

GDB のまとめ (最終回)

連載の最終回である今回は、GDB の概略について、そのまとめと解説をします。

(筆者)

1. GDB のまとめ

GDBでは、デバッグするプログラムをGDBを通じて動作させ結果を確認したり、データの中身を確認したりします。

動作環境はLinuxなどのUNIXをはじめとして、Windowsなどのほかの環境へ移植されていれば、その環境でも使用できます。

● パソコン 1 台でデバッグ可能な場合

パソコン上で、そのパソコンで動作するプログラムをデバッグする(スタンドアロン)場合です。必要なOS上でGDBを動作可能に設定し、デバッグ対象のプログラムを動作させて、動作とデータの中身を確認します。

注意すべきなのはプログラムのコンパイル時にデバッグを可能にするオプション(-g)を付けることです。

● x86系開発用パソコンでターゲット・マシンをデバッグ

ここでx86系開発用パソコンをホスト・マシン、ほかのアーキテクチャで動作するコンピュータ(パソコンや組み込み向けボード)をターゲット・マシンと呼ぶことにします。

二つのマシンはシリアル・ケーブルやEthernetなどで接続します(図1)。この状態でホスト・マシンからターゲット・マシンをデバッグすることをクロス・デバッグといいます。

ホスト・マシン上ではGDBが動作可能であり、ターゲット・マシンではスタブとしてGDBサーバが実行可能です(図2)。

ターゲット・マシンに開発環境やGDBそのものをインストールすることが不可能な場合に、この方法は有効です。

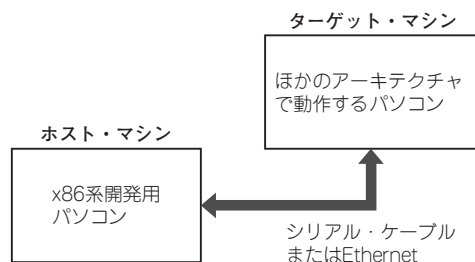


図1 二つのマシンの接続してデバッグする

2. クロス・コンパイル環境

ここまではクロス・デバッガの方式を説明しました。クロス・コンパイル環境の説明についても簡単に説明します。

● クロス・コンパイル環境の作成

先人のおかげでわれわれが手軽に使えるクロス・コンパイル環境として、Linuxを使用したものやCygwinを使用したものなどがあります。

ここではCygwinを使用して、いつも使っているWindowsパソコンにクロス・コンパイル環境を構築する方法を説明します。

CygwinはGNUの開発ツールを含む、UNIXのさまざまなフリー・ソフトウェアをWindowsへ移植したものです。実体はCygwin DLLというライブラリで、これがUNIXのコマンドとWin32APIを橋渡ししています。結果として、数多くのUNIXプログラムを大規模なソース・コードの修正なしに容易に移植することが可能になりました。

Cygwinはもともと米国Cygnus Solutions社によって開発と保守がなされていましたが、米国Red Hat社によるCygnus Solutions社の買収に伴い、現在はRed Hat社の製品となっています。もちろんフリー・ソフトウェアですから、Red Hat社以外の企業も参加しています。

動作環境はWindows 95/98/Me、Windows NT/2000/XPです。安定した動作が必要ならNT系を使用すべきでしょう。

Linuxを使うために専用のマシンが用意できる人はともかくとして、実験的にまたは学習用に使いたい人は専用のマシンを購入する余裕がないかもしれません。さらにデュアル・ブート

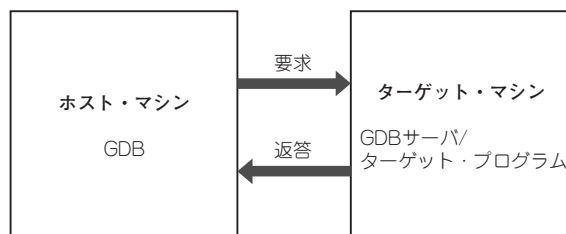


図2 二つのマシン間のデバッグのようす