

第7回 C 言語講座

2011/6/20

メインテーマ：ファイル操作、定数

○ファイル操作

外部のファイルからデータを読み込んだり、ゲームだったらハイスコアを保存したりする機能。

以下のルールで読み書きする。

- ①ファイルを開く（ここで読み取るか書きこむかを定める）
- ②データを読み書きする
- ③ファイルを閉じる

ここでは一番わかりやすいテキストファイルの読み込みを例に上げる。バイナリファイルの読み書きについては各自調べてみることに。

・定数① (#define)

#define 名前1 名前2 と書くと名前1が名前2にコンパイル時に置き換わる。

置き換えなので、配列の宣言時にも使える

例：#define MAX 500 →ソース中で MAX と書いたら 500 として扱われる。

int arr[MAX]; と書くのは OK。

```
#include <stdio.h>
#define STRING_MAX 100 //define定義。STRING_MAXが100として扱われる
int main(void)
{
    FILE *fp;          //ファイルのデータをもろう構造体
    char str[ STRING_MAX ]; //書き込んだりする文字列

    printf( "文字列を入力してください\n->" );
    scanf( "%s" , str );

    //書き込みモードで開く(ファイルがなかったら勝手に作る)
    fp = fopen( "test.txt" , "w" );
    //ファイルへの書き込み
    fprintf( fp , "%s" , str );
    //ファイルを閉じる
    fclose( fp );

    //読み込みモードで開く(ファイルがなかったらエラー)
    fp = fopen( "test.txt" , "r" );
    if( fp == NULL ) //ファイルがない場合の処理。そのまま読み込み関数と呼んではいけないので終了
    {
        printf( "ファイルがありません\n" );
        return -1;
    }
    //ファイルから文字列を読み込む
    fscanf( fp , "%s" , str );
    //ファイルを閉じる
    fclose( fp );

    printf( "%s\n" , str );
    return 0;
}
```

○リアルタイム入力とコンソール画面のクリア

kbhit() → キーボードが押されているか調べる関数

getch() → 押されたキーを調べる関数

system("cls") → コンソール画面をクリアする。

・定数②(const)

const を変数宣言の前につけると、その変数は値を変更することが出来なくなる。(初期化の際に値を設定する。)

#define との違いは#define がただの”置き換え”であるのに対し、const はあくまでも変数であるということ。

```
#include <stdio.h>
#include <conio.h> //kbhit , getchに必要
#include <windows.h> //systemに必要

int main(void)
{
    char input = 0;
    const int KEY_ESCAPE = 27; //定数

    while( input != KEY_ESCAPE ) //Esc = 27。エスケープが押されたら終了
    {
        if( kbhit() == 1 ) //キーボードが押されていたら
        {
            input = getch(); //押されているキーを取得
            if( input == 'c' ) //cが押されていたら画面をクリア
            {
                system( "cls" );
            }
            printf( "%c が押されました\n" , input );
        }
    }
    return 0;
}
```

演習問題

①

csv ファイルという Excel で読み書きできるファイルがある。

このファイルは各データがカンマで区切られている形式をしている。

そのため、プログラムで読み書きしやすく、Excel が使えることから編集も比較的簡単である。

以下の csv ファイルを作り、データを画面に表示せよ。

また、後ろの 2 つの値を x 座標、y 座標として原点からの距離 $\sqrt{x^2 + y^2}$ を算出せよ。

data.csv

```
obj1,100,50
```

ヒント：

csv ファイルの読み込みには `fscanf` 関数ができる。

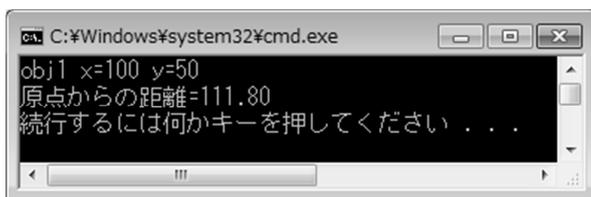
カンマで区切られた複数データを読み込むには `fscanf (fp, "%d,%d,%d",&a,&b,&c)` のように使うが、文字列を読み込む時だけ注意が必要になる。

その理由は普通に `%s` と書くと区切りで使っているカンマを文字列としてみなしてしまうため、それを回避するには `fscanf (fp,"%[^,],%d" , str , &a)` と書く。`%[^,]` はカンマ以外の文字を読み込むという意味である。

また、平方根の計算は `sqrt()` で行える。(`math.h` をインクルードすること。)

`sqrt` 関数の中身が `int` 型の演算になる場合、`float` などでキャストすること。

結果例：



```
C:\Windows\system32\cmd.exe
obj1 x=100 y=50
原点からの距離=111.80
続行するには何かキーを押してください...
```

②

リアルタイムで入力を受けつけ、数字キーが入力されたらその値を足して表示していくプログラムを作れ。

ただし、特定のキーが押されたらプログラム終了とする。(文字キーなら何でも良い)

入力されたキーが数字かどうかの判定は、前回使った `if(moji >= '0' && moji <= '9')` を使って良い。

結果例：



```
C:\Windows\system32\cmd.exe
入力 = 1, 合計 = 1
入力 = 2, 合計 = 3
入力 = 7, 合計 = 10
入力 = 9, 合計 = 19
入力 = 0, 合計 = 19
入力 = 8, 合計 = 27
入力 = 6, 合計 = 33
```

①

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    char str[100];
    int x,y;
    float distance;

    FILE *fp;
    fp = fopen( "test.csv", "r" );
    if( fp == NULL )
    {
        printf( "test.csvがありません" );
        return -1;
    }
    fscanf( fp , "%[^,],%d,%d", str,&x,&y );
    fclose(fp);

    distance = sqrt( (float)( x*x + y*y ) );

    printf( "%s x=%d y=%d\n", str,x,y );
    printf( "原点からの距離=%.2f\n", distance);

    return 0;
}
```

②

```
#include <stdio.h>
#include <conio.h>
#define KEY_ESCAPE 27

int main(void)
{
    int sum = 0;
    char input = 0;
    while( input != KEY_ESCAPE )
    {
        if( kbhit() )
        {
            input = getch();
            if( input >= '0' && input <= '9' )
            {
                sum += input - '0' ;
                printf( "入力= %c , 合計= %d\n" , input , sum );
            }
        }
    }
    return 0;
}
```