

第9回

3. ファイルとその応用

外部記憶装置に記録されたプログラムやデータを、ファイルと呼ぶ。

シーケンシャルファイルやランダムファイルへのデータの記録や読み出し、更新の手順について学習する。

(1) ファイルとレコード	ファイル	複数の関連したデータを一つに集めたり プログラムを外部記憶装置に保存したもの
	レコード	ファイルを構成する一塊のデータ ex. 個人カード
	フィールド	レコードを構成する個別の要素 ex. 氏名, 住所, 電話番号等

cf. ファイルという概念

ディスクファイル、画面、キーボード、ポート、テープ上のファイル等、入出力機能を提供するデバイスはすべてファイル・システム上のファイルである。

ストリームという概念

プログラマとファイル(入出力デバイス)の間にある論理的インタフェース。ストリームは「開く操作」によってファイルに結び付けられ、「閉じる操作」によってファイルから切り離される。

ファイル処理にはどんな内容があるか

- | | |
|------------|-------------------------|
| ①ファイルの作成 | 集められたデータをまとめて記録する |
| ②レコードの取り出し | ファイル内の必要なレコードを取り出す |
| ③レコードの更新 | ファイル内の特定レコードに変更を加え、書き込む |
| ④レコードの追加 | 新しいデータを新しいレコードとして追加する |
| ⑤レコードの削除 | 不用になったレコードをファイルから削除する |

cf. 削除の方法

- ・対象となるレコード内のデータの消去
- ・対象となるレコードそのものの削除
- ・削除を示すマークをつける

(2) ファイルのアクセス方法

《シーケンシャルファイル》

データが領域の先頭から順番に記録されているファイルのこと。

これらのデータを取り出すには、先頭から順番に取り出すことしかできない。

ex. 個人カード（レコード）が可変長で記録されているファイル

・シーケンシャルファイルの処理手順

①ファイル・ポインタの宣言

↓

②ファイルのオープン

↓

③ファイル処理

↓

④ファイルのクローズ

①ファイル・ポインタを宣言する。

```
FILE *ファイル・ポインタ
```

```
ex. FILE *fp;
```

注) stdio.h の中で、一般的なファイルの入出力のために FILE 型という構造体が定義されている。FILE 型は、サイズ、ファイルの現在位置、読み書きのどちらをされるのかといったアクセスモード、エラーあるいはファイル終端 (EOF) が起きたのかといった、ファイルに関するさまざまな情報を格納している。fp は FILE に対するポインタであり、ファイルを開くための fopen 関数は FILE 型へのポインタを返すことを意味している。

②ファイルを開く。

```
ファイル・ポインタ = fopen("ファイル名", "モード");
```

```
ex. fp = fopen("name.txt", "w");
```

ファイルオープン時のモード

モード	内 容	注 意
r	読み込みモード	ファイルがないとエラーとなる
w	書き込みモード	すでに同名のファイルがあれば重ね書きする。 なければ、新たにファイルをつくる。
a	追加モード	すでに同名のファイルがあれば、最後に追加する。 なければ、新たにファイルをつくる。
モードのあとに b を付けると、バイナリファイルが対象となる。 モードのあとに + を付けると、読み書き可能となる。		

③必要なファイル処理を行う。

基本的なファイル処理は、ファイルからの読み込み、ファイルへの書き込みである。

ファイル入出力関数

	内 容	内 容
(1)	fgetc(ファイル・ポインタ)	ファイルからの1文字読み込み
(2)	fgets(配列名, 最大文字数, ファイル・ポインタ)	ファイルからの1行読み込み
(3)	fscanf(ファイル・ポインタ, "入力フォーマット", 引数...)	ファイルからのフォーマット付き入力
(4)	fputc(文字, ファイル・ポインタ)	ファイルへの1文字書き込み
(5)	fputs(配列名, ファイル・ポインタ)	ファイルへの1行書き込み
(6)	fprintf(ファイル・ポインタ, "出力フォーマット", 引数...)	ファイルへのフォーマット付き出力

例(1) `int c;`

`c = fgetc(fp1);`

例(2) `char a[10];`

`fgets(a, sizeof a, fp1);`

例(3) `int i, j;`

`fscanf(fp1, "%d %d", &i, &j);`

例(4) `int c;`

`fputc(c, fp1);`

例(5) `char a[10];`

`fputs(a, fp1);`

例(6) `int i, j;`

`fprintf(fp1, "%d %d", i, j);`

④ファイルをクローズする。

```
fclose(ファイル・ポインタ);
ex. fclose(fp);
```

例題 3-3

出力結果 3-3 のように、生徒の氏名をキーボードから入力し、ファイルに書き込むプログラムをつくりなさい。入力終了は、Ctrl-d としなさい。

▼出力結果 3-3

データを入力してください。終了は、Ctrl-d

```
Kyoko Ishida↵
Ichiro Suzuki↵
Yoshiko Tanaka↵
Makoto Sasaki↵
Yoko Kobayashi↵
Ctrl-d
```

考え方

ファイル・ポインタを `fp1`、ファイル名を `name.txt` とし、`fopen` 関数においてモードを `"w"` とすることに注意する。文字列の入力には書式を指定して `scanf` 関数を用いる。

注) ファイルのオープンに失敗したときのエラー処理

C 言語では指定したファイルがなくてもエラーメッセージを出さない。そのため、プログラムでチェックする必要がある。以下に例を示す。

```
if ((fp1 = fopen("name.txt", "w")) == NULL) {
    printf("ファイルがオープンできませんでした\n");
    exit(1);
}
```

ここでは、fopen 関数はファイルがオープンできなかった場合に NULL を返すことを利用してエラー処理を行っている。exit(1)はそのときの処理を終了する関数であり、stdlib.h に収録されている。このため、プログラム先頭で

```
#include <stdlib.h>
```

と記述する必要がある。

《関数 exit() について》

関数 exit() は、C 言語でプログラムの実行を強制終了するとき呼び出す。この関数は stdlib.h で次のように宣言されている。

```
void exit(int status);
```

int 型引数の status の値は、慣習的には、0 という値が返されればすべて正常であり、0 以外の値が返されれば何らかの異常な状態が起こったことを示す。

▼プログラム 3-3

```
01     /* E3-3 */
02     /*シーケンシャルファイルへのデータの記録*/
03     #include <stdio.h>
04     #include <stdlib.h>
05
06     main()
07     {
08         FILE *fp1;
09         char last_name[15], first_name[15];
10
11         if ((fp1 = fopen("name.txt", "w")) == NULL) {
12             printf("ファイルがオープンできません。¥n");
13             exit(1);
14         }
15         printf("データを入力してください。終了は、Ctrl-d¥n");
16         while (scanf("%s%s", first_name, last_name) != EOF) {
17             fprintf(fp1, "%s %s", first_name, last_name);
18             fputc('¥n', fp1);
19         }
20         fclose(fp1);
21     }
```

練習問題 27

27-1. 1 から 10 までの数を, `fprintf` 関数を用いてファイル `number.txt` に書き込むプログラムをつくりなさい。但し, 各数字のあとに改行 '`\n`' を挿入すること。

27-2. 例題 3-3 で作成したファイル `name.txt` に自分の名前を追加するプログラムをつくりなさい。

《ヒント》ファイルの最後にデータを追加する場合には, `fopen` 関数によってファイルをオープンするときにモードを "`a`" にすればよい。

例題 3-4

例題 3-3 で作成したファイル `name.txt` を読み込んで、画面に表示するプログラムをつくりなさい。

▼出力結果 3-4

データを画面に表示します。

Kyoko Ishida

Ichiro Suzuki

Yoshiko Tanaka

Makoto Sasaki

Yoko Kobayashi

考え方

ファイル名を `name.txt` とし、`fopen` 関数においてモードを `"r"` とすることに注意する。

▼プログラム 3-4

```
01     /* E3-4 */
02     /*シーケンシャルファイルからのデータの読み出し*/
03     #include <stdio.h>
04     #include <stdlib.h>
05
06     main()
07     {
08         FILE *fp1;
09         char last_name[15], first_name[15];
10
11         if ((fp1 = fopen("name.txt", "r")) == NULL) {
12             printf("ファイルがオープンできません。 ¥n");
13             exit(1);
14         }
15
16         printf("データを画面に表示します。 ¥n");
17         while (fscanf(fp1, "%s%s", first_name, last_name) != EOF)
18             printf("%s %s¥n", first_name, last_name);
19         fclose(fp1);
20     }
```

練習問題 2 7

2 7-3. 練習問題 2 7-1 で作成したファイル number.txt を読み込んで、画面に表示するプログラムを fscanf 関数を用いてつくりなさい。

《ランダムファイル》

ファイルのサイズに関係なく、ファイルを構成するすべてのレコードが同じ容量であり、ファイルの途中にあるデータを直接読み書きできるファイルのこと。データの中で容量が最も大きいデータが書き込めるように、レコードの容量を決定する必要がある。あとから、レコードの容量を変更することはできない。

ex. 個人カード（レコード）が固定長で記録されているファイル

・ランダムファイルの処理手順

基本的にはシーケンシャルファイルと同様。

但し、ランダムファイル処理を行うには、ファイルの中のデータの位置を指定する必要がある。C 言語では、`fseek` 関数を利用してランダム処理ができる。

`fseek` 関数の書式は次の通り。

`fseek(ファイル・ポインタ, 移動量, 移動原点);`

移動量 : 移動量は `long` 型である

移動原点 : `SEEK_SET` 先頭から

`SEEK_CUR` 現在の位置から

`SEEK_END` 最後から

例題 3-5

出力結果 3-5 のように、生徒の氏名をキーボードから入力し、1 レコード 30 バイト (姓に 15 バイト、名に 15 バイト) のランダムファイルに書き込むプログラムをつくりなさい。

▼出力結果 3-5

データを入力してください。終了は、Ctrl-d

Kyoko Ishida↵

Ichiro Suzuki↵

Yoshiko Tanaka↵

Makoto Sasaki↵

Yoko Kobayashi↵

Ctrl-d

考え方

ファイル・ポインタを `fp1`、ファイル名を `name2.txt` とし、`fopen` 関数においてモードを `"w"` とする。また、データ読み込みは `scanf` 関数、書き込みの際は位置指定に `fseek` 関数、データ書き込みには書式を指定する `fprintf` 関数を用いる。

▼プログラム 3-5

```
01     /* E3-5 */
02     /*ランダムファイルへのデータの記録*/
03     #include <stdio.h>
04     #include <stdlib.h>
05
06     main()
07     {
08         FILE *fp1;
09         char first_name[15], last_name[15];
10         int i = 0;
11
12         if ((fp1 = fopen("name2.txt", "w")) == NULL) {
13             printf("ファイルがオープンできません。¥n");
14             exit(1);
15         }
16
17         printf("データを入力して下さい。最後は、Ctrl-d¥n");
18         while (scanf("%s%s", first_name, last_name) != EOF) {
19             fseek(fp1, i * 30L, SEEK_SET);
20             fprintf(fp1, "%15s%15s", first_name, last_name);
21             i++;
22         }
23         fclose(fp1);
24     }
```

例題 3-6

例題 3-5 において作成したデータファイル `name2.txt` の先頭から任意の位置のデータを表示するプログラムをつくりなさい。

▼出力結果 3-6

最初から何番目のデータを表示しますか (0 で終了) ? 3 ↵

データは Yoshiko Tanaka です。

最初から何番目のデータを表示しますか (0 で終了) ? 1 ↵

データは Kyoko Ishida です。

最初から何番目のデータを表示しますか (0 で終了) ? 0 ↵

考え方

ファイル・ポインタを `fseek` 関数によって読み込みたい場所へ移動し, `fscanf` 関数によって読み込む。

▼プログラム 3-6

```
01     /* E3-6 */
02     /*ランダムファイルからのデータの読み出し*/
03     #include <stdio.h>
04     #include <stdlib.h>
05
06     main()
07     {
08         FILE *fp1;
09         char last_name[15], first_name[15];
10         int i = 0;
11
12         if ((fp1 = fopen("name2.txt", "r")) == NULL) {
13             printf("ファイルがオープンできません。¥n");
14             exit(1);
15         }
16         while (1) {
17             printf("最初から何番目のデータを表示しますか (0 で終了) ? ");
18             scanf("%d", &i);
19             if (i == 0) break;
20             fseek(fp1, (i-1) * 30L, SEEK_SET);
21             fscanf(fp1, "%s%s", first_name, last_name);
22             printf("データは%s %s です。¥n¥n", first_name, last_name);
23         }
24         fclose(fp1);
25     }
```

練習問題 28

28-1. 下記のような電話帳がある。これをランダムファイル `name_tel.txt` に書き込むプログラムをつくりなさい。

```
Ichiro Suzuki      03-1111-2222
Hanako Yamato      042-387-5555
```

《ヒント》1人毎のデータを下記のような単位(レコード)で記録する。名と姓、電話番号の2項目についてそれぞれ `scanf` 関数を用いて入力する。また、データ入力の終了は `Ctrl-d` とする。データの書き込みは1人毎のデータをまとめて書式指定して、`fprintf` 関数を用いて行う。

名 15 バイト 姓 15 バイト 電話番号 15 バイト

▼出力例

名前を入力して下さい。終了は, Ctrl-d

Ichiro Suzuki↵

電話番号を入力して下さい。

03-1111-2222↵

名前を入力して下さい。終了は, Ctrl-d

Ctrl-d

28-2. 例題3-5において作成したデータファイル name2.txt の最後から4番目のデータと、その二つ後のデータを表示するプログラムをつくりなさい。

《ヒント》 name2.txt からのデータ読み出しは、fseek 関数で位置を指定してから、fscanf 関数を用いる。fseek 関数での移動原点の指定には、SEEK_END と SEEK_CUR を用いる。

▼出力例

最後から4番目のデータは Ichiro Suzuki です。

二つ後のデータは Makoto Sasaki です。

28-3. 練習問題28-1で作成したファイル name_tel.txt の先頭から任意の位置のデータを表示するプログラムをつくりなさい。

《ヒント》 データ読み出しは、書き込みと同じ書式を指定して fscanf 関数を用いる。

▼出力例

最初から何番目のデータを表示しますか (0で終了)? 1↵

Ichiro Suzuki 03-1111-2222

最初から何番目のデータを表示しますか (0で終了)? 0↵