

# 計算機プログラミング'95 # 9

久野 靖\*

1995.11.2

## 1 WWW と CGI と Java

今回は講義としては最終回なので、少しお遊びほくて、しかし最新!の話題として WWW 関係のをやりましょう。

### 1.1 GIF イメージを生成する

まず、WWW ページに入れるイメージは GIF という形式なのですが、計算機科学基礎で学んだ PPM 形式をプログラムで生成すれば、ppmtogif というフィルタを使って GIF に変換できます。たとえば、「真っ赤」な長方形を作ってみましょう。(もちろん、xpaint で描いた方が簡単ですがあくまでもプログラミングの練習です。)

```
/* t91.c -- generate all-red ppm image */

main() {
    int w = 200;
    int h = 100;
    int i, j;
    printf("P3 %d %d 255\n", w, h);
    for(i = 0; i < w; ++i)
        for(j = 0; j < h; ++j)
            printf("255 0 0\n");
}
```

これを GIF に変換して xv で観察するには…

```
% gcc t91.c
% a.out | ppmtogif >red.gif
% xv red.gif &
```

**練習 1** そのまま動かせ

**練習 2** 何か面白い図柄を生成してみよ。たとえば日の丸とか縞模様とか。

**練習 3** さらに、どこかの色の RGB 値を引数として渡すようにしてみよ。つまり「solid 255 0 0」で真っ赤になるわけ。

---

\*筑波大学大学院経営システム科学専攻

## 1.2 WWW ページの作り方

計算機科学基礎でやりましたから、ごく簡単に。

- ホームディレクトリの下に WWW というサブディレクトリを作り、誰でもたどれるようにしておく ( 「mkdir WWW; chmod a+rx WWW」 ) 。
- その中に 「なんとか.html」 や 「なんとか.gif」 や 「なんとか.cgi」 というファイルを置く。
- 「chmod a+r ファイル名」 (実行させるプログラムは 「chmod a+rx ファイル名」 ) によって保護を外しておく。

ごく簡単なページの例。

```
<HTML>
<HEAD><TITLE>K-Pro '95 Page</TITLE></HEAD>
<BODY>
<H1><IMG SRC="red.gif">「計算機プログラミング」の練習ページです</H1>
```

```
<P>いくつか例題を作ってみました。</P>
```

```
<UL>
<LI> <A HREF="colors.html">いろんな色を見よう</A>
<LI> 自分で色を指定しよう
<LI> そのほか?
</UL>
</BODY>
</HTML>
```

- <名前>とか</名前>でいろんな指定 (タグ)。ペアのタグとペアでないタグがある。
- 全体は<HTML>...</HTML>。その中に<HEAD>...</HEAD>と<BODY>...</BODY>。ヘッダ部には<TITLE>...</TITLE>のみ。
- 本体部は<P>...</P>(段落)が基本。
- <H1>...</H1>はレベル 1 の見出し。レベル 6 までである。
- <UL>...</UL>は箇条書。その中の各項目は<LI>で始める。
- <IMG SRC="...">はそこにイメージを埋め込む。
- <A HREF="...">...</A>はリンクで、そこを選択すると別のページへいく。

## 1.3 CGI について

リンクの中でも、「なんとか.html」ではなく「なんとか.cgi」を指しているものはプログラムを実行させる。これを「CGI スクリプト」と呼ぶ。

CGI スクリプトの先頭では、必ず「出力データの MIME タイプ」を書き出さなければならない。そのため、だいたいシェルスクリプトとして用意する。

```
#!/bin/sh
PATH=/u1/kuno/WWW/:/usr/local/X11R6/bin:/usr/local/bin:$PATH
echo 'Content-type: image/gif'
echo ''
solid $* | ppmtogif 2>/dev/null
```

これを「solid.cgi」という名前にしておく。一方、ページの方は次の通り。

```
<HTML>
<HEAD><TITLE>K-Pro '95 Color Demo</TITLE></HEAD>
<BODY>
<H1>いろいろな色をお見せします</H1>

<UL>
<LI> <A HREF="solid.cgi?255+0+0">赤</A>
<LI> <A HREF="solid.cgi?0+255+0">緑</A>
<LI> <A HREF="solid.cgi?0+0+255">青</A>
</UL>
</BODY>
</HTML>
```

練習 4 自分のプログラムもこうやって動かすようにしてみよ。

## 1.4 フォーム

フォームというのは、記入欄のある書式のようなもの。HTML でもフォームを作れる。

```
<HTML>
<HEAD><TITLE>K-Pro '95 Form Demo</TITLE></HEAD>
<BODY>
<H1>あなたの好きな色をお見せします</H1>

<FORM METHOD=POST ACTION="formsolid.cgi">
<UL>
<LI> 赤: <INPUT TYPE="text" NAME="red">
<LI> 緑: <INPUT TYPE="text" NAME="green">
<LI> 青: <INPUT TYPE="text" NAME="blue">
</UL>
<P><INPUT TYPE="submit" VALUE="提出"></P>
</FORM>
</BODY>
</HTML>
```

ここで ACTION= で指定したプログラムがフォームのデータを受け取る。このデータを分解するのは自分でやってもよいが、cgiparse というプログラムを利用すると簡単。

```
#!/bin/sh
PATH=/u1/kuno/WWW/kpro95:/usr/local/X11R6/bin:/usr/local/bin:$PATH
eval `cgiparse -init`      ←初期設定
eval `cgiparse -form`     ←フォームのデータをシェル変数に格納
echo 'Content-type: image/gif'
echo ''
solid $FORM_red $FORM_green $FORM_blue | ppmtogif 2>/dev/null
```

練習 5 自分でもフォームを使って動かすことにせよ。

練習 6 フォームから「メッセージ」と「回数」を打ち込んでもらい、その回数だけメッセージを反復するページを生成してみよ (Content-type: text/html とする)。

練習 7 フォームから何かのデータを打ち込んでもらい、それに基づいて計算をして結果を返すページを作成してみよ。

## 1.5 Java

さて、これまでやってきた CGI では、ブラウザでデータを入力するとそのデータはサーバに転送され、サーバ側でプログラムが動いて結果がページやイメージなどの形で送り返されていた。それだと、対話性の高いものは限界がありますね？

そこで最近脚光を浴びているのが「Java」という言語。言語は C++ を整理したようなものだが、プログラムをバイトコードに翻訳し、HTML のページなどと一緒にブラウザに送り込む。ブラウザは Java のインタプリタを持っていて、送り込まれたプログラムを実行する。

これをやるには、ブラウザとして netscape を使う。smf/smd/msg でしか動かない。これらのマシンで「ns」というと起動される。たとえば「おもしろページ」の中の「JDK 1.0 prebeta」を選んでさらにそのページにある「examples」をたどると色々なデモがある。こういう、ページの中に埋め込まれた Java の領域のことを「アプレット」という。

ではさっそく、とりあえず、簡単な例題を見てみよう。

```
import java.awt.*;
import java.applet.AppletStub;

public class Demo1 extends java.applet.Applet {
    Color c1 = new Color(200, 200, 0);
    Color c2 = new Color(250, 0, 50);
    Color c3 = new Color(0, 100, 200);
    Color c4 = new Color(0, 200, 100);
    Color c0 = c1;
    public boolean mouseEnter(Event e, int x, int y) {
        c0 = c2; repaint(10); return true;
    }
    public boolean mouseExit(Event e, int x, int y) {
        c0 = c1; repaint(10); return true;
    }
    public boolean mouseDown(Event e, int x, int y) {
        c0 = c3; repaint(10); return true;
    }
    public boolean mouseUp(Event e, int x, int y) {
        c0 = c2; repaint(10); return true;
    }
    public boolean mouseDrag(Event e, int x, int y) {
        c0 = c4; repaint(10); return true;
    }
    public void paint(Graphics g) {
        g.setColor(c0); g.fillRect(0, 0, size().width-1, size().height-1);
        g.setColor(Color.black);
    }
}
```

```
        g.drawLine(0, 0, size().width-1, size().height-1);
    }
}
```

見ればだいたい分かると思うが、イベントの種類ごとに別のメソッドになっていて、それが自動的に（実は親クラスにあるハンドラから）呼び出されるという仕組みである。これを動かすには `javac` (Java コンパイラ) が必要だが、それは何と `smb` でしか動かない。`smb` の窓をあけて

```
PATH=$PATH:/usr/local/libproj/java/bin
```

を実行してから、上のような内容のファイルが `Demo1.java` に入っているものとして

```
javac Demo1.java
```

によりコンパイルする。コンパイルし終わると `Demo1.class` というファイルができている。これを見るためには HTML ファイルがいる。

```
<HTML>
<HEAD><TITLE>Java Demo 1</TITLE></HEAD>
<BODY>
<H3>Java のデモその 1</H3>
<HR>
<APPLET CODE="Demo1.class" width=200 height=100>
</APPLET>
<HR>
</BODY>
</HTML>
```

たったこれだけ。（なお `<HR>` は横線を引くタグ。）

**練習 7** このプログラムをそのまま動かせ。次に、マウスをドラグすることで直線が引けるようにしてみよ。

**課題 6** 何か面白い CGI スクリプトを作って動かせ。

**課題 7** 何か面白いアプレットを作って動かせ。