

紙の仮想化を行う OS/omicron 第4版における「電紙」の設計

4H-3

早川栄一, 並木美太郎, 高橋延匡

(東京農工大学 工学部)

1. はじめに

表示一体型タブレットを用いた計算機システムは、人間の、「手書き」に近いインタフェースを提供できることから、最近、これを提供するいくつかのシステムが登場している。

しかし、これまでのシステムでは、「手書きインタフェース」が本質的に持つ、データの多様性、多義性、応答性に対して、十分な対応を行うことができなかった。このような理由から、我々は、手書きインタフェースを対象にしたOSである OS/omicron 第4版の開発を行っている。この中で、前述の手書きインタフェースの特色を支援するために、計算機によって紙を仮想化した「電紙」の概念と機構を実現した。

本報告では、OS機能の中で、「電紙」を実現する機構の設計について述べる。

2. 「電紙」：紙の仮想化

電紙は、紙の持つ性質を計算機によって仮想的に作り出した概念である。この電紙の設計方針は、次のとおりである。

- (1) 電紙の持つ属性情報を応用プログラムが定義できる
- (2) 電紙同士の属性の対応関係の管理と、属性変換の機構を備える
- (3) データの表現を可視化するための対応関係とその実行機構を備える

電紙の概念図を図1に示す。

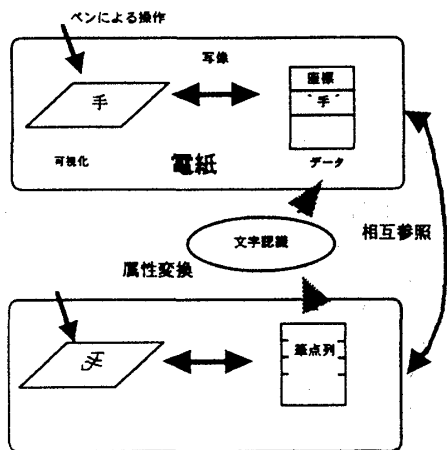


図1 電紙の概念図

さらに、電紙に対して属性を定める。属性は、電紙上に書かれたデータの性質を決定する。

電紙の属性としては様々なものが考えられるが、最低限の要素のデータ構造をOSで決定する必要がある。これらを基本属性と呼ぶ。基本属性は次の3つである。

- ・文字コード  
計算機資源を指定するために必要である。日本語情報処理のために2バイトコードを採用する。
- ・ビットマップ  
計算機内部で紙を表現する基本的な単位である。
- ・筆点列  
認識処理では、筆速などでストロークの時間を用いる。このために、ペンのスイッチ、座標、筆点の時間からなる筆点列を定義する。

これらの基本属性を備えた電紙を実現することによって、最低限の「紙」の仕様を満たす。

システムで統一したインタフェースを提供するためには、アプリケーションレベルで、これらの属性を定義して、システムに登録できる必要がある。例えば、矩形や円といった図形の可視化方法や、属性変換の方法などである。システムでは、アプリケーションの開発、実行段階でユーザが、これらの上に、必要な属性の定義を登録できる機構を実現する。

システム全体でのデータ管理の必要性から、「電紙」をOSレベルで導入する。

3. 電紙を提供するOSの設計

3.1 全体構成

OS/omicron 第4版の全体構成を図2に示す。

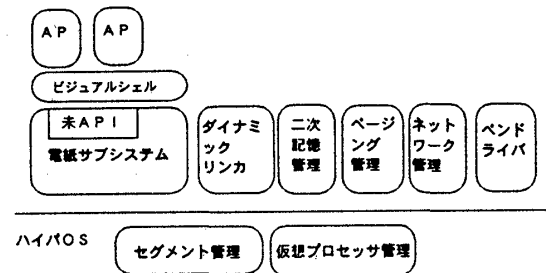


図2 OS/omicron 第4版の全体構成

OSは、複数の層で実現する。ハイパOS部は、複数のプロセッサ環境を想定していることから、ハードウェアを抽象化し、なおかつ、電紙の実行に必要な環境を提供する。ここでは、次の資源を管理する。

- ・セグメント - 電紙を構成する記憶単位生成、消滅、保護機構

- ・仮想プロセッサ - 実行単位  
生成, 消滅, アドレス空間の生成, 消滅, 通信, 同期
- ・割込みの仮想化  
物理割込みと手続き呼び出しとの対応

デバイスドライバや, 電紙サブシステム, ダイナミックリンクなどは, ハイバOSが提供する仮想プロセッサ上で動作する。

電紙サブシステムは, 電紙の機構を提供するサブシステムである。ここでは, 次のものの管理を行う。

- ・実行単位であるタスクと, 資源管理単位であるタスクフォースの提供
- ・セグメントと電紙の対応づけ
- ・属性を処理する手続きやデータへのリンク
- ・電紙へアクセスするためのAPI (Application Program Interface)

ハイバOS上のモジュールは, ダイナミックリンクを用いて結合する。これによって, モジュールの拡張性, デバッグ性を向上させることができる。

### 3.2 電紙サブシステム

電紙サブシステムは, 通常のOSにおけるプロセスに対応する, タスクフォースと呼ぶ資源管理単位を提供する。タスクフォースが提供するアドレス空間を図3に示す。

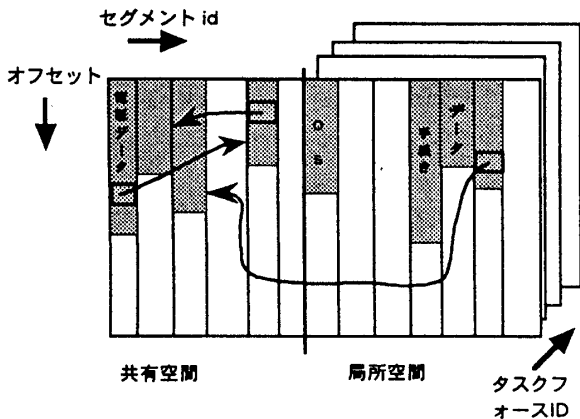


図3 アドレス空間

ここで, 電紙データは, 2次元アドレスのセグメントを一つの単位とする。手書きデータの多様性から, 紙面データは, 複数の小さなオブジェクトの集合体となる。さらに, それらは, 多様性, 多義性から, 複雑なリンクを伴ったデータである。このような性質を電紙上で実現するために, 電紙データをポインタによって結び付ける。このポインタは, セグメント+オフセットの形態を取る。

また, 電紙データは, 複数のタスクフォースによって, データを加工する。このために, 電紙データはすべてのタスクフォースから共有される空間におく。これは, すべてのタスクフォースから, セグメント+オフセットという形で, 一意にアクセスすることができるようになる。電紙データのリンクは複雑になることから, 単一記憶階層によって, ポインタを含んだ共有空間全体に永続的な性質を持た

せる。

これに対して, サブシステムや, アプリケーションなど, プログラムの実行に必要な手続きやプログラムデータは, 各タスクフォースに固有の空間にロードされる。

電紙サブシステムが提供する手続き (スーパーバイザコールにあたる) は, アプリケーションから, ダイナミックリンクによって呼び出される。

### 3.3 電紙の属性の実現

2. で述べたように, 電紙データを処理する手続きとして, 属性の変換と, 可視化の手続きがある。

これらは, 電紙についての属性によって定められる。電紙サブシステムは, 属性の変換表や属性の可視化表のエントリを用意する。属性の追加は, これらの登録に対する追加によって実現する。

属性処理関数のエントリは, システムで管理していて, アプリケーションからはダイナミックリンクの機構を通して呼び出される。属性表への動的な追加は, リンクテーブル (これもセグメントである) の拡張によって, 実現することができる。この機構を図4に示す。

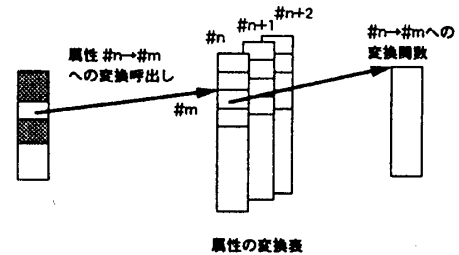


図4 属性処理関数の対応図

電紙への描画のAPIとしては, 我々の研究室で開発を行った, ペンインタフェースを対象にした, 未ウィンドウシステム [1] のAPIを用いる。

### 4. おわりに

本稿では, OS/omicon 第4版における電紙の設計について述べた。

紙の上に書かれたデータは複雑なリンクを持つ, という電紙の性質を実現するために, アドレス空間を, 電紙データと, プログラム, データに分けた。さらに, 電紙データにつく属性を変換, 可視化する機構を, サブシステムの機構として実現した。

現在, 我々は, 電紙の環境を提供する OS/omicon 第4版を, プロトタイプとして実現中である。

### 参考文献

- [1] 河又他: “表示一体型液晶タブレットを用いた “未” ウィンドウシステムの設計と実現”, 情処学HI研究会, 45-17, 1992.
- [2] 早川他: “手書きインタフェースを支援する OS/omicon 第4版の構成”, 情処シンポジウム論文集, Vol.92, No.7, 1992.