2013 年 7 月 4 日（木）実施

ファイル処理

ファイルとは

ファイル（file）は日常用語では紙などを綴じたものを表すが，コンピュータ用語では**データ**の集合体を指す言葉である。ファイルは例えば，文書ファイルやプログラムファイルのように，用途によって分類されることもあれば，また，テキストファイルやバイナリファイルのように，ファイルの作り方によって分類されることもある。

なお，ファイルに階層構造を持たせて，ファイルは**レコード**の集合体，レコードはデータの集合体とする場合がある。JIS（日本工業規格）用語でのファイルの定義は，このような階層構造に基づいている。レコードとは，様々な種類のデータ項目を一まとめにしたもので，例えば，住所録に於ける 1 人分のデータ（郵便番号，住所，氏名，電話番号等）がこれに当たる。C 言語では，レコードに相当するデータ構造を**構造体**によって表現する。同種のデータを一まとめにした配列では，データ型は 1 種類で共通であるが，構造体の各項目（メンバ）はそれぞれ別のデータ型となり得る。構造体に関する詳細は，次々回の教材で解説する。

ファイル操作

C 言語でファイル操作を行う際には，**高水準入出力関数**と呼ばれるライブラリ関数が利用できる。高水準入出力関数は OS に依存せずにファイルの読み書きを行えるもので，その中では**低水準入出力関数**と呼ばれる OS のシステムコールを利用するハードウェアよりのものを利用している。また，高水準入出力関数では，ファイルを直接操作せず，**バッファ**と呼ばれる一時記憶装置を介して，ファイルの読み書きを行う。このようなファイルへのアクセスを**ストリーム**と呼ぶ。

ここでは，C 言語で高水準入出力関数を用いたファイル操作を扱う。ファイル操作の一連の流れは次のようになる。

1) ストリームを開く　（読み込み，書き出し，追加等のオープンモードを指定する）
2) ストリームからの入力，ストリームへの出力
3) ストリームを閉じる

なお，ストリームを開く関数 fopen はストリームを制御する変数へのポインタを返すが，これを**ファイルポインタ**の初期値として設定し，それ以降はファイルポインタを用いる。入出力（読み込み，書き出し）は標準入出力の場合と同様，CPU 側から見た方向である。

例）FILE *fp = fopen("gakusei_data.txt","r");

ここで，FILE はストリームを制御するための情報を記録する変数の型であり，通常は構造体として stdio.h 中で定義される。そのメンバには，ファイル位置指示子やバッファへのポインタ等が含まれる。

fopen のオープンモードの 1 文字目には，次の 3 種類のうちいずれかを指定する。

1) r　読み込み（read）モード　　ファイルが存在しないか，読み込めなければ，失敗
2) w　書き出し（write）モード　　ファイルを新規に作成するか，既存のファイルがあれば，内容を廃棄し，長さ 0 に切り詰めて開く
3) a　追加（append）モード　　既存のファイルを追加書き出し用に開くか，ファイルを新規に作成する

fopen のオープンモードの 2 文字目または 3 文字目には，次の文字が指定可能である。両者を組み合わせることも可能である。

1) b　バイナリ（binary）モード　バイナリファイルとして開く
2) +　更新モード　　上述の r, w, a でストリームを開く際の扱いは引き継ぎ，読み込み，書き出し共に可能とする

例題 1（**ファイルからのデータの読み込み**）

次のソースプログラムをテキストエディタで入力して，prog12-1.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し，実行せよ。なお，実行する前に，1 個の整数値を書き込んだ"in.txt"を，プログラムと同じディレクトリに作成しておく必要がある。

```
/* prog12-1.c */
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int i;
    FILE *fin;

    if (NULL == (fin = fopen("in.txt","r")))
    {
        printf("ファイルが開けません。¥n");
        exit(1);
    }

    fscanf(fin, "%d", &i);

    printf("ファイルから読み込まれた整数値は %d です。¥n", i);

    fclose(fin);

    return 0;
}
```

【解説】

1. fopen がストリームを開くことに失敗した場合には，NULL を返す。

2. fin = fopen("in.txt","r") という式の値は，fopen が返して fin に代入された値となる。

3. fscanf は，第 1 引数に指定されたストリームからの入力を，入力書式文字列に基づいて後に続く変数に格納する。

例題 2（**ファイルへのデータの書き出し**）

次のソースプログラムをテキストエディタで入力して，prog12-2.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し，実行せよ。また，出来上がった"out.txt"の内容を確認する。

```
/* prog12-2.c */
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int i;
    FILE *fout;

    printf("整数値を入力してください：");

    /* 標準入力ストリームから整数値を読み込む */
    if (fscanf(stdin, " %d", &i) == 1)
        printf("標準入力から読み込まれた整数値は %d です。¥n", i);
    else
    {
        fprintf(stderr, "標準入力からの整数値の読み込みエラーです。¥n");
        exit(1);
    }

    if (NULL == (fout = fopen("out.txt","w")))
    {
         printf("ファイルが開けません。¥n");
         exit(1);
    }

    fprintf(fout,"i=%d¥n",i);

    fclose(fout);

    return 0;
}
```

【解説】

1. fscanf は代入された入力項目数を返す。（入力文字と書式とが一致しない場合は小さくなることがあり，0 にもなり得る）また，入力失敗の際には EOF マクロの値（負の整数）を返す。ここでは，正常に 1 個の変数にデータの格納が行われた場合には 1 が返されることを利用して

いる。

2. fscanf の入力書式文字列で" %d"の様に%d の前に空白が置かれているのは，ホワイトスペースを読み飛ばす（空白や改行を誤って入れても，その後に数値を入力できるようにする）ためのものである。

3. stdin は標準入力ストリームを表し，stderr は標準エラー出力ストリームを表す。

例題 3（ファイルへのデータの追加）

次のソースプログラムをテキストエディタで入力して，prog12-3.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し，実行せよ。なお，実行時には，プログラム第１引数にファイル名を指定して，　prog12-3 out.txt と入力する。out.txt 以外のファイル名を指定した場合には，最初の実行時にファイルが新規に作成される。実行後に，"out.txt"の内容を確認する。

```
/* prog12-3.c */
#include <stdio.h>
#include <stdlib.h>

int main(int argc,char *argv[])
{
    int i;
    FILE *fio;

    if (argc != 2)
    {
        printf("利用法：prog12-3 ファイル名¥n");
        exit(1);
    }

    if (NULL == (fio = fopen(argv[1],"a")))
    {
        printf("ファイルが開けません。¥n");
        exit(1);
    }

    printf("整数値を入力してください：");

    if (fscanf(stdin, " %d", &i) == 1)
    {
        printf("標準入力から読み込まれた整数値は %d です。¥n", i);
        fprintf(fio,"i=%d¥n",i);
    }
    else
    {
        fprintf(stderr, "標準入力からの整数値の読み込みエラーです。¥n");
        exit(1);
    }

    fclose(fio);
```

```
    return 0;
}
```

演習 1

10 個の整数データが各行に記述された入力用のファイル"input.txt"を開き，そこから読み込んだデータをソートした結果を，出力用のファイル"output.txt"を開いて書き込むプログラムを考える。この空欄 1)　　，2)　　，3)　　，4)　　を埋めてソースプログラムを完成させ，テキストエディタで入力して，ex12-1.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し，実行せよ。なお，"input.txt"は教材としてダウンロードしたものを用いる。

```
/* ex12-1.c */
#include <stdio.h>
#include <stdlib.h>

#define MAXDATA 10

void swap(int *, int *);
void sort(int [], int);

int main(void)
{
    int i;
    int point[MAXDATA];
    FILE *1)    , *2)    ;

    if (NULL == (fin = fopen("3)    ","r")))
    {
        printf("入力ファイルが開けません。¥n");
        exit(1);
    }

    for (i=0; i<MAXDATA; i++)
        fscanf(fin,"%d",&point[i]);
    fclose(fin);

    printf("データをソート中¥n");
    sort(point, MAXDATA);

    if (NULL == (fout = fopen("4)    ","w")))
    {
        printf("出力ファイルが開けません。¥n");
        exit(1);
    }

    for (i=0; i<MAXDATA; i++)
        fprintf(fout,"%2d:%4d¥n",i+1,point[i]);
    fclose(fout);

    return 0;
```

```
}

void swap(int *x, int *y)
{
    int temp=*x;
    *x=*y;
    *y=temp;
}

void sort(int data[], int n)
{
    int k=n-1;
    while (k>=0)
    {
        int i, j;
        for (i=1, j=-1; i<=k; i++)
            if (data[i-1]<data[i])
            {
                j=i-1;
                swap(&data[i], &data[j]);
            }
        k=j;
    }
}
```

提出物：

1) 例題 1 の出力結果をコピーして貼り付けたテキストファイル res12.txt
2) テキストファイル in.txt
3) テキストファイル out.txt
4) テキストファイル output.txt
5) 演習 1 のソースプログラムのファイル ex12-1.c の完成版