

MICROSOFT WINDOWS

西田 憲正 (株)アスキー マイクロソフト日本本部

はじめに

近年のコンピュータのハードウェアおよびソフトウェアの目覚ましい発展は、少数の技術者のみが関与していた過去のコンピュータ環境を大きく変革させてきた。よりコンパクトに、より高性能にという一つの流れがパーソナルコンピュータを生み、今日では極めて多くの人々が何らかの形でコンピュータを利用するようになってきた。すなわち、少数の専門家の複雑な手続きをこなしてコンピュータ処理を行っていた時代から、専門知識を持たない多くの人々が直接コンピュータを操作する時代へと変わったのである。

このような変化はパーソナルコンピュータのオペレーティングシステムにも影響を与え、二つの大きな流れを生んだ。

一つはオペレーティングシステム自体の日本語化の流れである。日本語のシステムメッセージは初心者への理解を容易にし、また、カナ漢字変換などの日本語入力機能によりコンピュータに蓄えられる情報の日本語化が促進された結果、出力情報もより利用しやすくなった。

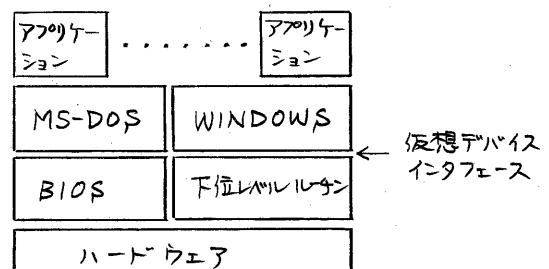
もう一つは、一つのスクリーン上に複数の異なる情報を同時に表示する、マルチウィンドウシステムの流れである。このシステムはすでに10年以上も前に種子は蒔かれていたが、パーソナルコンピュータの普及とユーザーからの要望が、マルチウィンドウシステムを開花させるエネルギーとなった。1983年11月に発表された米国マイクロソフト社の Microsoft Windows は、この流れから生まれた一種のオペレーティングシステムである。

Microsoft Windows の概要

Microsoft Windows (以下 Windows と呼ぶ) は、同じマイクロソフト社のオペレーティングシステムである MS-DOS の拡張システムであり、マルチウィンドウ管理、アプリケーションプログラム相互間のデータ交換機能、グラフィックスインタフェースなどを持つシステムソフトウェアである。Windows を走らせるためには、①米国インテル社の 8086 またはそれと互換性のある CPU、② 256K バイト以上の主メモリ、③ビットマップ方式のスクリーン、④マウスまたはその他のポインティングデバイス、⑤ 2 台以上のフロッピーディスク装置、の各ハードウェアが必要である。

Windows は右図に示すように、ソフトウェア階層上はオペレーティングシステムと同じレベルに位置する。ハードウェアへのアクセスは仮想デバイスインタフェースを介して行う。このインタフェースは、最小限の機能しか持たないハードウェアにも Windows を移植できるように配慮されており、

図1 ソフトウェア階層



また、高度の機能を持つハードウェアに対しこれを活用できるような柔軟性を持たせている。

Windows のスクリーン表示は図2のように作業領域とアイコン領域の二つの部分から成る。

スクリーンの大部分を占める作業領域は、すべてのオープンされたウィンドウ向けのスペースである。アプリケーションプログラムまたはユーザがウィンドウのオープンを指示すると、Windows はこの領域一杯にアプリケーションプログラムのウィンドウを設定する。

既にオープンされたウィンドウが存在する場合、Windows は作業領域を分割して新たなウィンドウと既存のウィンドウの全二が同時にみられるよう自動的に配置する。この表示技法は「タイルリング」と呼ばれる。ウィンドウの位置の移動やサイズの変更はユーザの指示によっても行うことができる。これに伴う作業領域内のウィンドウの再配置も Windows が自動的に行う。アプリケーションプログラム自身が配置やサイズ調整を行う必要はない。

スクリーンの下部のアイコン領域は、現在クローズしているウィンドウのアイコンを表示する領域である。アイコンとはアプリケーションを示す絵文字と考えればよい。アプリケーションプログラムはアイコンの状態であっても主メモリ中に存在しており、従ってタイマ割込み処理などを実行することができる。アイコンの形状はアプリケーションプログラムが制御する。ただし、アイコン領域内の配置については Windows およびユーザにより制御される。

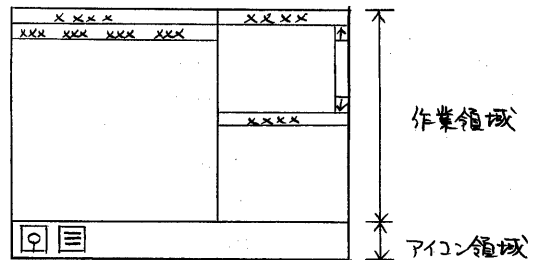
Windows とウィンドウクラス

ウィンドウは、アプリケーションプログラムの入出力インタフェースである。アプリケーションプログラムはスクリーン上の対応するウィンドウ内にテキストや図形を表示させ、またキーボード、マウスおよびタイマからの入力を受けとる。一つのウィンドウは一台の端末装置と同様に入出力機能を備えているが、システムの実際のハードウェア資源も完全に制御する必要のない点で端末装置とは異なっている。事実、一つのウィンドウは他のウィンドウとメモリ、スクリーン、キーボード、マウスなどのシステム資源を共有することができる。

ウィンドウのスクリーンイメージは、キャプションバー、特別コントロールアイコン、クライアント領域、およびオプションのメニューとスクロールバーなどで構成される。ウィンドウの実際の外見とふるまいは、そのウィンドウが属している「ウィンドウクラス」に依存する。

ウィンドウクラスは属性の集合体であり、アプリケーションプログラムが Windows 上で実行される場合に、そのウィンドウがどのように見えまたふるまうかをこの属性が定義する。ウィンドウクラスは、対応するウィンドウがメニューやスクロールバーを持っているか否か、ウィンドウの好ましいサイズと形状、ウィンドウアイコンの形状などを定義する。その他、ウィンドウクラスは、マウ

図2 Windows のスクリーン表示



スからの入力の読み取りやウィンドウ内でのテキストのスクロールなど、アプリケーションに特定の処理を実行する場合に使用するべき手続についても定義する。アプリケーションは、ウィンドウを作成するまえにそのウィンドウのウィンドウクラスを定義し登録しておかなければならない。

同一クラスのウィンドウは、スクリーン上に同時に幾つでも表示することができ。たとえば、二つのアプリケーションプログラムが同一のウィンドウクラスを持つ場合や、一つのアプリケーションが二度呼び出された場合、同一のウィンドウがスクリーン上に表示される。ただし、各々のウィンドウはそれぞれ独立であり、個々のウィンドウの動作はそのウィンドウに属しているアプリケーションプログラム以外には影響を与えない。

モジュール手続

Windows はアプリケーションプログラム内の手続を呼び出すことによりアプリケーションプログラムと対話する。その手続が必要な処理を実行する。実際には、アプリケーションプログラムはそれ自体で動作するのではない。Windows が必要に応じてアプリケーションプログラム内の手続をコールし、その手続の実行が完了すると直ちに制御を取り戻すことによりアプリケーションプログラムを制御する。この意味でアプリケーションプログラムは Windows のサブシステムである。

ほとんどのアプリケーションプログラムは、モジュール手続と呼ばれる三つの手続を持つ。"ロード手続"、"新インスタンス手続" および "インスタンス解放手続" がそれぞれである。

ロード手続は、アプリケーションプログラムがロードされた直後にコールされる。ここではそのアプリケーションプログラムが使用するウィンドウクラスの定義と登録を行わなければならない。

新インスタンス手続は、アプリケーションプログラムの実行準備のためにコールされる。ここでウィンドウの生成と、ディスクファイルのオープンやメモリの割付けなどの初期化処理を行う必要がある。

インスタンス解放手続は、アプリケーションプログラムの終了直前にコールされる。ファイルのクローズやメモリの解放など、アプリケーションプログラムの終了前に行うべきすべての処理を実施しなければならない。

クラス手続

ウィンドウクラスが登録されウィンドウが生成されると、Windows からそのウィンドウの "クラス手続" をコールすることが可能となる。

クラス手続は、Windows からの特定の要求に対して取りべき動作を定義する。たとえば、ウィンドウのサイズを変更する際に、Windows はそのウィンドウの "サイズ" クラス手続をコールする。これにより、Windows がそれ自身の動作を完了する前に、アプリケーションプログラムは自分が行いたいどんな処理でも実行することができ。また、クラス手続は Windows の規定の動作を変更するためにも使用することができる。

クラス手続には、"クリエイト"、"デストロイ"、"アイコン"、"ペイント"、"スクロール"、"インフォット"、"サイズ"、"フォーカス"、および

"データ"の9種がある。

以上、Windowsの概要とアプリケーションプログラムとの関連について述べてきた。以下でWindowsの特長について述べる。

アプリケーションプログラムのハードウェア独立性

Windowsの第1の特長は、アプリケーションプログラムがハードウェア仕様に依存しない点にある。

これまでのアプリケーションプログラムのほとんどは、ハードウェアを直接アクセスして線や円を描いている。スクリーンの解像度、物理座標とビデオRAMのアドレスとの関連、画素の縦横比などを知らずにビデオRAM上のビットをセットしたり、ハードウェアのカラーの実現方式に従って対応するビットをセット/リセットするというようなことをアプリケーションプログラムで行っている。ハードウェア仕様はそれぞれのシステムで異なっているため、あるシステム用のグラフィックスアプリケーションプログラムは他のシステムでは動作し得ない。

Windowsシステムにおいては、アプリケーションプログラムの出力はGDI (Graphics Device Interface)を通して行なうことになる。GDIは現在制定作業中のANSI-VDIに準拠したグラフィックスインタフェースプログラムであり、Windowsの中に組み込まれている。制御、出力、属性、および問い合わせの4グループ、90以上のプリミティブから成る。アプリケーションプログラムはGDIが提供する正規化座標系を用いることができる。実際の寸法に対応したデバイス座標系への変換はGDIが行なう。カラーについてもアプリケーションプログラムは論理的なRGB値を指定すればよい。このようにしてWindowsはアプリケーションプログラムにデバイスからの独立性を与え、したがって、Windows用のアプリケーションは、Windowsが搭載されているどのシステム上でも動作可能となる。

Windows自体の移植性

Windowsの第2の特長は、Windowsそのものの移植性の高さにある。これは先に述べた仮想デバイスインタフェースを定義することにより実現している。このインタフェースにおいて、ハードウェア機能はマウス、カーソル、キーボード、ディスプレイ、およびタイマの五つにグループ化されている。このうちのディスプレイ部では、最低二つのルーチン(ビットブロッック転送ルーチンおよびストリングブロッック転送ルーチン)を用意すればよい仕様になっている。このため、現在普及しているほとんどのパーソナルコンピュータに移植することが可能である。また、線、円弧、多角形などの機能を持つハードウェアにおいては、これらの機能を宣言し対応するインタフェースを定義することにより、Windows内部でのシミュレーションをバイパスしてハードウェアの機能を利用することができる。すなわち、Windowsは単に移植性が高いばかりでなく、ハードウェア機能を最大限に活用した効率のよいシステムに仕立て上げることにも可能となっている。

データ交換機能

Windowsの次の特長は、アプリケーションプログラム間で自由にデータを送受することができることだ。

先に述べたように、Windowsはスクリーンをいくつかに分割したウィンドウ内で個々のアプリケーションプログラムの実行を可能にしている。アプリケーションプログラムの出力は対応するウィンドウ内に限定され、他のウィンドウの内容に影響を与えない。したがって、プログラムの作成者やユーザは、他のプログラムとの関連を気にすることなく、自由にアプリケーションプログラムを製作したり利用することができるのである。このアプリケーションプログラムの独立性は、マルチウィンドウ環境下では必須の要件である。

この独立性と同時に、アプリケーションプログラム間でデータの受け渡しができるということも極めて重要なことである。実際の業務では、一つのデータを種々の形式で表現したり、複数のデータを寄せ集めてレポートにまとめ上げるといった事が多々ある。このようなことをパーソナルコンピュータ上で可能とすることがWindowsの主要な目的でもある。

Windowsが提供するデータ交換機能により、アプリケーションプログラムはデータの送受、削除、送信後削除といったことが行なえる。また、相互に解釈できるように、データ形式についても受け渡すことができる。基本的なデータ形式として、テキスト、SYLK、ビットマップ、DIFの4種があらかじめ定義されているが、他のデータ形式も登録すれば使用できる。Windowsには、データ交換における中間媒体の役割を果たす“クリップボード”も用意されている。

豊富な機能

最後に挙げるべき特長は、マルチウィンドウシステムのアプリケーションプログラムを作るうえで必要な様々な機能が用意されている点である。このような機能が豊富に整っているければ、良質のアプリケーションプログラムの開発が困難となる。ここでWindowsの機能の一端を紹介する。

ウィンドウ機能には、ウィンドウの作成、制御、ウィンドウとの対話、メニュー作成、特性リスト、入力制御といった機能グループがある。通常のスクリーン分割方式のウィンドウの中に子ウィンドウを作成したり、オーバーラップ方式のウィンドウを作成する機能も用意されている。メニューは文字だけでなくビットマップも登録できる。スクロールバーなども標準で用意されている。

GDIには先に述べた通り90以上のポリミティブがある。

その他、システムタイマにより割込をかける手続きや時間間隔を登録するルーチン、モジュール管理機能、メモリ管理機能などがある。

おわりに

パーソナルコンピュータのハードウェアの進歩には著しいものがあり、従来高価なシステムにしか無かった機能が次々と取り入れられている。その上で多くの有用なアプリケーションプログラムが生まれてきている。しかし、ユーザの立場から見ると、個々の業務の改善には役立つものの、システム全体の使いやすさや業務の統合性などの面では今一步の観がある。Microsoft Windowsは、こうした問題の解決手段を提供するものと考えられている。