

# Linux/Windows CE/T-Kernelのデバッグに必要な機能

辻 邦彦

従来の組み込みシステムは仮想記憶を使用せず、単一のメモリ空間を使ってプログラムが組まれていた。しかしLinuxやWindows CE, T-Kernel/SEでは仮想記憶を使った複雑なOSとなっている。このようなOSを採用したシステムをJTAGエミュレータを使ってデバッグするにはどんな機能が要求されるのか。ここではそのしくみについて解説する。  
(編集部)

近年の組み込みシステムでは、Linux/Windows CE/T-Kernelなどの大規模OSが採用されるようになってきました。ソフトウェアの規模が増大したために開発を効率化したいというのが主な理由です。JTAGエミュレータなどのICEを使わなくてもデバッグができるよう、これらのOSでは当初からデバッグ支援機能が含まれています。しかし実際の機器開発において、内蔵デバッグ支援機能だけでは機能不足になることがあります。そのために近年のICE、特にJTAGエミュレータでは、積極的にこれらOSへの対応が行われています。

本稿では、京都マイクロコンピュータ(以下KMC)のPARTNER-Jetを例にして、JTAGエミュレータにおけるOS対応について解説します。またOSは、ソース・コードなども含め、多くの読者の方が入手しやすいLinuxを中心に説明します。

## 1. 大規模OSを用いたときのソフトウェアの特徴

～何が今までの組み込みシステムとは違うのか?～

### ● ROM化されたプログラムではなく、ファイル化されたプログラム

Linuxなどを採用する以前の一般的な組み込みシステムでは、実行プログラムはROM化(もしくは同等の処理)されていました[図1(a)]。この場合、JTAGエミュレータなどのICEでのデバッグは難しくありません。通常なら、デバッグにROM化したプログラムをダウンロードし、デバッグしたい個所にブレークポイントを設定し、CPUを実行させるだけで、エミュレーション実行とデバッグが可能になります。

しかしLinuxでは、その多くのプログラムはROM化されていません。すべてのアプリケーション・プログラムは、OSがファイル・システムからメモリに読み込んで実行します。またモジュール化したデバイス・ドライバも、同じようにOSがファイル・システムからメモリへプログラムを転送して組み込みます[図1(b)]。

このため、今までのようにデバッグからデバッグしたいプログラムをダウンロードしてブレークポイントを設定して実行、ということができません。

### ● 仮想記憶を使って多重化されたメモリ空間

今までの一般的な組み込みシステムでは、メモリ空間はただ一つでした。CPUのスーパーバイザ・モードでしかアクセスできない特権空間を特定のアドレス領域に設けること

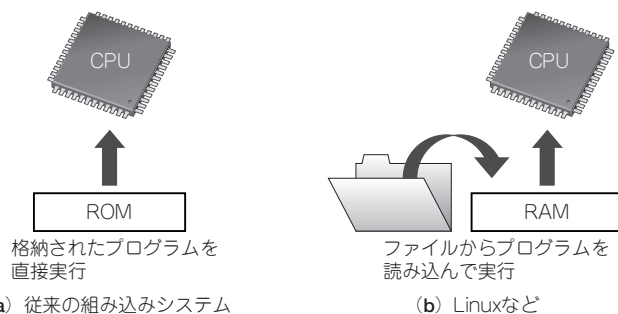


図1 プログラムの格納位置と実行位置