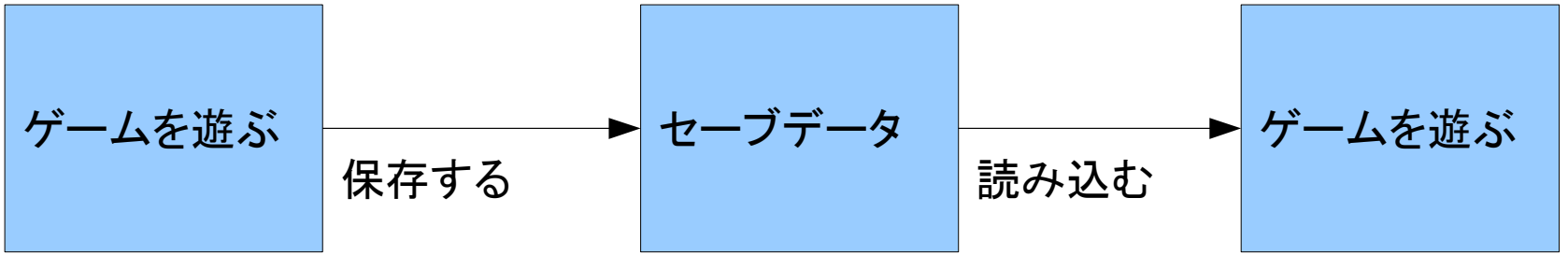
C言語講座Vol.7

2009年6月26日　CISC

ファイル

- プログラム中、メモリ上に計算した結果を置いているが、これはプログラムが終了したら消えてしまう。
- そこで、プログラム中で計算した結果など終了後も「保存しておくべき物」を、ファイルとして保存する。
- 設定ファイル、セーブデータ等。



C言語でのファイルの扱い

- 今回教えるのは、テキストファイル
- テキストデータ（＝文字列）を保存しておく。
- もちろん、数字だけの文字列としてもよい

- キーワード
  - ファイルポインタ
  - ファイルのオープンとクローズ

ファイルへの書き込み

```c
#include <stdio.h>

int main(void)
{
    FILE *fp;                   //ファイルポインタの宣言
    char str[100];

    fp=fopen("./text.txt","w");     //ファイルの「オープン」
    if(fp == NULL)                  //失敗したら終了
    {
        return 0;
    }

    printf("書き込む文字列を入力してください¥n=>");
    scanf("%s",&str[0]);

    fprintf(fp,"%s¥n",&str[0]);     //文字列の書き込み

    fclose(fp);                     //ファイルの「クローズ」

    return 0;
}
```

ファイルからの読み込み

```c
#include <stdio.h>

int main(void)
{
    FILE *fp;                   //ファイルポインタの宣言
    char str[100];

    fp=fopen("./text.txt","r");     //ファイルの「オープン」
    if(fp == NULL)                  //失敗したら終了
    {
        return 0;
    }

    while(fscanf(fp,"%s",&str[0]) != EOF)
    {
        printf("%s",&str[0]);
    }

    fclose(fp);                     //ファイルの「クローズ」

    return 0;
}
```

ファイルへの追記

```c
#include <stdio.h>

int main(void)
{
    FILE *fp;                   //ファイルポインタの宣言
    char str[100];

    fp=fopen("./text.txt","a");     //ファイルの「オープン」
    if(fp == NULL)                  //失敗したら終了
    {
        return 0;
    }

    printf("書き込む文字列を入力してください¥n=>");
    scanf("%s",&str[0]);

    fprintf(fp,"¥n%s",&str[0]);     //文字列の書き込み

    fclose(fp);                     //ファイルの「クローズ」

    return 0;
}
```

グローバル変数

- どこからでもアクセスできる変数

```
#include <stdio.h>

int num;            //グローバル変数

void input(void);

int main(void)
{
     input();
     printf("%d¥n",num);
     return 0;
}

void input(void)
{
     printf("数字を入力してください。");
     scanf("%d",&num);
}
```

多次元配列

- 2次元だけでなく、3次元でも4次元でも、使うことができる。

int型の要素１０個を持つ配列をさらに５個、連続的に確保する

```
int a[5][10];
```

```
int a[M][N];
```

↓

```
int a[M*N];
```

一次元の配列で同様のことを実現しようとすると、宣言はこのようになる。

初期化の方法

```
int a[2][3]={{1,2,3},
             {2,6,4}};

char str[3][100]={"Intel",
                  "AMD",
                  "IBM"};
```

アクセス方法は同様にこうなる。

```
a[i][j]
```

↓

```
a[i*M+j]
```

マクロ

- プログラム中の置き換えができる。
- 引数を持つこともできる

```
#include <stdio.h>
#define NUM 5          //NUMを見つけたら、それをに書き換える
#define f(x) (x+x)     //関数？も示せる

int main(void)
{
     int i;
     for(i=1;i<=NUM;i++)
     {
          printf("%d  ",i);
          printf("%d¥n", f(i) * f(i) );
     }
     return 0;
}
```

メモリアクセス

- malloc 指定バイト分メモリを確保
- sizeof　変数などのバイト数を返す
- free　　メモリの開放

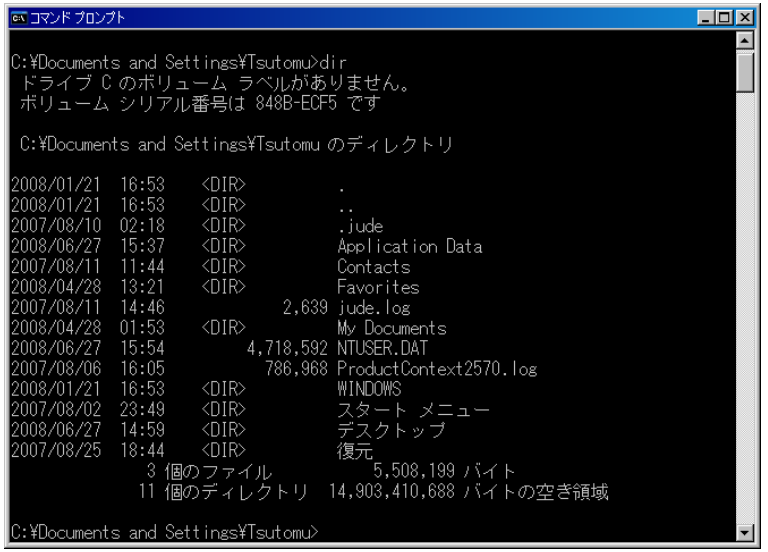
- 動的な領域確保などに使える。
- ポインタと密接な関係。
  興味のある人は、これをキーワードにしてネットを調べてみるとよろし。もちろん、質問に来ても可。

演習

- ファイルから数字を読み込んで、その和を計算し、画面に表示するプログラムを作れ。
  数字は、空白で区切られて書き込まれている。
  ファイルの終了（=EOF）で読み込みを終了する。

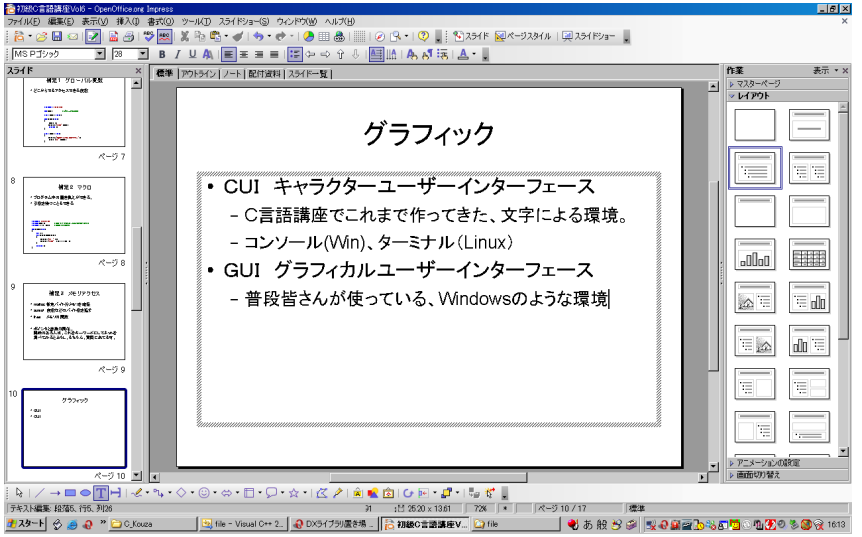
グラフィック

- CUI　キャラクターユーザーインターフェース
  - C言語講座でこれまで作ってきた、文字による環境。
  - コンソール(Win)、ターミナル（Linux）
- GUI　グラフィカルユーザーインターフェース
  - 普段皆さんが使っている、Windowsのような環境



GUI →

← CUI



Windowsでのグラフィックの取り扱い1

- めちゃくちゃ面倒
  左のプログラムが、
  Windowsにおける「最小の」
  プログラム
- グラフィックとか言う前に、
  初心者には読みずらい。
- キーワード
  - Win32API
  - API
  - GDI

```
/*Ｗｉｎｄｏｗｓプログラミングの最初の一歩*/

#define     STRICT
#include    <windows.h>

/*ウインドウプロシージャのプロトタイプ宣言*/
LRESULT CALLBACK WindowProc(HWND,UINT,WPARAM,LPARAM);

/*  アプリケーションエントリーポイント */
int WINAPI WinMain(HINSTANCE    hInstance,
                   HINSTANCE    hPrevInstance,
                   LPSTR        CmdLine,
                   int          CmdShow)
{
    HWND            hwnd;   /*  メインウインドウのウインドウハンドル   */
    MSG             msg;    /*  メッセージキューから取得したメッセージ */
    WNDCLASS        wc;     /*  ウインドウクラス登録用の構造体         */

    wc.style        =0;
    wc.lpfnWndProc  =WindowProc;
    wc.cbClsExtra   =0;
    wc.cbWndExtra   =0;
    wc.hInstance    =hInstance;
    wc.hIcon        =LoadIcon(NULL,IDI_APPLICATION);
    wc.hCursor      =LoadCursor(NULL,IDC_ARROW);
    wc.hbrBackground=(HBRUSH)(COLOR_WINDOW+1);
    wc.lpszMenuName =NULL;
    wc.lpszClassName="Hello";

    if(RegisterClass(&wc)==0)       /*  ウインドウクラス登録   */
        return  0;

    hwnd=CreateWindow(  "Hello",    /*  ウインドウ作成         */
                        "Hello World",
                        WS_OVERLAPPEDWINDOW,
                        CW_USEDEFAULT,
                        CW_USEDEFAULT,
                        CW_USEDEFAULT,
                        CW_USEDEFAULT,
                        NULL,
                        (HMENU)NULL,
                        hInstance,
                        0);
    if(hwnd==NULL)
        return  0;

    ShowWindow(hwnd,CmdShow);           /*  ウインドウの表示       */
    UpdateWindow(hwnd);                 /*  ウインドウの最初の更新 */

    while(GetMessage(&msg,NULL,0,0))    /*  メッセージループ       */
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return  msg.wParam;
}

/*  ウインドウプロシージャ */
LRESULT CALLBACK WindowProc(HWND hwnd,UINT message,WPARAM wparam,LPARAM lparam)
{
    switch(message)
    {
        case WM_DESTROY:
            PostQuitMessage(0);
            return  0;
    }
    return  DefWindowProc(hwnd,message,wparam,lparam);
}
```

Windowsでのグラフィックの取り扱い2

- API（アプリケーションプログラミングインターフェース）
- OSの機能を使うために、橋渡しをしてくれる。
- Windowsでグラフィックを扱うときに使う
  代表的なものは次のとおり
  - Direct X
    - ゲーム開発の際一般的に利用される。グラフィックだけではなく、入出力やサウンド、通信機能なども備える。
  - Open GL
    - フリーの環境として有名。ことグラフィックにこだわる方や、Win以外の方が使う。
  - GDI
    - 目の前のwindowはこれで描かれている。

簡単な方法

- 先ほどあげたものは、プロユース耐えるというか、そもそもそういう物。C言語だけでなく、C++などの知識が十分あって、大量のコードを読み下せる能力がないとついて行けない。
- そこで、DirectXなどを簡単に使えるように、仲立ちをしてくれる「ライブラリ」を使うとよい。
  - DX ライブラリ
    2Dのみだが、周囲の（既存の）知識を活用できる。
  - Luna
    国産では希少な、3Dも扱えるライブラリ
  - EL（Easy Link Library）
    その昔流行ったらしい、ライブラリ

ゲーム向けのテクニックについて

- 乱数の取り扱い
  →次のスライド
- コンソールでのリアルタイム動作の方法
  →別紙サンプルプログラム

乱数の使い方

- C言語での乱数の使い方

```c
#include <stdio.h>
#include <stdlib.h>    //乱数の関数のために必要
#include <time.h>      //時間取得のために必要

int main(void)
{
     int a;
     srand( (unsigned) time( NULL ) );   //現在の時刻で乱数テーブルを初期化

     a = rand();
     printf("取得した乱数は%dです。¥n",a);

     //1～の間にしたい場合
     a = rand()%10 + 1;
     printf("取得した乱数(1～)は%dです。¥n",a);

     return 0;
}
```

最終課題

- C言語を用いて、コンソールで動くゲームを作れ。
- 題材が思いつかない人は、次を使ってよい
  - スロットゲーム　じゃんけんゲーム　陣取りゲーム
- ゲームを作るにあたり、次の関数を紹介する。
  使い方は先ほどのスライド等を参照

関数名	仕様	例	例の意味	必要なヘッダファイル
rand	int rand();	num=rand();	Numにランダムな値を代入	stdlib.h
system	system(char *)	system("cls");	画面を消去する	windows.h
getch	int getch();	key=getch();	押されたキーの文字を代入	conio.h

<u>提出＆発表会は再来週金曜</u>
<u>質問などあれば会室にて随時受付</u>