

MC68000用小型OS: OS/οの開発

高橋延匡 武山潤一郎 並木美太郎 中川正樹

(東京農工大学 工学部 数理情報工学科)

1. はじめに

16ビット・マイクロプロセッサの情報処理能力の向上は目覚ましいものがある。同様に、LSIメモリに代表されるRAMのコスト・パフォーマンスは著しく改善されつつあるのは周知の事実である。

我々は、このようなコスト・パフォーマンスの良いハードウェア・リソースを豊富に利用した研究室内の使用を目的とした「研究用計算機システム」の開発の必要性を感じた。この研究用の計算機システムでは、研究対象として、

- (1) 人工知能の研究
- (2) 日本語情報処理の研究
- (3) 特に、オンライン手書き文字認識の研究
- (4) 計算機を用いた教育の研究

などがあり、それらの使用に耐える必要がある。

このような研究を実施するうえで、汎用大型計算機システムでは柔軟性に欠けるため十分に対処できない。そこで、上記の目的を達成するために、16ビット・マイクロプロセッサを用いた計算機システムの研究開発を昭和55年2月より学科内プロジェクト(プロジェクトPIE)として着手した。

本報告は、上記の応用を目的として開発しているオペレーティングシステムOS/ο(omicron)に関して、開発方針とその実現上の諸問題について述べる。

2. 研究の経過

プロジェクトPIEの発足から、現在までの主な研究内容と開発経過は以下のとおりである。

- (1) 並列処理系に関する研究方針(1980年2月)
- (2) 各種マイクロプロセッサの評価(1980年4月-同年6月)
- (3) システム記述言語の選定(1980年4月-1981年3月)^{1), 2)}
- (4) システム開発用のクロス・ソフトウェアの開発(1980年4月-1982年3月)⁴⁾
- (5) シングル・プロセッサ・システムのアドレス空間の構成法に関する概念設計(1981年4月-同年5月)³⁾
- (6) プロジェクトPIEのトータル・システムの概念設計(1980年8月-1982年1月)
- (7) 開発用シングル・プロセッサ・システムのハードウェア設計と試作(1980年7月-1982年12月)⁶⁾
- (8) 開発用シングル・プロセッサ・システムのソフトウェア・ツールの開発(1981年4月-1982年3月)⁵⁾
- (9) シングル・プロセッサ用OSの基本設計(1982年4月-現在)⁷⁾
- (10) システム記述用言語Cコンパイラの開発(1982年4月-現在)

3. OS/οの開発方針

我々が想定した応用に対処するために、我々は“自前の”OSを開発する決心をした。その問題意識は以下のとおりである。すなわち、

- (1) 人工知能の研究、文字認識の研究などへの応用を前提とする
- (2) 日本語情報処理に対して、標準的なアーキテクチャを考慮する
- (3) 時間的制御を容易にするため実記憶方式を前提とする
- (4) 特殊入出力装置の接続や専用の縮退したOSや拡張したOSの開発が容易にできるようにする
- (5) OS自身の開発はシステム記述言語を用いて行ない、将来のハードウェアの拡張に備える

などを満足させるためのOSを、我々は利用可能にする必要がある。そのようなOSを“自前で”用意する理由は、

- (1) 我々が自由に手を入れられるOSが存在しない
- (2) OSは、研究対象によって、“拡大や縮小”が自由に容易に行なえる必要がある
- (3) OSで実現されている「利用空間」で、使用者の「思考空間」が限定されるのは好ましくない

などが基本的なものである。特に、(3)は我々に強い開発の動機を与えた。また、

(4) OSの独立なくしては、計算機システムの独自の発展は期待できない

(5) 他のソフトウェアの互換性に束縛されると恐れているは、研究の進歩が期待できない。ソフトウェアの互換性は基本的には言語レベルで考えれば良い

と考えている。

4. 基本設計

上に述べた開発方針を満足するシステムを構築するために、我々は図4.1に示すハードウェア構成を採用した。本システムは実記憶方式であることから、できるだけ広いアドレス空間を確保したい。CPUとして採用したMC68000は、一様にアクセス可能なアドレス空間として16Mバイトを実現できる。また、構成法によっては、ユーザの領域とスーパーバイザの領域とを分離させること、および、プログラムの領域とデータの領域とを分離させることが、各々、独立に可能である。したがって、最大、16Mバイトの4倍まで実装可能である。このアドレス空間の広さが、MC68000を採用した主要因となった。

図4.1のシステム構成では、主記憶装置は、当面の応用を考え、2Mバイトを持たせる。さらに、本システムは、日本語情報処理の研究を行なうため、タブレット入力装置、高解像度プリンタ(レーザ・ビーム・プリンタ)および、高解像度ディスプレイ、漢字端末等の入出力機器を有する。また、40Mバイトの固定ヘッドディスク装置を2次記憶として接続することを前提としている。加えて本システムは、仮想フロッピーディスクを実現することを意図している。上記の他にも、OS/2の特徴の一つであるROMライブラリの作成のために、専用のROMライタがある。

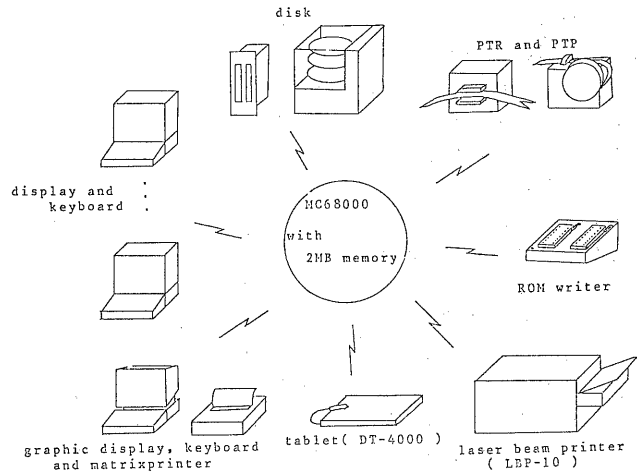


図4.1 ハードウェア構成

我々は、上記ハードウェア構成を前提として、以下の機能を開発している。

(1) 並列処理を実現するために、マルチ・タスクの機能を実現すること

人工知能の研究などの応用は並列処理には魅力的である。もちろん、現在のシステム構成(図4.1)では並列処理を本格的には実現できないが、入出力処理がある以上マルチ・タスクの機能を用意する必要がある。マルチ・タスクの実現の主目的は、マルチ・ユーザ・ジョブの実行というよりは、単一ジョブに対する並列処理の手段を提供することにある。

(2) 簡単な記憶保護の機能を実現すること

本システムは実記憶方式を採用した。その上でマルチ・タスクを実現するためには、システムの信頼性の見地からも記憶保護の機能が必要である。本システムでは、境界レジスタ方式の記憶保護方式を採用した。

(3) リロケータブル・リエントラブルなプログラムを標準とすること

実記憶方式で、マルチ・タスクを実現するには、プログラムはリロケータブルにすることが望ましい。また、同時に、ある種の並列処理の場合のように、同一プログラムの並列処理を実現するには、プログラムはリエントラブルであることが望ましい。したがって、OS/2では、上記のプログラム・モジュールを標準とする。これは、言語C処理系によって、実現することを前提としている。

(4) 必要最小限のOSの機能を実現すること

我々は、現実的な意味で、必要最小限の機能を持つOS(いわゆるOSの核の部分)を提供し、ユーザに対しては、応用対象に即した固有のOSを生成させるようにしたい。たとえば、従来にないような新しい機能を持つ機器の接続や、他システムとの交信などの機能の追加に対処できる必要がある。また、実記憶方式のため、ユーザのアドレス空間を、これらのシステムの機能によって実質的に減らしてしまっは困る。そこ

でOSは原則として、スーパーバイザの空間で動作することにする。

(5) 日本語情報処理の研究のため、16ビットのJIS漢字コードを標準とすること

日本語情報処理の研究(オンライン手書き文字認識を用いた日本語文書作成システムなど)を実現するために、本システムでは、内部コードとしてJIS漢字コード(JIS C6226情報交換用漢字符合系,1979)を採用した。その結果、日本語は何等特殊な処理ではなくなった。特に、ソフトウェア開発の全ライフサイクルを通して、日本語による設計仕様、プログラム仕様が計算機によって管理可能になり、ひいては、保守工程の品質の向上まで期待できる。

(6) システム記述言語として言語Cを採用したこと

我々は、ソフトウェアの全ライフサイクルの生産性、保守性を考慮し、システム記述言語として、言語Cを採用した。また、上記(3)の要請からも、言語Cコンパイラは、本システムの中核を成す。この言語Cコンパイラも言語Cで記述されている。⁹⁾

(7) 情報の統一的管理を目的としたファイル・システムを開発すること

OS/0で扱うプログラム、データ、テキストなどを統一的に扱うファイル・システムの設計が重要である。このファイル・システムの特徴として、ROMライブラリもファイル・システムの管理対象として統一的に処理される。また、仮想フロッピーディスクの概念を実現することにより、2次記憶装置の現実的な拡大を図る。

以上の基本設計を実現するOS/0の構成を図4.2に示す。

5. タスクとタスクフォース

前節で述べた開発方針の(1)を実現するために、OS/0では、通常のタスクの概念と、タスクをまとめて一つの実行環境を共有するタスクフォースの概念を導入した。

OS/0の実行環境は実記憶方式であるため、仮想記憶方式のように動的なリンクと、仮想CPUのペアで柔軟性の高いシステムを簡単には実現できない。

以下、OS/0の基本的な考え方について述べる。

タスクはプログラムによって実行手順を記述されたOSからみた一つの制御単位であり、

- (i) 手続き領域
- (ii) 静的変数領域
- (iii) 動的変数領域
- (iv) ヒープ領域

の4つの領域を主記憶上に持つ。手続き領域とデータ領域を分離することにより、リエントラビリティを、MC68000の8個のアドレスレジスタを表5.1の用途に割り当て、図5.1のように、各領域のベースレジスタに用いて、同時にリロケートビリティを実現している。

一般的なタスクは、上記の4種類の領域を主記憶上に確

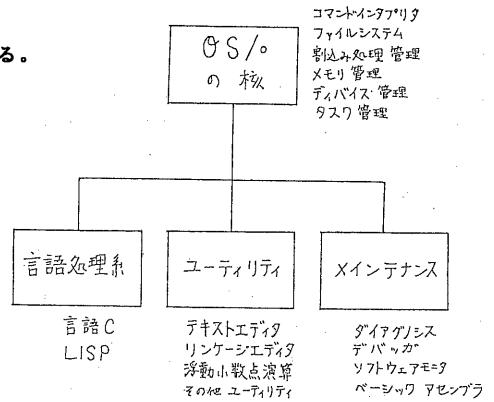


図4.2 OS/0の構成

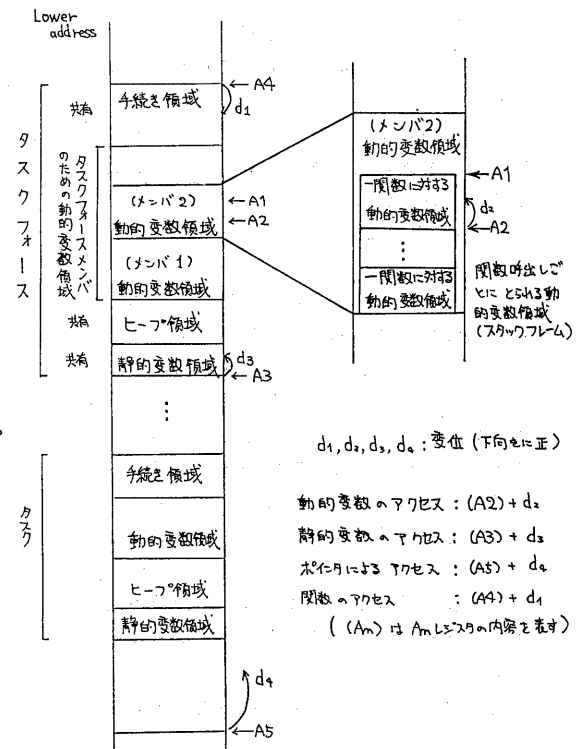


図5.1 実行環境

表5.1 アドレスレジスタの用途

アドレスレジスタ	用途
A0	作業用レジスタ
A1	スタックポインタ
A2	動的変数領域のスタックフレームのベースレジスタ
A3	静的変数領域のベースレジスタ
A4	手続き領域のベースレジスタ
A5	ユーザ領域ベースレジスタ
A6	A5の補数値
A7	使用せず

保し、手続き、データ(静的変数の初期値)をロードすることによって生成される。しかし、この形態のタスクとは別に動的に新しいタスクを生成したり、消滅できたり、一つの目的に対して複数のタスクで、それらが情報交換をしながら並列に実行する形態が考えられる。我々は、この形式の表現に一つのロードモジュールに含まれるサブルーチン(言語Cの上では関数)をタスクとして動的に生成する方式を考案した。タスクの生成時に、(iii)の動的変数領域のみ新しく主記憶上に確保し、残りの3つの領域は、タスク生成を行なったタスクの領域を共有する。このように生成された一群のタスクをタスクフォースと称し、そのタスクフォース内の個々のタスクを、一般的なタスクと区別する場合には、タスクフォースメンバという。タスクフォースメンバは、データ領域として、(ii)の静的変数領域と(iv)のヒープ領域を共有しており、これらの領域を介してタスク間相互の交信を実現する。また、このような構成から、タスクフォース内でオープンされたファイルは、自動的にタスクフォースメンバからアクセス可能になる。したがって、タスクフォースは、それに割り当てられたリソースを無条件に共有するタスクの集合とみることができる。

6. ROMライブラリの扱い方

MC68000 CPUアーキテクチャの特徴である、アドレス空間の用途別分離が、ハードウェアの設計によっては実現可能である。すなわち、スーパーバイザ領域とユーザ領域の分離、および各領域をさらに、プログラム領域とデータ領域に分離できる。本システムでは、図6.1のようにアドレス空間を構成している(詳細は参考文献[3]を参照していただきたい)。

このアドレス空間の特徴を生かすため、我々は積極的にROMライブラリの導入を考えた。このROMライブラリは簡単な関数計算のようなものから、言語Cコンパイラまで考慮する必要がある。言語Cコンパイラの場合のように、静的変数領域が必要なものは、手続き領域(ROM)はスーパーバイザ領域にとるが、データ領域はユーザ領域にとる。また、これは一つのタスクとして取り扱うように設計した。我々は使い勝手の点からもROMライブラリを充実させたいと考えている。

7. ファイル・システムと入出力処理の特徴

OS/2の応用分野に日本語情報処理がある。とくに、オンライン手書き文字認識の研究を最初のOS/2のユーザと考えている。そのため、タブレット入力装置やレーザ・ビーム・プリンタ(参考文献[8]を参照していただきたい)、パーソナル・コンピュータ(端末)などの入出力装置、および、フロッピーディスクや固定ヘッドディスクなどの2次記憶装置の統一的な管理が必要となる。

OS/2のファイル・システムは上記の応用も考え、以下の特徴を持たせた。

(1) 入出力を仮想化すること

OS/2は仮想記憶方式ではないため、ダイナミック・リンク方式や、MULTICSで提唱された「ファイル = セグメント」というような統一的な管理は実現しにくい。しかし、入出力装置や2次記憶装置を、ダイナミック・リンクの代わりに、オープンする手続きを実行することによって、すべてファイルとして一様に扱うことが可能である。

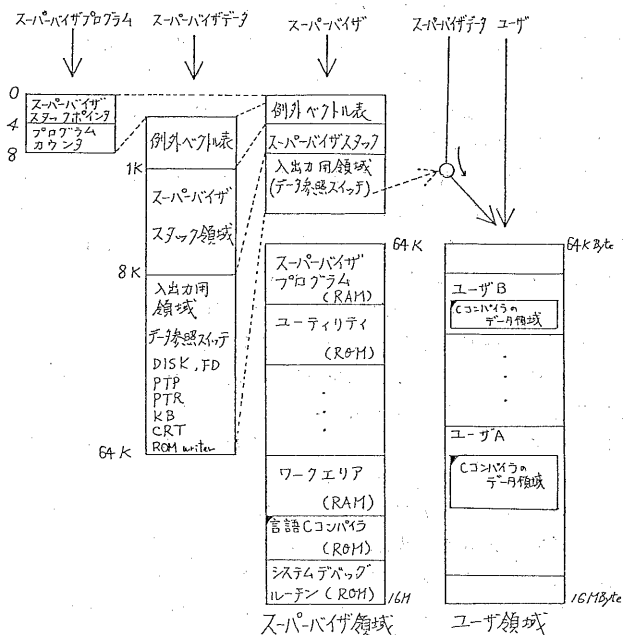


図6.1 本システムのアドレス空間

(2) ファイル・システムを木構造にすること
 OS/0のファイル・システムは、木構造を採用した(図7.1参照)。基本的には、ファイル名を一意に決定できることと、ファイルの保護の方式が自然な形で実現できるからである。実際のアクセスでは、ディレクトリを“/”で区切ってアクセスパスを示す。たとえば、ディレクトリ“言語”の下にある“C”というファイルを指定する場合には、

/ system / 言語 / C

とすれば良い。また、ユーザがユーザ・ディレクトリから指定する場合は、先頭の“/”は不要である。

なお、ファイルの共有はパーミッションを用いて行なう。保護はパスワード方式を拡張したプライベート情報に関する質疑応答方式によってなされる。これはプライベート情報(例えば自分の生年月日、恋人の名前など)をユーザが任意個登録しておき、アクセスする際に質問するのである。情報の個数、難易度によって機密の度合いをユーザが指定できる利点がある。

(3) 仮想フロッピーディスクを導入すること

OS/0は固定ヘッドディスク(40Mバイト)を2次記憶装置として使用するが、その容量には限度がある。そこで交換可能なフロッピーディスク装置も接続する。しかし、これはアクセス速度が遅い。そこでフロッピーディスクを使用する際は、その内容をすべて一時的に固定ヘッドディスク上にコピーして、ユーザは、そのコピーに対して操作を行なう。使用後は、再びフロッピーディスクに戻してやる。このように、フロッピーディスクをあたかも固定ヘッドディスクのように扱うことから、我々はこれを仮想フロッピーディスクと呼んでいる。なお、重要な特徴として、図7.2に示すように、ハードディスク上でもディスク上でもファイルのディレクトリを木構造にしていることである。これにより、固定ヘッドディスクの接続されていないシステムでもシステムの互換性は保てる。

8. おわりに

OS/0は、ほぼ設計が完了し、現在、インプリメンテーションの段階である。大学の研究室のように、人間の世代交替が激しい環境で、信頼性のある設計や保守等を、ソフトウェア工学の面から再度検討を加えたい。

本研究は、当学科の共同研究として開始された。本報告の内容はプロジェクトPIEの参加者各位の討論を

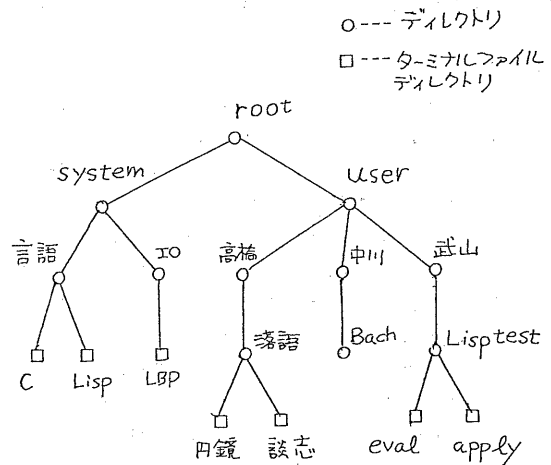


図7.1 ファイル・システムの構造

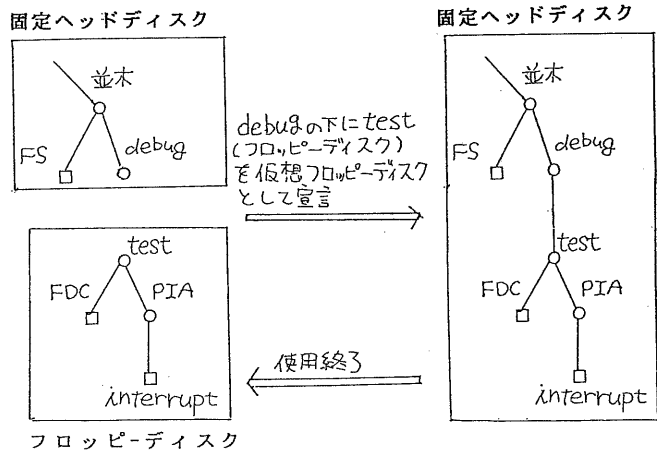


図7.2 仮想フロッピーディスク

土台に成立している。ここに共同研究者各位に心からの感謝の意を表す。さらに、本研究の基礎を築いた、大学院生、武部桂史(現日立)、林努(現日立)の寄与に対しても感謝の意を表す。

参考文献

- [1] 武部桂史, 岩崎俊夫, 志村隆則, 林努, 高橋延匡, “システム記述言語に関する一考察”, 情報処理学会第22回全国大会, pp.177-178, 1981.3
- [2] 岩崎俊夫, 志村隆則, 林努, 武部桂史, 高橋延匡, “MC68000のシミュレータの開発と、PL/IとPASCALの比較”, 情報処理学会第23回全国大会, pp.951-952, 1981.10
- [3] 武部桂史, 鶴澤繁行, 中川正樹, 阿刀田央一, 高橋延匡, “MC68000アドレス空間の問題点と構成方針”, 情報処理学会マイクロコンピュータ研究会資料19-1, 1981.12
- [4] 高田直樹, 本間幹男, 高田正之, “MC68000のクロスベーシックアセンブラの開発”, 情報処理学会23回全国大会, pp.953-954, 1981.10
- [5] 林努, 武部桂史, 五十嵐智, 鶴澤繁行, 高橋延匡, “MC68000開発用ソフトウェアツールESPについて”, 情報処理学会第24回全国大会, pp.955-956, 1982.3
- [6] 五十嵐智, 石丸知之, 細井雄次, 鶴澤繁行, “MC68000開発支援システム”, 情報処理学会マイクロコンピュータ研究会資料25-2, 1982.12
- [7] 林努, 高橋延匡, “MC68000用OSの基本設計(1) - ファイルシステムの設計 -”, 情報処理学会マイクロコンピュータ研究会資料25-3, 1982.12
- [8] 真鍋俊彦, 阿刀田央一, 高橋延匡, “マイクロプロセッサを用いたレーザビームプリンタのインテリジェント化 - ハードウェアの方式設計について -”, 夏のプログラミングシンポジウム, 1982
- [9] 中川正樹, 篠田佳博, 藤森英明, 高橋延匡, “MC68000ユニ&マルチ・プロセッサ・システム用システム記述言語C処理系の開発”, 情報処理学会計算機システムの制御と評価研究会資料21-7, 1983.12