhtmlのウィンドウの中で見るためには、FORTH自身が次の作業を行うようにFORTHを変更する必要がある。

① FORTHの起動時メッセージへ追加

html文書形式を出力し、はじめのhtmlタグを出力してから通常の起動メッセージを出すようにする。なお、html文書形式、はじめのhtmlタグと終わりのhtmlタグに関しては第1.4節のcgiプログラムを参照のこと。

② 終了時メッセージを出して終了するワードの追加

終わりのhtmlタグを出力してから終了するワードhtml-byeを追加する。

forth32w.exeにこれらの①、②の変更を加えたものをforth32l.exeとする。

### 2.4 FORTHを動かすcgiプログラム

Webサーバー上でFORTHを起動し、それに(B)の作業を行わせるプログラムは次のforth32.cgiである。

```
#!/usr/bin/perl
$str=system ("cmd /Cc : /forth/forth32l.exe /i 1000 bnp-pi
bnp. html-bye");
exit;
```

これをホームディレクトリのkeiji07に入れておく。このとき、コマンドプロンプトから

http : //localhost : 80/keiji07/forth32.cgi

と入力し実行する。実行の結果はhtmlウィンドウの中に表示される。

### 2.5 テキストエリアに出力

第2.4節で示したcgiプログラムはFORTHの出力を表示するが、改行に関して問題がある。

普通のFORTHでは、例えば

.time cr .date

というソースを実行すると、時刻表示(.time)の後、改行(cr)し、日付表示(.date)する。ところで、第2.2節の文(B')を次のように変更して実行してみる。

```
$str=system("cmd /Cc : /forth/forth32l.exe /i .time
  cr .date html-bye");
```

このとき、時刻表示の後、続いて日付が表示される。つまり、crはその役目を果たしていないことになる。

改行が正しく行われないという問題を解決するには、FORTHの出力先をテキストエリア(TEXTAREA)とすればよいことが実際に試みて確かめられる。出力先をテキストエリアとするために、第2.3節の①と②を次のように変更する。

① 起動時のメッセージ出力

次のようなテキストエリアの開始タグも出力する。

<textarea rows="25" cols="80">

② 終了時メッセージ出力と終了(html−bye)

次のようなテキストエリアの終了タグも出力する

</textarea>

## 3. FORTHを対話的に動かす仕組み

第2節でFORTHをとりあえず動かす仕組みを導入した。この仕組みでは、実行するソースをあらかじめcgiファイルに書き込んでおくことが必要であり、対話的にFORTHを動かすとはいえない。しかも、localhost以外にはファイルを書き変えることはできない。ところで、FORTHには、前節で用いたオプション/i でソースを指定する方法の他に、ファイルからソースをロードするという自動実行の方法がある。本節では、後者の方法を使ってFORTHを対話的に動かす仕組みを導入する。

### 3.1 ファイルのロード

FORTHではソース・プログラムをblk形式のファイルに保存することができる。そのようなblk形式のファイルを起動時にロードすると、そのファイルに書かれたソースが実行される。例えば、testhtml.blkに次のように書き込んでおくとする。

1000 bnp-pi bnp. bye

コマンドラインから次のように入力し実行するとする。

forth32w testhtml.blk

このとき、forth32wはtesthtml.blkをロードし、ソースを実行する。結局、1000桁分のπを計算しそれを表示し終了することになる。

このようなファイルtesthtml.blkがあるとき、Webサーバー上のcgiプログラムには、文(B'')の代わりに次のような文を置くことになる。

```
$str=system("cmd /Cc : /forth/forth32l.exe
        testhtml.blk");    ・・・文(B''')
```

この場合には、別なソースを実行するには、cgiプログラムではなく、testhtml.blkの内容だけを変更すればよいことになる。

### 3.2 ソースの入力と実行

本来のblkファイルは任意個のblockを持つことができる。ただし、1つのblockは1024バイトである。簡単にするため、Webサーバー上では、1個のblockを持つblkファイルのみを扱うものとする。さて、本来のFORTHでは、blkファイルを直接に編集するのであるが、Webサーバー上のFORTHではそのような訳には行かない。

Webサーバー上でFORTHを対話的に動かすために次のような仕組みを作る。

① htmlウィンドウ上のFORMにソース入力用のテキストエリアを設ける。なお、テキストエリアのサイズは1024バイトとする。

② FORMにある実行(Run)ボタンをクリックすると、テキストエリアの内容が、testhtml.blkに書き込まれた後で、

```
$str=system("cmd /Cc : /forth/forth32l.exe
        testhtml.blk");
```

が実行されるようにする。

このような仕組みでは、testhtml.blkは作業用ファイルであり、我々が直接編集するのはhtmlウィンドウのFORMにあるtextareaである。

### 3.3 自動終了のためのワードの修正

対話的な実行には、forth32l.exeに2つの修正が必要である。

① ソースが尽きたとき、forth32lが終了すること

本来のFORTHは、1000 bnp-pi bnp.のようなソースを実行しても終了せず、byeを実行して終了する。

Web上でテキストエリアにソースを入力するとき、いちいちソースの尻尾にhtml-byeを書くのは不便である。この不便さを解消するため、ソースが尽きたときにhtml-byeを実行し終了するように自動実行のワードautoexecの仕様を改める。

② エラーが起きたとき、forth32lが終了すること

ソース中のワードの名前が間違っていると、エラーが起こる。この場合も終了するようワード?errorの仕様を改める。

### 3.4 対話的に処理を行うプログラムforth32.cgi

プログラムforth32.cgiのリストを付録に示す。このプログラムの実行の流れは次の通りである。

① 起動後、FORMデータを取り込む

② プログラム自身から呼ばれた(Runボタンのクリックにより)のであれば、

・ソースをtesthtml.blkへ書き込む
・forth32l.exeを起動し、testhtml.blkをロードしてソースを実行する
・実行結果を新しいウィンドウへ出力する
・待機する
・ウィンドウを閉じて前のウィンドウへ戻る

③ FORMを表示し、testhtml.blkを表示する

④ ソースが入力されてRunボタンが押されるのを待つ

⑤ Runボタンが押されたならば、プログラム自身を起動し、そこにFORMデータを送り①へ行く

forth32.cgiを起動した後、我々のすることは、入力ウィンドウでソースを入力すること、Runボタンを押すこと、出力ウィンドウを閉じて始めの入力ウィンドウに戻ること、および、これら一連の作業を繰り返し行うことである。

## 4. 終わりに

Windows上のFORTHであるforth32w.exeを元としてforth32l.exeを作り、それを用いるWebサーバー上のプログラムforth32.cgiを作成した。このforth32.cgiは1024バイト分のソース入力用テキストエリアを持ち、blockが1個のblkファイルを開いた状態のforth32w.exeとほとんど同様の働きをする。

一方、拡張メモリに起動されるFORTHであるforthex.expからforthex1.expを作り、それを用いるWebサーバー上のプログラムforthex.cgiを作成した。forthex.cgi

もforthex.expとほぼ同様の機能を持っている。

二つのFORTHを比べると、froth32.cgiの方がforthex.cgiよりはるかに軽快に動作する。一方、使用するメモリーのサイズの違いから機能はforthexp.cgiの方が大きいといえる。例えば、1000000 bnp-pi（πの1,000,000桁計算）は、forthexp.cgiでは実行可能であるが、forth32.cgiでは実行不可能である。

今回は筆者が開発したWindowsのFORTHだけを取り上げた。ところで、Linux上にもgforth[5]のようなFORTHがある。今後Linux上のWebサーバーでgforthを動かすことを試みたいと考えている。

## 付録　forth32.cgiのリスト

```
#!/usr/bin/perl
require "jcode.pl";

#＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝ユーザ設定＝＝＝＝
$CHARSET = 'Shift_JIS';
$DATAFILE = 'testhtml.blk';
#＝＝＝＝＝＝＝＝＝＝＝＝＝＝メインプログラム＝＝＝＝
loadFormdata();
if (exists $FORM{'mode'}) {
        if ($FORM{'mode'} eq 'write')
 { $str2=$FORM{'source'};
$str1='';
for $i(1..1024){ $str1.='';}
$str2.=$str1;
$str3=substr($str2,0,1024);
$str4=$str1.$str3;
#ソースの書き込み
open(FILE, ">$DATAFILE") or printErrorPage
 ("書き込み用ファイルが開けません。");
eval{ flock(FILE,2)};
print FILE $str4;
close FILE;

$str=system("cmd /Cc : /forth/forth32l testhtml.blk ");
exit;
}}
printPage();
exit;
#＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝ソースの出力＝＝＝＝
sub   printPage
  {
my $str1='';
if(-e "$DATAFILE") {
open(FILE, "<$DATAFILE");
eval{ flock (FILE, 1)};
@DATA = <FILE>;
close FILE;
my $strall=$DATA[0];
$str1＝substr($strall,1024,1024);
$str1＝~s/>/&gt;/g;
$str1＝~s/</&lt;/g;
}

print <<END;
Content-type：text/html；charset=$CHARSET
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.01//EN">
<html>
<head><title>FORTH32w</title></head>
<body bgcolor="#008080" text="#000000">
<div align="center">
      <img src="myforth.JPG" > </div>
<form action="$ENV{'SCRIPT_NAME'}"
method="POST">
<div align="center">
     <textarea cols="64" rows="20" wrap="hard"
        name="source"> </textarea></div><br>
     <div align="center">
       <input type="hidden" name="mode"
            value="write">
     <input type="submit" value="Run">
     <input type="reset" value="Clear">
     </div>
</form>
<p> SCR #1 : $str1</p>

</body>
</html>
END

}
#＝＝＝＝＝＝＝＝＝＝＝＝エラーページ出力＝＝＝＝
sub   printErrorPage
{
print qq(Content－type：text/html；
  charset=$CHARSET¥n¥n);
print qq(<!DOCTYPE HTML PUBLIC
  "-//W3C//DTD HTML 4.01//EN">¥n);
print qq(<html>¥n);
print qq(<head><title>FORTH32w
```

```
  </title></head>¥n) ;
print qq(<body><h1>エラー
      </h1><p>$_[0]</p></body>¥n) ;
print qq(</html>¥n) ;
exit ;
}
#＝＝＝＝＝＝＝＝＝フォームデータ取り込み＝＝＝＝
sub   loadFormdata
{
my    ($query, $pair);
if (’REQUEST_METHOD’} eq ’POST’)
    {
  read(STDIN, $query,
    $ENV{’CONTENT_LENGTH’}) ;
    }
else {
  $query = $ENV{’QUERY_STRING’} ;
    }
foreach $pair (split(/&/,$query)){
  my ($key, $value) = split(/=/,$pair);
  $value=~tr/+// ;
  $value=~s/%([0-9a-fA-F]
          [0-9a-fA-F]) /chr(hex($1))/eg ;

$value = jcode : : sjis($value) ;
$value=~s/>/&gt ; /g ;
$value=~s/¥x0D¥x0A/ /g ;
$value=~tr/¥t// ;
$FORM{$key} = $value ;
  }
}
```

## 参考文献とソフトウェア


1）「FORTHと多数桁演算」，北海道医療大学情報センター年報，第2巻，22ページ，2004年.

2）ActivePerl-5.8.3.809-MSWin32-x86.msi

3）apache_2.0.49-win32-x86-no.ssl.msi

4）高橋大吾：「10日でおぼえるPerl/CGIプログラム入門教室」，翔泳社，2001年.

5）http : //www.jwdt.com/~paysan/gforth.html

# 32-bit FORTH in a Web Server

Ichiya SADAKATA*

**Abstract :** 32-bit FORTH in a DOS window, "forth32w.exe" and "forthex.exp", have been developed for several years by the author. "forth32w.exe" works in the DOS memory and executes such a task as the 1000-digit calculation of pi at high speed.

The purpose of the present study is to make 32-bit FORTH that works in a Web server. First, modifying "forth32w.exe", we made "forth32l.exe" that outputs html-tags when starting. Next, we made "forth32.cgi" that has the following perl's sentences :

```
#!c : /usr/bin/perl
$str=system("cmd /Cc : /forth/forth32l. exe testhtml. blk") ; exit ;
```

We can start "forth32.cgi" by inputting the following :

```
http : //localhost : 80/forth32.cgi
```

After starting, "forth32l.exe" loads source from "testhtml.blk" and executes it. This "forth32.cgi" is very simple. But, to execute a different source, it is necessary to rewrite "testhtml.blk".

To execute FORTH interactively, we have greatly renewed "forth32.cgi". The new "forth 32.cgi" works as follows :

(1) it opens a html-window that has a textarea and a "Run" button ;
(2) it waits for input of source (for instance, 1000 bnp-pi bnp.) ;
(3) after source is input and the "Run" button is clicked, "forth32.cgi" outputs the source into the work file "testhtml.blk". Then, "forth32l.exe" loads the source from "testhtml.blk" and executes it.
(4) it prints the output of FORTH in a new window and waits for close of the window.
(5) it goes to (2).

**Key words :** 32-bit FORTH, Web server, cgi-program, interactive execution

* Department of Integrated Human Sciences