



Object BLVD オブジェクトの散歩道

例題でわかる！ .NET Framework

第9回 「gcc/g77で.NET用DLLを作る」 ～ DLLアレルギーを解消しよう！

吉田 弘一郎 YOSHIDA, Koichiro

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:
 - C、MingW、Fortran

Level

Samples

・この記事で取り上げたソースコードおよびサンプルプログラムは、付録CD-ROMの¥DOTNET¥SAMPOディレクトリに収録しています。

・SAMPO09.SLN
今回紹介したサンプルのソリューションファイル

¥MINGW
gccでコンパイルしたgccdll.dllとソース(C)

¥MINGW2
gccでコンパイルしたgccdll2.dllとソース(C)

¥MINGW3
G77でコンパイルしたg77dll.dllとソース(Fortran)



MingWは使えるぞ!

新しいプログラム言語を用いて効率的なプログラムを書くというのは、非常に楽しいことです。特にその言語の将来性が期待できる場合には、値上がり保証された株を買うような嬉しさが伴います。しかし、新しい言語に対しては、常にそれに反対する声が敵陣営から巻き起こり、下手をすればこちらの決心がぐらついてしまいます。私の場合、アメリカにいるからだと思いますが、ハイテクブームの終焉時の反Microsoft陣営の惨状を見つ、またJavaの限界を実感し、それよりはるかに正常な市場を形成しているWindowsへの思い入れが一層強くなったのです。

日本の状況が、このようなものではないことも十分に理解しておりますが、「オープンソース」のようなキレイゴトが企業の本音であるとは信じられません。Microsoftに対抗するための捨て身的手段としての「オープンソース」は不純です。

しかし、正直な話、Visual Basic .NETとかC#という.NET系の新しい言語でプログラムを書こうとすると、「.NETで何ができるのか」という利点よりも「.NETで何ができないのか」という欠点のほうが気になります。今、持っている機能は失いたくないという本能的な恐れです。実際、この連載でも「.NETでも従来どおりのことができる」ということが重要な部分を占めてきました。「書式付き入出力」などがよい例ですが、この場合にも「.NETに従来どおりのことをさせてくれる」のはDynamic Link Library (DLL) でした。JavaでもJNI (Java Native Interface) を用いればDLLにアクセスできますが、.NETの場合には、はるかに自然にそれができます。.NET言語のDLLとの親和性は、もっと強調されるべき美点であると思います。

MingW って？

先月も Visual C++ .NET を用いて DLL を作成し、それを Visual Basic .NET で使用しました。これは、Visual

Studio .NET がみなさんのお手元にある前提での話でしたが、Visual Basic .NET一本槍のみなさんはどうすればよいのでしょうか。でも、心配はまったくご無用、Visual C++ .NETを用いなくても、無料で手に入るCコンパイラで、簡単にDLLを作ることができるのです。Windowsに移植されたGNUのCコンパイラには何種類ありますが、その中にWindowsとの親和性抜群なものがあるからです。

Visual Basicユーザーの方は「反UNIX派」が多いと期待しますが、ここではあまりUNIX臭のないMinimalist GNU Win32、略してMingWを使いますから大丈夫。このMingWは、GCC^[注1]を含めた一連のGNUツール群で、非常に軽快で使いやすいので、私も結構活用しています。実際、最近のGCCのバージョンアップは、早い話が「グジャグジャ」で、あちこちで悲鳴が上がっているようです。でも、このMingWのGCCに話を限れば、従来の品質を保っているように思えるのです。最新版のGCCの品質が、Linux上よりもWindows上で優れているなんて、実に興味深い話であると思います。

MingWは<http://www.mingw.org/>で入手できます。12MBほどの最新のMingWのセットアッププログラムをダウンロードして走らせればよろしい。今月の記事の執筆時(5月中旬)では、MinGW-2.0.0-3.exeになっていました。ちなみに私は「C:\MingW」にインス

注1) 従来GCCと言えばGNU C Compilerのことでしたが、最近ではGNU Compiler Collectionを意味します。このGCCに含まれるのが、C/C++コンパイラであるgcc、Fortran 77のg77などです。C++する場合にも、g++ではなくgccで構いません。

トールしています。このセットアッププログラムは、Windowsのレジストリはいじらないようです。「C:\MingW」にインストールするつもりが気が付いたら「C:\MingW\MingW」になってしまったので、ここのファイルやフォルダをエクスプローラで「C:\MingW」

に“よいしょ”ともってきました。乱暴ようですが、MingWの場合はこれでよいのです。このあとでしなければいけないのはPATHに「C:\MingW\bin」を入れることだけです。

ディレクトリ「C:\MingW\bin」には、図1のようなファイルがありました。

図1：MingWで提供される実行ファイル



Column

Cygnus社とMingW

GCCの開発を担当していたCygnus社がRed Hat社に買収されたのは何年前のことです。同社はWindows用にもGCCを含むGNUツールを移植しCygwin Toolsとして提供していましたが、これは今でもRed Hat社のWebサイトから無料で入手できます。

このCygwin ToolsはWindowsでUNIXゴッコをするためのものであり、X Windowまでサポートするかなり本格的なものです。これを用いれば、Windows上で無料でXターミナルを使うこともできるわけです。市販のXターミナルは安くはないので、ありがたいといえばありがたい話です。もっとも、PCで簡単にLinuxできる時代になりましたから、本格的にPC-UNIXする場合にはCygwinよりもLinuxを用いるほうが何かと楽であろうと思います。

これとは対照的に、MingWは「WindowsでGCCする」ためのものです。Cygwinのように、UNIXのコンソールを用いるのではなく、普通のWindowsのコンソール上で直接gccやg77を用いるのです。そして、普通のWin32プログラムを作ってくれます。つまり、UNIX機能をエミュレートするDLLなどは不要なのです。ですから、今月のように、無料で使える普通のCコンパイラとしては、このMingW版のgccが最適なのです。