従来のPC/AT用BIOSを置き換えるUEFI仕様の概要

藤原 尚伸，高橋 泰博

UEFI仕様のBIOSは，32ビット以上のCPUに対応して開発された．UEFIを利用すると，大容量ハード・ディスクの対応や，クロス・プラットホームであること，事前処理を最小限にできる，BIOSの設定領域の拡大，といったメリットがある．本稿では，UEFIの概要を解説し，例として64ビットCPUでの処理の流れを紹介する．

1．BIOSの概要と問題点

x86のPC/AT互換BIOSは，その基本構造を大きく変えずに互換性を維持しながら，新しいCPUやデバイス（VGA，CD-ROM，PCI，USBなど），PnPやACPIといった新しい仕様に対応してきました．CPUは16ビットから64ビットまで拡張されましたが，BIOSのインターフェースは，基本的には8086のころから存在する16ビットのリアル・モードのままです（PCI BIOSなど32ビット・インターフェースもあるが），OS起動時もこのモードで動作します．

リアル・モードで使用可能なアドレス領域は1Mバイトです．BIOSは原則1Mバイト以下のこの空間（**図1**）でOSに対してランタイム・インターフェースを提供します．そのためコード・サイズをできるだけ小さくできるよう，Cなどの高級言語ではなく，主にアセンブリ言語を用いて作成されてきました．これは，BIOSに対して構造化やオブジェクト指向などのプログラミング手法を適用する妨げとなり，BIOSそのものの拡張性の足かせとなっていました．そこでUEFI仕様でBIOSのフレームワーク自体を再定義する取り組みがされました．

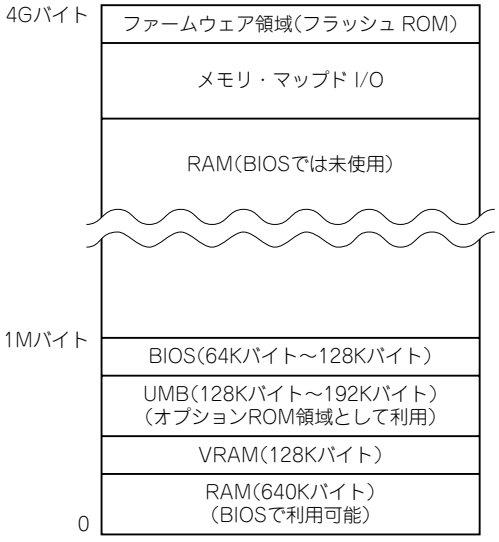

図1　従来のBIOSのメモリ利用例

従来のBIOSは，アドレス領域が1Mバイトしかなかった8086のころからの構造を互換性維持のために引き継いでいる．1Mバイトの狭い領域で新しいデバイスや機能のサポートを行うため，領域の使用を節約する努力が必要．
互換性の問題により，1Mバイトより上の領域を積極的に利用できず，ほとんどが未使用である．一番上のファームウェア領域にはBIOSが格納されているフラッシュROMのイメージが見える．BIOSはここから1Mバイト以下のBIOS領域にコードをコピーして実行する．

図2 UEFIのメモリ利用例

1Mバイト以上のアドレス領域を主に利用し，領域不足に悩まされない．UEFIとしてアドレス1Mバイト以下の領域は利用されないが，CSMで従来のBIOSとの互換性維持の目的で利用される．