

参考文献：「パソコン用手作り外部インターフェース」、土井滋貴著、CQ 出版社

■ マルチスレッド

● スレッドを使ったプログラム

Visual Studio .NET 2003 を起動。

ファイル > 新規作成 > プロジェクト

Visual C++プロジェクト > Win32 コンソールプロジェクト

プロジェクト名: List203

場所: D:\takase\Control

新しいソリューション名: List203

[OK]

概要 ・ コンソールアプリケーション

[完了]

次のようにソースを入力する。

```
// List203.cpp : コンソール アプリケーションのエントリ ポイントを定義します。
```

```
//
```

```
#include "stdafx.h"          // <windows.h>
#include <stdio.h>
#include <conio.h>            // getch() の宣言
#include <windows.h>
#include <process.h>         // スレッド用
```

```
// このコード モジュールに含まれる関数の宣言を転送します :
```

```
DWORD WINAPI      thProc(LPVOID);
LRESULT CALLBACK  WndProc(HWND, UINT, WPARAM, LPARAM);
```

```
HINSTANCE hInstance;      // スレッド側でも利用するのでグローバル変数とする
```

```
void main()
```

```
{
```

```
    // WinMain() と異なり main() はシステムからインスタンス・ハンドルを得られない
```

```
    // 代わりに指定されたウィンドウに関する情報を取得する GetWindowLong() を使う
```

```
    hInstance = (HINSTANCE)GetWindowLong (HWND_DESKTOP, GWL_HINSTANCE);
```

```
    WNDCLASSEX wcex;      // 新しくつくるウィンドウクラス用構造体
```

```
    memset (&wcex, 0, sizeof (WNDCLASSEX));
```

```
    wcex.cbSize = sizeof (WNDCLASSEX); // WNDCLASSEX の大きさ
```

```
    wcex.lpfnWndProc = (WNDPROC)WndProc; // このクラスの持つウィンドウプロシージャ
```

```
    wcex.hInstance = hInstance;
```

```
    wcex.hCursor = LoadCursor (NULL, IDC_ARROW);
```

```
    wcex.hbrBackground = (HBRUSH)GetStockObject (WHITE_BRUSH);
```

```
    wcex.lpszClassName = "wc00"; // szWindowClass;
```

```

if (!RegisterClassEx(&wcex)) return:    // ウィンドウクラスの登録

// メッセージループのスレッドを起動
DWORD id;
CreateThread(NULL,          // pointer to thread security attributes
             0,             // initial thread stack size, 0はデフォルト
             thProc,        // pointer to thread function
             NULL,         // argument for new thread
             0,             // creation flags
             &id);         // pointer to returned thread identifier

printf("%n コンソールからスレッドを使ってウィンドウを作成\n");
printf("終了するには何かキーを押してください。%n\n");
getch();
}
/**
 * メッセージ・ループのためのスレッド、ここでウィンドウを作る
 */
DWORD WINAPI thProc(LPVOID)
{
    MSG msg;

    CreateWindow("wc00",          // ウィンドウクラスの名前
                 "コンソール&マルチスレッド版 hello", // キャプションの内容
                 WS_OVERLAPPEDWINDOW | WS_VISIBLE, // ウィンドウの属性、クラスの名前
                 CW_USEDEFAULT, CW_USEDEFAULT, // 位置は指定しない
                 400, 300, // ウィンドウの大きさ
                 HWND_DESKTOP, // 親はデスクトップ
                 NULL, hInstance, NULL); // メニューハンドルなし、インスタンス、

    // メイン メッセージ ループ :
    while (GetMessage(&msg, NULL, 0, 0)) // メッセージループ
    {
        DispatchMessage(&msg);
    }

    return 0;
}

```

```

//
// 目的 : メイン ウィンドウのメッセージを処理します。
// WM_COMMAND - アプリケーション メニューの処理
// WM_PAINT   - メイン ウィンドウの描画
// WM_DESTROY - 中止メッセージを表示して戻る
//
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    switch (message)
    {
        case WM_PAINT:
            {
                PAINTSTRUCT ps;
                char *p = "hello windows";
                BeginPaint(hWnd, &ps);
                TextOut(ps.hdc, 10, 10, p, (int)strlen(p));
                EndPaint(hWnd, &ps);
                break;
            }
        case WM_DESTROY:
            PostQuitMessage(0);
            break;
        default:
            return DefWindowProc(hWnd, message, wParam, lParam);
    }
    return 0;
}

```