Prologue

PC/AT 互換機とは異なる千差万別の組み込みアーキテクチャ

組み込み Linux が難しいと言われる理由

横田敬久

Linux 入門特集に入る前に，ここでは PC/AT 互換機で動作する Linux と，組み込みシステムで動作する Linux の違いについて，いくつか整理してみましょう．（筆者）

1. x86 系 CPU 用 Linux と非 x86 系 CPU 用 Linux

● x86 系＝ PC/AT

x86 系 CPU にもいろいろなものがありますが，上位品種で拡張された命令を使わなければ機械語レベルでの互換性が保たれます．また，CPU ID を判定することで，拡張された命令が使えるかどうかをプログラム自身が判別することも可能です．

CPU が x86 系である場合，現在はそのほとんどに PC/AT 互換アーキテクチャが採用されています（**写真 1**）．8086 や V30 などの 16 ビット CPU であれば，PC/AT 互換アーキテクチャではない独自アーキテクチャの組み込み向け CPU ボードも存在しました．しかし，386 以降の 32 ビット CPU で独自アーキテクチャというハードウェアを（Intel Mac を除き），筆者はほとんど見たことがありません．

PC/AT 互換アーキテクチャは，レガシ・インターフェースであればある程度アドレスが固定されます．また，PCI デバイスのように環境によって割り当てられるアドレスが異なる場合は，デバイスを検索したりベース・アドレスを取得したりするための手続きが規定されています．これにより，バイナリ・レベルで同一のプログラムを実行することができます．

このような状況から，x86 系 CPU 用 Linux とは PC/AT 互換機用 Linux であると言い換えることができます．CPU にもプラットホーム（コンピュータ・アーキテクチャ）にも互換性があるので，KNOPPIX のように 1 枚の CD-ROM から Linux を起動させるようなシステムを実現できるわけです．

● 非 x86 系 CPU には標準プラットホームがない

しかし，非 x86 系 CPU の場合はそのような標準プラットホームがありません．

まず第 1 に，CPU のアーキテクチャが異なれば機械語の命令レベルで互換性がありません．

第 2 に，たとえ CPU が同一もしくは機械語レベルの互換性がある場合でも，プラットホーム・レベルで互換性がないと，一般的には同一のプログラムを実行することができません．

● 非 x86 系 CPU 用 Linux

このような状況から，非 x86 系 CPU 用 Linux は，CPU およびプラットホームごとに，それぞれ個別にオペレーティング・システム（OS）を用意しなければならない状況になっています．そのため PC/AT 互換機のように，インストールするだけですぐに起動する「○○ Linux」というようなディストリビューションがないのです．

もし手元に，Linux が起動する非 x86 系 CPU ボードがあるとすれば，それは誰かが Linux を移植したからだといえます．

2. Linux カーネルとは

カーネルとはメモリやタスク管理機能など，OS の核の機能を示します．

カーネルの世代はバージョン番号で管理されています．現在の Linux の最新バージョンは 2.6 と呼ばれるものです．組み込み用途では安定度を重視して（「枯れている」と呼ぶこともある），カーネル 2.4 が使われているシステムもあります．

カーネルを構造的に分類すると，カーネルとデバイス・ドライバや OS 各種機能などを異なるメモリ空間で動作させるマイクロ・カーネルと，カーネルやデバイス・ドライバを同一のメモリ空間で動作させるモノリシック・カーネルに分けられます．Linux カーネルはモノリシック・カーネルに分類されます．

Linux ではカーネル起動時に必要なデバイス・ドライバも，カーネルの中に組み込んでおきます．例えば，ハード・ディスク・ドライブ（HDD）の中に後述するルート・ファイル・システムを確保する場合は，その HDD へアクセスするためのデバイス・ドライバをあらかじめカーネルに組み込んでおく必要があります．そしてカーネルやドライバをひとまとめにして，vmlinux や zImage などのファイル名でカーネル・イメージ・ファイルを作成します．Linux ではこのカーネルのイメージ・ファイルのことを，単にカーネルと呼ぶこともあります．

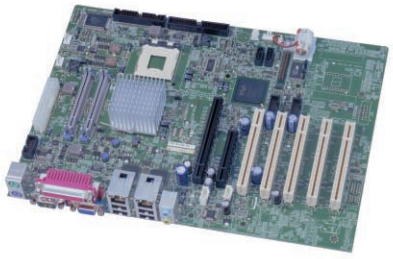
カーネル・イメージ・ファイルは，後述するブート・ローダでメモリにロードされます．

3. ルート・ファイル・システムとは

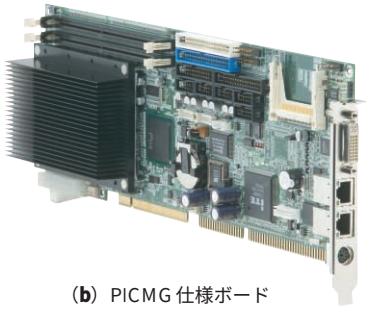
ルート・ファイル・システムとは，Windows でいえば C ドライブに相当する，システムを起動させるために必要なファイル群を格納するファイル・システムです．

一般的には HDD などのストレージ上にルート・ファイル・システムを構築します．しかし非 x86 系システムでは，HDD などのようなストレージだけでなく CPU ボードに実装されてい

KEYWORD ―組み込み Linux，x86 系 CPU，PC/AT アーキテクチャ，カーネル，ルート・ファイル・システム，シェル，ライブラリ，ブート・ローダ



(a) ATX仕様ボード



(b) PICMG仕様ボード

(c) CompactPCI仕様ボード

写真1　PC/AT互換アーキテクチャを採用したボード
ボード形状はまったく異なるものの，いずれもアーキテクチャに互換性がある．

るフラッシュROMに確納する場合も多いようです．特に今回の特集では，最小構成のLinuxシステムを構築するという意味から，外付けのストレージを使わず，ボード上のフラッシュROMにルート・ファイル・システムを作成する事例を解説しています．

また，フラッシュROMには圧縮したイメージを保持し，実際のシステム稼動時はRAM上に展開して読み書きを可能にすることもあります．さらにはネットワーク上に確保することもできます．

PC/AT互換機ではストレージとしてHDDなどを使うのが一般的なので，ファイル・サイズなどの容量的な面はあまり問題とされません．しかし，組み込みシステムではストレージの容量に制限がある場合もあるので，ルート・ファイル・システムのサイズは小さいことが望まれます．

4. シェル/ライブラリとは

● シェルとは

シェルは，MS-DOSでいえばCOMMAND.COM，Windowsでいえば Windows Explorerに相当する，ユーザがOSを操作するときに必要なコマンド入力インターフェース，あるいはグラフィカル操作インターフェースのことを言います．

組み込みシステムでは画面表示装置を持たないものも多いので，シリアル・インターフェースをコンソール入出力ポートとして使い，ターミナルからコマンドを入力してOSを操作します．Linuxのシェルとしてはbashが標準的に使われています．

● ライブラリとは

プログラムが複数ある場合，その中に共通の処理が含まれることが多々あります．そのような共通部分を取り出してひとまとまりとし，各プログラムはそれを呼び出すような構造とします．この共通処理の塊を共有ライブラリと呼びます．Linuxでは共有ライブラリとしてglibcなどが一般的に使われます．

ルート・ファイル・システムのサイズを小さくするには，ライブラリもサイズの小さいものを用意する必要があります．組み込み用途では基本的な機能のみを提供するライブラリで十分なことから，サイズの小さいuClibcなどが好まれるようです．

5. ブート・ローダとLinux

● PC/AT互換機のブート手順

ブート・ローダとはOSをストレージなどの記憶媒体から読み出し，メモリ上にロードして実行を移すシステム起動用のプログラムを指します．

PC/AT互換機にはBIOS(Basic Input/Output System)があり，これがブート・ローダの役目も担当します．BIOSは起動対象デバイスのブート・セクタをメモリに読み込み，そこにジャンプします．ブート・セクタにはOSのインストール時にそれぞれのOS用のローダを格納しておくので，そこから実際のカーネルがロードされてOSの起動処理に入ります．

このようにPC/AT互換機では，2段ロケットのようなイメージでOSがロードされます．

● 非x86系システムのブート手順

しかし，非x86系システムではBIOSが存在しないものも多く，それぞれ独自の方法でOSを起動させることが多いようです．また組み込みシステムでは，RedBootやU-Bootといったブート・ローダを採用しているものも多く，これらのブート・ローダがカーネル・イメージをメモリに転送します．

最小構成システムではHDDなどのストレージを持たない場合も多く，その場合はルート・ファイル・システムをCPUボード上のフラッシュROMに格納しますが，カーネル・イメージ・ファイルも同じようにフラッシュROMに格納します．カーネル・イメージ・ファイルがフラッシュROMにある場合，ブート・ローダの基本作業は，フラッシュROMからRAMへのメモリ間転送だけとなり，非常に簡単なブート・ローダでOSを起動させることができます．

よこた・たかひさ

News Flash ―― Honeywell，宇宙ミッション用のOSとしてWind River Linuxを採用
Honeywell Aerospace社は，NASAのNew Millennium Program Space Technology 8(ST8)におけるディペンダブル・マルチプロセッサ用OSとして，Wind River Linuxを採用した．宇宙機の中でデータ分析を行い，制御方針などを決めることが可能となる．