

解説

マイクロコンピュータ用オペレーティング
システムの動向†

加藤 肇彦†† 石川 知雄††

1. はじめに

筆者らが本誌第25巻第3号に、マイクロコンピュータ用オペレーティングシステム（以下 OS）の現状と動向の解説を寄稿してから5年余が経過した¹⁾。その間に OS は長足の進歩を遂げると同時に、応用分野の重点が機器組み込み制御から OA や情報処理へ移行するともなっており質的にも変化してきた。前稿では単なる願望あるいは筆者らの独断による予測でしかなかったことの多くが、OS 技術者、方式設計者、半導体技術者の努力によって実現してきている。本稿では前稿以降の OS の発展動向について報告する。

2. OS 関連ハードウェアの機能強化

2.1 CPU アーキテクチャ

半導体の高集積化の結果、周辺素子でサポートしていた機能が CPU のアーキテクチャに取り込まれてきた。特に縮小命令セットコンピュータ (RISC) では、命令縮小の結果発生したチップ領域の余裕を、OS サポートに活用している²⁾。以下に OS 関連の CPU アーキテクチャの機能強化を紹介する³⁾⁻⁵⁾。

(1) 割り込み機能

直接アクセス可能なアドレス空間が4ギガバイト以上に拡張されるにともない、メモリへのアクセス権のレベルを分け、上位レベルのメモリ領域へのアクセスをハードウェアで禁止する、リング保護機構をもつ CPU が増加している。32ビット CPU の大半は4レベルまでのリングをもち、最上位のレベルは特権モードでのみアクセス可能になっている。スタックへの退避と回復の対象は、プログラムカウンタのほか、CPU

の状態やモードを制御し反映する情報をプロセッサ状態語 (PSW) としてひとまとめにしたものになっている。現在実行中のプログラムのリングレベルは、PSW 中のリングフィールドで指定される。

タスクコントロールブロック (TCB) は従来は OS だけで管理されていたが、現在は CPU に一部が包含されている。図-1は G_{MICRO} 32/200 の TCB (コンテキスト制御ブロック) の構造を示している。ユーザの処理タスクの処理状況 (コンテキスト) を表す情報をひとまとめにして主記憶に格納し、その先頭アドレスを CTXBB という専用レジスタで指している。TCB の形式をハードウェアで固定することによって、コンテキストスイッチを自動化し高速化している。

見かけ上複数の CPU が存在するように CPU を稼働させる仮想計算機システムの実現のために、各仮想 CPU ごとのベクタテーブルのベースアドレスを保持するベクタベースレジスタが用意されている。

従来からあったシングルステップトレース割込みに加えて、命令またはオペランドのアドレスによるブレークポイント設定機能が CPU に包含され、エミュレータによる実機デバッグ作業の一部が CPU のみでも可能となった。

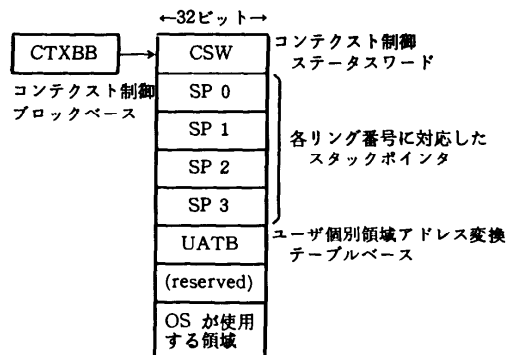


図-1 G_{MICRO} 32/200 の TCB (コンテキスト制御ブロック)

† Trends of Microcomputer Operating Systems by Hatahiko KATO (Space Systems Division, Hitachi Ltd.) and Tomo ISHIKAWA (Electronic Communication Engineering Dpt., Musashi Institute of Technology).

†† (株)日立製作所宇宙技術推進本部
††† 武蔵工業大学電子通信工学科

(2) メモリ管理機能

メモリ管理はキャッシュ制御と主記憶管理に大別される。キャッシュは主記憶の内容の一部のコピーを保持する、CPU 内蔵または外付けの高速メモリであり、CPU が主記憶を直接アクセスする回数を減らすことにより命令の実行速度を高めている。キャッシュと主記憶の内容不一致を防ぐために、キャッシュに書き込んだ内容がある時点で主記憶に転送する必要がある。現存する CPU では、キャッシュと主記憶に同時に書き込むライトスルー方式を採用して、常にキャッシュと主記憶の内容一致を保ち、タスク切替え時の主記憶書替えを不要化することによって切替えを高速化し、また、マルチプロセッサ動作時のメモリ共有を可能にしている。命令フェッチとオペランドフェッチの動作が同一のバスサイクルを奪い合うことによるパイプライン処理の乱れを防止するために、命令キャッシュとデータキャッシュを分ける、分散キャッシュ方式を取ることが多い。

キャッシュの存在は通常ユーザからは意識されない。ただしタスク切替えにともなう TCB の更新時にはキャッシュの内容がパーージされること、マルチプロセッサ動作時に他の CPU のキャッシュに取り込まれたセマフォにアクセスしないよう、アクセス前にモニタが必要であることは、OS ユーザの意識すべき点である。

32ビット CPU は主記憶管理機構 (MMU) を内蔵することが多い。外付け MMU に比較すると CPU に内蔵することにより MMU 操作命令がプログラムシNTAXの中に取り込まれ、パイプラインフローが高速化できる。MMU は動的アドレス変換機構 (DAT) により論理アドレスをフィールドに分割し、各フィールドの指示するテーブルの先頭アドレスを順次索引し、数段階の変換過程の後に物理アドレスを得る。主記憶をアクセスするたびに毎回 DAT が動作すると変換時間のオーバーヘッドが大きくなるため、現在頻繁にアクセスしているページ (通常 4 キロバイト) の物理アドレスを対応表 (TLB) に十数エントリから数十エントリ連想記憶しておいて、高速に変換する。アクセスしようとした

ページが主記憶上に存在しない場合はページフォルト割込みが発生し、DMA コントローラによって補助記憶から転送することにより、仮想記憶が実現できる。

図-2 は MMU の構造と機能を示している。以上に示したマイクロコンピュータの MMU の諸機能は、大型コンピュータの仮想記憶管理機構に範を取って生まれてきたものである⁹⁾。16ビット CPU までの仮想記憶は、狭い論理アドレス空間内にその数倍の実記憶媒体を収容するための技術であったのに対し、32ビット CPU では大型コンピュータと同様に、少ない実記憶媒体をいかに多くみせるかに重点が移ってきている。

(3) 命令語

上記の OS サポートの各機能に対し、これら进行操作するための命令が強化され、また、専用命令化によってコンテキストスイッチが高速化されている。

PSW 中のリング指定やブレークポイント設定のフィールド操作命令は、特権命令となっている。

タスクのキューに関する命令が追加された。キュー型データとして表現された TCB リストへの挿入と削除、リスト内データの探索を専用命令化し、OS の負担を軽減している。

TCB の設定と参照はそれぞれ専用の特権命令によって実行される。

キャッシュ制御の特権命令は、キャッシュ制御レジ

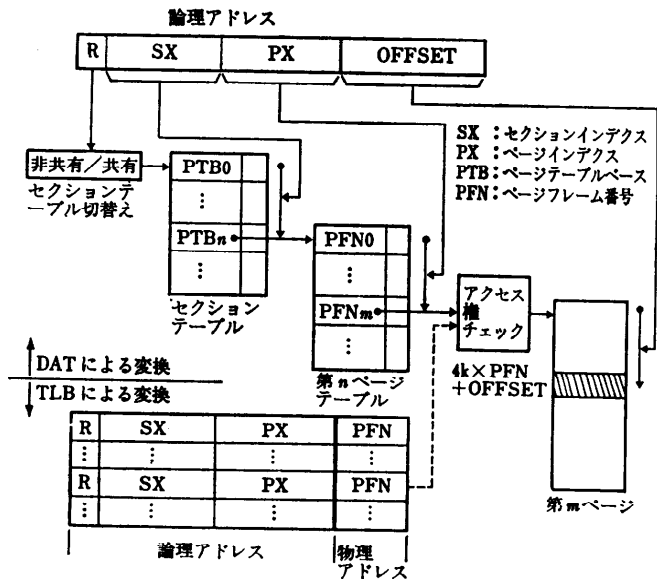


図-2 MMU の構造と機能

スタをセットすることによって、命令キャッシュとデータキャッシュに対して独立にクリア、アクセスの禁止と許可を指定できる。

主記憶管理用の命令としては、DAT が使用するテーブルに対するロードとストア、ならびに TLB の無効化の特権命令がある。さらに実アドレス参照の特権命令、アクセス権のチェックなどの命令がある。

演算用コプロセッサの使用が広まると、複数タスク間でのコプロセッサのアクセス競合が問題になる。従来はテストアンドセット命令によるか、OS がセマフォを利用して、コプロセッサを先に獲得したタスクに占有させることによって、他タスクからの干渉を回避するなどの純ソフトウェア的な対策が取られていたが、最近ではコプロセッサの内部状態を退避/回復する命令が新設され、リエントラント性を保つことにより複数タスクによる同時多重使用が可能になった。これらの命令はコプロセッサによって実行されるが、CPU の OS 関連命令と同様の特権命令としての扱いを受け、特権違反は CPU によって検出される。これはコプロセッサの内部状態をユーザコンテキストの一部と見なした観点から新設された機能である。

2.2 OS サポート用周辺要素

前稿で紹介した OS サポート用周辺要素の機能の大半はすでに CPU のアーキテクチャに吸収され、わずかに割り込み制御要素だけが機能を強化されて残っている。処理すべきタスク数と処理要求源の増大に対処するために、多段に従属接続されて、割り込み要求が入ると自分の要求レベルと一致している場合は受け付け、不一致の場合は次段に送るダイジェンチェーン機能によって、割り込み要因を拡張することができる¹⁰⁾。

3. OS の理論と言語の発展

3.1 OS の理論

理論面の研究はその後も定常的に続けられている。タスクの同期と順序づけに関しては、タスク動作の半順序関係に注目し、これに束論を適用した処理結果の同定法、並行ソフトウェアのベトリネットによる動作解析などが報告されている^{11), 12)}。

デッドロックの解析に関しては、タスクの並行動作条件の分類とその回避法、デッドロック回復のための強制終了プロセス決定問題などが報告されている^{13), 14)}。

分散処理 OS への関心の高まりを反映して、分散処理ソフトウェアの複雑性と、ソフトウェアメトリクスに関する報告がなされている¹⁵⁾。

3.2 OS 記述用言語

OS 記述用言語としては、現在 C がもっとも広く実用されているが、Ada が徐々に普及しはじめており、3.1 に示した研究の大半は Ada を記述言語としている。rendezvous と呼ばれるタスク間同期機能や、例外処理を指定する raise 文などの機能を利用して、Ada で記述された本格的 OS が出現する時期も速くはないと思われる¹⁶⁾。他にデータ抽象機能をもつ言語 CLU の利用、前稿で紹介した Modula-2 の実用化がみられる^{17), 18)}。

4. OS の機能強化

4.1 リアルタイムモニタ

利用者側の要求と、CPU の機能拡張と性能向上、高速化に適した RISC の出現を反映し、リアルタイムモニタには以下に示す動向がみられる。

まず、機能拡張の傾向がある。応用プログラムが大規模化し、フレキシブルディスクを前提にした非常駐タスクの管理、内蔵 MMU を利用した主記憶管理、コプロセッサの競合回避などの必要性に対応している。

機能拡張の結果としての OS の肥大化を回避するためにモジュール化し、非常駐タスクや主記憶管理などの機能を必要としない小規模応用向けに、核部のみを切り出せる構造にしている¹⁹⁾。その結果 CPU の改良やスケジューリングアルゴリズムの工夫とあいまって、応答性はかえって向上している。

小規模応用を指向するために、提供媒体が ROM からフレキシブルディスクに移行している。これはユーザが自分でシステムジェネレーションして応用システムに組み込み、CPU、周辺要素、ゲートアレイ化した専用回路、応用プログラムなどとともに同一チップ上に集積して、特定用途向 IC (ASIC) を構成することを想定している。

以下に主要なリアルタイムモニタの動向を紹介する。

(1) iRMX²⁰⁾

米国インテル社により 80286/80386 対応の iRMX 286 が開発された。iRMX 86 の機能に加え、80286 のプロテクトモードを利用して各種の保護機能が強化され、誤った領域へのアクセス防止やスタック領域のオーバフローの検知が可能になった。アクセス可能なアドレス空間が 16 メガバイトに拡張された結果、デバイスドライバの数やメモリアールのサイズを増すことが可能になった。さらにユーザが自分でコマンドラ

インインタプリタを作成する機能も付加された。

80386 フルサポート版については、ファイル管理やパラメータチェックなどの機能を省略して軽量高速化したカーネル部である、iRMK がまず提供される。

(2) MTOS^{21),22)}

米国インダストリアルプログラミング社によって、前稿以降 80186, 80286, MC 68010, MC 68020 などにも搭載された。さらにダイナミックアロケーション機能と、多値セマフォなどのタスク間同期機能を強化した MTOS-UX/68 K が発表されている。MTOS-UX/68 K はこのほかに、あるタスクから主記憶中または補助記憶中にある別のタスクを呼び出したり、UNIX ふうのシステムコールを生成する C コンパイラによるサポート機能が強化されている。

(3) VRTX^{23),24)}

前稿では米国ハンターアンドレディー社によって各 16 ビット CPU 向けに開発された、コンパクトなりアルタイムモニタとして紹介した。同社はその後レディシステムズ社と名称を変え、各 CPU の後継機種や、宇宙用 CPU である 1750 A, さらに Am 29000 や Gmicro シリーズなどの 32 ビット CPU に順次搭載中である。従来の単値セマフォに替えて、多値セマフォによる複数資源の競合回避が可能になったほか、実行時間がタスク数に依存しないように、スケジューリングアルゴリズムも改良されている。しかもコードは 8 キロバイト以下に抑制してある。Ada ランタイム環境を実現した上位互換版として ARTX がある。

(4) pSOS+²⁵⁾

米国ソフトウェアコンポーネツグループ社によって MC 68000 系の各 CPU と MC 88000 に搭載され、80386, Gmicro シリーズにも搭載予定である。シリコン化を極限まで追求した結果、約 13 キロバイトのコードは完全にポジション非依存であり、任意のアドレスに割付け可能である。システム初期設定時に参照すべきアドレスは Anchor と呼ばれるポインタによって指示する。マルチプロセッサ用の pSOS+m, ファイルハンドラ pHILE+, UNIX 用ドライバ pUCP+, アセンブリレベルデバッグ pROBE+, ソースレベルデバッグ pPREVIEW, C ランタイムサポート pREP/C などがファミリーを形成している。

(4) ITRON 仕様 OS^{26),27)}

後述の TRON (The Real-time Operating system Nucleus) プロジェクトのリアルタイムモニタであり、まず既存の各 CPU に搭載された、日本電気の RX

116 (V 20/25/30/40/50 用), 同 RX 616 (V 60/70/80 用), 富士通の REALOS (80286 用), 日立製作所の HI 68 K (MC 68000 用) と HI 16 (HD 641016 用), 三菱電機の MR 32 (32032/3232 用) がある。さらに TRON 仕様の各 CPU への搭載が進められている。

ITRON 仕様はシングルチップ用/8 ビット用のサブセットである μ ITRON 仕様をもつ。日立製作所が H 8/500, 富士通が F*MC に、三菱電機が MELPS 7700 にそれぞれ μ ITRON 仕様 OS を搭載した。これらはいずれも ITRON 仕様の内部状態の数を減らし、内部テーブルを簡略化することによってメモリ量を削減し、かつ応答時間を短縮した OS である。

4.2 デバッグ用 OS

前稿で一節を割いたデバッグ用 OS は、そのアイデンティティを失い、次節に示す汎用 OS に吸収された。

4.3 汎用 OS

パーソナルコンピュータあるいはワークステーションの操作性向上の要求から、現在はずべての汎用 OS がマルチタスク機能を備え、複数のプログラムのローディング、実行、結果出力を並行して実行できるようになった。また、マルチタスク機能に支えられたマルチウィンドウ機能によって、画面間でデータを交換しながら複数業務を進めることが可能になった。

卓上出版への応用を想定して、日本製の OS はひろんのこと、外国に起源をもつ OS も日本語処理へのインタフェースを備えている。

一部の OS ではマルチユーザー機能を備えているが、ホストと端末の規模の差があまり大きくないパーソナルコンピュータやワークステーションの場合は、分散処理のための LAN (ローカルエリアネットワーク) のサポートがまず要求され、ほとんどの OS がこれを装備している。

以下に、各汎用 OS の現状と動向を簡単に紹介する。

(1) CP/M 系ならびに FlexOS^{28),29)}

前稿で紹介した各ヴァリエーションはその後米国デジタルリサーチ社による再編成の後、マルチユーザ/マルチタスク機能を有する Concurrent DOS 386, 拡張メモリ空間をサポートする Concurrent DOS XM, シングルユーザ/シングルタスクで ROM 化可能な DR DOS などのファミリーへ発展した。Concurrent DOS 386 のメモリ管理がパーチャル 86 モードで動作する以外はすべてリアルモードで動作する。リアルタイム機能をもつ FlexOS 386 は 80386 のプロテクト

モードで動作し、32ビットリニアアドレスを利用してプロセス数は無制限、ウィンドウ管理、階層構造のディレクトリファイル、MS-DOS 3.X 互換性などを有する。

(2) MS-DOS³⁰⁾

前稿以降米国マイクロソフト社によって、1.2メガバイトのフレキシブルディスクとハードディスクのファイル管理を強化した V 3.0、ならびに MS-NET-WORK に対応した V 3.1、さらにウィンドウライクシェルをサポートした V 4.0 が開発された。V 4.0 では MS-Windows のアプリケーションが、後述の OS/2 のプレゼンテーションマネージャへの移植性をもつように配慮されている。同時に、OS/2 のコミュニケーションマネージャとの双方向互換機能をもつ。

(3) OS/2^{31),32)}

米国マイクロソフト社によって 80286/80386 ベースのパーソナルコンピュータの OS として開発された。16メガバイトの実記憶装置と1ギガバイトの仮想記憶空間をサポートし、マルチタスク機能を有する。MS-DOS では外付けであったマルチウィンドウ機能がプレゼンテーションマネージャとして標準装備され、LAN もサポートされた。データベースマネージャの標準装備により、リレーショナルデータベース SQL が利用可能になった。一定の制約のもとで MS-DOS の応用プログラムを利用できる MS-DOS 互換モードをもつ。BIOS やハードウェアへのアクセスが、従来のソフトウェア割込みをユーザが直接実行する方法から、統一的なファンクションコールによる方法に変わり、ハードウェア間ならびに OS/2 のバージョン間での応用プログラムの互換性が向上した。

(4) UNIX ならびに UNIX 系^{33),34)}

UNIX とは特定のインプリメンテーションによる単一の OS を指すのではなく、AT & T の UNIX としての検定に合格した OS 群に対する総称である。市場の大半の CPU に搭載され、高機能ワークステーション用としては事実上の標準 OS となっており、通産省のシグマ計画にも採用されている。最近各社から発表されている RISC はワークステーションの CPU として使用されることが多く、ユーザは UNIX を通じて RISC を利用していることになる。

ライセンスをもたないために UNIX の名を冠してはいないが、機能的に UNIX と等価な UNIX ライク OS も多く、さらに大半の汎用 OS は変換ユーティリティなどの手段により UNIX となんらかのファ

イルパスを有している。また、最近では優先レベル別処理などの機能を付加することにより、リアルタイム処理を可能にした例も増えている。

バージョン6をベースに主として大学で普及した BSD、バージョン7から発展した企業版の System V、ならびに独自の起源をもつ OSF 版などの拡張版が並立しており、各グループ間で統一のための調整が進められている。

(5) OS-9³⁵⁾

米国マイクロウェアによって開発され、現在は MC 68000 用、MC 68020/68030 用 (MC 68881 演算プロセッササポート付) が発表されている。さらにそれぞれに対し SPU (System Protection Unit) または MMU を利用してメモリ保護を行うバージョンがあり、メモリアロケーションを16バイト単位で管理する。8ビット用に対してセクタ数増加によるディスクアクセス高速化と、RAMディスクの複数化によるユーザ RAM ディスク空間の分離などの改良点がある。

コンパクトディスク制御用 CPU である MC 68070 に搭載されて、CD-I インタフェース規格の論理フォーマットをサポートする CD-RTOS も開発された。

(6) TRON 仕様 OS³⁶⁾

東大の坂村健氏がリーダーとなり、統一プロジェクトとして各社で開発が進められている OS 群であり、社団法人 TRON 協会により運営されている。同協会より提示されたコマンドインタフェース、ファイルインタフェース、パラメータ名などに準拠し、4.1 に紹介したリアルタイム用の ITRON 仕様、教育・ビジネス・ソフトウェア開発用の BTRON 仕様、通信処理用の CTRON 仕様、これをネットワークで統括する MTRON 仕様があり、TRON 仕様 CPU やバス仕様とともにサブプロジェクトとなっている。

(7) S1-OS³⁷⁾

米国マルチソリューションズ社が開発した OS で、システム記述言語 SL (Systems Language) で記述することによる CPU 間移植性、ハードウェア依存部をドライバとして分離することによるシステム間移植性、モジュール構造による目的別適応性などに特長がある。このため組込みリアルタイム制御の43キロバイトの最小構成からワークステーション用の320キロバイトのフルセットまで、柔軟に対応できる。その結果、たとえばリアルタイムプログラムの実行環境とその開発環境を、同系統の OS で統一することができる。

(8) THEOS³⁹⁾

前稿では OASIS の名称で紹介した米国シオス社のマルチユーザマルチタスク OS である。最新版 THEOS 386 V 3.0 は 80386 のプロテクトモードで稼働し、32ビットアドレスとデマンドページング機能を利用して、ユーザ当たり4ギガバイトの論理アドレス空間と、最大2ギガバイトの物理アドレス空間をサポートする。64個の2値セマフォによりタスク間の同期を取ることができる。コマンドインタプリタ、EXEC ジョブ制御言語、UNIX ふうのファイル管理機能、画面エディタ、MS-DOS アプリケーションインタフェースなどがモジュール化されて移植を容易にしている。

(9) その他の汎用 OS

上記以外にも、最近新しい試みが進められている。西独 GMD 社と Bielefeld 大学の EUMEL は ELAN 言語で記述されており、これが EUMEL ϕ という仮想計算機の機械語に翻訳され、各ハードウェア用のドライバによって実行される³⁹⁾。

米国 Carnegie-Mellon 大学の Mach は UNIX を基本としながら、密結合のマルチ CPU に処理を分散して構造を単純化している⁴⁰⁾。

東京農工大学の OS/omicron は内部コードに漢字を含めた JIS 全2バイトコードを採用して日本語処理を容易にし、核部、ユーザ拡張部、アプリケーション部に分けて、ユーザによる目的別構成を可能にした⁴¹⁾。

5. 今後の動向

5.1 高機能化と操作容易化

当初ハードウェア利用効率向上の手段であったマルチタスク機能は、現在はオペレータの作業時間短縮のために利用されている。応用プログラム機能の OS への取り込みは、日本語処理機能やマルチウィンドウから、今後高機能グラフィック、通信、データベース、光ディスクサポートなどに遠心的に及んでゆく。

前稿では iRMX がいち早くオブジェクトの概念を採用したことを紹介したが、高機能化にともなう複雑化に対処するために、オブジェクト指向の設計法が普及してゆく。これによる不要情報の隠ぺいは操作を容易化し、安全性を高める。また、昨今問題になっているコンピュータウィルスへの耐性も向上させる⁴²⁾⁻⁴⁶⁾。

操作容易化の別の方向として、サポートツールの充実がある。ワークステーション用 OS の統一ファイル形式のもとにリアルタイム OS と応用プログラム開

発用ユーティリティが、一貫したサポートツールとして整備されてゆく。たとえばレディーシステムズ社の CARDtools と RTAda は、それぞれ VRTX の上で稼働する応用プログラムの仕様を Ada のソースコードに変換し、VRTX のシステムコールを発生するオブジェクトコードを生成する。また、モトロー社の VMEexec では、リアルタイムモニタと UNIX 開発環境の間に RTEID という基準を設け、これを満たすすべてのリアルタイムモニタをカーネルとして利用できる⁴⁰⁾。

5.2 シリコン化の加速

前稿で予測したシリコン化の動向が現時点で顕著になってきている。OS が ROM や周辺 LSI としてシリコン化され、さらに CPU アーキテクチャに吸収されるという求心的傾向は OS の遠心的傾向と対をなしており、半導体の高集積化技術と設計検証技術の進歩によって今後一層加速される。その結果、OS の流通コストは半導体の L コープに乗って急激に低下する。

シリコン化の問題点を指摘する声も当然聞かれる。OS の柔軟性が低下し、その自由な発展を阻害しないかという懸念は理解できるものであるが、OS の定形化によるソフトウェア人口の OS 以外への指向や、ソフトウェアの互換性向上といった利点のほうが損失を上まわると考えられる。

5.3 少数系統への集約化

少数系統への集約化として、モジュール化が一層進行する。必要なモジュールを組合せて目的別構成をとることにより、リアルタイムモニタから開発用 OS までを同系統の OS で統一できる。

集約化はまた、シリコン化のもたらす結果とみることもできる。一貫したコンセプトによる OS と CPU の同時開発、CPU 情報の早期入手、OS 側からの要求の CPU アーキテクチャへの反映など、半導体プロセスと密接に連携しなければ、OS のシリコン化は不可能である。また、シリコン化は LSI のマスク権と OS の著作権を同時に必要とするため、これら両方を備え得る開発元が有利な位置に立ち、このことが OS の集約化を助長する。今後は OS の開発はソフトウェア開発環境と半導体プロセスなどの装置産業を前提とした、複合形態へと移行してゆく。

6. おわりに

人類がプログラム内蔵型コンピュータを完成してから、高水準言語コンパイラの必要性を認識し、それを

完成させるまでに10年を要した。OSの実用化はそのさらに5年後である。しかし今では大型コンピュータではコンパイラとOSはハードウェアの一部であるかのように密接不離の存在であり、これらを備えてはじめて実用的なコンピュータの名に値する。

マイクロコンピュータはOSを備えるまでに5年しかかからなかった。マイクロコンピュータはある範囲までは大型コンピュータの歴史をなぞりながら、時間スケールを3分の1以下に短縮して追隨してきた。今やマイクロコンピュータは集積度とアーキテクチャの充実度で大型コンピュータに肉迫している。マイクロコンピュータ用OSは大型コンピュータ用OSの難型を脱し、本格的OSの地位を獲得したばかりではなく、加えて新しいコンセプトを実現している。これには新しい起源をもつマイクロコンピュータの自由度が幸いしている。今後データ幅64ビット、集積度500万トランジスタのCPUが普及すれば、現在の大型コンピュータ用OSをしのぎ、完全に独立したOSの世界を形成する。

本稿では相当ハードウェアに踏み込んだ記述を含めたが、これはOSを従来のような“与えられたハードウェア資源を有効に利用するソフトウェア”という技術的な捉え方でなく、“ユーザが目的を達しやすい環境を提供するための、ソフトウェアとハードウェアの融合体”という、よりユーザ寄りの捉え方をした結果である。コストや集積度のような技術的制約が大幅に緩和された現在では、このような見方が許されるのではなかろうか。半導体技術と方式設計技術がOSを通じてユーザに益するための方向づけが本稿の目的である。

前稿ではOSの仕様やCPUのアーキテクチャは一種の文化であると述べた。TRONコンセプトのOSとCPUの誕生は、日本にも文化としてのOSが育ちつつあることを示している。他からの単なる移入ではなく、その国の技術的風土に根ざして育ったものは、文化として定着してゆく。TRONプロジェクトの今後を見守りたい。

最後に、絶えず新しい情報をご提供いただいた国内外のOS関係者、ならびに議論をいただいた当学会マイクロコンピュータとワークステーション研究会の委員の皆さまにお礼申し上げます。

参考文献

- 1) 加藤, 筒井, 茶木: マイクロコンピュータ用オペレーティングシステムの現状と動向, 情報処理, Vol. 25, No. 3, pp. 232-243 (1984).
- 2) Marshall, T. and Tazelaar, J. M.: *Worth the RISC, BYTE*, Vol. 14, No. 2, pp. 245-249 (1989).
- 3) H 32/200 オペレーションアーキテクチャマニュアル, 日立製作所, 東京 (1988).
- 4) F 32/200 オペレーションアーキテクチャマニュアル, 富士通, 東京 (1988).
- 5) *MC 68030 Enhanced 32-Bit Microprocessor User's Manual*, Motorola Inc., Tempe, USA (1987).
- 6) *80386 System Software Writer's Guide*, Intel Corp., Santa Clara, USA (1988).
- 7) 河本, 矢野, 鈴木, 藤井, 椎葉: V 60/V 70 マイクロプロセッサと高信頼化システム, 情報処理学会論文誌, Vol. 30, No. 1, pp. 58-71 (1989).
- 8) 岩村, 得丸, 岡本: TRON 32 ビットマイクロプロセッサ TX シリーズ, 東芝レビュー, Vol. 43, No. 11, pp. 909-912 (1988).
- 9) 高橋: OSと命令セットアーキテクチャ, 情報処理, Vol. 29, No. 12, pp. 1463-1473 (1988).
- 10) *GMICRO F 32/IRC ユーザーズマニュアル*, 富士通, 東京 (1988).
- 11) Cheng, J. and Ushijima, K.: *Partial Order Transparency in Monitoring Tasking Behaviors of Concurrent Ada Programs*, *Proc. Int. Comput. Symp.*, pp. 583-588 (1988).
- 12) 孫, 落水: ペトリネットによる並行ソフトウェアシステムの設計時動作解析, 情報処理学会論文誌, Vol. 29, No. 8, pp. 782-789 (1988).
- 13) Cheng, J. and Ushijima, K.: *Detecting Tasking Communication Deadlocks in Concurrent Ada Programs*, *IEEE Proc. Int. Comput. Sci. Conf.*, pp. 138-145 (1988).
- 14) 赤津: デッドロック回復における強制終了プロセス決定問題の考察, 信学論 (D), Vol. J 71-D, No. 12, pp. 2536-2542 (1988).
- 15) Shatz, S. M.: *Towards Complexity Metrics for Ada Tasking*, *IEEE Trans. Softw. Eng.*, Vol. 14, No. 8, pp. 1122-1127 (1988).
- 16) 程, 荒木, 牛島: Ada 並列プログラムの事象駆動型モニタ EDEN の開発と応用, 情報処理学会論文誌, Vol. 30, No. 1, pp. 12-24 (1989).
- 17) 久野, 佐藤, 鈴木, 中村, 二瓶, 明石, 関: CLU マシンシステムの開発, 情報処理学会論文誌, Vol. 29, No. 10, pp. 966-974 (1988).
- 18) Chirlian, P. M. (武市訳): *Modula-2 入門*, アスキー出版局, 東京 (1986).
- 19) Yokohata, S., Sugita, Y., Akita, H., Ishikawa, T. and Aoki, H.: "A-RMS", *An Advanced*

- Modular Structured Realtime Operating System for Industrial Control, IECON '88*, pp. 883-888 (1988).
- 20) *iRMX 286 REFERENCE MANUAL*, Intel Corp., Santa Clara, USA (1986).
- 21) *MTOS-UX User's Guide*, Industrial Programming Inc., Jericho, USA (1984).
- 22) Ripps, D.L.: *A Guide to Real-Time Programming*, *ibid.* (1987).
- 23) *VRTX 32/68020 USER'S GUIDE*, Ready Systems Corp., Sunnyvale, USA (1987).
- 24) *VRTX 32/386 USER'S GUIDE*, *ibid.* (1987).
- 25) *pSOS+シリーズソフトウェア概説書*, オーティシー, 東京 (1988).
- 26) *RX 616 カーネルユーザーズマニュアル*, 日本電気, 東京 (1988).
- 27) *HI 68K V 2.0 ユーザーズマニュアル*, 日立製作所, 東京 (1987).
- 28) *DR DOS User's Guide*, Digital Research, Monterey, USA (1988).
- 29) *FlexOS User's Guide*, *ibid.* (1987).
- 30) *MS-DOS V 4.0 Programmer's Reference*, Microsoft, Redmond, USA (1988).
- 31) Kogan, M.S. and Rawson III, F.L.: *The Design of Operating System/2*, *IBM Syst. J.*, Vol. 27, No. 2, pp. 90-104 (1988).
- 32) Shaw, R.H.: *Using the OS/2 Environment to Develop DOS and OS/2 Applications*, *Microsoft Syst. J.*, Vol. 4, No. 1, pp. 77-88 (1989).
- 33) 近藤, 西門, 福岡, 野瀬, 村田: 2050 ワークステーションにおける WGC システム, 情報処理学会第 41 回オペレーティングシステム研究会資料, 41-1 (1988).
- 34) 市瀬, 水橋, 古城, 永作, 門田: UNIX におけるリアルタイム性の導入に関する一考察, 第 54 回マイクロコンピュータとワークステーション研究会資料, 54-4 (1989).
- 35) Dibble, P.: *OS-9 INSIGHTS*, Microware Systems Corp., Des Moines, USA (1988).
- 36) 小特集「TRON」, 情報処理, Vol. 30, No. 5 (1989).
- 37) S1-OS 概説書, シーイーシー, 東京 (1986).
- 38) *THEOS System Reference Manual*, THEOS Software Corp., Walnut Creek, USA (1987).
- 39) *EUMEL USER MANUAL*, Univ. of Bielefeld, Bielefeld, W. Germany (1985).
- 40) Walmer, L.R. and Thompson, M.R.: *A Programmer's Guide to the Mach User Environment*, Dept. Comp. Science, Carnegie-Mellon Univ., Pittsburgh, USA (1988).
- 41) 鈴木, 小林, 田中, 中川, 高橋: OS/omicon における日本語プログラミング環境, 情報処理学会論文誌, Vol. 30, No. 1, pp. 2-11 (1989).
- 42) 高橋, 太田, 白鳥, 野口: オブジェクトの概念に基づいたネットワーク OS の設計法, 信学論(D), Vol. J71-D, No. 10, pp. 2128-2139 (1988).
- 43) 土居, 児玉: 並行オブジェクト指向言語 ABCL/C+ によるオペレーティングシステム核の記述, コンピュータソフトウェア, Vol. 5, No. 4, pp. 12-26 (1988).
- 44) 大特集「オブジェクト指向プログラミング」, 情報処理, Vol. 29, No. 4 (1988).
- 45) 特集: オブジェクト指向コンピューティング, コンピュータソフトウェア, Vol. 6, No. 1 (1989).
- 46) *REAL TIME EXECUTIVE INTERFACE DEFINITION SPECIFICATION*, Motorola Inc., Tempe, USA (1988).

(平成元年2月20日受付)