

2V-8

UNIX*とイメージ処理

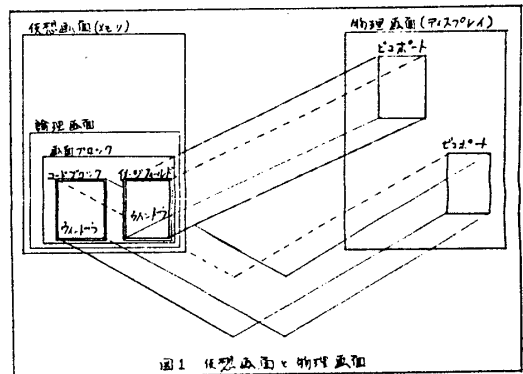
山田 操
(株) 東芝 青梅工場

最近UNIXをOSとしたワークステーションが各社から出されているが、このワークステーションの最も大切な機能の一つとして、イメージ処理があげられよう。ところが、このUNIXシステム上にイメージ処理機構を構築する上で解決しなければならない問題点がいくつか存在する。主な問題点をいくつかピックアップしてこれらについて考える。

1. UNIXとウィンドウマネージャとの親和性
2. マルチウィンドウと仮想ターミナルの一体化
3. ビットマップディスプレイ上でのコードデータ処理
4. UNIXシステム下でのイメージデータの取り扱い
5. マルチウィンドウ下での日本語処理機構

1. UNIXとウィンドウマネージャとの親和性

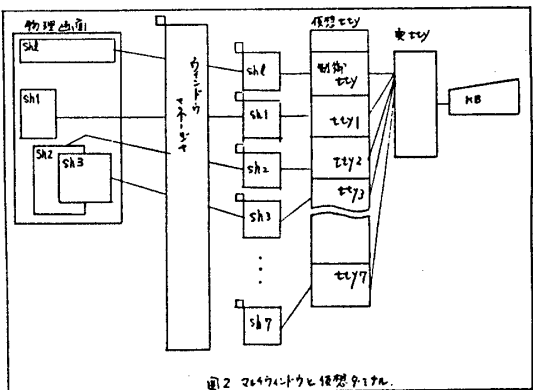
ウィンドウマネージャの取り扱う画面制御のファクタとして、物理画面、仮想画面、論理画面、画面ブロック、コードブロック、イメージフィールド、ウィンドウ、ビューポートなどがあり、これらを制御するためのインターフェースをユーザプログラムに提供する必要がある。UNIXには、もちろんこのようなインターフェースは存在しないので新しく提供する必要があった。その実現方法としては、UNIXの基本的なシステムコールであるread/write、ioctlなどを利用する方法もあるが、この種のインターフェースでは、画面制御を行うときの制御情報量から考えてムリがある。これらの多量の制御情報を一度に渡せるインタフェースを新しく作成し制御した方が、今後の拡張性からみても得策であった。このインタフェースにより、ユーザプログラムレベルで、容易に画面制御の各ファクタの定義および変更はもちろん、イメージデータ、コードデータの取り扱いをも容易にしている。このユーザインタフェースは二種類存在し、一つはユーザプログラムが何らかの回答をシステム側から期待している場合で、もう一つは何も回答を期待しない場合である。ユーザプログラムは、基本的にはこの二種類のシステムコールを使用し、実際に行いたい処理をコマンド形式でウィンドウマネージャに要求することで画面制御を行うことができる。



2. マルチウィンドウと仮想ターミナルの一体化

UNIXの仮想tty構造を利用してマルチウィンドウをサポートする方法を考える。ここで言うマルチウィンドウとは、画面上の各矩形の中で、それぞれ独立して、シェルが動けるようなウィンドウマネージメントができることとする。

仮想tty構造の環境とシェルレイヤとの関係については、UNIXの基本機能として含まれているので説明は省略する。各仮想端末上で、それぞれシェルの動く環境を提供するのが仮想tty管理であり、これを同時に画面上に表示しながら、それぞれの処理を実行させるのがマルチウィンドウ管理である。(従って、マルチウィンドウ管理では、物理的に一つの画面をいくつかに分割して、それぞれを仮想端末として割り当てることになる。) このとき、仮想端末のオープンの方法および一度オープンした矩形のサイズの変更、カレントな仮想端末の定義、カレントの切り替え方法(シェルレイヤ、マウス、ファンクションキー等)がポイントとなる。



*UNIXはAT&Tが開発しライセンスしているOSです。

UNIX and Image Processing
Misaō YAMADA
TOSHIBA, Ltd.