

3Y-7

**統合化オフィスワークステーションの開発(Ⅳ)**  
 - UNIXライクOSにおけるリアルタイム機能と高速化の実現 -  
 塩田博行 , 笠井建治 , 松山誠 , 高久典夫  
 富士ゼロックス株式会社

1. はじめに

6060ワークステーションは、JStarの優れた操作性を継承したOA向け統合化ワークステーションである。ここでは、本システムの基本ソフトウェアであるオペレーティングシステム(OS)について説明する。

6060ワークステーションでは、複数ウィンドウの同時実行や、イーサネット、プリンタとの同時処理を効率よく行なうため、マルチプロセス機能のOSが必須である。そこで、ベースOSとしてコンパクトでしかも基本機能を備えたUNIXライクOS Idris (Whitesmiths社製、UNIX V6, V7相当)を採用した。

しかし、本OSだけでは任意のプロセス間通信機能、メモリ共有機能がなく、また、高速性能も十分に得られないため、ベースOSに種々の改造、改良を加えた。以下、6060ワークステーションで行なった機能拡張、高速化方式、及びエンドユーザ向けの保守性の向上策について述べる。

2. 開発方針

- (1) OS機能の拡張に際し、ベースOSのリソースアクセス方式と整合性のあるインタフェースを提供し、アプリケーション・プログラム側の負担を軽減する。
- (2) ベースOSとの共存を重視し、元来の機能を損なうことなく機能の拡張を図る。
- (3) エンドユーザでもシステム保守が容易にできるように、システム全体をコマンドレス化する。

3. 機能拡張

マルチウィンドウ・システムにおいて、任意のウィンドウ間のデータコピー、プリント処理、通信処理中にプロセスの同期、排他制御等を効率よく実現するためにはリアルタイム処理機能が必須である。また、メモリの利用効率を向上させるためには、オーバレイ機能や共有ライブラリの機能が必要であり、これらの機能を新たに開発した。

3.1 リアルタイム機能

任意のプロセス間で、同期、排他制御等を行なう機能は従来より多くのマシンで実現されているが、ここではUNIXの考え方に合う使用方法を実現し提供した。表1に拡張機能を示す。

(1) 同期制御

図1を例に説明する。まず、プロセスAでは事象用制御ブロックをgetecb()で確保し、そのidを得る。プロセスBも同様の名称でgetecb()することにより、同一事象の受け渡しを決定する。idはシステム全体でユニーク

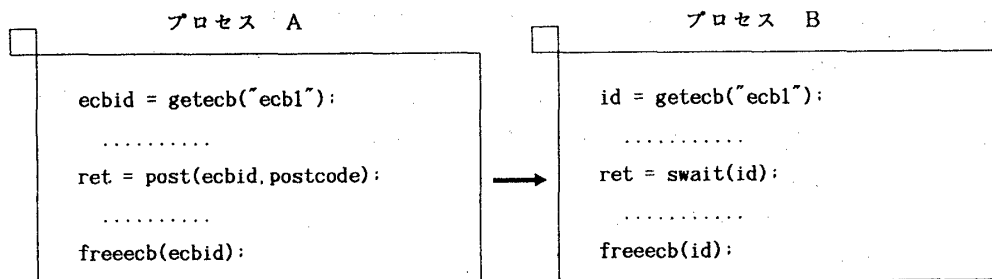


図1. 同期制御の例

な番号である。post(),swait()は、そのidを基に同期制御を行なう。終了時には、freecb()でそのリソースを解放する。他の拡張機能も基本的には同様であり、UNIXのopen()してread()/write()する使用方法と同じである。また、これら拡張機能を使用しているプロセスが異常終了しても、デッドロック等が発生しないように、OS内でリソースの解放、終了処理を行なっており、システムの保全性を向上させている。

#### (2) 排他制御

リソースをシリアルに使用する時、本排他制御機能を利用できる。

#### (3) メッセージ送受信

比較的短いメッセージを任意のプロセス間で送受信する時、処理の同期を取りながらデータの受け渡しを行う。

#### (4) タイマの登録/削除

ミリ秒単位でタイマ登録ができ、結果として事象通知を受け取る。

#### (5) 共有メモリ

任意のプロセス間で比較的大量のデータの受け渡しに利用出来る。

#### (6) 固定プライオリティ

プロセスの実行を平等にサービスするのではなく、特定のプロセスに高いプライオリティを設定できる。

表1. 拡張機能一覧

分類	拡張機能
同期制御	getecb, freeecb, post swait, mwait
排他制御	getrcb, freercb enq, deq
メッセージ 送受信	getmcb, freemcb sendmsg, recvmcg
タイマ 登録/削除	timer, tcancel
共有メモリ	getshm, freeshm
固定プライ オリティ	nice(lxxで高プライ オリティ)

### 3.2 オーバレイ制御機能

1プロセス当たりのメモリサイズを小さくし、メモリの有効活用を図れる。

### 3.3 共有ライブラリ機能

本システムでは文字、図形表示やウィンドウ制御用のライブラリを数多く必要とする。これらを各アプリケーション毎に持つと、ディスク上、メモリ上に共にスペースオーバーヘッドが大きくなるため、共通に使用されるライブラリ群を共有メモリの一部に常駐し、各アプリケーション・プログラムからは、最小限のライブラリ・インタフェースで共有ライブラリを呼び出す方式とした。これにより、実行速度を低下させることなく、メモリ利用効率のよい共有ライブラリを実現することができた。

### 4. プログラム起動の高速化

UNIXファイルシステムにおいては、ディスクファイルはブロック単位にアクセスが行なわれることから、プログラム起動が一般的に遅くなる。しかし、予めメーカーで提供するプログラムやデータは、最初の導入時には比較的連続した領域にロードされることから、それらを一括してロードできるように改良した。これにより、既存のファイル構造を損なうことなく、従来方式に比べて約3~4倍の高速性能を実現することができた。

### 5. システムの保全性の向上

一般的に、UNIXシステムはシステムの構築、管理、立上げ手順等がプログラム開発者向きに作られており、その操作に専門的知識が要求される。しかし、6060ワークステーションでは、一般のエンドユーザがシステム保守を容易に行なえるように、システム全体をコマンドレス化した。すなわち、システムの構築、バックアップ、保守等を全てメニュー形式で選択し、簡単な操作で使用できるようにしている。

主な機能として、次のものがある。

- (1) 各種ソフトウェアの導入/削除/表示
- (2) ディスクボリュームのバックアップ/リストア
- (3) ファイルシステムのチェック
- (4) OSの再ジェネレーション

### 6. おわりに

UNIXライクなOSに種々の機能拡張を行ない、機能的、性能的にも満足のいくオフィスワークステーションとしてのベースOSを実現することができた。今後、更に高性能化、高信頼化を進めていく予定である。

#### [参考文献]

阿部、他：『統合化オフィスワークステーションの開発(I)~(VI)』情報処理学会第33回全国大会