

アプリケーション、プロセス、スレッドの関係性から役割分担まで

# CPUとOSによる実行環境の抽象化①…基礎知識

内田 公太

表1 実行環境の抽象化におけるCPUとOSの役割分担  
汎用OSの抽象化は複雑でCPUに求める機能が多い

項目	汎用OSの場合	組み込みOSの場合
システム・コール	CPUによるOSとアプリケーションの権限分離+専用命令による呼び出し	関数呼び出し
アドレス変換	CPUによるアドレス変換機構を活用	なし
仮想記憶	CPUによる例外トラップ+OSの例外ハンドラ	なし

ソフトウェアの実行環境では、CPUだけでなくOSも重要な役割を果たします。本章では、CPUとOSの関係、OSが何をするのかを説明します。

本章におけるOSという言葉は、カーネルの意味で使っています。典型的には、メモリやタスクの管理を提供するようなソフトウェアです。あるいはCPUの特権モードで動作するソフトウェアという説明でもよいかもしれません<sup>注1</sup>。

## 実行環境の抽象化

### ● アプリケーションに実装の詳細を隠すこと

CPUとOSが協力し、抽象化された実行環境をアプリケーション(プロセス)に提供します。ここで言う抽象化とは、実装の詳細を隠すという意味です。例えば、ファイルが保存されている場所がSSDなのかDRAMなのか、ネットワーク・ソケットなのかを区別することなく、ただwriteシステム・コールにファイル・ディスクリプタとバッファとバイト数を渡せば任意のバイト列を書き込める、というようなことです。

注1: この説明は、CPUに動作権限を変える機能があり、それを活用するOSを動かすときだけ当てはまります。

注2: 組み込みと言ったとき、ラズベリー・パイのようにPCと同じ構成のコンピュータ、あるいはスマートフォンやタブレットなどを使った専用システムを指すことがあります。しかし、これらはどちらかといえば汎用OSの世界です。本章ではArduinoや1個数百円程度のマイコンを使ったシステムを組み込みと考えます。

あるいは、OS上で他にどんなアプリケーションが動いているかを意識させず、自アプリケーションとOSだけが存在するかのように見せる、という抽象化もあります。物理メモリ上にはさまざまなデータやプログラムが存在し、空き領域は飛び飛びになっています。抽象化された実行環境であれば、アプリケーションはそんなことを気にせずOSに必要なメモリ量を要求するだけで連続した空きメモリ領域を確保できます。

### ● 汎用OSの抽象化は複雑でCPUに求める機能が多い

CPUとOSがどう役割分担するかは、汎用OS(PCやスマートフォン向け)と組み込みOS(マイコン向け)で大きく異なります。例外はあると思いますが、おおよそ表1のようになっています。

本章では、より複雑で説明すべきことが多い汎用OSと、それを動かすためのCPUに着目します。

#### ▶ 汎用OSの場合

汎用OSでは、動作するアプリケーションが決まっていけないので、信頼できないアプリケーションを動かす可能性があります。そのため、OSとアプリケーション、あるいはアプリケーション同士を分離し、不正なアプリケーションからOSや他のアプリケーションを保護する必要があります。そのため、CPUに求める機能が多くなります。

#### ▶ 組み込みOSの場合

組み込み<sup>注2</sup>の世界は、動作するアプリケーションが事前に決められます。そのため、アプリケーションを含めてシステム構築時に検証され、信頼されたアプリケーションのみが動きます。アプリケーションが不正な動作をしないという前提であれば、OSと同じ権限で動作させても問題ありません。そうすることでOSやアプリケーションの保護に関するオーバーヘッドを削減できるので、性能の低いCPUを選択でき、コストを下げられます。