

無償ツールを組み合わせ
クロス開発環境を構築

セルフ・コンパイルと クロス・コンパイル



ここでは、Windowsが動作するパソコン上にクロス・コンパイル環境を構築し、玄箱向けのオブジェクト・コードを生成する方法を紹介する。クロス・コンパイル環境を構築する方法としては、パソコンで稼働するCygwin上にGNUWingというソフトウェアを導入する方法、およびbinutilsやGCC、newlib、GDBを個別にインストールする方法などがある。

(編集部)

岸 哲夫

いくら玄箱が便利で使いやすいといっても、CPU(Power PC)の動作周波数は200MHz、RAM容量は64Mバイト、フラッシュROMは4Mバイトという非力な環境です。この上で、アプリケーション・ソフトウェアのソース・コードをコンパイル(セルフ・コンパイル)した場合、かなりの処理時間がかかることでしょう(p.56のコラム1を参照)。できれば、これは避けたいところかもしれません。

単純な比較はできませんが、約10年前に出荷されたApple社の「Power Macintosh 7600」に200MHz動作のPowerPCが搭載されていたことを考えると、玄箱の非力さを理解していただけたと思います。

前章で紹介したように、玄箱にVine LinuxやDebian Linuxを導入すると、コンパイルなどの作業を手軽に行え

ようになります。しかし、「それでも時間がかかるのは嫌だ」という方のために、ここでは一般のパソコンで稼働するCygwin上にGNUWingというソフトウェアを導入したり、GCCやGDBなどを個別に導入したりして、玄箱向けのオブジェクト・コードを生成(クロス・コンパイル)する方法を指南します。

なお、ターゲットとなるCPU(実行用CPU)と同じCPUの上でコンパイルし、オブジェクト・コードを生成することを「セルフ・コンパイル」^{注1}、ターゲットとなるCPUとは異なるCPUの上でコンパイルして、ターゲットCPU向けのオブジェクト・コードを生成することを「クロス・コンパイル」と呼ぶことにします(図1)。

組み込みシステムでは、ターゲットとなる機器にディスプレイやキーボードなどの入出力装置が付いているとは限りません。また、CPUの種類によっては、ソフトウェア開

注1：ネイティブ・コンパイルと呼ぶこともある。

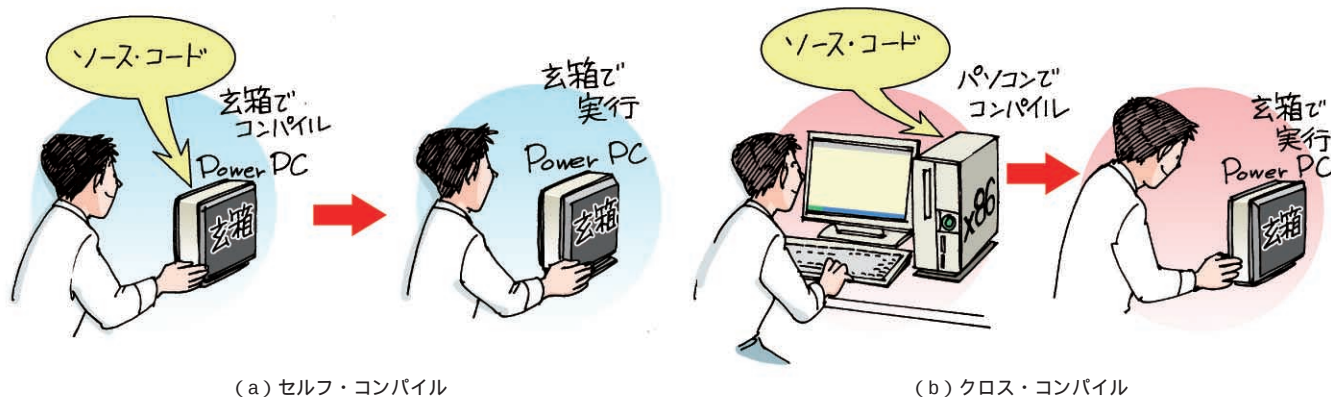


図1 2種類のコンパイル方式

開発用CPUと実行用CPUが同じ場合は「セルフ・コンパイル」、開発用CPUと実行用CPUが異なる場合は「クロス・コンパイル」になる。

発環境が必ずしも整備されていないものもあります。そのため、クロス・コンパイル手法やクロス開発環境がよく利用されています。

● UNIX 系ツールを Windows 上で利用できる Cygwin

ここで、Cygwin について簡単に説明します¹⁾。Cygwin は、GNU の開発ツールを含む、UNIX 系のさまざまなフリー・ソフトウェアを Windows に移植したものです(図2)。実体は Cygwin DLL というライブラリで、これが UNIX のコマンドをシミュレートして、Win32 API に橋渡ししています。結果として、大規模なソース・コードの修正なしに、数多くの重要な UNIX プログラムを容易に移植することが可能となります。

Cygwin は、もともと Cygnus Solutions 社によって開発および保守されていました。その後、Red Hat 社による Cygnus 社の買収に伴い、現在は Red Hat 社の製品となっています。また、Cygwin は GPL(The GNU General Public License)²⁾ に従ったオープン・ソースのフリー・ソフトウェアです。

動作環境は、Windows 95/98/Me、Windows NT/2000 (Win32 環境)、および Windows XP です。安定した動作を望む場合は、NT/2000/XP の環境を推奨します。

現在では容易に Linux を入手できますし、インストールも簡単です。それにもかかわらず Cygwin を使用するメリットとして、以下の項目が挙げられます。

● Linux 専用のコンピュータ・ハードウェアが不要

Linux を使うために専用のコンピュータ・ハードウェアが用意できれば良いのですが、ちょっとした実験や学習の場合には、わざわざ専用のハードウェアを購入する余裕がないケースがあります。パソコン上で Windows と Linux のデュアルブートの設定を行うという手もあるのですが、この作業を煩雑に感じる人もいることでしょう。また、

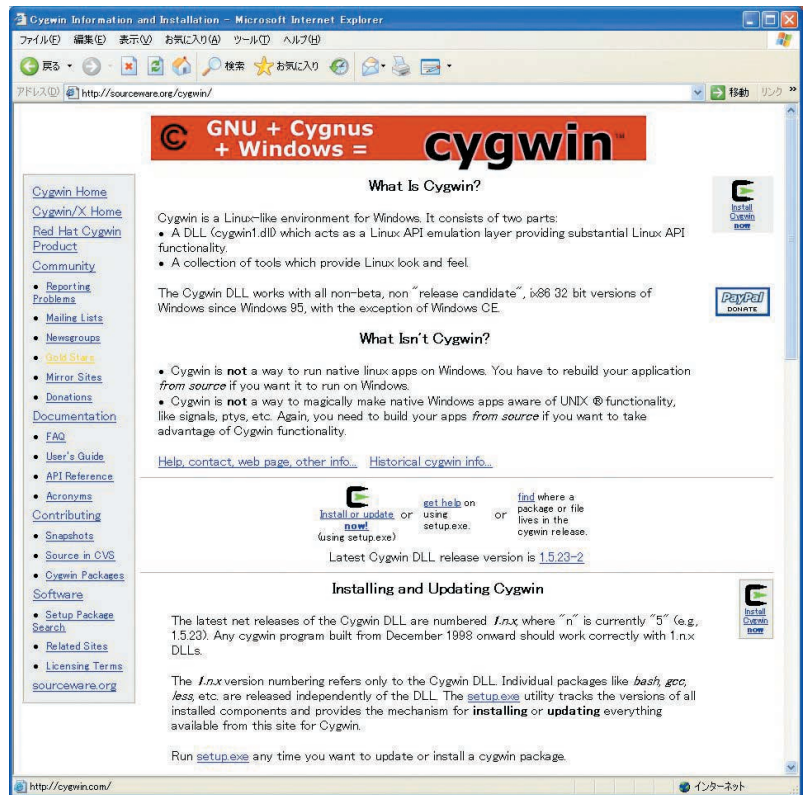


図2 Cygwin の Web サイト

URL は <http://sourceware.org/cygwin/>

デュアルブートのためのツールはときどき誤動作を起こし、場合によってはブートできなくなることもあります。

これに対して、Cygwin であればそのようなことを心配せずに使えます。Windows が稼働するパソコンのハード・ディスクに空き容量があれば、インストール可能です。

● Windows のコマンド・ライン環境を強化できる

例えば一連のバッチ処理を行いたいと考えても、Windows のシェルは MS-DOS 時代からほぼ変わっていません。市販のジョブ・スケジューラを導入してもよいのですが、これだとコストがかかり、ある程度の慣れも必要です。そのようなとき、コマンド・ライン環境の充実した UNIX 互換のシェルを利用できる Cygwin を導入することには、大きなメリットがあります。

● Windows 上で GNU ツールを使いたい

Windows 上でプログラム開発を行う場合、手軽に開発

注2：GPLとは、FSF(Free Software Foundation)が提唱するフリー・ソフトウェアのライセンスの一形態。ソフトウェアの再配布や改変を行っても構わないが、その際にソース・コードを開示する義務が発生する。ライブラリなどの動的な(実行時の)リンクに限り、GPLに従わないソフトウェアの利用を認める「LGPL(Lesser GPL)」というライセンスもある。