

# 14章「CGI, SSI, FORM」

中島康彦

## §14. 1 CGIとSSI(サーバ内共通の設定)

`/usr/local/etc/httpd/`

```
▶ conf/srm.conf      ... 設定ファイル
  DirectoryIndex      index.html index.shtml index.htm index.cgi
  AddType              application/x-httpd-cgi .cgi(CGI)
  AddType              text/x-server-parsed-html .shtml(SSl)
```

- ▶ CGI(Common Gateway Interface)を使うファイルは, 拡張子を `.cgi` にする.
- ▶ SSI(Server Side Include)を使うファイルは, 拡張子を `.shtml` にする.

## §14. 2 CGIとは

---

ファイルの内容を表示するのではなく、ファイルの実行結果を表示する仕掛け

- ▶ `date.ccc` ... わざと間違えてみる

```
#!/bin/csh
echo 'Content-type: text/plain'
echo ''
date
```

- ▶ ファイルモードが `-rw-r--r--` の場合

```
% chmod 644 date.ccc
```

```
#!/bin/csh
echo 'Content-type: text/plain'
echo ''
date
```

- ▶ ファイルモードを `-rwxr-xr-x` としても同じ

```
% chmod 755 date.ccc
```

```
#!/bin/csh
echo 'Content-type: text/plain'
echo ''
date
```

---

## §14. 2 CGIとは(続き)

---

拡張子を `.cgi` とすることにより、CGI機能が有効となる。

- ▶ `date.cgi`

```
#!/bin/csh
echo 'Content-type: text/plain'
echo ''
date
```

- ▶ ファイルモードが `-rw-r--r--` の場合エラー

```
% chmod 644 date.cgi
```

```
403 Forbidden
```

```
Your client does not have permission to get URL /date.cgi from this server.
```

- ▶ ファイルモードを `-rwxr-xr-x` とすると動作する

```
% chmod 755 date.cgi
```

```
Sun Sep 17 09:40:00 JST 2000
```

---

## §14. 3 SSIとは

---

WEBサーバ側でファイルの実行結果をHtmlファイルに埋め込む仕掛け

- ▶ `date.html` ... わざと間違えてみる

```
<html>
  現在時刻は
  <font color=0000ff>
  <!--#exec cgi="date.cgi"-->
</font>
</html>
```

現在時刻は

---

## §14. 3 CGIとは(続き)

---

拡張子を `.shtml` とすると, SSI機能が有効となる.

- ▶ `date.shtml`

```
<html>
  現在時刻は
  <font color=0000ff>
  <!--#exec cgi="date.cgi"-->
</font>
</html>
```

- ▶ `date.cgi` が存在しない場合エラー

現在時刻は [an error occurred while processing this directive]

- ▶ `date.cgi` が存在しファイルモードが `-rwxr-xr-x` であれば正常に動作する.

現在時刻は Sun Sep 17 10:28:50 JST 2000

---

## §14. 4 応用 ... サーバ負荷の表示

```
/usr/bin/uptime
```

```
10:48AM up 2:08, 1 user, load averages: 0.02, 0.10, 0.10
```

```
/usr/local/etc/httpd/cgi-bin/local/load
```

```
#!/bin/csh
echo 'Content-type: text/plain'
echo ''
echo 'load<font color=ff0000>'
/usr/bin/uptime|sed 's/^. *averages: //g'
echo '</font>'
```

```
Content-type: text/plain
```

```
load<font color=ff0000>0.02, 0.10, 0.10</font>
```

loadを使ったHtml

```
<html>
<font color=0000ff><!--#exec cgi="/cgi-bin/local/load"--></font>
</html>
```

```
load0.07, 0.02, 0.03 /15%
```

---

## §14. 5 応用 ... ディスク使用率の表示

```
/bin/df
```

Filesystem	1K-blocks	Used	Avail	Capacity	Mounted on
/dev/da0sla	516062	70152	404626	15%	/
mfs:27	515598	10	515588	0%	/tmp
/dev/da0sle	4129310	1859552	1939414	49%	/usr
/dev/da0s1f	930509708	55966010	874543698	6%	/usr0
procfs	4	4	0	100%	/proc

```
/usr/local/etc/httpd/cgi-bin/local/df
```

```
#!/bin/csh
echo 'Content-type: text/plain'
echo ''
echo -n '/<font color=ff0000>'; /bin/df -k / |tail +2|sed 's/[ ][ ]*/ /g'|cut -d ' ' -f 5;echo '</font>'
echo -n '/tmp<font color=ff0000>'; /bin/df -k /tmp |tail +2|sed 's/[ ][ ]*/ /g'|cut -d ' ' -f 5;echo '</font>'
echo -n '/usr<font color=ff0000>'; /bin/df -k /usr |tail +2|sed 's/[ ][ ]*/ /g'|cut -d ' ' -f 5;echo '</font>'
echo -n '/usr0<font color=ff0000>';/bin/df -k /usr0|tail +2|sed 's/[ ][ ]*/ /g'|cut -d ' ' -f 5;echo '</font>'

Content-type: text/plain
/<font color=ff0000>15%</font>
/tmp<font color=ff0000>0%</font>
/usr<font color=ff0000>49%</font>
/usr0<font color=ff0000>6%</font>
```

dfを使ったHtml

```
<html>
<font color=0000ff><!--#exec cgi="/cgi-bin/local/df"--></font>
</html>
```

```
/tmp0% /usr49% /usr06%
```

## §14. 6 応用 ... 最終更新日の表示

---

変数LAST\_MODIFIED

変数を使ったHtml

```
<html>
更新<!--#echo var="LAST_MODIFIED"-->
</html>
```

```
更新Friday, 25-Feb-05 08:27:39 JST
```

---

## §14. 7 応用 ... アクセスカウンタの表示

---

ファイルのロック(lockf)

```
lockf [-t seconds] file command
ファイルを読み出し、1を加えて書き込むまでをロックする。
```

```
/usr/local/etc/httpd/cgi-bin/local/count
```

```
#!/bin/csh
echo 'Content-type: text/plain'
echo ''
lockf -t 10 /tmp/lockf csh -c '@ a= `cat countfile` + 1;echo $a | tee countfile'
```

ファイルcountをコピーする

```
% cp /usr/local/etc/httpd/cgi-bin/local/count .
```

ファイルcountfileを作ってから実行してみる

```
% echo 0 > countfile
% ./count
Content-type: text/plain
1
```

countを使ったHtml

```
<html>
参照<!--#exec cgi="count"-->回
</html>
```

参照1回

---

## §14. 8 FORMによるメール送信(Htmlファイル)

```
<form method="get" action="/cgi-bin/local/mail">
  <input name="name" size=10 maxlength=20 value="氏名">
  <input name="addr" size=10 maxlength=40 value="返信先">
  <select name="subject">
    <option value="message" selected>message
    <option value="request" >request</select>
  <input type="submit" value="send">
  <input type="reset" value="clear">
  <textarea name="comments" rows=2 cols=50></textarea>
</form>
```

### 入力例

```
name           = ユーザ
addr           = xxx@i.econ.kyoto-u.ac.jp
subject        = message
comments       = テスト
```

---

## §14. 8 FORMによるメール送信(cgi-bin/local/mail)

変数QUERY\_STRINGに以下が1行に格納されている

```
name=%A5%E6%A1%BC%A5%B6&addr=xxx%40i.econ.kyoto-u.ac.jp&
subject=message&comments=%A5%C6%A5%B9%A5%C8
```

この文字列からメールを作成する.

```
#!/bin/sh
echo 'Content-type: text/plain'
echo ''
name=`echo $QUERY_STRING|sed -e 's/^.*name=/' -e 's/&.*//'\`
addr=`echo $QUERY_STRING|sed -e 's/^.*addr=/' -e 's/&.*//'\`
subject=`echo $QUERY_STRING|sed -e 's/^.*subject=/' -e 's/&.*//'\`
comments=`echo $QUERY_STRING|sed -e 's/^.*comments=/' -e 's/&.*//'\`
echo 'Following mail has been sent.'
echo ''
(echo "From: $name <$addr>";echo "Subject: $subject";echo;echo $comments) | ./convert
(echo "From: $name <$addr>";echo "Subject: $subject";echo;echo $comments) | ./convert |
/usr/sbin/sendmail x@y.jp
```

```
From: ユーザ <xxx@i.econ.kyoto-u.ac.jp>
Subject: message
```

```
テスト
```

送信メールの画面表示, および, sendmailを使ったx@y.jp宛の送信 (本番では文字コードに留意)

## §14. 8 メール送信(cgi-bin/local/convert.c)

実はconvert前のメッセージは以下のようにになっている

```
From: %A5%E6%A1%BC%A5%B6 <xxx%40i.econ.kyoto-u.ac.jp>  
Subject: message
```

```
%A5%C6%A5%B9%A5%C8
```

この文字列から正しいメッセージを作成する.

```
main()  
{  
  int c, o;  
  int stat = 0;  
  while ((c = getchar()) != -1) {  
    if (stat == 0) {  
      if (c == '%') stat = 1;  
      else if (c == '+') putchar(' ');  
      else          putchar(c);  
    }  
    else if (stat == 1) {  
      stat = 2;  
      if (c >= '0' && c <= '9') o = c-'0';  
      else if (c >= 'A' && c <= 'F') o = 10+c-'A';  
    }  
    else {  
      stat = 0;  
      if (c >= '0' && c <= '9') putchar(o*16+(c-'0'));  
      else if (c >= 'A' && c <= 'F') putchar(o*16+(10+c-'A'));  
    }  
  }  
}
```

---

## §14. 9 他のURL(Goo)による検索

変数QUERY\_STRING中のMTフィールドに検索文字列を格納する.

```
<html>  
<form method="get" action="http://www.goo.ne.jp/default.asp?." target=frame3>  
  <a href=http://www.goo.ne.jp/>Goo</a>  
  <input name="MT" size=30 maxlength=60>  
  <input type="submit" value="go">  
  <input type="reset" value="clear">  
</form>  
</html>
```



The image shows a simple web search interface. On the left, there is a small logo that says "Goo". To its right is a rectangular text input field. Further to the right are two buttons: one labeled "go" and one labeled "clear".

## §14. 10 他のURL(Yahoo)による検索

変数QUERY\_STRING中のpフィールドに検索文字列を格納する。

```
<html>
<form method="get" action="http://search.yahoo.co.jp/bin/search?." target=frame3>
  <a href=http://www.yahoo.co.jp/>Yahoo</a>
  <input name="p" size=30 maxlength=60>
  <input type="submit" value="go">
  <input type="reset" value="clear">
</form>
</html>
```



The screenshot shows a search form with a text input field containing the word "Yahoo". To the right of the input field are two buttons: "go" and "clear".

今日はここまで