

配列とファイル操作の練習 (その1)

山本昌志*

2004年11月12日

1 配列とファイル処理の復習

1.1 配列

文字や数字等のデータを処理することがコンピューターの仕事と考えることができる。処理すべきデータは、全て、コンピューターのメモリー¹に記憶しなくてはならない。それにアクセスするためには、プログラマーはその記憶場所に名前を付けなくてはならない。加えて、C言語の場合²、効率よくデータを扱う³ためにデータの型も指定する必要がある。最初に学習した単純型のデータ構造の場合、

```
int a;  
double x;
```

のように宣言すると、

- a と名付けられた整数型のデータ領域が一つ用意される。
- x と名付けられた倍精度実数型のデータ領域が一つ用意される。

となる。このデータ構造では、変数名、たとえば a や w を指定することで、その領域のデータを入出力できる。

単純型のデータ構造の場合、一度に確保できるメモリーの領域は1個なので、大量のデータを扱うのは不向きである。そこで、大量のデータを扱うために、配列というデータ構造が考えられた。

これは、同じ型のデータを任意の個数宣言し、配列名と自然数⁴でアクセスすることができ、便利である。配列を使うためには、

```
int b[10000];  
double y[10000];
```

*国立秋田工業高等専門学校 電気情報工学科

¹主記憶 (メインメモリー) と呼ばれ、直接 CPU がアクセスする。プログラムも実行時ここに格納される。

²C言語に限らず多くのプログラミング言語で、変数の型の宣言は必要である

³コンパイラーやハードウェアの都合である。

⁴ここでは、0も自然数に含める。

のように宣言をする。こうすると、

- 配列名 `b` の整数型のデータ領域が 10000 個用意される。用意されるデータ領域は、`b[0] ~ b[9999]` である。
- 配列名 `y` の倍精度実数型のデータ領域が 10000 個用意される。用意されるデータ領域は、`y[0] ~ y[9999]` である。。

となる。このデータ構造では、配列名と添え字 (インデックス)、たとえば `b[1234]` や `y[45]` を指定することで、その領域から値を入出力できる。

配列型のデータを取り扱う場合、繰り返し文とともに使われることが多い。次の例のようにである。

```
for(i=0; i<=9999;i++){
    y[i]=3.1415*b[i]
}
```

添え字が 1 つのものを一次元配列と言い、それ以上のものを多次元配列と言う。C 言語では多次元配列を使う場合、

```
int hoge_1[100], hoge_2[100][100], hoge_3[100][100][100];
double huga[10], huge[10][10], hugo[10][10][10];
```

のように宣言を行う。これらも、配列名と複数の添え字で、そこにあるデータにアクセスする事ができる。3次元以上ももちろん可能である。

当面、変数 (データ構造) の宣言は、`main` 関数の最初を書くことと憶えて欲しい。`main` 関数の途中では宣言しない。実際は、他の場所でも宣言できるのであるが、それはより進んだ場面で学習する。

1.2 ファイル操作

コンピューターで大量のデータを操作する場合、ハードディスク上のファイルの取り扱いが必須である。ハードディスク上のファイルを取り扱うプログラムは簡単で、

1. FILE 型の変数、ファイルポインタの用意
2. ファイルのオープン
3. ファイルの読み書き
4. ファイルのクローズ

を記述すれば良い。

1.2.1 ファイル出力

簡単な例として、次の

- ファイル名は、"test_out.txt"とする。
- そこに、2004 と整数を書き込む。

ようなハードディスクにデータを書き込むプログラムを示す。このプログラムは、以下のように書く。

```
#include <stdio.h>

int main(void)
{
    FILE *fp_write;

    fp_write = fopen("test_out.txt","w");

    fprintf(fp_write,"%d", 2004);

    fclose(fp_write);

    return 0;
}
```

簡単である。ファイルポインタの宣言とオープンとクローズはおまじないと思えば良い。実際に、ファイルにデータを書き出す部分は、fprintf 関数を使う。これは、ディスプレイに出力する printf 関数とほとんど同じである。

1.2.2 ファイル入力

次の例は、ハードディスクからデータを読み込むプログラムである。

- 読み込むファイルは、1.2.1 節で作った"test_out.txt" とする。
- ファイルに書かれている整数を書き込み、変数 a に格納する。

このプログラムは、以下のように書く。

```
#include <stdio.h>

int main(void)
{
```

```
FILE *fp_read;
int a;

fp_read = fopen("test_out.txt","r");

fscanf(fp_read,"%d", &a);

fclose(fp_read);

return 0;
}
```

難しいことは何も無い。おまじなの部分は、データ出力と同じである。ファイルからデータを読み出す部分は、`fscanf` 関数を使う。これは、キーボードから読み込む `scanf` 関数とほとんど同じである。

1.3 エラー処理付きファイルオープン

実用的なプログラムでは、エラー処理は必須である。ファイル操作のプログラムの場合、ファイルが無い等の理由でオープンできない場合がある。このようなときには、エラー処理として

```
if((fp_read = fopen("test_out.txt","r"))==NULL){
    printf("ファイルが開けません\n");
    return 1;
}
```

と書く。この例は、ファイル読み込みの場合であるが、書き込みの場合も同じようにエラー処理を書く。これはファイル処理をする場合のオープンのパターンと覚えておいて欲しい。

2 本日の学習内容

本日と来週は、配列とファイルの取り扱いについて、学習する。今まで学習してきたことをよく思い出して、プログラムを作成してください。本日は、ヒントを与えないので、今まで配布したプリントと教科書をよく見て、チャレンジしてください。自分で調べ、考え、プログラムを書き、テストしてエラーを取り除く作業を繰り返すことにより、プログラミングの技術は上達する。

3 練習問題 (プログラム作成)

これから、練習問題を与える。今週と来週で全のプログラムを作成せよ。

3.1 ファイル出力

1.2.1 節に示したファイル出力のプログラムを、以下のように改造せよ。

- オープンの時のエラー処理を追加せよ。
- for 文を用いて、ファイルに 1~10000 までの整数を書き出せ。

プログラムができあがったならば、コンパイル・実行を行い、できあがったファイルの内容を確認せよ。

3.2 ファイル入力

1.2.2 節に示したファイル出力のプログラムを、以下のように改造せよ。

- オープンの時のエラー処理を追加せよ。
- ファイル (/tmp/1e/exercise2.txt) に 10000 の実数がかかれていいる。このデータを倍精度実数型の配列に格納せよ。
- ファイルのフォーマットが分からない場合は、ファイルを見よ。
- そして、その合計値をディスプレイに表示させよ。

プログラムができあがったならば、コンパイル・実行を行うこと。

3.3 温度のデータ処理

前回のプリントで示した次のプログラムを作成せよ。

- 11月の毎日の1時間毎の気温のデータファイル (/tmp/1e/temperature.txt) がある。
 - 各行には、その日の1時間毎の24個のデータがある。0時~23時までである。
 - 行数は30行で、11月1日から11月30日を表している。
- 日毎の最高気温と最低気温、平均気温をディスプレイに書き出す。
- 11月の最高気温と最低気温、平均気温をディスプレイに書き出す。

3.4 数値ファイルの清書

教科書、p.186のLesson 5-1のプログラムを作成せよ。

- 問題で与えられるべきファイルは、/tmp/1e/read1.txtにある。
- 結果のファイル (result1.txt) は、自分のディレクトリー、できればプログラムを実行したディレクトリーに保管すること。

プログラムができあがったならば、コンパイル・実行を行い、出来上がったファイルを確認すること。

Lesson 5-1

あるファイル (read1.txt) には、整数値のデータ (0 以上、最大 9999) が次のように、1 行に 8 つ、4 行書かれている。

【read1.txt】

```
31 124 1221 3 8 9 14 100
9 109 601 1200 9 120 910
412 81 8 132 317 32 2000 3000
3000 1 2526 7832 2311 4874 2127 8921
```

⇒

【result1.txt】

```
31 124 1221 ...
9 109 601 ...
412 81 8 ...
3000 1 2526 ...
-----
3452 315 4356 ...
```

このデータを縦の列で数値が右揃えになるようにし、5 行目に「-」で線を引き、6 行目に合計値が書き込まれているファイル (result.txt) を作成するプログラムを作りなさい。

3.5 アクセスカウンター

教科書、p.186 の Lesson 5-2 のプログラムを作成せよ。

- アクセス数を記述したファイルは、/tmp/1e/up.log にある。

プログラムができあがったならば、コンパイル・実行を行い、アクセス数の表示を見よ。

Lesson 5-2

ホームページを閲覧しているときに、アクセスカウンターと呼ばれる「そのページが何回閲覧されたか」を表示する仕組みがある。これは、ホームページにアクセスしたときにあるプログラムが動くようになっていて、そのプログラムが何回実行されたかを調べている。すなわち、実行回数をファイルに保存しておき、プログラムを実行するときにはそのファイルを読み込んで実行回数に 1 を加えてまた保存するという行っているわけだ。

このようなときに利用できるアクセスカウンタープログラムの基本として、プログラムの実行回数をカウントし、「 回目の実行です」と表示するプログラムを作成しなさい。

3.6 総合応用問題

教科書、p.186 の Lesson 5-3 のプログラムを作成せよ。

あるライブコンサートは全席指定で、

A 席	(0～9 の 10 席)	5000 円	(税込み)
B 席	(10～29 の 20 席)	4000 円	(税込み)
C 席	(30～79 の 50 席)	3700 円	(税込み)

の 3 種類の席が用意されている。このチケットの予約システムの基本プログラムを作りなさい。
プログラムは実行することに

- 現在の空席状況 (合計の空席数と空席の番号) が表示され、
- 座席番号を入力すると予約を行い、金額を表示する

という処理を行う。

2 回目以降に実行したときには、前の予約が反映されていなくてはならない。

このような座席予約システムのアルゴリズムを考え、実際にプログラムを完成させなさい。

4 レポート

4.1 内容

このプリントの練習問題のプログラムを全て完成させること。実行の確認ができたならば、それをプリントアウトして、レポートとして提出する。

自宅のプログラムを作っても良い。そして、実行の確認がとれたならば、それをプリントアウトし、レポートとして提出しても良い。

4.2 レポート 提出要領

提出方法は、次の通りとする。

期限	11 月 26 日 (金)PM5:00 まで
用紙	A4
提出場所	山本研究室の入口のポスト
表紙	表紙を 1 枚つけて、以下の項目を分かりやすく記述すること。 授業科目名「情報処理 I」 課題名「課題 配列とファイル処理」 1E 学籍番号 氏名 提出日
内容	ソースプログラム (プリントアウトのみ、手書きは不可)