

4P-1 入出力ドライバ分離を可能とする ビルディングブロック型OS構築方式

中 箴 恵 丈 近 藤 恵

((株)日立製作所システム開発研究所)

1. はじめに

ワークステーションの普及に伴い、多種多様な入出力デバイスのサポートに対するニーズが強い。一方、従来の入出力ドライバー体型OS (Operating System)構築方式ではOS自身が入出力ドライバを容易に追加することのできる構造を持っていないため、ユーザが新規に開発した入出力ドライバをシステムに組み込むことが困難である。そこで入出力ドライバを分離し、ハードウェア構成に合わせて入出力ドライバを逐次的に追加することのできるビルディングブロック型OS構築方式を開発した。本報告では、その方式の特徴および実現方法を述べる。

2. 従来方式の問題点

ワークステーションに接続する入出力デバイスの種類が多く、またユーザ自身が開発した入出力デバイスを接続する機会も多いことから入出力ドライバ組み込みの自由度が要求されている。しかし、従来、ワークステーションのOSを構築する場合、メモリ管理やタスク管理等のOS基本部分であるカーネルと入出力制御を行う入出力ドライバの全てを準備してからコンパイルし、一括してリンクするドライバー体型の構築方式を採用していた。従来方式では以下のような問題点があり、入出力ドライバの追加や組合せ変更が容易ではない。

- (1) 入出力ドライバの組合せを事前に決定し、その組合せに適合したOSを準備しておくことが不可能である。なぜなら、入出力ドライバn種類とカーネル1種類があると、入出力ドライバの組合せに応じてOSを準備したとすると、 $n \times C_n$ 種類のOSが必要となり非現実的な数になる。
- (2) ドライバー体型ではOSを構築する場合、カーネルや入出力ドライバのオブジェクト形式プログラムが不可欠であるが、ユーザがそれらのオブジェクト形式プログラムを入手することは困難である。
- (3) 新たな入出力デバイスの接続等、ハードウェア構成の変更が生じてユーザサイトで入出力ドライバの追加あるいは組合せ変更を行うことが不可能である。

そこで入出力ドライバを分離し、逐次的に追加することのできるビルディングブロック型OS構築方式を開発した。

3. ドライバ分離方式の基本的な考え方

ワークステーションのようにマルチユーザ・マルチタスク環境下で動作するシステムにおいて入出力ドライバを開発する場合、入出力ドライバは、入出力デバイスを制御するだけでなく、応用プログラムとの同期制御、割り込み制御あるいはタイマ監視制御等、OSの基本部分であるカーネル制御機能を利用しなければ開発不可能である。入出力ドライバとカーネル間で相互の機能を呼び出すために互いの外部名称を参照している。分離して開発された入出力ドライバとカーネル間の外部名称のアドレス解決のために以下のようなメカニズムを利用している。

- (1) ドライバからカーネルへのインタフェース (図1参照)

入出力ドライバをカーネルから分離して開発する場合、入出力ドライバの参照するカーネル外部名称のアドレス解決が問題となるが、それを以下のような2段階アドレス解決法にて解決する。

- (a) 入出力ドライバの参照するカーネル外部名称(カーネルのルーチンやデータ)を番号に変換するドライバ実行時ライブラリと入出力ドライバを一緒にコンパイルする。コンパイル時点ではカーネル外部名称は、アドレス未解決の状態である。

- (b) 入出力ドライバ実行時にドライバ実行時ライブラリは、ソフトウェア割り込み命令を利用しカーネルに番号を渡してカーネル外部名称に相当するアドレスを解決する。

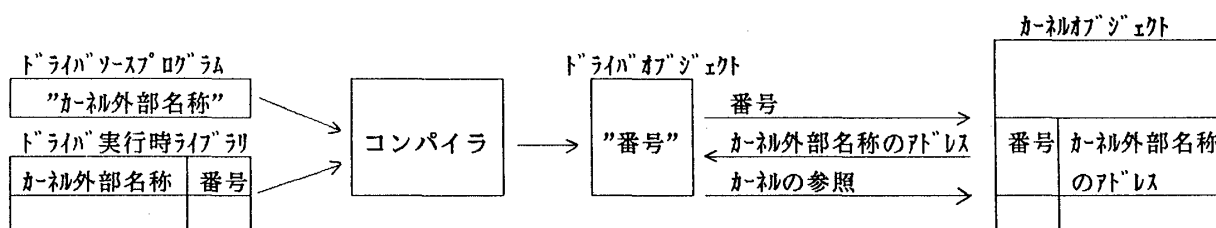


図1 ドライバ分離方式における2段階アドレス解決法

(2) カーネルからドライバへのインタフェース

カーネルの参照する入出力ドライバの外部名称のアドレスは、以下のように解決する。

- (a) システム立ち上げ時にカーネルは、各ドライバを初期化する。その時点でカーネルと各ドライバのハンドシェイクにより各入出力ドライバの外部名称（ドライバの read、write 制御ルーチン等）のアドレスを求めて、カーネルはドライバ制御テーブルを作成する。

4. 入出力ドライバ分離の実現方式

入出力ドライバの開発から組込みまでの概要を図2に示す。図に示すように本方式は、以下の3つのフェーズからなる。

(1) コンパイル

カーネルのドライバインタフェースを規定するドライバ実行時ライブラリと一緒に入出力ドライバをコンパイルし、カーネルと分離した形態でオブジェクトファイルを作成する。

(2) オブジェクトファイルの追加

入出力ドライバの名称や動作空間などのドライバ構成定義情報を記述した構成定義ファイルを用意することにより、入出力ドライバのオブジェクトファイルをシステムに組み込む。

(3) ローディングと実行時のアドレス解決

構成定義ファイルに従って、カーネルと必要とする全ての入出力ドライバのオブジェクトファイルをメモリ空間にローディングする。

カーネルが入出力ドライバを初期化した時点で各ドライバからのアドレス情報の報告を受けて、カーネル内にドライバ制御テーブルを作成する。

入出力ドライバがカーネルのサービスを要求した時点でドライバ実行時ライブラリがカーネルに割り込んでアドレス解決を行う。

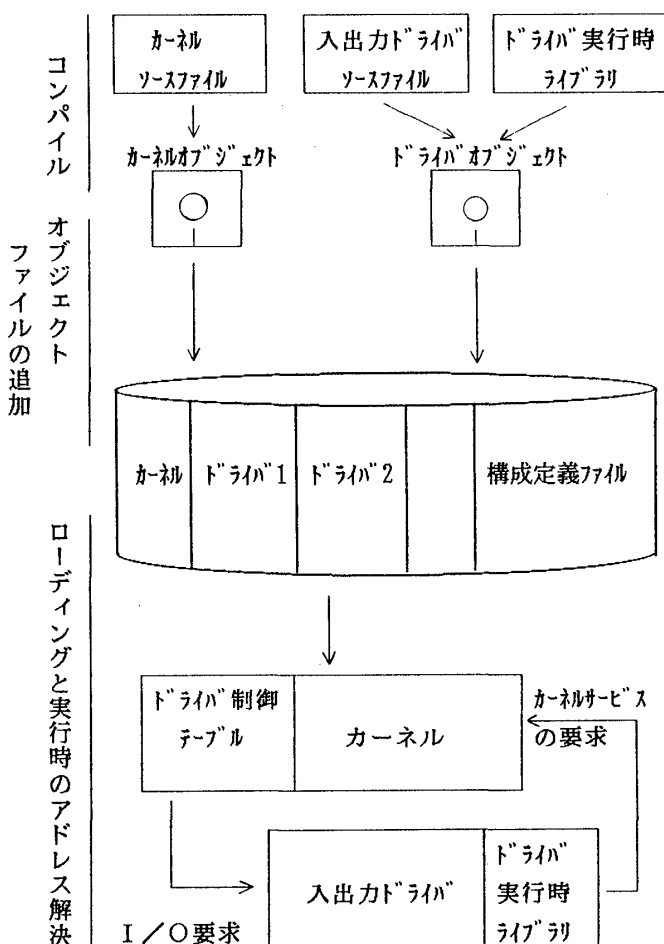


図2 入出力ドライバの組み込み方式

5. おわりに

本方式におけるドライバ分離方式は、以下の特徴を持つ。

- (1) ユーザが開発した入出力ドライバをそのドライバの構成定義を記述したファイルを付加することにより、逐次システムに追加することができる。
- (2) カーネルのオブジェクト形式プログラムが入手できなくても入出力ドライバの開発が可能である。
- (3) ハードウェア構成に応じて稼働中のシステムに入出力ドライバの追加あるいは組合せ変更が可能である。