

14. ファイルの入出力

(1) 入出力の準備

キーボード入力やディスプレイ表示は、標準入出力と呼ばれ、黙っていても使える入出力装置として扱われています。人間にたとえると、目や口のように生まれながらにして備わっている器官と考えればよいでしょう。

一方、ディスクファイルは、ノートや本の類と考えることができます。置いてある棚から取り出してきて、まず開きます。そして、読んだり、書き込みます。終わったらノートや本を閉じて元の場所に戻します。すなわち、ディスクファイルを使うには、以下の指定を行います。

- どのデバイスの、どのフォルダの、どのファイル(どの部屋の、どの棚の、どのノート／本か)を使うかを指定します。
- 書き込むのか、読み込むのか？を指定します。これを入出力モードといいます。
- 書き込みや読み込みの前に、ノートや本を開くように、Open(開く)という操作を行います。
- 終わったら、ノートや本を閉じるように、Close(閉じる)という操作を行います。

(2) オープンとクローズ

まず、入出力用のバッファや制御用の構造体へのポインタを示す変数を宣言します。

```
FILE *fp;
```

この構造体をストリームといいます。ファイルを開くには、次のように書きます。

```
fp = fopen(ファイル名, モード);
```

ファイルを閉じるには、ストリーム変数を使って

```
fclose(fp);
```

と指定します。

ファイル名を定数で指定するとき、フォルダの前後につける¥記号は、以下のように¥を2文字続けて書くことに注意して下さい。

```
C:¥¥myFolder¥¥myFile.txt
```

モードは、次の記号で指定します。

記号	意味
"r"	読み込み用
"w"	書き込み用
"a"	追加用
"r+"	既存ファイルを更新用として
"w+"	新規ファイルを更新用として
"a+"	ファイルの終わりから更新用として
"rt"	テキストファイルを読み込み用として
"wt"	テキストファイルを書き込み用として

なお、更新用とは、読み込みと書き込みを同時に行うような入出力です。

プログラムの誤りによってファイルを壊すことも多いので、特殊なプログラムでなければ、あえて避けたほうがよいでしょう。更新型の典型であるメインメモリのほうが高速ですので、通常のプログラムでは、メモリ内で更新して、結果を別ファイルに出力するほうを推奨します。

(3) 代表的な入出力関数

特に説明のない限り、以下の引数名は、次のような値を示すものとします。

```
char *path : ファイル名
char *mode : 入出力モード
FILE *fp   : ストリームへのポインタ
char *str  : 文字列
char ch    : 文字
```

■ int fgetc(fp)

ストリームから 1 文字を入力。ファイルエンドやエラーのとき EOF を返却。

■ int fputc(ch, fp)

ストリームに 1 文字を出力。エラーのとき EOF を返却。

■ int fgets(str, n, fp)

復改文字(¥n)に出会うか、n-1 文字読み出して str に書き込む。ファイルエンドやエラーのとき EOF を返却。

■ int fputs(str, fp)

ストリームに文字列を出力。エラーのとき EOF を返却。

■ int remove(path)

ファイルの削除。エラーのとき 0 以外の値を返却。

■ int rename(oldpath, newpath)

ファイルの削除。エラーのとき 0 以外の値を返却。

■ int fseek(fp, offset, org)

org (int) で指定した位置 (0 : 先頭, 1 : 現在位置, 2 : 終り) から, offset(long) のバイト数分だけファイルの位置を移動。

■ int fprintf(fp, char *format, ...)

int fscanf(fp, char *format, ...)

書式付き入出力。ファイル入出力 (fp を指定する) における printf, fscanf と考えればよい。

(4) ファイル出力の例

キーボードから入力された文字列をテキストファイルに出力する例を示します。

実行すると、最初、出力ファイル名を聞いてきますのでファイル名を指定します。なお、ファイル名の入力最後の[CR]コードが入力されてしまいますので、ダミー読みを行っています。

[出力プログラム例]

```
#include "stdafx.h"
int _tmain(int argc, _TCHAR* argv[])
{ FILE *fout; char fname[80]; int ch;
  printf("出力ファイル名: ");
  scanf("%s", fname);
  /* scanf 最後の CR] を無視 */
  /* するためのダミー読み */
  ch=getchar();
  if((fout=fopen(fname, "wt")) !=NULL)
  { ch=getchar();
    while(ch !=EOF)
    { if(fputc(ch, fout)==EOF) ch=EOF;
      else ch=getchar();
    }
    fclose(fout);
  }
}
```

(5) ファイル入力の例

テキストファイルから読み込み、ディスプレイに表示する例を示します。

```
#include "stdafx.h"
int _tmain(int argc, _TCHAR* argv[])
{ FILE *finp; char fname[80]; int ch;
  printf("入力ファイル名: ");
  scanf("%s", fname);
  if((finp=fopen(fname, "rt")) != NULL)
  { while((ch=fgetc(finp)) != EOF)
    { printf("%c", ch);
    }
    fclose(finp);
  }
}
```

(6) 出力の際の常套手段

ファイル名を間違えて指定したために、大事なファイルを壊してしまった経験は、だれでもありませんね。そこで、間違えて上書きしても復旧できるように、以下のような処理を行います。

- ① ファイルが既に存在していたら、上書きするかどうかを問い合わせる。
- ② それでも間違える可能性があるため、既に存在しているファイルの拡張子を変えて残し、新規ファイルとして出力する。

```
[プログラム例]
#include "stdafx.h"
#include "conio.h"
static FILE *fout;
void fname_change(char name[])
{
    int i, ip; char new_name[81];

    /* ファイル名の最後の位置を見つける */
    for (i=0; (i<76) && (name[i] !=0); i++)
        new_name[i]=name[i];
    ip=i--;

    /* 拡張子の位置を見つける */
    /* ドット(.)が見つからなければ */
    /* 拡張子なしとする */
    while ((i>=0) && name[i] !='.') i--;
    if (i>0) ip=i;

    new_name[ip] = '.';
    new_name[ip+1] = '$'; /* 拡張子. $$$ */
    new_name[ip+2] = '$';
    new_name[ip+3] = '$';
    new_name[ip+4] = 0;

    /* 拡張子. $$$のファイルを削除 */
    remove(new_name);

    /* 出力ファイルと同じ名前のファイルを
       拡張子. $$$の名前に変更 */
    rename(name, new_name);
}
```

```
int ex_open(char name[])
{
    char ch;

    /* 読み込みモードでオープンできたら、
       ファイルありと判断する */
    if ((fout=fopen(name, "r")) !=NULL)
    {
        fclose(fout);

        /* 確認用メッセージ出力 */
        printf
        ("¥n 同一ファイル名があります。");
        printf("上書きしますか(Y/N)?");
        do ch = _getch(); /* 1文字入力 */
        while (ch!='y' && ch!='Y' &&
              ch!='n' && ch!='N');

        if (ch=='n' || ch=='N') return 1;
        fname_change(name);
    }
    if ((fout=fopen(name, "wt"))==NULL)
        return 2;
    return 0;
}

int _tmain(int argc, _TCHAR* argv[])
{
    char fname[80];
    printf("ファイル名:"); scanf("%s", fname);
    if (ex_open(fname)==0)
    {
        /* この部分にファイル出力処理を
           記述する。以下はテスト用 */

        fprintf(fout, "¥n Test Dayo");
        fclose(fout);
    }
}
```

(7) 演習

テキストファイルをコピーするプログラムを作成しなさい。

なお、出力するファイルの同一ファイルがあったら、上書きするかどうかを確認するようにしてください。