

プログラミング言語基礎

[その1 2]

1. ファイル操作 fopen関数, fclose関数.
2. ファイルからデータを読み込むプログラムを作る
 - fscanf 関数
 - 文字列
3. ファイルにデータを書き込むプログラムを作る
 - fprintf 関数

http://www.ipc.fukushima-u.ac.jp/~p058
 福島県立医科大学 物理 吉田 宏
 E-Mail: yoshidah@fmu.ac.jp

2004/7/6

プログラミング言語基礎(その1 2)

1

1. ファイル操作 ファイル操作とは...

データをファイルから読み込む

- ① ファイルを開く
- ②-1 ファイルを読み込む
- ③ ファイルを閉じる

- ① fopen関数
- ②-1 fscanf関数
- ③ fclose関数

データをファイルへ書き込む

- ① ファイルを開く
- ②-2 ファイルに書き込む
- ③ ファイルを閉じる

- ① fopen関数
- ②-2 fprintf関数
- ③ fclose関数

2004/7/6

プログラミング言語基礎(その1 2)

2

①ファイルを開く

fopen関数

FILE *fin; /* 読み込むファイルをメモリー内に確保する
 為に、ファイルポインタを宣言する */

```
fin=fopen("xxxxx.dat","r");
/* ファイルxxxxx.datをオープンモードOpMで
   ファイルポインタに取り込む */
```

ファイル名

オープンモード
 r ファイルを読み込むとき
 w ファイルに書き込むとき (上書き)
 a ファイルに追加するとき (挿入)

2004/7/6

プログラミング言語基礎(その1 2)

3

③ファイルを閉じる

fclose関数

fclose(ファイルポインタ);

```
FILE *fin,*fout;
fin=fopen("input.dat","r");
fout= fopen("output.dat","w")
.....
fclose(fin);
fclose(fout);
```

fopen関数とfclose関数は対で使う

2004/7/6

プログラミング言語基礎(その1 2)

4

2.ファイルからデータを読み込む

fscanf関数

使い方は基本的にscanfと同じ

fscanf(ファイルポインタ, 入力書式, 入力変数)

cf. scanf(入力書式, 入力変数)

```
fin=fopen("data","r");
if(fin==NULL) {
    printf("ファイルが見つかりません\n");
    exit(1);
}
fscanf(fin,"%d %s %d %d %lf",&a,&s,&b,&c,&x);
```

↑ 文字列を読み込むとき

↑ 文字列

2004/7/6

プログラミング言語基礎(その1 2)

5

②-1 ファイルを読み込む

fopenを使って読み込みモードでファイルを開こうとしたとき
 指定したファイルがない!

```
fin=fopen("data","r");
```

dataというファイルがないとき
 finに"NULL"が代入される

プログラムをexit関
 数で強制終了させる

```
if(fin==NULL) {
    printf("ファイルがない");
    exit(1);
}
```

このようにしておくこと安心してプログラムを実行できる

Try 課題5.1&5.2

2004/7/6

プログラミング言語基礎(その1 2)

6

```
#define N 5 tmp.dat
```

整数型 文字型 整数型 整数型 実数型
 num[N] name[N][20] kokugo[N] sansuu[N] hyouka[N]

1	浅井清和	56	45	2.3
2	井上ひかる	45	66	2.5
3	宇野泰三	91	21	4.1
4	遠藤芳美	54	25	5.4
5	岡村文雄	87	41	7.3

↑ iが0からN-1になるまで繰り返す

```
fscanf(fin,"%d %s %d %d %lf",&num[i],name[i],
    &kokugo[i],&sansuu[i],&hyouka[i]);
```

2004/7/6

プログラミング言語基礎(その1 2)

7

文字列

数値の配列と同様に、各成分に1つの半角文字が格納されて
 いるもの。

```
char s[15]; /* 文字列の宣言 */
```

意味: s[0], s[1], ..., s[14]の文字が宣言された。文
 字列の最後の成分には必ず'0'という記号が入力されるの
 で、15個分の文字を確保するときは、

```
char s[15+1]; /* 15文字までの入力が可能 */
```

とする。各文字には1バイトのメモリーサイズが確保され
 る。日本語入力の場合は2バイト必要。

```
char s[15]="mojiretsu";
s[0] s[1] s[2] s[3] s[4] s[5] s[6] s[7] s[8] s[9] s[10] s[11] s[12] s[13] s[14]
m o j i r e t s u 0
```

2004/7/6

プログラミング言語基礎(その1 2)

8

ex0501.c で使われている配列

- ① char name [N] [20];
- ② int num [N], kokugo [N], sansuu [N];
- ③ double hyouka [N];

- ① 20個の文字の並びを確保する文字列 name をN個宣言。
- ② N個の成分を持つ整数型のベクトル num, kokugo, sansuuを宣言。
- ③ N個の成分を持つ倍精度実数型のベクトル hyoukaを宣言。

2004/7/6

プログラミング言語基礎(その1 2)

9

3.ファイルにデータを書き込む

fprintf関数

使い方は基本的にprintfと同じ
fprintf(ファイルポインタ,出力書式,出力変数)

cf. printf(入力書式,入力変数)

```
FILE *fout;

if((fout=fopen("result","r"))!=NULL){
    printf("同名のファイルがあります。¥n");
    fclose(fout);
    exit(1);
} else fclose(fout);
fout=fopen("result","w");
/* または fout=fopen("result","a"); */
fprintf(fout,"%d %s %d %d %f", a,s,b,c,x);
```

Try 課題5.3

2004/7/6

プログラミング言語基礎(その1 2)

10

文字列

②-2 ファイルに書き込む

fopenを使って書き込みモードでファイルを開こうとしたとき
指定したファイルがない!

fout=fopen("result","w") ← 上書きモード

resultというファイルがないとき
新たにresultというファイルが作成される

fopenを使って書き込みモードでファイルを開こうとしたとき
指定したファイルが既にある

以前のデータが消える

消えると困るとき

fout=fopen("result","a") ← 追加モードにする

2004/7/6

プログラミング言語基礎(その1 2)

11

書き込もうとしたファイルが 既にあるときの対処法②

fout=fopen("result","r") ← 読み込みモード

```
if(fout!=NULL){
    printf("既に同名のファイルがあります。¥n");
    fclose(fout);
    exit(1);
} else fclose(fout);
```

ファイルが既にあるか
どうかを確認する

```
fout=fopen("result","w");
.....
.....
```

2004/7/6

プログラミング言語基礎(その1 2)

12

データを読み込んで、計算をして、 結果を出力するプログラム

1. データファイル1を開く	データの読み込み	f1=fopen("ファイル1","r");
2. データを読み込む	読み込み	fscanf(f1,"書式",.....);
3. データファイル1を閉じる		fclose(f1);
4. 各データに対して計算をする		kekka[i]=T[i]*T[i]/(R[i]*R[i]*R[i]);
5. 計算結果を保存するファイル2を開く		f2=fopen("ファイル2","w");
6. 計算結果を書き込む	データの書き込み	fprintf(f2,"書式",.....);
7. ファイル2を閉じる	書き込み	fclose(f2);

2004/7/6

プログラミング言語基礎(その1 2)

13

planet.dat

惑星	公転周期	軌道半径
水星	0.240850	0.371
金星	0.615210	0.724
地球	1.000040	1.000
火星	1.880890	1.508
木星	11.862230	5.193
土星	29.457720	9.516
天王星	84.013000	19.140
海王星	164.790000	30.080
冥王星	248.400000	36.780

① 文字列title[0] 文字列title[1] 文字列title[2]
② 文字列 p[i] 実数 T[i] 実数 R[i]

同じパターン

繰り返しの制御文

planet.dat の読み込み例

```
#define N 9

FILE *fin,*fout;
char title[3][15],p1[N][10];
double T[N],R[N],kekka[N];
int i;

fin=fopen("planet.dat","r");

1 fscanf(fin,"%s %s %s", /* 各列のタイトルの読み込み */
2 title[0],title[1],title[2]);

for(i=0;i<N;i++){
    fscanf(fin,"%s %lf %lf",p1[i],&T[i],&R[i]);
} /* データの読み込み */
```

2004/7/6

プログラミング言語基礎(その1 2)

15

Try 課題5.4