

Windows Vista 時代の デバイス・ドライバ開発

第6回

WDF(KMDF)の概要と解説

日高 亜友, 川出 智幸, 相良 徹

第6回目となる今回はサンプル・ドライバをビルドし、インストールするまでの手順をステップごとに解説する。実際にUSBフラッシュ・メモリを使ったKMDFのサンプル・ドライバを、別のUSBターゲットに置き換えてテストする方法についても解説しているので参考にしてほしい。
(編集部)

今回からWDF(Windows Driver Foundation)について解説します。WDFは、Windows VistaやWDK(Windows Driver Kit)とともにMicrosoft社が導入する新しいドライバ・モデルを実現するためのデバイス・ドライバ・フレームワーク(プログラミング・インターフェースであるライブラリ、クラス)と、それをサポートする開発ツールが含まれます。

Windows NT3.1の登場から現在まで、Microsoft社が提供してきた32ビット系Windowsのデバイス・ドライバを記述するためのデバイス・ドライバ・インターフェース(DDI)は、それほど大きく変わりませんでした。あえて挙げれば、Windows2000が登場したときに、プラグ&プレイと電源管理をサポートする仕組みを追加するとともに、Windows98系とデバイス・ドライバのプログラミング・インターフェースを共通化するために導入された、WDM(Windows Driver Model)が大きな変更点でした。そしてWindows Vistaのリリースに合わせて公開されたWDKでは、従来のWDMに加えて、WDFによるデバイス・ドライバの記述と実装を新しくサポートするようになりました。

今までのデバイス・ドライバ・モデルには、汎用的なWDMのほかに、Windows NT時代から続く、ストレージやネットワーク、オーディオなどの各デバイス・クラスに対応した、多くのデバイス・クラス固有のドライバ・モデルがありました。途中で追加されたWDMは、非同期入力やドライバ階層化、プラグ&プレイ、電源管理、WMI(Windows Management Instrumentation)といった機構を提供しています。

デバイス・クラス固有のドライバ・モデルは一般に、Microsoft社が作成したポート・ドライバと、独立系ハードウェア・ベンダが作成したミニポート・ドライバの組み合わせで構成されています。このような従来のデバイス・

ドライバ・モデルには、次のような問題点がありました。

- 汎用WDMドライバ・モデルは低レベルで複雑である
- DDIは元々はOS(オペレーティング・システム)記述用で、デバイス・ドライバ開発用途を目的として設計されておらず、またバージョン管理もされていない
- ミニポート・モデルの数が多すぎる
- ほとんどのドライバをカーネル・モードで実行する必要がある
- ドライバ・モデルの数が多いため、テストと検証ツールの効率的な開発と運用が難しい

なかでもプラグ&プレイと電源管理機能は、同じ目的を実現するために同種のカーネル・レベル・コードがさまざまなプログラマによって書かれています。それが開発者の負荷でもあり、トラブルの原因ともなっていました。このような反省点からWDFが検討され、以下の目標を掲げたドライバ・モデルとして開発されました。

- ドライバ・モデルをシンプルで柔軟にする
- ドライバ・モデルとDDIをOSから独立させる
- バージョン管理と複数バージョンの共存サポート
- 一貫性と拡張性の確保
- 多くのデバイス・ドライバをユーザ・モードで作成可能にする
- 高水準言語の使用を可能にするドライバ・モデルの導入
- DDIは検証と解析を簡易化できるように設計
- 将来はドライバ部分を隔離させることを可能にするドライバ・モデル

これらの目標がすべて実現されたわけではありません。Microsoft社の技術者によると、ネットワーク・デバイスやディスプレイ・アダプタなどの一部を除いて、すべてのドライバはユーザ・モードで実装されるべきだという考え

表1 WDF がサポートする OS

アーキテクチャ	wlh	wnet	wxp	w2k
KMDF				
UMDF				

wlh Windows Vista と Windows Server™ Longhorn (x86, x64, ia64)
 wnet Windows2003 Server SP1(x86, x64, ia64)
 wxp Windows XP SP2(x86)
 w2k Windows2000 SP4(x86)
 サポート
 未サポート

方があるようであり、その意味でも WDF の導入は、Windows におけるユーザ・モード・ドライバ導入の第一歩であると言えます。

1. WDF の概要

WDF は、カーネル・モードで動作するドライバのための KMDF(Kernel-Mode Driver Framework)とユーザ・モードで動作するドライバのための UMDF(User-Mode Driver Framework), それらをサポートする開発ツールから構成されます。基本的に、KMDF は Windows2000 以降の OS をサポートし、UMDF は Windows XP 以降のサポートを行います(表1)。

カーネル・モードで動作する KMDF は、WDM で実現してきた機能を置き換え、なおかつデバイス・クラス固有のドライバにも対応させることを目標としています。KMDF や UMDF も、WDK(ビルド 6000)に含まれるバージョンはそれぞれ 1.5 です。WDF の理解にあたって一番分かりにくいので注意することは、まだ WDK/WDF が開発途中であり、その使用には制約や条件があることです。WDK に含まれる PREfast や SDV といったツールは、WDF での利用を前提に開発されたものですが、これらの利用についても、まだいくつかの制約があります(表2)。

WDF 共通のデバイス・ドライバ・モデルを記述するために最適化されたオブジェクト・モデルが、KMDF と UMDF には導入されています。しかし、モデルは共通でも実装の形態は異なります。WDM と同じく KMDF は C で記述する .SYS のコードとして実装しますが、UMDF は VC++ で記述する DLL であり、これらの間ではソース・コードの流用すらできません。

表2 WDF がサポートするドライバ・モデル

デバイス・クラスまたは ドライバ・モデル	KMDF	UMDF	SDV	PREfast
ウイルス・フィルタ				
特定クラスの USB や IEEE 1394 ドライバ				*
ディスプレイ・アダプタ				
モデム				
HID				
非 PnP ドライバ				
ISA, PCI, PCMCIA, USB, SD				
NDIS, TDI				
プリンタ				
スキャナ, WIA				
SCSI				
ビデオ・キャプチャ				

* カーネル・モード・ドライバ固有の警告メッセージを無視する必要がある

2. KMDF

本節では、主に KMDF について解説していきます。KMDF をよく知るために、まず UMDF と比較してみましょう。

UMDF は、WDF の導入で新たに用意された、ユーザ・モードでデバイス・ドライバを動作させ、またそれを記述するための仕組みですが、その実体は COM 形式の DLL です。

しかし、アプリケーションからは直接呼ばれず、常にカーネルから呼び出される点が通常の COM とは異なります(図1)。

一方、KMDF の実体は、従来のカーネル・モード・ドライバを記述する仕組みである WDM を利用して、その上

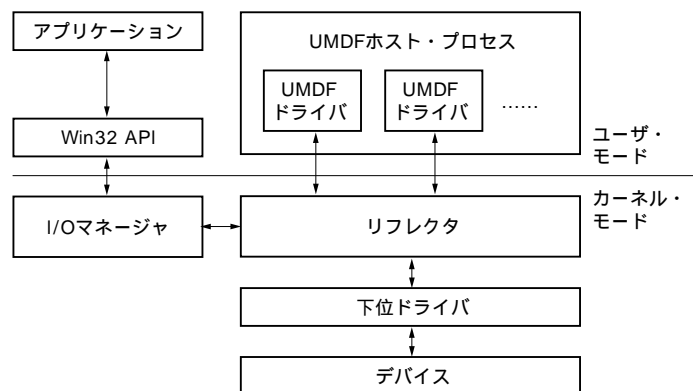


図1 UMDF ドライバの構成概要