

マイコン・プログラマのための
Linux 超入門

第11回

同じソースでいろんなチップ/ボード依存回路が動かせる
最新のしくみデバイス・ツリー

宗像 尚郎

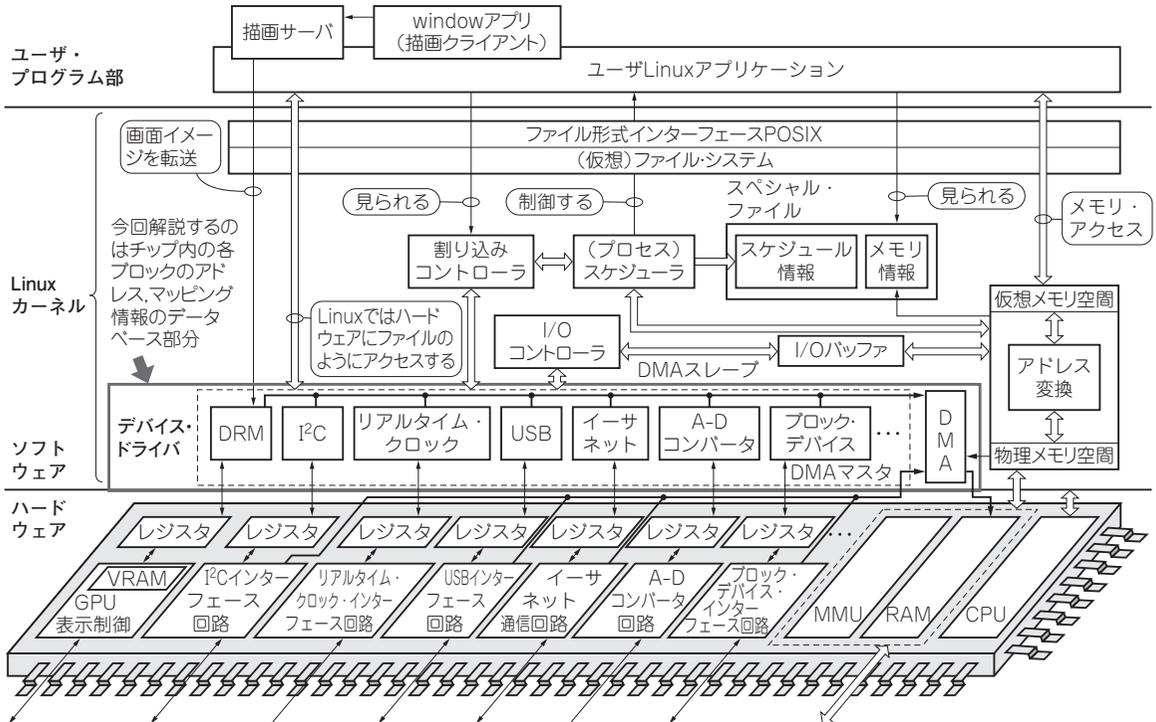


図1 今回解説すること…多様なハードウェアを一つのカーネルでサポートするためのしくみ
カーネル起動時にはチップやボードの固有のコンフィグレーション情報をカーネルに伝える必要がある

Linuxは、x86やARMなどのいろいろなCPUアーキテクチャで動作します。さらに、パソコンや組み込みCPUボードなどいろいろなボードでも動作するマルチプラットフォームに対応したOSです。今回は、Linuxがマルチサポートを実現するしくみや、ARM製品をサポートする上で対応が必須となったデバイス・ツリーについて紹介します(図1)。(編集部)

● OSの原則…特定マシン向けのソフトウェア

コンピュータの基本ソフトであるOSは、原則として特定のCPUアーキテクチャ向けに開発されます。

たとえば、Windowsは基本的にインテル・アーキテクチャのCPUが搭載されたコンピュータ向けのOSです。Mac OSも、そのバージョンがリリースされた

時に製品に使われているCPUのアーキテクチャでしか動きません。

WindowsパソコンやMacの場合は、ハードウェアとOSの間にBIOSと呼ばれるハードウェア抽象化プログラムが入っているので、OSからみるとハードウェアは共通のものとして考えることができます。

● LinuxはいろいろなCPUやボードでも使える…マルチプラットフォーム対応

一方、Linuxカーネルはマルチ・プロセッサ・アーキテクチャ、マルチプラットフォームで動作することを前提に設計されているので、特定のプロセッサや特定のボード構成向けという前提はありません。

表1は三つのLinuxカーネルのバージョンについて

第1回 電源投入からカーネルが起動するまでの動作(2013年11月号)

第2回 仮想アドレスのメカニズム(2013年12月号)

第3回 高性能処理のキモ! プログラム実行順序決定のメカニズム(2014年1月号)