

## スーパーパソコン「アポロDOMAIN」のアーキテクチャー

日本ラッド株式会社 技術開発部  
山村紀夫 堀尾章一

### 1. まえがき

アポロDOMAINは、1980年に、アポロ社にて開発された、32ビットスーパーパーソナルコンピュータである。CPUには、16ビットマイクロプロセッサ「MC68000」が2個使われている。OSは、UNIXと同じ思想にもとづく「AEGIS」が提供されており、対話性にすぐれている。またディスプレイは、ビットマップ方式で、きわめて高解像度の画質を得ることができ、画面を複数の短形エリアに分割して、最大15個の処理を、同時に実行することができる。さらに複数のアポロDOMAINは、同軸ケーブルを介して、容易に、リング状のローカルネットワークを構成することができる。ネットワーク内の各資源は、「オブジェクト」と呼ばれ、ツリー構造のディレクトリーに集約される。これによって、システム資源の共有性を高めている。

### 2. アポロDOMAINの設計思想

アポロDOMAINには、いくつかの設計思想がある。

まず、各々のユーザーごとに、専用のCPUが存在する。これは、応答が速く、操作しやすいコンピュータを提供するためである。

第2に、ユーザーの専用CPUは、高性能ローカルネットワークを介して、互いに接続される。これは、ネットワークのモジュール化により実現される。ユーザーの専用CPUは、それだけで、1つのシステムになるだけでなく、ネットワークに接続すれば、それ全体の資源を共有することのできる、1つの完結したシステムになる。

第3に、高度の抽象性にもとづいたアーキテクチャーを構築している。これはネットワーク内の各オブジェクトを、96ビットの、ネットワーク仮想アドレス空間の中にマップして、それらをツリー構造のディレクトリーに集約する、ということにより実現される。

第4に、現代の最先端技術を利用している。これは、VLSI、ウィンチエスターディスク、さらにIEEE標準案マルチバス等の採用により実現される。

### 3. ノードの構成

ノードとは、単一のアポロコンピュータのことをいう。アポロノードは、図1に示すように、CPUキャビネット、ディスプレイ装置、キーボードの3つで構成される。

CPUキャビネットには、CPU、メインメモリー、ネットワークインターフェイス、3つのRS232CシリアルI/Oポートなどが入っている。オプションとして、ウィンチエスターディスク、フロッピーディスクドライブなどがある。また、キーボードには、標準ASCIIキーの他に、画面コントロール用の特殊キーがある。そして、CPUキャビネットとディスプレイ装置はフラットケーブルで、ディスプレイ装置とキーボードはコイルケーブルで、各々接続される。

標準的なアポロノードは、512KBメインメモリー、33MBウィンチェスターディスク、1MBフロッピーディスクドライブ、縦型グリーンディスプレイ、およびキーボードで構成される。これらのコンポーネントは、アプリケーションの用途に応じて、次のように変えられる。

- メインメモリーは3.5MBまで拡張できる。
- ウィンチェスターディスクは66MBもある。
- ディスプレイは、横型ブラック&ホワイトもある。この場合、キーボードにはタッチタブレットが付く。
- 本年度中には、4または8プレーンの、横型カラーディスプレイが提供される。

また、以下の周辺機器を取付けることができる。

- 300行/分または600行/分のドットラインプリンター。
- 60字/秒のバトミントンプリンター。
- 300MB外部記憶装置。これは1台のノードに2台まで取付けられる。
- 1600BPI自動ロード磁気テープ装置。

その他、上記以外のオプションとして以下のものがある。

- 浮動小数点高速処理ハードウェア装置と4KBのキャッシュメモリーが入っているパフォーマンスボード。
- IEEE-P796互換デバイスコントローラー作成用のGPIOソフトウェアパッケージ。

#### 4. ノードの内部構造

アポロノードの内部構造は、図2のようになっている。

CPUは、2台のモトローラ68000で構成される。2台のモトローラ68000をA、Bとすると、通常は、Aのみが動き、システムを管理している。その間、Bはidle状態になっている。Aでメモリ管理上のトラブル(例えば、ページ不在など)が発生すると、Aは中断され、トラブル解決のために、Bにシステムの制御が渡される。Bによってトラブルが解決されると、制御は改めてAに渡され、処理が続行される。

このCPUはメモリ管理装置(MMU)に接続される。MMUはCPUからの

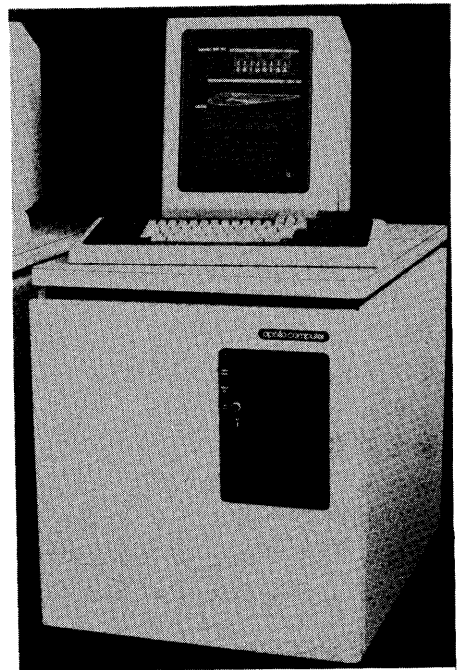


図1. アポロノードの外観

24ビット仮想アドレスを、22ビット物理アドレスに変換する。MMUは実際には2つの部分からなり、1つはCPU、1つはI/Oシステムのためのものである。メモリは、誤り訂正符号(ECC)で保護されている。I/Oシステムは、さらに2つの部分に分けられる。1つはウィンチェスターディスクやネットワークコントローラといった、アポロシステムに欠くことのできない周辺機器で、これらはブロックマルチプレクサーチャンネルに接続される。他の周辺機器、例えば、ラインプリンター、磁気テープ装置、外部記憶装置、さらにユーザーが用意した周辺機器などは、ロードパリアフェラルアダプター(マルチバスコントローラ)に接続される。

ディスプレイシステムには、各辺が1024ビットの正方形からなる、128KBのビットマップ・ディスプレイメモリがある。常時800X1024ビットのエリアが、30(B&G)または60(B&W)サイクルでリフレッシュされ、画面に表示される。残りのエリアはキャラクタフォントテーブルとして使用される。キャラクタフォントテーブルには、画面に表示される各種ASCII文字のパターン情報が入っている。さらに、ビットブロックトランスファーがある。これはビットムーバーとも呼ばれ、画面上の1つの短形エリアを別の場所に移動したり、ディスプレイメモリとメインメモリ間のブロック転送を行ったりするハードウェア機構である。データ転送率は、32Mビット/秒である。ディスプレイメモリ

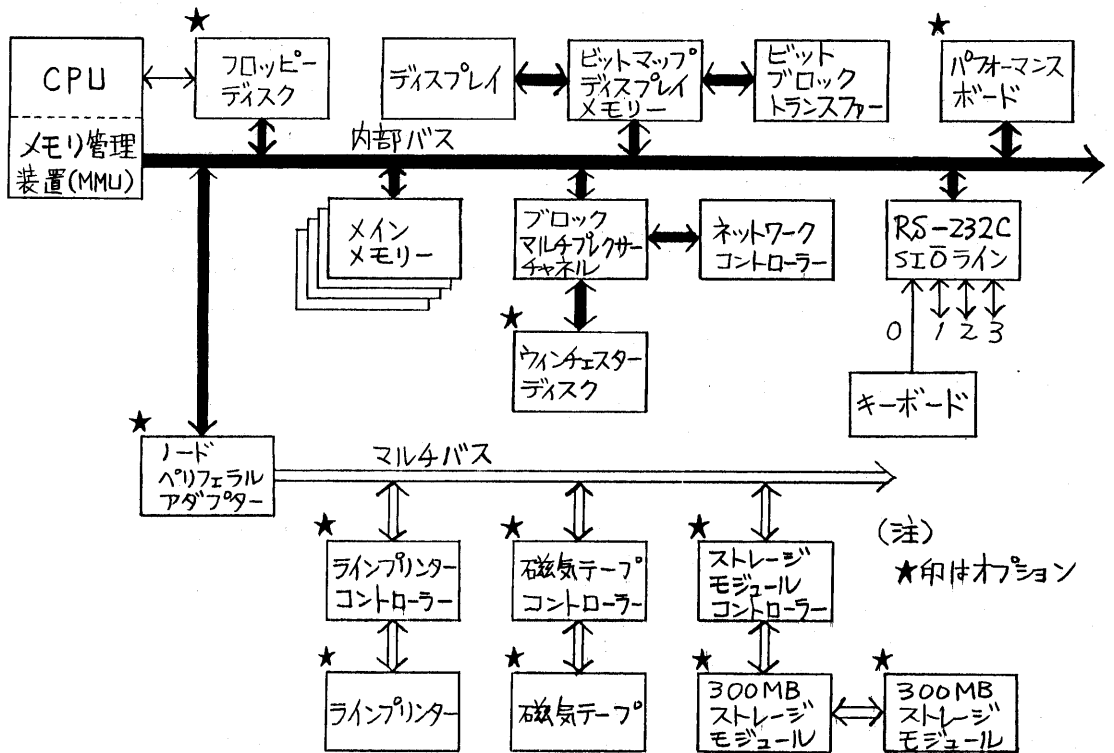


図2. アポロノードの内部構造

ーとメインメモリーが分かれていることにより、CPUによるメインメモリーアクセス、ディスプレイメモリーからの画面リフレッシュ、さらにビットムーバーによるブロック転送は、同時に、互いに干渉することなく行なわれる。

RS-232Cシロラインには、4つのポートがある。ポート0は受信専用で、キーボードとの接続に使われる。他の3つのポートは各々、50~19.2Kボーマでの転送幅を指定できる。

## 5. システム環境

システム環境は3つの特長がある。第1にネットワークのモジュール化、第2に高性能ローカルネットワーク環境、第3にネットワーク対話機能の極大化である。

### 5.1 システム構造

ネットワークレベルでのモジュール構造により、ユーザーは、特定のアプリケーションに応じて最も経済的なシステムを構築することができる。また、システムの拡張や変更の際に際して、大がかりな再プログラミングが不要であることも意味する。そしてアポロのMAINネットワークに接続されたユーザーは、それぞれ相互に交信したり、ネットワーク内の共有プログラムや共有データファイル、さらに周辺機器にアクセスしたりすることが、容易にできる。

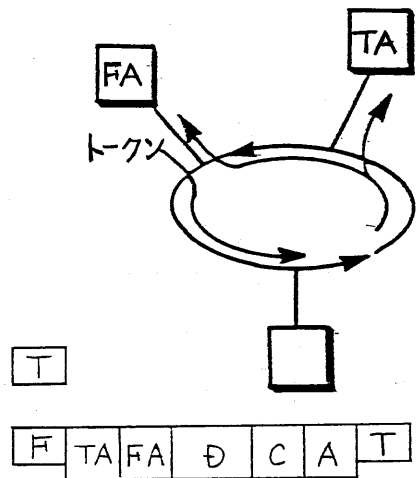
### 5.2 リングネットワーク

アポロネットワークは、リング構造トークン方式を採用している。

リング構造は、一般に高いデータバンド幅(12Mビット)と長い距離(ノード間は、最大1000m)を可能にし、光ファイバーのような新しい技術への移行も容易である。

トークンは、特別にエンコードされたビット列で、ネットワーク通信を調停する機能を持つ。リング上にはただひとつのみ存在する。どのノードからも送信要求がない場合、トークンはリングを循環している。そして、あるノードに送信要求が入力されると、そのノードはトークンを占有し、そのノードだけが、ネットワーク内でメッセージを送信できる。

リング上でのメッセージフォーマットを図3に示す。図からわかるように、ア



- T : トークン (= 011111100)
- F : フレーム (= 011111101)
- TA : 受信先アドレス
- FA : 送信元アドレス
- D : 送信データ (1KB)
- C : CRC フィールド
- A : ACK フィールド

図3. メッセージフォーマット

プロシステムでは、2アドレスパケット伝送機構を用いている。パケットの同期をとるために、いくつかの特別なフラグ文字がある。これらは、6個の連続したビット1とそれに続く2ビットの識別ビットで構成される。トークンもこのひとつであるが、最終ビットで、他のフラグ文字と区別され、その最終ビットを変えるだけで、1つのノードが1つのトークンを占有できる。このことが、各ノードのバッファリングを最小にし、ネットワークの応答を最大にしている。

また、ネットワーク内のあるノードがダウンしていたり、電源が切れていたとしても、他のノードは何ら支障なく、そのノードを除いたシステム資源を共有できる。しかし、ノードを接続する同軸ケーブルが切れている時は、これにあてはまらない。この場合、各ノードはスタンドアロンで動く。リング上を走るネットワーク信号は、ノードごとに増幅されて、1方向に循環するだけで、あるノードで迂回したりはしない。

## 6. プロセス環境

プロセス環境には、主に3つの特長がある。第1に、プログラムやデータファイルのような共通の実体が、オブジェクトと呼ばれるような抽象体に統一されることである。第2に、ネットワークワイドの仮想メモリーを実現するために、デマンドページシステムがあるということである。第3に、シェルプログラミングによって、プロセス間のストリーミングが効率よく処理されることである。

### 6.1 システムネーム空間

プロシステム内で生成されるネーム空間はいくつもある。主なものを図4に示す。

ユーザーグローバルネーム空間は、ユーザーがプログラムを実行したり、ファイルにアクセスするために、端末から入力する空間である。これは後述するディレクトリーのパス名を意味する。

システムグローバルネーム空間は、OSがネットワークレベルで用いるネーム空間である。これは64ビットのUIDと32ビットのオフセットで構成される、96ビットのアドレス空間である。

UIDは、オブジェクトの生成時刻とノードの製品番号を符号化したもので、オブジェクトの名前を表す。従って、いついかなる時も、同一の

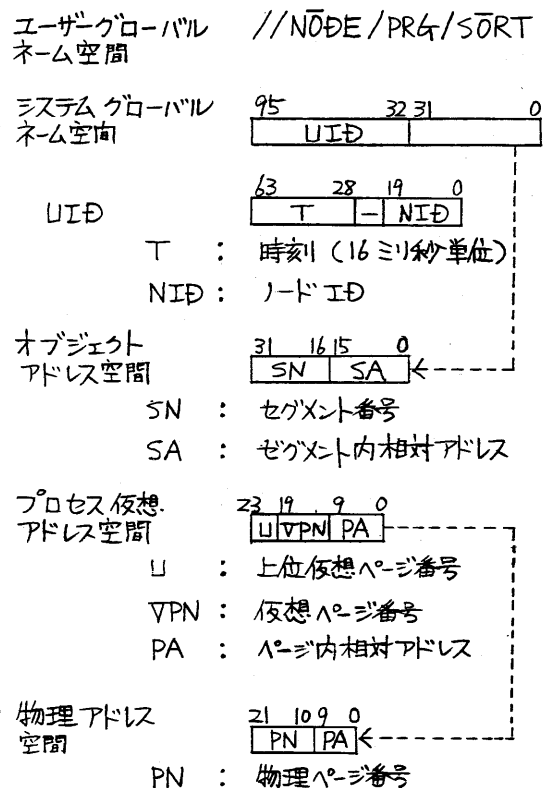


図4. システムネーム空間

UIDが生成されることはない。

オブジェクトアドレス空間は、32ビットであり、ひとつのノードに存在する、プログラムやファイル、さらにOSやその他の実体を表わすものである。ノードごとに、4kBの容量を持つ。

プロセス仮想アドレス空間は、24ビットで表現され、モトローラ68000がプロセスを実行するアドレス空間である。

プロセス仮想アドレス空間は、MMUによって、22ビットの物理アドレス空間にマップされる。物理アドレス空間の割付けは、1024Bのページ単位で行なわれる。

## 6.2 システムの相関関係

アポロシステムでは、ユーザーコマンドの実行は、以下の手順で行なわれる。

まず、ユーザーが1つのコマンド(パス名)を入力すると、これはネームサーバーによってUIDに変換される。ネームサーバーは、パス名とUIDを対応づけるOSのコンポーネントである。

次に、OSマッピングストラクチャーが、UIDから、オブジェクトアドレス空間を求め、それをプロセスアドレス空間にマップする。オブジェクトアドレスとリモートディスク内のデータブロックを物理的に対応づけるのは、ディスクストラクチャーが行なう。いったんプロセス仮想アドレス空間に、マッピングが行なわれれば、CPUが要求しない限りデータは転送されない。

そして、MMUがセットアップされ、これはプロセス仮想アドレスを物理アドレスに変換する。そしてプロセスの処理が行なわれる。もし、処理の途中でページ不在が発生すると、ページングシステムが、ネットワーク内のリモートディスクから、要求ページを検索し、物理メモリにそれを転送する。

## 6.3 メモリ管理装置(MMU)

MMUは、モトローラ68000からの24ビット仮想アドレスを、アポロノードの22ビット物理アドレスに変換するハードウェア装置である。

仮想アドレスを物理アドレスに変換する方法は、おおよそ次の通りである。図5に示すように、24ビット仮想アドレスは、3つのフィールドに分割される。

- 上位仮想ページ番号(U)
- 仮想ページ番号(VPN)
- ページ内相対アドレス(PA)

10ビットの仮想ページ番号は、ページ変換テーブルのインデックスとして使われる。ページ変換テーブルは、12ビットのポインターを持ち、これは直接物理メモリ内のページを指す。これはまた、ページフレームテーブルのインデックスとしても使われる。

このページフレームテーブルは、各ページごとの保護および統計情報、上位仮想ページ番号、PIDを持つ。これらの情報が合致していれば、メモリ参照が行なわれる。もし合致していなければ、メモリ参照はアボートされ、ページ番号の特定値に対応する検索が、ページフレームテーブルのすべてのエントリーに対

して実行される。この検索は、リンクを追跡して行なわれる。リンクは、ページフレームテーブルの有効エントリをチェーンしている、12ビットのポインターである。そしてMMUは自動的に要求ページを検索し、ページが見つければ処理を続行する。もし見つからなければ、CPU割込みをおこす。

#### 6.4 シェルプログラミング

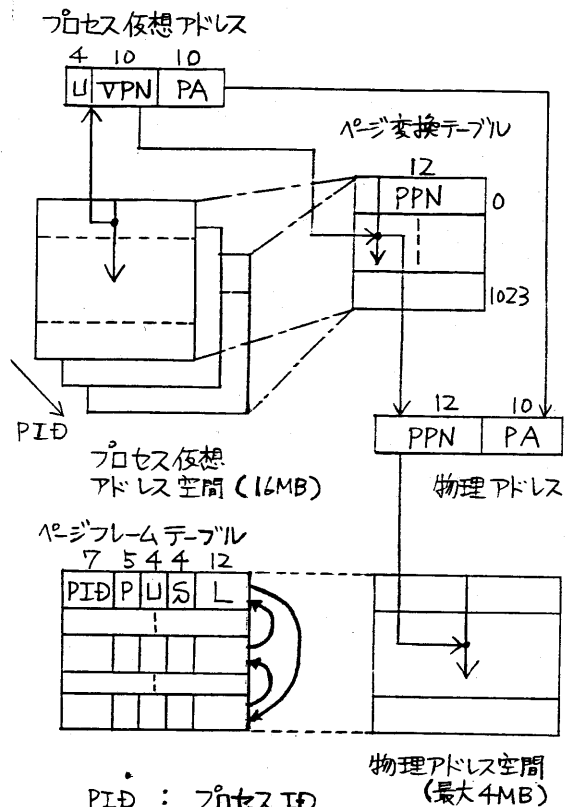
コンピュータの発展と相まって、大量の、基本機能を持つプログラムモジュール群が、ソフトウェアツールとして蓄積されてきた。アプリケーションは、これらのソフトウェアツールの集合を介して、データのストリームを相互結合することにより、容易に作成することができる。これがシェルプログラミングである。

シェルプログラムは、FORTRANとかPASCALのような従来のプログラム言語よりも、高位レベルのプログラム言語である。各シェルプログラムに該当するシェルコマンドを入力するだけで、これらを実行することができる。また、ストリームE/Oを用いて、プロセスからファイルへの処理、プロセスからプロセスへの処理を、容易に行なうことができる。さらにアポロでは、これらの処理を同一ノード内だけでなく、ネットワークを介しても、取扱うことができる。

現在アポロは、高級言語として、FORTRAN77とPASCALをサポートしている。(本年度中には、LISP、C、ADAが予定されている。) これら高級言語により作成されたプログラムは、すべてシェルコマンドとして、システムに登録されるので、実行可能なプログラムはシェルコマンドになる。

#### 7. ユーザー環境

ユーザー環境は、大きく2つの特長がある。第1に、システムが参照できるすべてのオブジェクトは、ネットワークを超越した、ツリー構造のディレクトリに集約されることである。第2に、高解像度のディスプレイを使って、同時処理が行なわれることである。



- PID : プロセスID
- P : 保護情報
- U : 上位仮想ページ番号
- S : 統計情報
- L : リンク

図5. メモリ管理機構

## 7.1 ディレクトリー

ディレクトリーは、図6に示すように、ネットワーク上にあるすべてのオブジェクトの位置を表わした、ユーザーグローバルネーム空間である。各々のオブジェクトは、その位置を示すパス名を入力することで参照できる。

ディレクトリーの頂点は、ネットワーク全体を集約する部分で、各ノードのディレクトリーの最上位にある。つまり、各ノードのディレクトリーは、ネットワーク全体のディレクトリーを構成するサブディレクトリーである。

これに対して、ディレクトリーの最下位は、プログラムやデータファイル、さらに周辺機器といった特定のオブジェクトを表わす。中間のディレクトリーは、ある共通の関連を持つオブジェクトの集合を表わすのに用いられる。

オブジェクトの位置を示すパス名はいくつかの型がある。まず2つのスラッシュで始まるパス名である。これは、あるローカルノードから、ネットワーク内の任意のオブジェクトを参照するのに用いられる。例えば、ノードEから、ノードAにあるオブジェクトDを参照する時は、//A/B/C/Dを指定する。

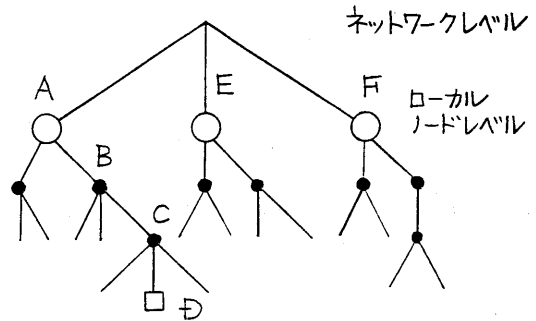
特定のユーザーノードで、そのノードにあるオブジェクトを参照するには、1つのスラッシュで始まるパス名を用いる。最初のスラッシュに続く名前は、そのノードにカタログされている、第1レベルのディレクトリーを指す。例えば、ノードAにおいて、オブジェクトDを参照する時は、/B/C/Dを指定する。

さらに、ユーザーは、ディレクトリーの任意の位置に、ユーザー自身の作業ディレクトリーを設定できる。この作業ディレクトリーの下にカタログされているオブジェクトを参照するには、スラッシュのないパス名を用いる。例えば、ノードAにおいて、今、作業ディレクトリーがBであるとすると、オブジェクトDを参照する時は、C/Dを指定する。

ただし、ネットワーク内のどのノードであっても、またその作業ディレクトリーがどこであっても、//A/B/C/Dは、ノードAにあるオブジェクトDを指す。また、ノードAにおいて、その作業ディレクトリーがどこであっても、/B/C/Dは、オブジェクトDを指す。

## 7.2 同時処理

同時処理の概念は、従来のタイムシェアリングシステムでは利用できない、アポロシステム独自の新しい概念である。タイムシェアリングでは、1台の端末からは、ある時間にひとつの処理しかできず、1つの処理から他の処理へ切替える



パス名

//A/B/C/D : ネットワークレベル

/B/C/D : ローカルノードレベル

C/D : 作業ディレクトリーレベル

図6. ディレクトリー構造



と、前の文脈は失われ、後で再生する必要があった。アポロシステムでは、ディスプレイマネージャによって、ユーザーは、画面を複数の短形エリアに分割して、それらを同時に表示できる。各々の窓に、関係ある、または無関係なアプリケーションの出力を表示することができる。

### 7.2.1 ディスプレイマネージャ

ディスプレイマネージャは、アポロシステムを構成する一番外側の層であり、CRT画面の表示内容を制御する。従ってアポロシステムは、従来のプログラミングレベルに2つの層を追加している。前に述べたように、シェルプログラミング環境が、多くのプログラムの処理(順次処理やパイプライン処理)を調整する。このシェルの出力は、パッドと呼ばれる仮想ターミナルに書かれる。そして、このパッドの内容は、ディスプレイマネージャにより、画面上にある窓を通して表示される。

ディスプレイマネージャを起動する方法は2つある。1つはディスプレイマネージャコマンドを入力する方法である。もう1つは、キーボード上の特殊キーを押す方法である。特殊キーは、ディスプレイマネージャコマンドで定義されており、これを押すと、一連のディスプレイマネージャの基本機能が実行される。ユーザーは、複雑なディスプレイマネージャ機能を実行するように、これらの特殊キーを再定義することができる。

### 7.2.2 ユーザー処理環境

ノードを立上げると、画面には、ディスプレイマネージャ用の窓と、ひとつのプロセス用の窓が表示される。ディスプレイマネージャの窓は、入力エリアと出力エリアがある。入力エリアに、ディスプレイマネージャコマンドを入力する。プロセスの窓は、タイトル、出力エリア、入力エリアの3つから成り、シェルコマンドは、入力エリアの\$マークのあとに入力する。そして、その出力内容は、出力エリアに表示されていく。

画面上には、最大15個の窓を、同時に表示できる。そして、その窓を重ね合わせることはもちろん、上下左右に移動したり、窓の大きさを変えたりすることができる。また出力エリアの内容を、上下左右にスクロールすることもできる。

画面上のひとつの窓が、ひとつのプロセスに対応する。そして、各々のプロセスでは、最大16MBまでのプログラムを実行することができる。

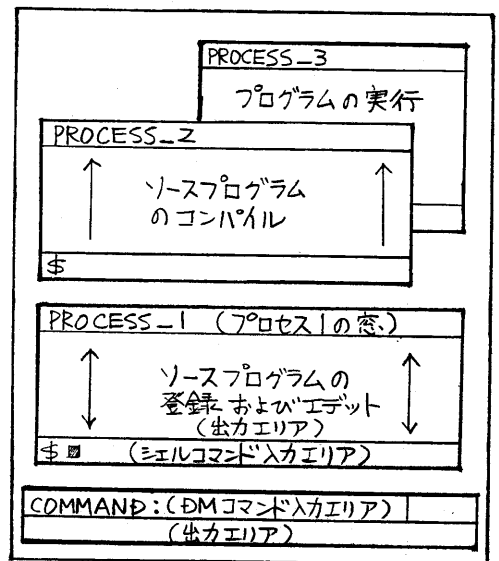


図7. 同時処理画面

## 8. 要約

アポロDOMAINシステムは、専用ミニコンピューターとタイムシェアリングの両者の長所である、対話機能と資源の共有機能を兼ね備えた、1980年代のシステムである。

アポロDOMAINシステムは、また、ローカルネットワークによって相互に接続された、多数の高性能専用コンピューターから構成される。ネットワークを構成する各ノードは、ネットワークワイドのデマンドページ仮想メモリーシステムを実現し、非常に大きな仮想アドレス空間を持つプロセスを、多重処理にて実行することができる。

システム上で実行できる言語は、FORTRAN77やPASCALがあり、32ビット処理を実現する。

ネットワーク上のすべての資源は、オブジェクトとして、ディレクトリーに管理され、適切なアクセス権が与えられていれば、ネットワーク内のどのユーザーからも、平等にアクセスすることができる。

ユーザーのディスプレイターミナルは、多重フォントテキストや図形を表示でき、また多数の窓に分割して、それぞれ独立したプログラムの出力を表示することができる。

## 9. 参考文献

「アポロDOMAIN情報処理システムアーキテクチャー」  
アポロ社 (日本ラッド(株), 丸紅ハイテック(株), 丸紅(株))

「APOLLO SYSTEM USER'S GUIDE」  
アポロ社 (日本ラッド(株), 丸紅ハイテック(株), 丸紅(株))

「アポロDOMAIN情報処理システム入門書」  
日本ラッド(株), 丸紅ハイテック(株), 丸紅(株)

「UNIXシステム入門」  
石田晴久著 bit 共立出版(現在, 連載中)

「ソフトウェア作法」  
木村 泉著 共立出版

「輸入され始めた専門家向き高性能パーソナルコンピューター」  
日経エレクトロニクス 1982.2.15号

「見え始めた分散処理 ソフトウェアの新しい波」  
日経コンピューター 1982.4.15号

「Apollo DOMAINシステムの概要」  
吉川徹男著 ビジネスコミュニケーション 1982.5月号