

# マイクロコンピュータオペレーティングシステム

## Microcomputer Operating System

マイクロコンピュータ応用システムの大規模化・複雑化に伴い、OSの役割と重要性が高まっている。このような中において、システム作成者側でOSを自主開発するには、コストやリスクの面で問題があること、システム利用者側で既存ソフトウェアが利用でき、かつ実績のあるOSへのニーズとが合致して、汎用OSが多く利用されようとしている。

汎用OSのうち、8080及び8086のCP/M<sup>\*1)</sup>と互換性があるシングルユーザーシングルタスク機能をもつCP/M-68K<sup>\*1)</sup>、ミニコンピュータやメインフレームの共通OSになる可能性があり、マルチユーザーマルチタスク機能をもつUNIX<sup>\*2)</sup>、リアルタイム機能をもつRMSが、68000応用システムで実用化されている。

尾石辰郎\* *Tatsurô Oishi*  
 矢部栄一\* *Eiichi Yabe*  
 菅原正彦\* *Tadahiko Sugawara*  
 茶木英明\* *Hideaki Chaki*  
 筒井茂義\*\* *Shigeyoshi Tsutsui*

### 1 緒言

OS(オペレーティングシステム)及びその応用ソフトウェア開発の労力を軽減したいというニーズと、応用ソフトウェアを広範に利用したいというニーズから、汎用OSへの期待は大きい。一方、16ビットマイクロコンピュータ採用のきっかけは、8ビットマイクロコンピュータで実行していたものを高機能化するためと、ミニコンピュータで実行していた機能を小形化・低価格化するためと大きく二つに分かれる。これに対応する16ビットOSには、メモリ管理、CPU(中央処理装置)管理、デバイス管理、ファイル管理、タスク管理など多くの機能をもつことが望まれ、シングルユーザーOS、マルチユーザーOS、リアルタイムOSに分けて利用されている。シングルユーザーOSは、あまり多くの機能を備えずコンパクトさを特長とし、マルチユーザーOSは、不特定ユーザーの利用がその根底にあるので、十分なOSの機能が求められ重装備になってくる。リアルタイムOSは、多少のカスタマイズで済むものから重装備が必要なものまで千差万別である。ここでは、各各に対応し、かつ68000で活用できるOSとしてCP/M-68K<sup>\*1)</sup>、UNIX<sup>\*2)</sup>及びRMS(Realtime Monitor System)について述べる。

### 2 CP/M-68K

シングルユーザーシングルタスク機能をもつCP/Mは、その優れた移植性のため、パーソナルコンピュータの標準OSとしてゆるぎない地位を築いている。発売当初の8080、Z-80内蔵機種用、その後の8086内蔵機種用を合わせると、販売部数は200万部を突破している。また、CP/M上で作られた言語プロセッサ、ユーティリティ、応用ソフトウェアは膨大な数に上る。既存のCP/M下で動作するこれらのソフトウェア資産を16ビット68000内蔵機種で利用できるようにすることがCP/M-68K開発の主目的であり、既にCP/M-68Kのセットメーカーは40社を数えている。

### 2.1 機能及び特徴

#### (1) 既存CP/Mとの互換性

CP/M-68Kは8ビットマイクロコンピュータ8080、Z-80用CP/M2.2、8086用CP/M-86 1.1に対して上位互換性をもっている(表1)。ディスクメディアの片面単密度8inフロッピーディスクについては統一した形式を採用しており、既存CP/Mとファイルが可換になっている。また、システム機能[BDOS(Basic Disc Operating System)及びBIOS(Basic Input/Output System)]もマイクロプロセッサの機能に依存するわずかな項目を除いて同一である。コマンドやコマンドのもつサブコマンドの指定方法も同一で、親和性をもたせてある。

#### (2) 既存応用ソフトウェアの活用が可能

CP/M-68K下で動作する高級言語C、CBASIC、FORTRAN

表1 CP/M-68Kの仕様 CP/M-68Kは、128kバイトのRAM、フロッピーディスク2台及びCRTディスプレイ1台があれば利用できるようになる。

項目	仕様	
ハードウェア構成	最小	(1) 68000 CPU (2) 128kバイト RAM (3) フロッピーディスク装置 2台 (4) CRTディスプレイ 1台
	拡張	(1) 16Mバイトまで主記憶拡張可能 (2) プリンタ装置 1台 (3) 補助入力装置 1台 (4) 補助出力装置 1台
ディスク装置(最大)	(1) 16ドライブ(A~P)まで (2) 1ファイルの大きさ 32Mバイト (3) 1ドライブの大きさ 512Mバイトまで	
OSの機能	(1) 単一ユーザー単一タスク (2) BDOS機能 46種類 (3) BIOS機能 22種類	
コマンドOSの大きさ	(1) 組込みコマンド 7種類 (2) トランジェントコマンド 16種類 (1) 24kバイト(BDOS+CCP)+BIOSの大きさ	
互換性	CP/M 2.2及びCP/M-86 1.1に対して以下の互換性がある。 (1) 8in片面単密度フロッピーディスク形式 (2) BDOS、BIOS機能(ただし、ハードウェア依存項目を除く。) (3) コマンドの指定方法	

注: 略語説明 OS(オペレーティングシステム), CPU(中央処理装置), CRT(Cathode Ray Tube), BDOS(Basic Disc Operating System), BIOS(Basic Input/Output System), CCP(Command Control Processor)

\*1) CP/M, CP/M-68Kは、デジタルリサーチ社の登録商標である。  
 \*2) UNIXは、米国ベル研究所のオペレーティングシステムの名称である。

及びS-PL/Hが開発されており、これら高級言語で記述された応用ソフトウェアを、そのまま又は若干の修正で利用できる(図1)。

(3) コンパクトなOSで操作が容易

言語プロセッサ使用時には、メモリ容量128kバイトを必要とするが、OS本体部の大きさは24kバイトでコンパクトにまとまっている。また、複数コマンドの結合入力機能やエラー回復処理機能の支援など、マンマシンインタフェースの強化を図っている。

(4) 各種マシンへの移植が容易

図2に示すように、CP/M-68Kの構造はハードウェアに依存する部分と依存しない部分とに分けることができる。依存する部分はBIOS[Basic I/O(Input/Output) System]といい、各システム固有のキーボード、CRTディスプレイ、ディスク装置その他のI/Oポートを制御するルーチンを含む。このBIOSをそれぞれのマシンに合わせて独自に作れば、CP/M-68Kのその他の部分はどんなマシンでも共通に利用できる。

(5) 68000のアーキテクチャを有効活用

68000は16Mバイトのメモリ空間を直接アクセスできる。そこでOS本体部やユーザープログラムを16Mバイトのメモリ空間のどこにでも配置し、実行できるようになっている。このほか、例外処理機能の強化、ユーザープログラムをスーパーバイザモードで動作可能にしているため、OSレベルのプログラム開発ができるなどの機能を強化している。

(6) C言語コンパイラを支援

CP/M-68Kでは標準として、C言語コンパイラを用意している。このC言語コンパイラはC言語の標準入出力機能を実行時ライブラリとして実現しており、アセンブラをほとんど意識することなくプログラムの開発が行なえる。また、UNIX系OSの標準言語CがCP/M-68Kで利用できることにより、UNIX系応用ソフトウェアをCP/M-68Kで利用できる。

2.2 CP/M-68Kの移植方法

前述したように、マシンに合わせてBIOSを開発すればCP/M-68Kを移植できる。BIOSは22種類の機能から成る。これらの機能は、(1)OSスタート時のマシン初期化機能を含むシス

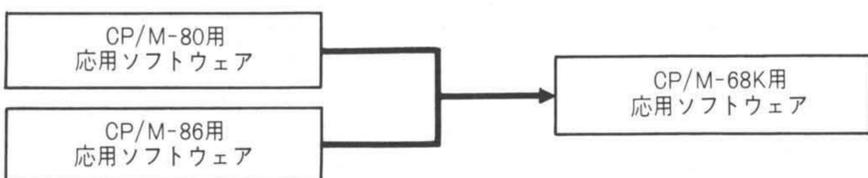
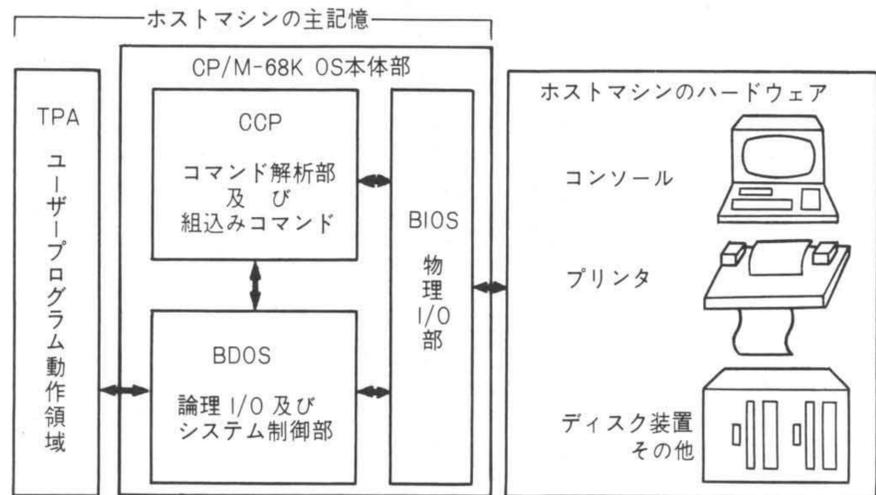


図1 応用ソフトウェアの活用 既存のCP/M-80用応用ソフトウェア及びCP/M-86用応用ソフトウェアがそのまま又は若干の修正で利用できる。



注：略語説明 BDOS(Basic Disc Operating System)  
BIOS(Basic Input/Output System)  
I/O(入出力装置)

図2 CP/M-68Kの構造 ハードウェアに合わせてBIOSを独自に作れば、CP/M-68Kを移植できる。

表2 BIOS実現時の検討項目 BIOS実現時には、入出力装置、ディスク形式及びシステムをどのように制御するかを検討しなければならない。

分類	検討項目	関連BIOS機能
1文字入出力	<ul style="list-style-type: none"> <li>●ターミナルとの接続方法</li> <li>●プリンタとの接続方法</li> <li>●補助入出力装置との接続方法</li> <li>●論理入出力装置と物理入出力装置との対応方法(IOBYTEサポート)</li> </ul>	# 2 (CONST) # 3 (CONIN) # 4 (CONOUT) # 5 (LIST) # 6 (AXO) # 7 (AXI) # 19 (GET IOBYTE) # 20 (SET IOBYTE)
ディスク入出力	<ul style="list-style-type: none"> <li>●ディスク台数</li> <li>●ディスク属性                             <ul style="list-style-type: none"> <li>スキューファクタ、セクタ数/トラック</li> <li>セクタ数/ブロック、ディレクトリ数</li> <li>ディスクの先頭からのスキップトラック数</li> </ul> </li> <li>●ディスクコントローラとの接続方法</li> <li>●ブロックング/デブロックングの方法</li> <li>●バッファのフラッシュ方法</li> </ul>	# 8 (HOME) # 9 (SELDSK) # 10 (SETTRK) # 11 (SETSEC) # 12 (SETDMA) # 13 (READ) # 14 (WRITE) # 16 (SECTTRAN) # 21 (Flush Buffers)
システムコントロール	<ul style="list-style-type: none"> <li>●マシンの初期化</li> <li>●ユーザープログラム領域の大きさ・範囲決定</li> <li>●例外ベクタの設定</li> </ul>	# 0 (INIT) # 1 (WBOOT) # 18 (GET MRT Address) # 22 (SET Exception Vector)

テムコントロール機能7種、(2)キーボードやターミナルとの1文字入出力機能6種、(3)ディスク装置に対する物理入出力機能8種、の三つに大別できる。BIOSプログラムは全体を一つのモジュールとし、22種類の機能に対応した関数形手続きで構成、TraP # 3で呼び出されるBIOSのエントリーポイントから各機能の手続きへ制御を移すことになる。BIOSプログラムを実現するに当たっての検討項目を表2に示すが、ディスク属性に関する情報はDPH(ディスクパラメータヘッダ)とDPB(デバイスパラメータブロック)のテーブルで定義しなければならない。BIOSプログラム開発後、H680SD200などのCP/M-68Kマシンを使用してアSEMBル又はコンパイル後、マシンに依存しない核の部分と結合すればCP/M-68Kを生成できる。

3 UNIX

UNIXは、ベル研究所で開発されたミニコンピュータ用の会話形OSで、TSS(タイムシェアリングシステム)モードによるマルチユーザーシステムをサポートしている。UNIXは、そのシンプルな構造や豊富なツール、使い勝手の良さなどから優れたソフトウェア開発環境として評価が高く、現在ではメインフレームやマイクロコンピュータへ広がっている。特に16ビットマイクロコンピュータ68000では、UNISOFT社のUNIPLUSなど相当数のUNIXが開発されて、強力なグラフィック機能とかOA用支援ツールなどの整備が行なわれている。今後UNIXは、CP/M系のOSとともに16ビット及び32ビットマイクロコンピュータの世界標準OSの一つになると見られている。

3.1 機能及び特徴

(1) 階層構造のディレクトリ

ファイルの名称と実体を結びつけるディレクトリは、図3に示すように、階層構造になっている。ファイルはすべて最上位のルートディレクトリから始まる階層構造のディレクトリの枝の先端につながっている。一つのファイル実体は、複数のディレクトリにつなぐことができ、すなわち同一のファイル実体を異なる名前で指定することができる。

入出力装置も一種のファイルとしてディレクトリの下につ

ながれており、通常のファイルと同様に扱われる。端末起動時には、カレントディレクトリと呼ばれるあらかじめ決められた位置のディレクトリがユーザーに与えられ、そこを起点としてファイル名の指定を行なえる。

(2) 単純なデータ構造

UNIXのシステムが直接取り扱うファイルは、1バイト単位のデータ列であり、ファイル編成、アクセス方法、レコードタイプ、レコード長などといったファイル構造に関する考え方はもっていない。ソースプログラムが格納されているテキストファイルは改行文字で各行が区切られているが、改行文字を何にするかなどはシステム側ではなく、エディタやコンパイラがその構造を管理する。

このようにシンプルなデータ構造をもつことにより、ファイル構造に起因する種々の制約から解放され、後述するパイプ機能や数多くの有益なアプリケーションを可能としている。

(3) 標準入出力と切替え

UNIXで実行されるユーザープログラムは、3個の入出力ファイルをオープンして起動される。これらは特に指定しない場合は、標準入出力装置(ログインした端末)がそれぞれに対応するファイルとして割り当てられる。3個の入出力ファイルのうち、2個はそれぞれ標準入力、標準出力で、これらは通常のファイル入出力用として使用される。残り1個は標準エラー出力で、エラーメッセージ出力用に使われる。

いずれの場合も、ユーザープログラムの起動時にコマンドライン上でそれぞれの標準入出力に対応するファイルを指定することにより、ログイン端末以外のファイルに切り替えることができる。すなわち、ユーザープログラム上では実行時の入出力ファイル(装置)については特に意識する必要はなく、実行時に目的に合った入出力ファイルをコマンド上で指定すればよい。

(4) パイプ

UNIXでは、ファイルに対する入出力と同様の手順により、プロセス(ユーザープログラムの実行単位)間の通信が可能で、この経路をパイプと呼ぶ。

このパイプを利用し、ユーザープログラム間でのデータの授受が容易に実行できる。UNIXがサポートするコマンドの多くは、標準入力からの入力データを処理し、標準出力へ出力する。このようなコマンド間で、先に実行したコマンドの出力結果を次のコマンドの入力として使う場合は、パイプでこれらのコマンドを連結させる。これにより、各コマンドは並列に起動され、直前のコマンドの標準出力へ出力されたデ

ータはパイプを通して次のコマンドの標準入力から入力される。

(5) shell

shellは、UNIX下でユーザープログラムの一つとして動作するコマンドインタプリタで、ログイン端末ごとに別々のshellが割り当てられ、ユーザーのキーインするコマンド列の解析及びコマンド(ユーザープログラム)の起動を行なう。このとき、ワイルドカード(任意の文字列に置き換え可能な)文字によるファイルのサーチ、標準入出力の切替え、パイプによる連結なども行なう。

コマンドファイルの場合は、shell言語と呼ばれる一種のプログラム言語でコマンド手続きを記述することができる。shell言語では、if-then-else, while-doなどの条件文や繰返し文によるコマンドの実行制御、専用の変数への実行結果の代入、パラメータの変更などが可能で、エラーの有無による実行順の変更とか、既存コマンドの組み合わせによる独自のコマンドの作成などができる。

3.2 UNIXの移植方法

(1) 開発環境

UNIX移植を行なう場合、始めにその移植作業用の開発環境を整備する必要がある。移植は、ターゲットマシン用のクロスCコンパイラ、クロスアセンブラ、クロスリンカージェネリタなどによるクロス環境で行なう。一般的には、VAX11上のUNIX〔特に4.1BSD〕で行なわれる場合が多い。このクロスシステム上で改造作業を行ない、ターゲットマシン用のオブジェクトプログラムを作成する。

次に、このオブジェクトプログラムをターゲットマシン上に移すが、このために回線接続、フロッピーディスクなどによるダウンロード手段が必要である。ターゲットマシン側には上記出力を読み込むローダとデバッガが必要である。

(2) ハードウェア環境

UNIXを搭載し、ある程度の性能を得るためには次のようなハードウェア構成が最小限必要であろう。

- (a) メモリ 512kバイト
- (b) メモリ管理用ハードウェア
- (c) ハードウェアタイマ
- (d) CRTコンソール 1台
- (e) ハードディスクドライバ 1台(20Mバイト)

このほか、ダウンロードの方法によってはフロッピーディスクドライバ又は回線接続用ポートなどが必要である。

(3) 移植作業

UNIXをターゲットマシン上に移植するには、カーネル(核)の中でハードウェアに依存している各種入出力ドライバ、割込み処理及びメモリ管理部分の改造を行なう。入出力ドライバはカーネル内のデバイス管理テーブルを介してつながっており、該当する装置のドライバをターゲットマシンの仕様に従って変更する。メモリ管理は、ユーザープログラムの実行制御を行なうプロセス管理部の中にある。VAX11では二重のページテーブル方式によるメモリ管理を行なっているが、既に発売されている16ビットマイクロコンピュータUNIX搭載マシンでも、これと類似したページテーブル方式を採用しているものが多い。これは改造量を極力少なくするためであり、また他の機能への影響も少ない。UNIXでは、ユーザープログラムのデータ領域、スタック領域の拡張のためにメモリの割付けを動的に行ない、領域が不足するとスワッピングを行なっている。これをターゲットマシンのメモリ管理に合わせて改造を行なうが、性能の低下を防ぐためスワッピングの多発とか、メモリ内容の移動量の増加がないように十分注意し

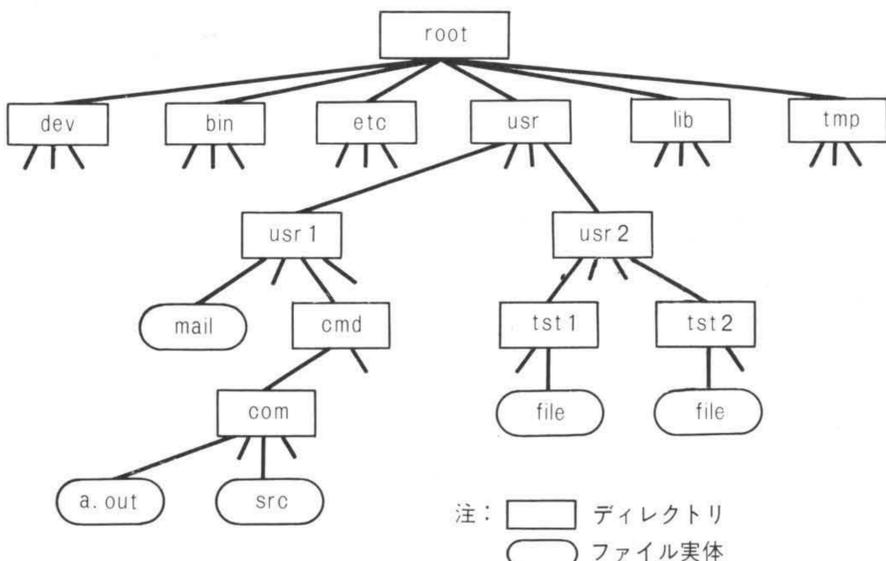


図3 階層構造のディレクトリ ファイルは、ルートディレクトリから始まる階層構造のディレクトリの枝の先端につながっている。

なければならない。

#### 4 RMS

リアルタイム制御にマイクロコンピュータを応用しようとするユーザーに対して、OSの開発の労力を軽減するために、リアルタイムOS、RMSを販売している。

##### 4.1 RMS利用による利点

RMSを利用することにより、以下の利点が得られる。

- (1) タスクと呼ばれる独立した処理プログラムの登録・抹消・優先度の変更が可能で、ユーザー独自のOSが容易に作成でき、また拡張性も優れている。
- (2) ユーザーは、高い応答性が必要なシステムでの複雑なアルゴリズムをRMSにゆだねることにより、独立した機能をもつタスクだけを作成するだけでよい。

上記利点により、ユーザーの開発工数が削減できる。

##### 4.2 68000RMSの特長と機能

68000RMSの特長を以下に述べる。

- (1) RMSは、日立シングルボードコンピュータ用にROM(Read Only Memory)で提供され、ユーザー実機に組み込みやすくなっている。
- (2) 割り込みやI/O装置など、ユーザーシステムによって異なるハードウェア構成は、テーブルの形で定義するだけでよく、ハードウェアに対する適応性が高い。
- (3) テーブルで吸収できないユーザー仕様についても、任意に入出力制御・割り込み処理プログラムをRMSに組み込めるように配慮されている。

表3 RMS機能一覧表 RMSにはタスク管理、入出力管理の核になる部分のほかにデバッグ管理、ファイル管理機能を含む。

品 種	HD6800用 RMS	HD6809用 RMS	HD68000用 RMS	
提供媒体及び容量	マスクROM, 4kバイト	マスクROM, 8kバイト	EPRM, 40kバイト	
タスク管理	最大タスク数	32	255	
	タスク制御	●起動・停止 ●POST/WAIT	●起動・停止 ●POST/WAIT ●ENQ/DEQ ●禁止・解除	●起動・停止 ●POST/WAIT ●ENQ/DEQ ●禁止・解除 ●中断・解除 ●メッセージ交換 ほか
	排他制御資源数	—	255	255
	タイマ	タイマ数 255 最小単位 10ms	255 10ms	255 1ms
入出力管理	入出力装置数	64	255	255
	標準入出力装置	●コンソール (RS232Cレベル) (TTLレベル) ●電子式卓上計算機タイプコンソール	●コンソール (RS232Cレベル) (TTLレベル) ●電子式卓上計算機タイプコンソール	●コンソール (RS232Cレベル) (TTLレベル) ●プリンタ (セントロインタフェース) ●フロッピーディスク
デバッグ機能	コマンド数	9	19	23
	デバッグ機能	●タスク制御 ●タスク状態表示 ●メモリダンプ ●メモリ変更 ほか	●タスク制御 ●タスク状態表示 ●メモリダンプ ●メモリ変更 ●ブレイクポイント ●事象トレース ほか	●タスク制御 ●タスク状態表示 ●メモリダンプ ●メモリ変更 ●ブレイクポイント ●事象トレース ●メモリ転送 ほか
ファイル管理機能	オプション	計 画 中	標 準	

注：略語説明 RMS(Realtime Monitor System), ROM(Read Only Memory)  
EPRM(Erasable and Programmable Read Only Memory)  
TTL(Transistor Transistor Logic)

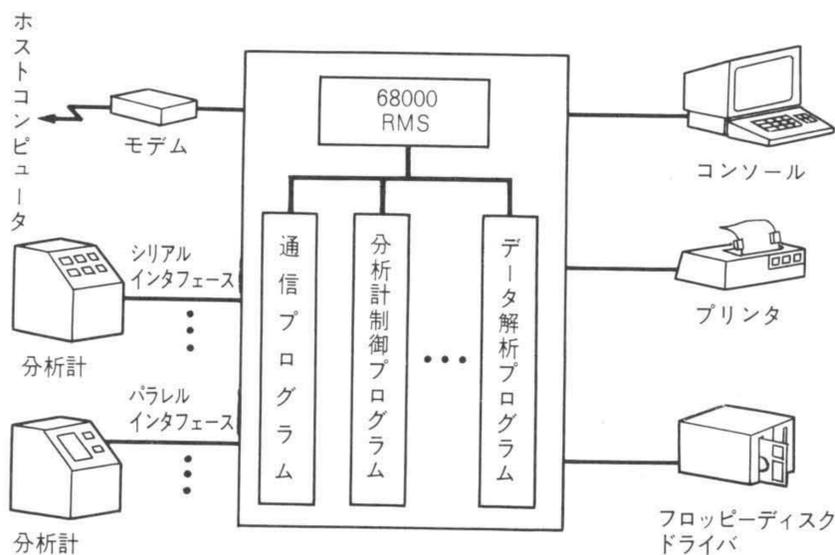


図4 68000RMS応用例 68000RMSを用いた臨床検査システムの例である。RMSを核とし、物理入出力は入出力ドライバとしてRMSに組み込み、論理的制御及びデータ処理はタスクとして登録している。

- (4) 高速割り込みに対するユーザーのダイレクト処理が可能である。
- (5) オンラインデバッグが可能である。また、ファイル管理機能を用い、フロッピーディスクとの入出力が可能である。

表3にRMSの機能一覧を示す。

##### 4.3 68000RMSの応用例

検査システムにRMSを用いた例を図4に示す。複数の分析計から非同期に発生するデータを実時間で多重処理し、ホストコンピュータとの通信により、情報の集中管理を行なう例である。

また、システム管理者は、コンソールからシステムの状態を見ることができ、プリンタ、フロッピーディスクに分析結果を出力させることも可能である。更に、RMSの資源管理機能を用いることにより、安全性の高いシステムとなっている。

本システムは、RMSの多重処理機能、資源管理機能、タイマ管理機能及び各種入出力管理機能を有効に活用している例である。またRMSは販売管理ほかのOA用途にも応用できる。

#### 5 結 言

68000応用システムを効率良く開発するため、汎用OSを利用する機会が増えている。68000汎用OSで代表的なCP/M-68K, UNIX, RMSの機能・移植方法について述べた。今後リアルタイム機能、マルチウインドウなどのユーザーフレンドリー機能、ネットワーク機能、データベース機能、日本語処理などの機能拡張、多目的に利用でき軽量OSを目指すためのOSのモジュール構造化、更にOSの機能をプロセッサ内に含めたシリコンOSへ発展すると思われる。

#### 参考文献

- 1) 尾石, 外: CP/M-68Kの特徴と機能, インタフェース, pp. 182~190(1983-10)
- 2) 茶木, 外: CP/M-68K移植の基礎, インタフェース, pp. 253~268(1983-10)
- 3) 加藤, 外: 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, pp. 232~243(1984-3)
- 4) 石川, 外: 16ビット・マイクロプロセッサ68000用OS CP/M-68Kの開発, 日経エレクトロニクス, pp. 147~173(1983-4)