

2P131 Windows 環境における計測プログラミング

(広島市大・情報) 藤原久志, 石渡 孝

【序】近年我々は、画像計測における必要性から、Windows 環境で C 言語による計測プログラミングを行っている。その経験を通じて MS-DOS 環境から Windows 環境に移行する際の「障壁(メッセージ、スレッドについての理解)」を認識した。本発表では、この二点についての解説を行い、さらにサンプルプログラム、参考文献を示す事にする。これらの情報を参考にすれば、MS-DOS 環境と同様な感覚で C 言語による計測プログラミングが可能になると考えている。なお、サンプルプログラムのソースコードは、近日中に公開予定の研究室のホームページ内に掲載する予定であり、そのアドレスは学会当日にポスターに明記する。

【Windows プログラムの特徴】

1：基本構成(メッセージの役割) 一般的な Windows プログラムの根幹は、WinMain 関数とウィンドウのコールバック関数(ウィンドウプロシージャと呼ばれる)である。

全ての Windows プログラムの実行は、WinMain 関数の呼び出しで始まる。この関数の役割は、ウィンドウの生成、そしてオペレーティングシステム(OS)からのメッセージの受け取りと処理である。OS より送られるメッセージは、何らかのイベント(キーを押した、ウィンドウ中のボタンが押された等)に対応している。WinMain 関数は、受け取ったメッセージを適宜加工し、OS を介してウィンドウプロシージャに伝える。

ウィンドウプロシージャは、受け取った種々のメッセージに対応した実際の処理を行う。図 1 にその例を示す(この例は、図 2 に示したサンプルプログラムの中から選んだ)。図中の「IDC_BUTTON1」は、図 2 の「表示」ボタンが押された事に対応するメッセージであり、これに対応して測定スレッド関数が起動することになる。ウィンドウプロシージャはコールバック関数の一種であり、OS がメッセージを伝える必要があるときに呼び出される。そしてウィンドウプロシージャは、内部に記述された switch 文に応じて処理を判定し実行する。この処理後にウィンドウプロシージャは終了し、次の呼び出しに備える。

```
case WM_COMMAND:
    switch(LOWORD(wp)){
        case IDC_BUTTON1:
            //測定スレッド起動
            threadHandle = (HANDLE)_beginthreadex(0, 0, Measurement,
                (LPVOID) &measure, 0, (unsigned *) &threadID);
            return TRUE;
    }
```

図 1 ウィンドウプロシージャ内の記述例(抜粋)

2：スレッド 上では「受け取ったメッセージに応じて、ウィンドウプロシージャが実際の処理を行う」と述べた。しかし、その処理が複雑な場合がある(計測プログラミングは、正にその好例である)。このとき、その処理をウィンドウプロシージャ内に全て記述してしまうと、この処理を行っている間プログラムはメッセージに回答できなくなる(ボタンを押しても反応しない等)。このような場合には、スレッド関数に処理すべき内容を記述し、ウィンド

ウプロシージャではその関数を呼び出すことだけを行う(図1参照)。スレッドとは、実行中のプログラムの中の1つの処理の流れのことであり、Windows環境では1つの実行プログラムの中に複数の流れ(マルチスレッド)を持つ事ができる。ウィンドウプロシージャは、スレッド関数を呼び出した後に速やかに終了できるので、次のOSの呼び出しに備えることができる。一方、スレッド関数は、これを呼び出したウィンドウプロシージャが終了した後も、独立して測定を続けることが可能である。このスレッド関数は、内部に適切な終了条件を設定しておくことで、ユーザーの希望する機会に終了させることができる(終了条件の実装については、サンプルプログラムのソースコードを参照のこと)。

【サンプルプログラムの概要】

サンプルプログラムは、Visual C++ 6.0環境でコーディングした。クラスライブラリは使用せず、Win32 API 関数とC言語の標準ライブラリ関数を使用した。

プログラムの主な機能は、「表示ボタン」を押すと正弦波形を計算し描画する、

「記録ボタン」を押すとこれらに加えて描画したデータをファイルに保存する、の二つである。「正弦波形の計算コード」を各自が必要とする「計測機器の制御(データ取得)コード」に置き換えてやる事で、独自の計測プログラムに書き換えることができる。また、その他に以下の機能も実装している：高性能パフォーマンスカウンタを利用した計時機能；プログラムからの情報をテキスト表示する「メッセージ画面(デバッグ、エラー発生時に有用)」；書き込んだテキストをファイル保存する「メモ画面(計測条件などの保存に利用可能)」。

Windows環境においてC言語で自由に計測プログラムを構築する能力は、独自の新しい計測システムを開発するための、ひとつの大きな武器となるはずである。

【参考文献】

- 1) 土井滋貴、那須靖弘、上田悦子、Win32 API 完璧マスタ(CQ出版、東京、2001)。
- 2) 北山洋幸、技術者のためのVisual C++実践プログラミング技法(技術評論社、東京、1999)。
- 3) Marshall Brain、Ron Reeves(訳：ドキュメントシステム)、Win32システムサービスプログラミング(ピアソン・エデュケーション、東京、2002)。
- 4) Herbert Schildt(監訳：山本信雄)、Windows 2000プログラミング標準講座(翔泳社、東京、2000)。
- 5) Charles Petzold(訳：株式会社ロングテール、長尾高弘)、プログラミングWindows第5版(上・下巻)(アスキー、東京、2000)。

(補足)文献1は、今回のプログラム作成で一番参考にした本である。また文献2は、関数の戻り値や引数の記述が見易く、スレッド・同期の記述が充実しており、文献1を補う形で参考にした。ファイル入出力、通信については、文献3に詳しい記述があり、この本と文献1、2を揃えれば、出発点として充分である。その後には、Windowsプログラミングの一般的知識ならば文献4、画像の取り扱いならば文献5を参照すればよい。

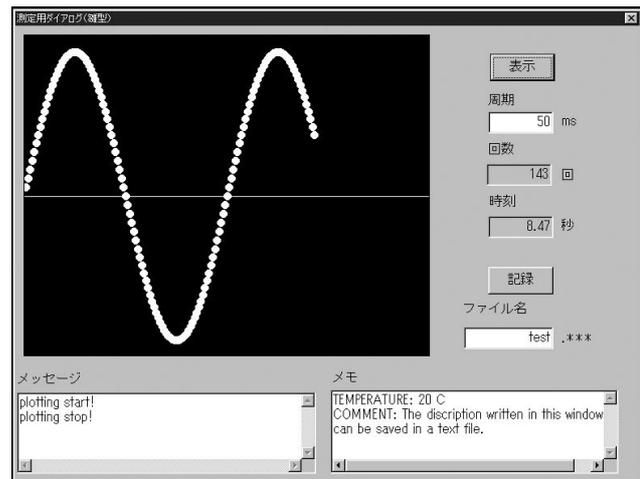


図2 サンプルプログラムの実行画面