

4N-2 OS/omicron ウィンドウ・システムの設計

堀 素史, 並木美太郎, 高橋延匡
(東京農工大学 工学部 数理情報工学科)

1. はじめに

オペレーティングシステム(以下, OS)のアーキテクチャを考える場合, マンマシンインタフェースやデバイスインタフェースは計算機の性格の一部を表わす重要な領域である。ユーザにとって使い勝手が悪かったり, アプリケーションを構築する際に適切なデバイスインタフェースが用意されていないと, どんなに高性能な計算機であってもその生産性や使いやすさは低下してしまう。そのため, ワークステーションなど, 計算だけでなく知的生産の道具としても利用される計算機のOSは, この部分を積極的に支援しており, ウィンドウ・システムなどの形で実現している。

OS/omicron(omicron)ウィンドウ・システムは, 我々が開発しているOS, OS/omicron [1] 上で動作するウィンドウ・システムであり, マンマシンインタフェースやデバイスインタフェース自身の研究が行いやすいように, これらを司る部分の機能の追加や変更が容易に行える構成となっている。

2. 設計方針

OS/omicronウィンドウ・システムは次のような設計方針をとっている。

2.1 インタフェース部分を自由に構築できる構成

デバイスと人間との間のマンマシンインタフェースの研究においては, デバイス自身の特性の研究もその対象となる。そこで, ウィンドウのような仮想デバイスにおいて, スクロールバーの表示形態やメニューの方式, 表示位置など, ウィンドウの視覚構成を自由に定義できる設計とする。この結果, 仮想デバイスの特性自身の研究が行えるようになる。

同時に, ウィンドウ・システムとアプリケーションとの間のデバイスインタフェースにおいても, 呼び出しの仕方や機能を自由に登録できる設計とする。

これらの機能により, マンマシンインタフェースやデバイスインタフェースの研究のサポートが行なえる。

2.2 デバイスの抽象化によるアプリケーション開発のサポート

ウィンドウ・システムがデバイスを管理し, アプリケーションに対してデバイスインタフェースを提供する際, ハードウェアに近い形式と共に, より抽象度の高い形式

も提供する。これにより, アプリケーション側が独自にデバイスの形式を設計する必要がなくなり, アプリケーションの開発が行いやすくなる。また, 体系化されたユーザインタフェースの構築が容易になる。

2.3 ハードウェアに独立したシステム

ウィンドウ・システムが管理するデバイスは同一マシン内のデバイスに限らないようにして柔軟性や拡張性を持たせている。実際には次のようなデバイスを扱える設計にする。

- ・マルチディスプレイ
- ・ネットワークを通した他のマシンのデバイス
- ・通信回線を通したインテリジェント端末上のデバイス
- ・異なるOS上のデバイス

3. OS/omicronウィンドウ・システムの構成

OS/omicronウィンドウ・システムは, OS/omicronがOSの機能を拡張するため設けているユーザ拡張部に置かれ, ウィンドウ・システムを利用するユーザからのスーパーバイザコール(以下, SVC)によって機能を利用される。

OS/omicronウィンドウ・システムは, 図1のような構成であり, 大きく2つに分けることができる。

3.1 アプリケーションインタフェース部

アプリケーションとのインタフェースをとる部分であり, SVCハンドラとライブラリから構成される。

(1) SVCハンドラ

SVCハンドラは, ウィンドウ・システムの機能をアプリケーションに提供するデバイスインタフェースを司る部分である。このインタフェースは仮想化されておりマシン独立なものである。したがって, このSVCを利用して作成されるアプリケーションは, 他のマシンへの移植に際して変更の必要はない。

(2) ライブラリ

ライブラリとは, 仮想デバイスドライバが提供するハードウェアの形式に近いデバイスインタフェースを抽象化し, より高機能なSVCを提供する部分である。したがって, SVCはハードウェアの仕様に近い低レベルなものから, ライブラリレベルのものまで幾つかの階層に分かれる。

ライブラリレベルのSVCは変更が容易な構成とし, この部分を入れ替えることによってユーザインタフェース

A design of "OS/omicron window system"

Motofumi HORI, Mitarou NAMIKI and Nobumasa TAKAHASHI

Department of Information Science, Faculty of Technology,
Tokyo University of Agriculture and Technology

の部分の変更が可能となる。SVCの例を表1. に示す。

3. 2 デバイスハンドラ部

デバイスハンドラ部は、物理デバイスドライバ、仮想デバイスドライバからなる。

(1) 物理デバイスドライバ

物理デバイスドライバは、実際に個々のデバイスを制御する部分である。デバイスには、キーボード、マウス、ディスプレイ、タブレットなどが挙げられる。

(2) 仮想デバイスドライバ

仮想デバイスドライバはマシン独立なインタフェースをSVCハンドラに提供する部分である。また、物理デバイスドライバはそのデバイスについての制御のみを行っているが、仮想デバイスドライバではこれらを統合して扱うようにする。つまり、マルチディスプレイを扱えるようにしたり、マルチタスク、マルチプロセッサでの利用による排他制御を実現する。

さらに、そのマシン固有のデバイスのみを仮想化するだけでなく、通信回線で接続されているデバイスを仮想化することも想定している。つまり端末としてパーソナルコンピュータを接続しているような場合、その端末上のデバイスを扱えるように物理デバイスドライバを端末上にインプリメントすれば、そのデバイスを仮想デバイスドライバが吸収してそのマシン上のデバイスと同じように扱うことが可能になる。

ネットワークシステムに接続されている別のマシン上のデバイスを利用するのは、OSのネットワーク機能レベルで行われるので、ここではマシンに直接接続されるようなデバイスの仮想化を行っている。

4. おわりに

OS/ο ウィンドウ・システムは、マンマシンインタフェースやデバイスインタフェース自身の研究のサポートや、アプリケーションのより深い要求に答えられるように、これらの機能を司る部分を容易に変更できるような設計とした。この結果、我々のようにOSやそれを取り巻く環境そのものを研究しているものにとっては、より優れた開発環境が提供されることになるであろう。

参考文献

[1] 高橋延匡：“OS/οmicron の設計思想”，情報処理学会第29回全国大会，1984． 9

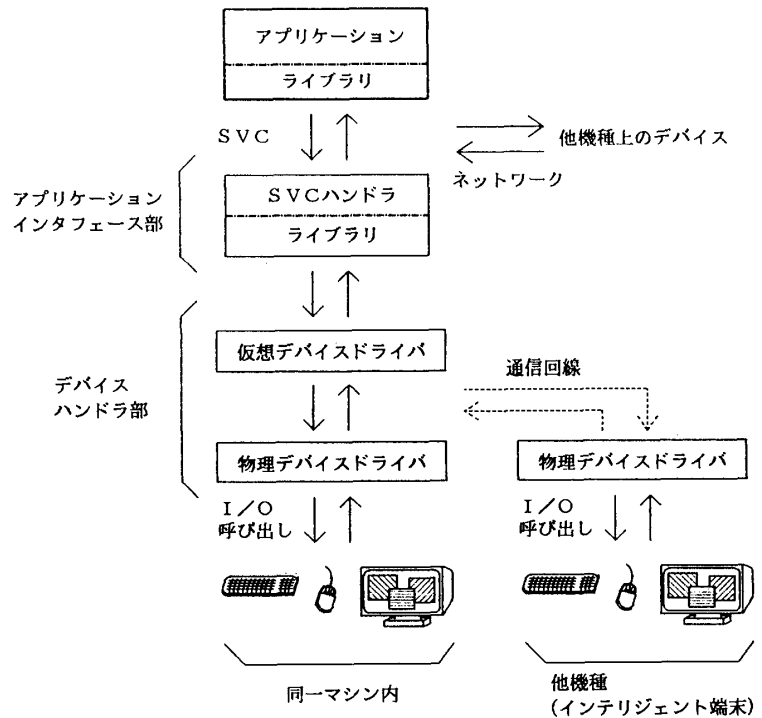


図1. OS/οウィンドウ・システムの構成

表1. SVCの例

基本セット	ライブラリセット
ウィンドウのオープン	ウィンドウ内のスクロール
ウィンドウのクローズ	スクロールバーの表示
ウィンドウのリサイズ	メニューデータのセット
アクティブウィンドウの切り替え	メニューのオープン
フォントデータの取得	メニューへの割り込み取得
ビットイメージデータの出力	図形の出力
ビットイメージデータの移動	文字の出力
マウス割り込みの取得	アイコンの定義
タブレット座標の取得	アイコンの出力
キーコードの取得	指示割り込みの取得
液晶タブレットへの出力	