結果の出力方法とgnuplot（数学購読）

４月２５日

1　プログラム結果の出力

プログラムを作成するときには, その目的に応じて

- 何をどうやって計算し（アルゴリズムの決定）,
- データの入力，データの出力をどうするか.

について常に考えなければならない.　出力や入力はプログラムが行う計算アルゴリズムと同じくらい重要な要素であり, 適切な出力と入力のないプログラムはそれ自体で価値がない. そこで，今回はプログラムにおける計算結果の「出力」について学ぶことにする.（入力についてはもう少しいろいろ学んだ後に解説するので今回は見送る.）

一言にプログラムの「出力」といってもその方法は多様である.

- 計算結果を「数字・文字」として「画面」に出力する.
- 計算結果を「数字・文字」として「ファイル」に出力する.
- 計算結果を「数字・文字」として別の「プログラム」に出力する.
- 計算結果を「グラフ・表」として「画面」に出力する.
- 計算結果を「グラフ・表」として「ファイル」に出力する.

勿論これら以外にも，様々な出力方法や形態があるはずだが, 上記が標準的なものであろう.

今回の実習で，我々は**計算結果を「数字・文字」で「画面」や「ファイル」に出力する方法**について学ぶことにする. 計算結果を「グラフ・表」として出力するプログラムを作成するためには, もう少し高度な知識が必要な

ので, ここでは解説しない. その代わりに近年はファイルを「グラフ・表」として処理し計算するプログラムが市販アプリケーション (Excel など）で多数存在しており, 我々としてはそれらのアプリケーションで処理できる形式で, 結果をファイルに出力できさえすれば，それらを利用することで図や表にすることができる. 今回は図を作成するアプリケーションとして，**gnuplot** と呼ばれるコマンドを利用することとし，その方法について学ぶ.

2　C言語の printf 関数を使いこなす

C言語／ UNIX の体系を学ぶ上で, まず押さえておくべき概念は**標準入力・標準出力**という概念である. その言葉の通り, 標準入出力は UNIX が定める標準的な入力と出力である. 具体的にはコンソール「画面」にデータを出力するのが標準出力, コンソール「画面」からデータを入力するのが標準入力である.

前回の実習では詳しく解説せずに使ったが, C言語で用意されている printf 関数は「数字・文字」をコンソール「画面」（標準出力）に出力する関数である. その用法は基本的には man コマンドを利用して調べればよいが, 最近は Web ページでも詳しい解説がある. たとえば

```
http://www.linux.or.jp/JM/html/LDP_man-pages/man3/printf.3.html
```

などには詳しい解説がある. しかし, これを読んだだけでは，実感としてわかりずらいので prog2-1.c を入力してコンパイルして，その動作を見ることにしよう.

ステップ：prog2-1.c を入力してコンパイルし, 実行結果と Web ページの解説を比較せよ. また, このプログラムをいろいろ修正して, 様々な出力方法について動作を確認してみよ.

3　C言語プログラムにおけるファイル出力

次にプログラムの中で計算結果として得られた「数字・文字」を「ファイル」に出力する方法を学ぶ. まずは典型的な出力方法として, **標準出力をファイルに出力する**という方法がある. 例えば prog2-1 というコマンドの標準出力結果をファイル data.txt に出力する場合はコマンドラインから

```
prog2-1 > data.txt
```

とすればよい. これまで prog2-1 の結果は画面に出ていたが, この指定によって画面には何も出力されず, 出力はすべてファイル data.txt の中に出力される. 記号 > はリダイレクト出力とよばれ, 最も簡便で使い安い方法である.

さて, 標準出力は便利だが基本的には出力はすべて一つのファイルにしかできないという特徴を持つ. 一方でプログラムによっては計算結果を複数のファイルに別々に書き込みたい場合も多々ある. そこで, C 言語のプログラムの中でファイルへの出力を制御する関数がある. その代表的な関数は fopen, fprintf, fclose の三つであり, 以下でその使い方を学ぶ.

ファイルにデータを書くという作業は以下のようなステップで行われる.

1. ファイルを作成する，あるいはすでに存在するファイルを開く. (fopen 関数)
2. そのファイルにデータを出力する. (fprintf 関数)
3. ファイルを閉じる. (fclose 関数)

これらのそれぞれの方法についての説明は以下の Web ページに詳しい.

```
http://www.linux.or.jp/JM/html/LDP_man-pages/man3/fopen.3.html
http://www.linux.or.jp/JM/html/LDP_man-pages/man3/fprintf.3.html
http://www.linux.or.jp/JM/html/LDP_man-pages/man3/fclose.3.html
```

例によってここでは, prog2-2.c を入力しコンパイルすることによって, その働きについて学ぶことにする.

ステップ: prog2-2.c を入力してコンパイルし, 実行結果を見よ.

基本的には次のステップでファイルの読み書きが行われる.

- FILE 変数 ファイル作成のために必要なデータ. ファイルの読み書きに必要なデータがここに記述される. 変数を宣言（定義）する時は必ず

  ```
  FILE *fp;
  ```

  と先頭に*をつけて宣言する. （この*の意味はここでは理解しなくてもよい)

- **fopen 関数**: ファイルを開く. 基本的には次の書式に従う.

  ```
  fopen( ファイル名, ファイルのオープンモード  );
  ```

ファイル名 は入出力を行うファイルの名前を指定する.

ファイルのオープンモード はファイルをどのような目的で開くのかを指示する. 次のオプションがある.

"w" ... ファイルを書き込み (Write) モードで開く. このモードでファイルを開くとデータはすべて上書きされる.

"r" ... ファイルを読み込み (Read) モードで開く.

"a" ... ファイルを追加書き込み (Append) モードで開く. このモードでファイルを開くとすでにあるデータの最後から追加の書き込みを行う.

その他 **"r+"**, **"w+"**, **"a+"**などがあるが, ここでは説明しない. なお, この関数の戻り値は FILE 型の変数 (へのポインタ) なので,

```
FILE *fp;
fp = fopen( "output.dat", "w");
```

と書くことによって, output.dat という名前でオープンしたファイルの情報 (ファイルに書き込むデータそのものではなくて, ファイルアクセスに必要な名前やサイズなどの情報が fp に書き込まれている.)

もし fopen がファイルのオープンに失敗したら, fp には NULL という値が戻ってくるので, if 文によってファイルエラー処理を行う. (prog2-2.c を見よ)

- **fprintf 関数**: ファイルにデータを書き込む. 基本的な書式はほとんど printf 関数と同じだが, どのファイルに書き込むかを最初に指示しなければならない.
- **fclose 関数**: ファイルをクローズする. 作法としてオープンしたファイルは必ずクローズしなければ正しい結果が得られないことがあるので注意する.

4 gnuplot による図形表示

与えられた関数や, プログラムによって作成されたデータファイルを図やグラフとして出力するコマンドに gnuplot がある. 使い方は次の Web ページを参照するのが便利である.

http://lagendra.s.kanazawa-u.ac.jp/ogurisu/manuals/gnuplot-intro/
http://t16web.lanl.gov/Kawano/gnuplot/

なお, 最初のページは北大を卒業した先生が作成しているページである.

ステップ：最初の Web ページに従って gnuplot の様々なコードを実行せよ.

gnuplot を使ってデータを図示する場合は，gnuplot がどのような「ファイル形式」をどのように表示するかということを理解しておかねばならない．すなわちプログラムの中における printf や fprintf による「出力」方法は，gnuplot が受け付ける「入力」に合うように考えて決定しなければならない．

5　4 月 2 5 日課題

課題 1　関数 $y = \sin x$ を区間 $[0, 2\pi]$ で出力するプログラムを書き, そのデータを gnuplot の plot で表示してみよ.

課題 2　関数 $z = \sin x \cos y$ を領域 $(x, y) \in [0, 2\pi] \times [0, 2\pi]$ で計算し, その結果をファイル output2.dat に出力するプログラムを書き, そのファイルを使って gnuplot の splot で曲面表示せよ.

6　この実習で用いるソースコード

```
/* prog2-1.c printf 関数の出力 */
#include <stdio.h>
int main( int argc, char *argv[] )
{
    double a,b,c;
    int i,j,k;

    /*  文字列は単にその文字を書けばよい */
    printf( "This is a pen.\n" );

    /*  整数型データの表示  %d   */
    i=1000;
    j=-10;
    k = 45;
```

```
    printf( "i = %d\n", i );
    printf( "i = %8d\n", i );
    printf( "i = %08d\n", i );
    printf( "j = %+8d\n", j );
    printf( "j = %+08d\n", j );
    printf( "k = %u\n", k );
    printf( "k = %x\n", k );
    printf( "j = %x\n", j );

    /* 実数型データの表示  %f, %e */
    a = 0.1;
    b = -1.0e-1;
    c = 123456.014289;

    printf( "a = %f\n", a );
    printf( "a = %+f\n", a );
    printf( "b = %+f\n", b );
    printf( "a = %lf\n", a );
    printf( "a = %.3f\n", a );
    printf( "a = %e\n", a );
    printf( "a = %.8e\n", a );
    printf( "a = %+e, b= %+e\n", a, b );
    printf( "a = %x\n", a );
    printf( "a = %+x\n", a );
    exit(0);
}

/* prog2-2.c のプログラム */
#include <stdio.h>
#include <math.h>
int main(int argc, char *argv[] )
{
    int i,n;
    double x[200],y[200];
    FILE *fp;
    /*output.dat という名前のファイルを書き込みモードでオープン */
    fp = fopen( "output.dat", "w" );
    if( fp==NULL ){
```

```
      printf( "File open fails\n" );
      exit(1);
    }

    n = 100;
    for(i=0;i<n;i++){
      x[i] = (double)i/(double)n;
      y[i] = sin(x[i]);
    }
    for(i=0;i<n;i++){
      fprintf( fp, "%10.8lf %10.8lf\n", x[i], y[i] );
    }
    fclose( fp );
    exit(0);
}
```