第1章

コンパイラ，アセンブラ，リンカのインストールから使い方まで

GNUツールを使ったクロス開発環境の構築

本稿では，GCCなどのGNUツールを使った場合の開発フローやソースの入手，ビルドやインストールの方法，そしてコンパイラ，アセンブラ，リンカの使い方について解説する．本誌付属のDVD-ROMにはCygwin環境で動作するビルド済みGCCなどを収録しており，すぐにインストールして使うことも可能である．（編集部）

山際 伸一

はじめに

A 社の CPU には A 社の開発環境が，B 社の CPU には B 社の開発環境が用意されています．一般に CPU には互換性がなく，開発ツールにも互換性はありません．初めて使う CPU は，その CPU アーキテクチャを理解し，開発ツールの使い方も覚えなければならないわけです．

しかし GNU ツールで開発環境を構築すると，CPU を変更しても開発環境が大きく異なることはありません．プログラマは CPU アーキテクチャの理解に専念でき，使い慣れた開発ツールの環境を使い続けることができます．GNU ツールの信頼性とその知名度から，最近では CPU ベンダも GNU ツールを無視できない状況になっています．

本稿では，この GNU ツールを使ってプログラム開発を行う上で必要なプログラム開発環境の構築方法と，知っておきたい基本事項を押さえていきます．本特集の後述の各章では，これらの基本事項を踏まえた上で説明を進めます．読んでいて「あれ？ どうだったかな？」というような疑問が出たときに，この章に戻って読み返すことをお勧めします．

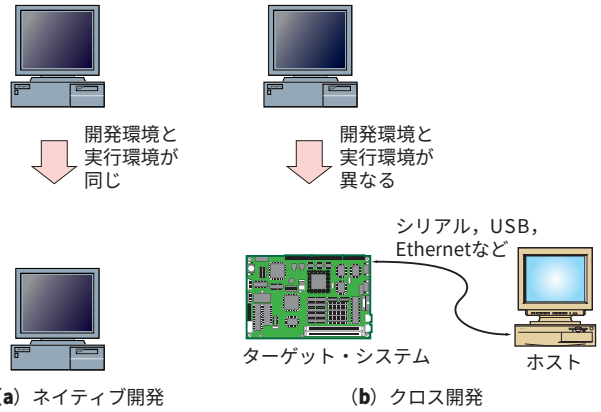

図1　ネイティブ開発環境とクロス開発環境

1. GNU ツールで組み込みプログラミング

● 組み込み開発環境とは

プログラム開発環境には大きく二つの環境があります（**図1**）．一つはネイティブ開発環境です．プログラム開発を行う環境と動作させる環境が同一のものです．例えば Windows 上で Visual Studio を使って Windows 向けのプログラムを開発するようなものです．

もう一つがクロス開発環境と呼ばれるもので，プログラムを開発する環境と動作させる環境が異なる場合です．組み込み開発の多くはこちらの環境となります．プログラム開発に用いられるマシンのことをホスト・マシン，そのプロセッサのことをホスト CPU と呼びます．また，ホスト・マシンで作成されたプログラムを実行するマシンをターゲット・マシン，そのプロセッサをターゲット CPU と呼びます．

クロス開発環境では，CPU アーキテクチャが異なるため，ホスト CPU で開発したターゲット CPU 向けの機械語バイナリを直接実行することはできません．そこで，動作確認を行うための仕組みが必要となります．ホスト・マシンとターゲット・マシンをシリアル・ケーブルや JTAG ケーブルなどの通信ケーブルで接続し，ホスト・マシンで

作成したプログラムをターゲット・マシンにダウンロードして実行します．

● モニタ（GDBスタブ）

一般的にシリアル・ケーブルによる接続では，ターゲット・システム側にも通信プログラムが必要です．これを一般にモニタと呼びます．このモニタはあらかじめターゲット・システムに書き込んでおく必要があります．

本特集では，このモニタに相当するプログラム（GDBスタブ）を各種ターゲットCPUごとに用意し，ターゲットCPUやターゲット・ボードに説明されている方法でモニタを書き込んでおきます．

なお，GDBスタブの構造や作成についても解説しているので，クロック周波数や接続に使用するシリアル・ポートのチャネルを変更するといった改造も容易だと思います．

● ROM化

開発したシステムが，最終的に開発時と同じように通信ケーブルを接続した状態で使うのであれば問題はありませんが，多くの場合，ターゲットは単体で動作するものになるでしょう．そうなると，作成したターゲットCPU向けプログラムは，半永久的に消えないメモリに書き込まれていなければなりません．ターゲットにはROMが搭載されていることが多く，そのROMからCPUが起動するように設計されているのが普通です．

開発時は，ターゲットCPUの初期化に必要な処理をモニタが実行しているので，ユーザの作成したプログラムには初期化処理がなくてもシステムは動作します．しかしROM化した場合はROMモニタが行っていた処理と同じ処理を組み込んでおかないと，ユーザのプログラムは起動しません．

また，一般にROMモニタを介してダウンロードされるプログラム領域は，ROM化時のアドレスと異なるので，プログラムのアドレス・マップを再設定する必要があります．

● GNUクロス開発ツールを使えるようになろう！

以上の説明で，クロス開発を行うための環境はネイティブ開発環境とは異なるツールを使ったり，ROMモニタなどの基本プログラムが必要となることが理解できたと思います．GNUツールを用いたターゲットCPU向けのプログラム開発では，クロス・コンパイラやクロス・アセンブラ，クロス・リンカ，デバッガといった一連のクロス開発ツールを無償で手に入れることができます．しかし，それらはベンダではサポートされていないことが多く，ユーザに多くの基本的な知識が要求されます．本特集は，GNUクロス開発ツールが使えるようになることを目標としています．

2. GNUツールを使ったプログラム開発フロー

まず，ターゲットCPUで動作するプログラムを作成する過程について理解していきましょう．ここでは，ターゲットCPUで実行可能なバイナリを作成するまでにどのようなツールを用いるかについて解説していきます．

GNUツールを使った場合，**図2**に示す開発フローをたどります．この開発フローはホスト・マシンで実行されます．

● プログラムのコンパイルとアセンブル

C，C++などの高級言語で書かれたソース・プログラムはコンパイラを介して，アセンブリ言語で書かれたソー

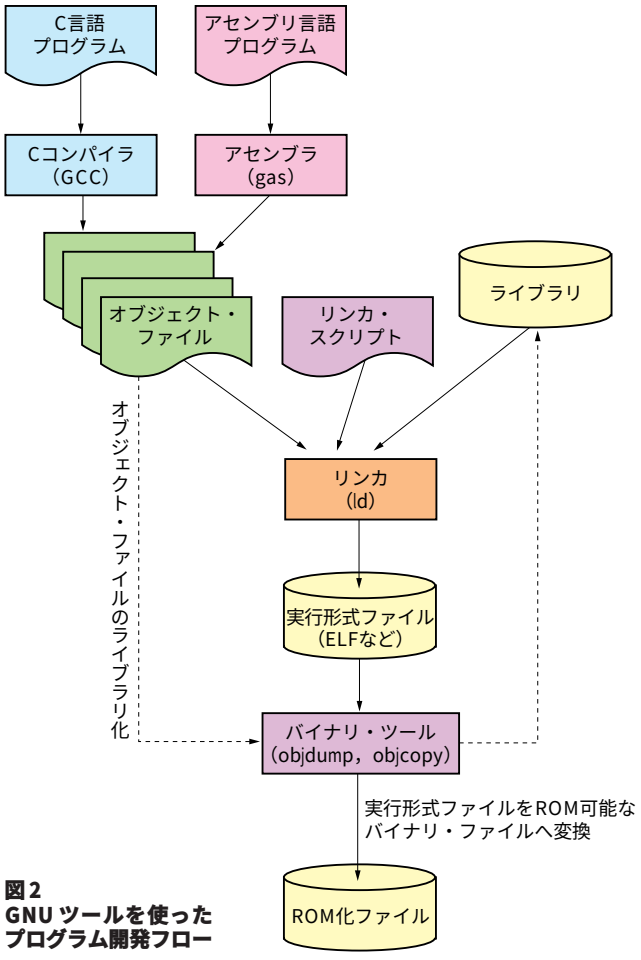

図2
GNUツールを使ったプログラム開発フロー

New Products ―― MontaVista，携帯機器向けLinux OSの新版を発表
MontaVista Software社は，携帯機器向けOS「MontaVista Mobilinux 5.0」を発表した．セキュリティ機能や消費電力管理機能などを強化した．携帯電話や携帯型医療機器，無線POS端末などに利用できる．Mobilinuxは，スマートホンなどで多くの採用実績がある．