

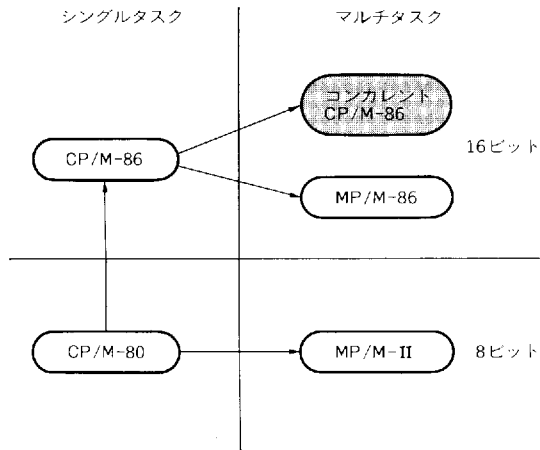
# コンカレント CP/M-86 の特徴と機能

多賀谷 哲也

8088/8086 用シングルユーザ・マルチタスク OS のコンカレント CP/M は、仮想コンソール、リアルタイム・プロセス、パスワードによるファイル保護、タイム・スタンプ機能などの特徴をもつ。ファイル・システムも高速化/大容量化(最大 128 M バイト/ファイル、512 M バイト/ドライブ)した。MP/M-86 への上位互換性もある。すでに IBM-PC 上で動いているが、本稿はマニュアルによる机上評価である。(編)

最近 16 ビットのパソコンが続々と発表されている。多くは CP/M-86 や MS-DOS を採用しているが、それらは高機能化してきたハードウェアを充分活用しているとはいえない。近ごろ発表されたコンカレント CP/M-86(Concurrent CP/M-86)は、その不満を解決するものとして注目を集めているが、それに関する情報は現在のところ非常に少ない。

〔図1〕 デジタル・リサーチ社の OS 群  
シングルタスク マルチタスク



## 1 コンカレント CP/M-86 とは…

コンカレント CP/M-86 はデジタル・リサーチ社の最新の 16 ビット・マイコン(8086/8088)用 OS で、CP/M-86 や MS-DOS がシングルユーザ・シングルタスクの OS であったのに比べ、シングルユーザ・マルチタスクの機能をもっている。コンカレントは“並行”と訳されることが多く、複数のプログラムの実行が時間的に重なり合うことをいう。さてコンカレント CP/M-86 の発表によりデジタル・リサーチ社は図 1 に示す OS のラインアップをもつことになった。

コンカレント CP/M-86 は MP/M-86 をシングルユーザ用に手直したものであると考えられるが、社長の Gary Kildall 氏はつぎのように位置づけている。

「コンカレント CP/M-86 は図 2 に示すように、他の OS の境界領域に位置しており、MP/M-86 と同じようにリアルタイム核をもっていてプロセス制御やインゲストリアル・コントロールにも応用が可能な OS である」

コンカレント CP/M-86 は、現在、IBM-PC 上で稼動しているが、その最小ハードウェア構成としては 2 台以上のディスク・ドライブ、256 K バイト以上の RAM が要求されている。IBM-PC は日本では未発表なので本稿は、机上でのマニュアル<sup>2,3)</sup>調査をもとにしている。

## 2 コンカレント CP/M-86 の特徴

さて、コンカレント CP/M-86 を大まかに理解するために、その代表的な機能をながめてみよう。

### (1) マルチタスキング

複数のプログラムを同時に走らせることができるの

で、16ビットCPUの採用により高性能になったハードウェアの能力を充分発揮させられる。たとえば、ソフトウェア開発を行うときに、プログラムをエディットしながら、同時に別のモジュールをコンパイルし、さらにプリンタへ出力することが可能になる。

### (2) 仮想コンソール

並行に走る各プログラムにそれぞれ仮想コンソールが割り当てられる。ある時点では仮想コンソールのどれか一つが実コンソールに接続されている。ユーザはキーボードからの入力により、実コンソールに接続される仮想コンソールをつぎつぎと切り替え、そのプログラムと対話したり、出力を見たりすることができる。これは、ちょうどテレビのチャンネルを回して番組を選ぶのと同じイメージである(図3参照)。

### (3) ファイルの共有

ファイルを複数のプロセスから同時にオープンして共有することができ、ファイル単位、レコード単位でのロック機構が用意されている。

### (4) リアルタイム・プロセス

プライオリティ・ドリブンでスケジューリングを行うリアルタイム核がシステムを管理しており、リアルタイム・アプリケーションやローカル・エリア・ネットワークにも対応できる。

### (5) パスワードによるファイル保護

ファイルには8文字以内のパスワードを割り当てられ、パスワードを知らないユーザからのアクセスを、読出し、書込み、削除の各レベルで制限することができる。また、パスワード・プロテクトの制限を受けずに任意のファイルにアクセスができるスーパー・ユーザ的な存在を許す機構もある。

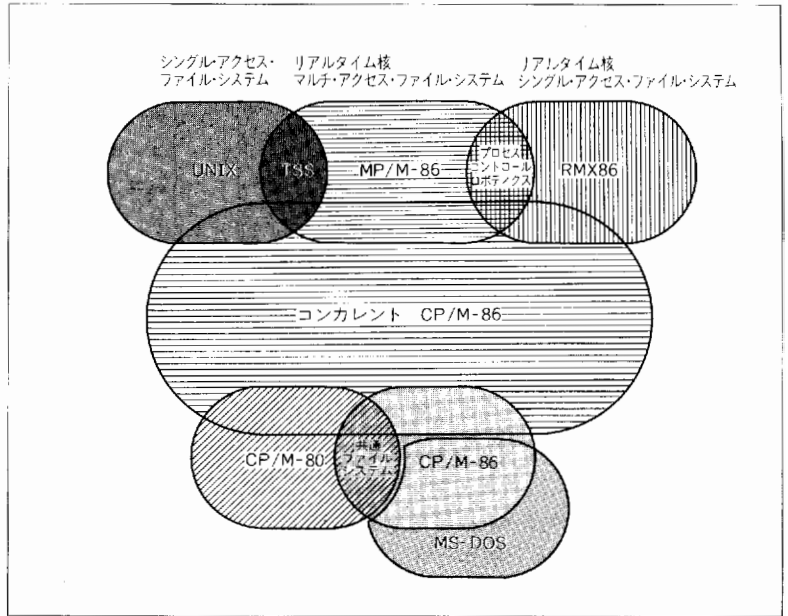
### (6) ファイルの日時スタンプ

リアルタイム・クロックを利用して、ファイル・ディレクトリにファイルの作成/最終アクセス、最終更新の日付を記録することができ、ファイルの管理に役立つ。

### (7) 大容量ファイル・システム

1ファイルあたりの容量とドライブあたりの最大容量が、CP/M-86の8Mバイト、128Mバイトから32Mバイト、512Mバイトにそれぞれ増えて、システムがサポートできるディスクの最大容量は8Gバイトにも

【図2】コンカレントCP/M-86の位置<sup>1)</sup>



達するようになった。

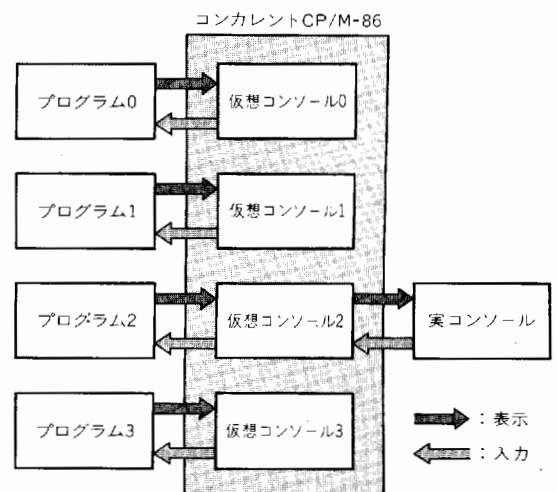
### (8) 高速ファイル・アクセス

RAM上にディスク・イメージをもたせる仮想ドライブ(Mディスク)機能やBDOSのマルチセクタI/Oファンクションなどの採用により、CP/Mの弱点の一つであったファイル・アクセスの遅さが克服されている。

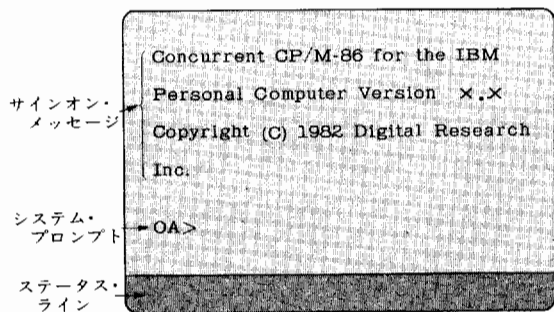
### (9) MP/M-86 への上位互換性

MP/M-86とファンクション・コール体系、コマンド体系、ファイル形式はほぼ同一であるので、コンカレ

【図3】仮想コンソールの概念



[図4] システム起動時のコンソール画面



ント CP/M-86 の上で開発されたソフトウェアは容易に MP/M-86 上へ移植できる。

### 3 コンカレント CP/M-86 の概要

この章では、おもに CP/M-86 との違いを中心にしてコンカレント CP/M-86 の概要について述べる。

#### 3.1 システムの起動

マニュアルによれば、「システムのブート・アップのためにはブート・ディスクとシステム・ディスクの2枚のフロッピー・ディスクが必要」となっている。それぞれをドライブ A, ドライブ B に挿入してパワーオンすると、図4のサインオン・メッセージとシステム・プロンプトがコンソール上に表示される。プロンプト

[図5] コンカレント CP/M-86 の FCB フォーマット

dr	f1	f2	...	f8	t1	t2	t3	ex	cs	rs	rc	d0	...	dn	cr	r0	r1	r2
00	01	02	...	08	09	10	11	12	13	14	15	16	...	31	32	33	34	35
dr	ドライブ・コード(0~16)																	
f1...f8	ファイル名																	
	f1'	MP/M 1.1 Default Open																
	f2'	Partial Close Default																
	f3'	Ignore Close Checksum Errors																
	f4'	Disable FCB Checksum Verification																
	f5'																	
	f6'																	
	f7'																	
	f8'																	
t1...t3	ファイル・タイプ																	
	t1'	Read Only (1)/Read Write(0)																
	t2'	SYS(1)/DIR(0)																
	t3'	ARCHIVE(1)																
ex	カレント・エクステント番号																	
cs	FCBチェックサム																	
rs	reserved																	
rc	エクステント内レコード数																	
d0...dn	reserved																	
cr	カレント・レコード番号																	
r0,r1,r2	ランダム・レコード番号																	

(注) X'はXのバイトの最高位ビット(ビット7)を表す

の1文字目はユーザ番号を示し、2文字目はデフォルト・ドライブ番号を示している。ユーザ番号はユーザを識別するためのもので0~15の値をとる。システム・リセット時に0となっているが、USER コマンドで変更することができる。ファイルの生成時にカレント・ユーザ番号がそのファイルの属性として与えられる。通常、あるユーザは他のユーザ番号のファイルにアクセスすることはできない。ユーザ番号の概念は CP/M Ver.2 以降のものであるが、CP/M(-86)ではあまり表面には出てこなかった。しかし、OS の性格上 MP/M-86 やコンカレント CP/M-86 ではシステムの表面にはっきりと出てきている。

デフォルト・ドライブの切替えは CP/M と同じように、ドライブ名:で行う。

コンソールの最下行には、システムのステータス・メッセージ、カレント・ユーザ番号、現在日時が表示される(ステータス・ライン)。

#### 3.2 仮想コンソール

ユーザからみてコンカレント CP/M-86 が従来の OS ともっとも違っているのは仮想コンソールの概念である。仮想コンソールが実コンソールに割り当てられているプロセスをフォアグラウンド・モードと呼び、そうでないものをバックグラウンド・モードと呼ぶ。

仮想コンソールは、物理モード、ダイナミック・モード、バッファ・モードのいずれかの状態にある。物理モードは、仮想コンソールが実コンソールに接続された状態のことで、仮想コンソール上に出力された文字は、直接、実コンソール上に表示される。ある時点では仮想コンソールのどれかは物理モードになっている。

ダイナミック・モードは RAM 上の仮想スクリーン・バッファを仮想画面としてとらえ、その上で画面を更新していく。もちろん、カーソル・コントロール文字やスクロールなどもサポートされる。ダイナミック・モードにあった仮想コンソールが選択される(フォアグラウンドになる)と、そのときのバッファの内容が実コンソールに表示される。

バッファ・モードでは、仮想コンソールに出力された文字は、すべてディスク上のファイルに記録される。選択されたときには、ファイルから文字が読み出されてきて、コンソール画面に順々に表示される。すなわち、このモードではダイナミック・モードの場合とはちがって、バックグラウンド・モードにあった間に発生した出力をその経過どおりに見ることができるのである。

コンソール入力に対しては、コンカレント CP/M-86 は各仮想コンソールに対して先読み(type ahead)パッ

[表1] インターフェース属性

ファンクション名	属性	機能
OPEN FILE	f5' = 1 f6' = 1	Open in Unlocked Mode Open in Read Only Mode
MAKE FILE	f5' = 1 f6' = 1	Open in Unlocked Mode Assign password to file
CLOSE FILE	f5' = 1	Partial Close
DELETE FILE	f6' = 1	Delete file XFCBs only

ファをもっており、バックグラウンドのプログラムはそこから先読みされた文字を取り出して実行できる。バッファにデータがなくなった時点で、コンソールにメッセージを出し、オペレータに知らせる。

### 3.3 ファイル管理

システムが高機能化し、複数のプロセスが同時に走るようになれば、ファイルの共有化、ファイルの保護などが重要なポイントになる。この点で、コンカレント CP/M-86 のファイル管理は CP/M のそれと比べるとずいぶん複雑になっている。

#### ● FCB

図5にコンカレント CP/M-86 の FCB フォーマットを示す。CP/M-86 のそれとほとんど同じであるが、属性ビットの構成だけが異なる。

##### (1) ファイル属性

- ① DIR/SYS 属性…通常のファイルは DIR 属性であるが、SYS 属性にするとシステム・ファイルになる。
- ② Read Write/Read Only 属性…ファイルごとに読出し、書込み共に可能 (Read Write) か、読出しのみ可能 (Read Only) か指定できる。

[図6] XFCB フォーマット

dr	file	type	pm	s1	s2	rc	password	ts1	ts2			
00	01	...	09	...	12	13	14	15	16	...	25	29

dr            ドライブ・コード(0~16)  
file          ファイル名  
type         ファイル・タイプ  
pm          パスワード・モード  
             ビット7:Readモード  
             ビット6:Writeモード  
             ビット5:Deleteモード  
s1,s2,rc     reserved  
password     パスワード(暗号化)  
ts1          ファイル生成/最終アクセス日時スタンプ  
ts2          最終更新日時スタンプ

[表2] パスワード・プロテクト・レベル

パスワード・プロテクト・モード	パスワードを与えない場合のアクセス許可		
	read	modify	delete
Read	×	×	×
Write	○	×	×
Delete	○	○	×

③ ARCHIVE(アーカイブ)属性…ファイルが変更されていないことを示す属性である。BDOS の WRITE が実行されると自動的にこの属性はリセットされる。PIP の [A] オプションではこの属性がオフのファイルのみコピーされる。

##### (2) コンパチビリティ属性

コンカレント CP/M-86 のファイル・システムに加えられた新しい制限により、CP/M や MP/M で開発されたプログラムがそのままでは走らない可能性がある。この属性は、その制限を状況に応じて取り払うためのものであるが、詳しくは省略する。

##### (3) インターフェース属性

この属性は表1に示す BDOS コールに対してのみ意味をもつ。

#### ● XFCB

ディレクトリの拡張として、XFCB というエントリをディレクトリ領域にもつことができる。XFCB はファイルのパスワード・プロテクトと日時スタンプを管理する(図6)。XFCB はファイル生成とは別に作る (WRITE FILE XFCB) こともできるが、ディレクトリ・ラベルの制御でファイル生成時に自動的につくる (MAKE FILE) こともできる。

[図7] パスワード・プロテクトの例

```

OA > SET MYFILE.TXT[PASSWORD = XYZ]
OA > TYPE MYFILE.TXT; XYZ
      This is MYFILE.
      END OF TEXT
OA > TYPE MYFILE.TXT
      Bdos Err On A: Password Error BDOS Function
      15 FILE; MYFILE.TXT
OA > ERA MYFILE.TXT
      Not Erased; MYFILE.TXT Password Error
      Password ?
      (ここでパスワード"XYZ"を入力すればERAは実行される。ただし入力
       はエコーされない。)
OA >
    
```

〔図8〕 デフォルト・パスワードの使用例

```
OA> PIP; XYZ SAMPLE.TXT; XYZ=TEST.DOC; XYZ
(a)
```

```
OA> SET [DEFAULT=XYZ]
OA> PIP SAMPLE.TXT=TEST.DOC
(b)
```

#### (4) パスワード

コンカレント CP/M-86 では、パスワードでファイルの保護をおこなっている。パスワードはファイルごとに SET コマンドで指定できる。パスワード・プロテクトされたファイルにアクセスする場合は、セミコロンのあとにパスワードをつけてファイル名指定をおこなわなければならない。違反した場合はエラーの表示後に状況に応じてパスワードの入力を求めてくる。この様子を図7に示す。

またパスワード・プロテクトは表2に示すように三つのレベルがあり、SET コマンドで指定できる。ファイルが同じパスワードをもつ場合は、デフォルト・パスワードを指定すれば、ファイル名指定にいちいちパスワードをつけなくともアクセス可能になって便利である(図8の(a)と(b)は意味的に同じ)。

#### (5) 日時スタンプ

XFCB 内にファイルの生成、最終アクセス、最終更新の日時を記録できる。このうち生成と最終アクセスは同じフィールドを使うので片方しか選択できない。

### ● ディレクトリ・ラベル

ディレクトリ領域には XFCB 以外にもう一つ、ディ

〔図9〕 ディレクトリ・ラベル・フォーマット

dr	name	type	dl	s1	s2	rc	password	ts1	ts2		
00	01	...	09	12	13	14	15	16	...	25	29

dr           ドドライブ・コード(0~16)  
name         ディレクトリ・ラベル名  
type         ディレクトリ・ラベル・タイプ  
dl           ディレクトリ・ラベル・データ・バイト  
              ビット7: ファイル・パスワード・プロテクト・オン  
              ビット6: 最終アクセス日時スタンプ・オン  
              ビット5: 最終更新日時スタンプ・オン  
              ビット4: XFCB自動生成  
              ビット0: ディレクトリ・ラベルあり  
s1,s2,rc     未使用  
password     ディレクトリ・ラベル・パスワード(暗号化)  
ts1          ディレクトリ・ラベル生成日時スタンプ  
ts2          ディレクトリ・ラベル更新日時スタンプ

レクトリ・ラベルという特別なエントリをつくることのできる(図9)。ディレクトリ・ラベルはディスクにただか一つしかなく、ディスクの識別子としての名前が与えられる。また、ディスク内のファイルに対するパスワード・プロテクトのオン/オフ、日時スタンプのオン/オフ、XFCBの自動生成などをコントロールする。ディレクトリ・ラベル自身にパスワードを与えると、それへのアクセスはすべてパスワード・プロテクトされる。また、ディレクトリ・ラベル内にファイルのパスワード・プロテクトをオフするビット(dlビット7)をもっているため、ディレクトリ・ラベルのパスワードは“スーパー・パスワード”的な存在である。

### ● ファイル・オープン・モード

ファイルの共有を制御するために、三つのファイル・オープン・モードが用意されている。これらはFCBのインターフェース属性で選択される。

#### (1) ロック・モード

ロック・モードでオープンしたファイルは、他のプロセスからオープンすることができなくなり、クローズするまでそのプロセスに専有される。ロック・モードがデフォルトのオープン・モードである。

#### (2) アンロック・モード

アンロック・モードでオープンされたファイルは、他のプロセスからもアンロック・モードでオープンすることができ、ファイルはプロセス間で共有される。共有しているプロセス間では、BDOSのRECORD LOCK, UNLOCK や TEST AND WRITE RECORD ファンクションを使って、アクセス競合を避けながらレコード更新処理ができる。

#### (3) リード・オンリ・モード

このモードでは、一つのファイルを書込み禁止で多数のプロセスが同時にオープンできる。

### 3.4 コンカレント CP/M-86 のコマンド

コンカレント CP/M-86 のコマンドには、RSP、ビルトイン・コマンド、ランジェント・コマンドの3種類がある(図10)。CP/Mとはちがってほとんどがランジェント・コマンドである。

#### (1) RSP(Resident System Process)

RSPはシステム・ジェネレーション時に、ユーザの指定によりOSの一部となったコマンドで、メモリに常駐する。

#### (2) ビルトイン・コマンド

- ① USER コマンド…カレント・ユーザ番号の表示および変更をおこなう。
- ② PRINTER コマンド…カレント・プリンタ番号の表示および変更をおこなう。

### (3) トランジェント・コマンド

トランジェント・コマンドは、ファイル・タイプに CMD をもつファイルで、ディスク内に存在し、呼ばれたときにディスクからメモリへロードして実行される。コマンド・ファイルをディスク上でサーチするときにはつぎの順にさがす。

- ① デフォルトまたは指定ドライブのカレント・ユーザ番号の CMD ファイルをさがす。
- ② そのファイルがなければ、システム・ドライブのカレント・ユーザ番号の CMD ファイルをさがす。
- ③ このファイルがなければ、デフォルトまたは指定ドライブのユーザ番号 0 の CMD ファイルをさがす。
- ④ このファイルもなければ、システム・ドライブのユーザ番号 0 の CMD ファイルをさがす。
- ⑤ それでもなければ、つぎのエラー・メッセージを出す。

ファイル名：? Can't Find Command

ただし②③④の場合は SYS 属性をもったファイルでなければならない。逆に、(システム・ドライブの) ユーザ番号 0 の SYS 属性をもつファイルはどのユー

[図11] ABORT コマンドの実行例

0A > TYPE DOCUMENT.TXT — 仮想コンソール1で実行

Dear Sir:

The company is pleased to inform you tha

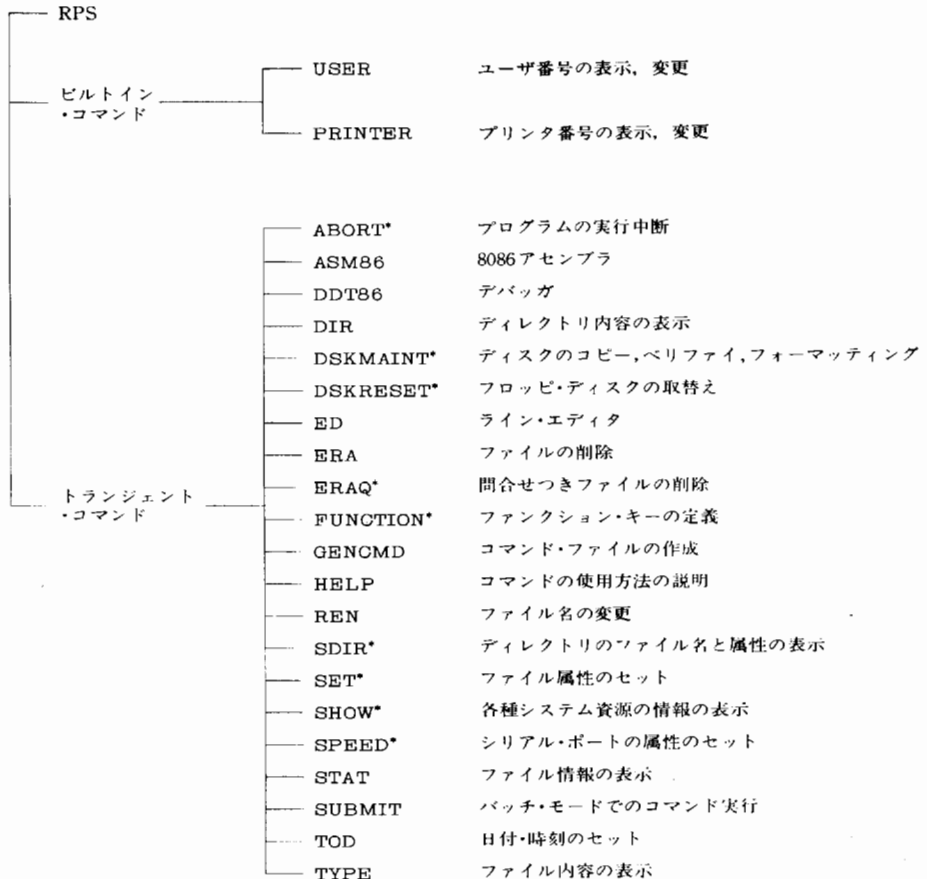
5B > ABORT TYPE 1 <cr> — 仮想コンソール3で実行  
5B >

ザ番号からも実行することができるので、共通的に使われるプログラムをそこに格納しておけば、各ユーザごとに重複してファイルをもたなくてすむようになり、ファイル・スペースが節約できる。

コンカレント CP/M-86 のトランジェント・コマンドの多くは、CP/M-86 のそれとまったく同じかほぼ同じものである。ここでは、コンカレント CP/M-86 に特有のもの(図10の\*印)だけを解説する。

#### (1) ABORT コマンド

[図10] コンカレント CP/M-86 のコマンド



(注) \*印はコンカレント CP/M-86 特有のもの

[図12] DSKRESET コマンドの使用例

```
OA > DSKRESET A:
OA > DSKRESET B:

Disk reset denied, Drive B: Console2 Program PIP
```

ドライブBの交換要求が出されたが、仮想コンソール2のPIPがドライブB上でファイルをオープンしているので要求は拒否された

ABORTは指定したプログラムの実行を中断させる。このとき、コンソール番号を指定して、別の仮想コンソールで走っているプログラムを止めることができる。ただし、止めたいプログラムが同じ仮想コンソールで走っているときは、実コンソールとの接続を先に断つてからABORTしなければならない。実コンソール上のプログラムはCTRL-Cで止めることもできる。

図11にABORTの実行例を示す。

(2) DSKMAINT コマンド

DSKMAINTはファイルのコピー、ベリファイ、フロッピ・ディスクのフォーマットなどをおこなうコマンドらしいが、マニュアル中には詳しい記述はない。

(3) DSKRESET コマンド

DSKRESETはフロッピ・ディスクの交換をおこなうときに使うコマンドである。マルチタスクで走っているときは複数のプロセスが一つのディスク上でファイルをオープンしている可能性がある。したがって、あるプロセスでフロッピ・ディスクの交換が必要になっても、オペレータは他のプロセスとの関係ですぐに交換してよいかどうか分からない。そんなときにDSKRESETコマンドを使う。このコマンドを出したときに他のプロセスがそのディスクを使っていなければ、

[図14] FUNCTION コマンドの使用例

```
OA > FUNCTION <cr>
FUNCTION utility, v 1.0 10/9/81
F1: dir <CR>
F2: dir b: <CR>
F3: stat <CR>
F4: stat b: <CR>
F5: pip <CR>
F6: pip b: = a: *.* [v]
F7: stat *.* <CR>
F8: stat b: *.* <CR>
F9:
F10:

Home: <ESC> H
| : <ESC> A
PgUp:
<--- : <ESC> D
--->:

End: END
| : <ESC> B
PgDn:
Ins :
Del : <DEL>

Function key?: F7
--> STAT *.* CTRL-G <ENTER>
Function key?: CTRL-C
OA >
```

ファイル・システムはリセットされ、交換OKとなる。さもなければ図12のようなエラーが表示され、ディスク交換要求は拒否される。DSKRESETをおこなわずにディスクを交換すれば、ドライブは自動的にRead Onlyになる。

(4) ERAQ コマンド

ERAQはファイルを削除するコマンドで、機能的にはERAコマンドと同じだが、各ファイルを削除する前に図13に示すように削除してよいかどうかをたず

[図13]

ERAQ コマンドの使用例

```
1A > ERAQ C: *.*CMD

C: ASM86 CMD ?y
C: FUNCTION CMD ?n
C: SDIR CMD ?y
C: DSKRESET CMD ?y
1A > DIR C: *.*CMD
C: FUNCTION CMD
1A >
```

[図15] SDIR コマンド・フル・フォーマット表示例

```
OB > SDIR

Directory For Drive B: User O
```

Name	Bytes	Recs	Attributes	Prot	Update	Create
GENLED DAT	109K	873	Dir RW	None	08/05/81 14:01	08/01/81 09:36
RECEIPTS DAT	59K	475	Dir RW	None	08/08/81 12:11	08/01/81 09:40

```
Total Bytes = 168K Total Records = 1348 Files Found = 2
Total 1K Blocks = 168 Used/Max Dir Entries For Drive M: 2/64
```

〔表3〕SDIR コマンドのオプション

SYS	SYS 属性のファイルのみ表示
RO	RO 属性のファイルのみ表示
DIR	DIR 属性のファイルのみ表示
RW	R/W 属性のファイルのみ表示
XFCB	XFCB をもつファイルのみ表示
NONXFCB	XFCB をもたないファイルのみ表示
USER = n	ユーザ番号 n のファイルを表示
USER = ALL	すべてのユーザ番号のファイルを表示
USER = (0, 1, ..., 15)	指定したユーザ番号のファイルを表示
DRIVE = d	ドライブ番号 d のファイルを表示
DRIVE = ALL	すべてのドライブのファイルを表示
DRIVE = (A, B, ..., P)	指定したドライブのファイルを表示
FULL	フル・フォーマットで表示
LENGTH = n	1 ページに n 行のディレクトリを表示
SIZE	サイズ・フォーマットで表示
FF	ハード・コピー用に改変キャラクタを出力
MESSAGE	ドライブ番号、ユーザ番号をヘッダに表示
NSORT	ディレクトリをアルファベット順に表示しない
EXCLUDE	指定したファイル名以外へファイルを表示
HELP	ヘルプ機能
SHORT	ショート・フォーマットで表示

ねてくる。

(5) FUNCTION コマンド

FUNCTION はファンクション・キーの機能を定義するコマンドである。あまり重要ではないので説明

〔図16〕

SET コマンド・ディレクトリ・ラベル・パスワード設定の実行例

```

OA > SET (PASSWORD = SECRET)

Label for drive A:

Directory      Passwds      Make      Stamp      Stamp      Stamp
Label          Req'd       XFCBs     Create     Access     Update
-----
A: Label              off        off         off         off         off
Password = SECRET
OA > SET [PROTECT = ON]

Password ? (パスワード"SECRET"を入力する。ただしエコーされない)
Label for drive A:

Directory      Passwds      Make      Stamp      Stamp      Stamp
Label          Req'd       XFCBs     Create     Access     Update
-----
A: Label              on         off         off         off         off
OA >
    
```

〔表4〕

SET コマンドのパスワード・セット・オプション

SET [PASSWORD=SECRET]	ディレクトリ・ラベルのパスワードを"SECRET"にセット
SET [PASS=<CR>]	ディレクトリ・ラベルのパスワード・プロテクトを解除
SET [PROTECT=ON]	ファイル・レベルのパスワード・プロテクトをオン
SET [PROTECT=OFF]	ファイル・レベルのパスワード・プロテクトをオフ
SET file[PASSWORD=XYZ]	ファイルのパスワードを"XYZ"にセット
SET file[PROTECT=READ]	ファイルのパスワード・プロテクト・レベルを READ にセット
SET file[PROTECT=WRITE]	ファイルのパスワード・プロテクト・レベルを WRITE にセット
SET file[PROTECT=DELETE]	ファイルのパスワード・プロテクト・レベルを DELETE にセット
SET file[PROTECT=NONE]	ファイルのパスワード・プロテクトを解除
SET [DEFAULT=XYZ]	デフォルト・パスワードを"XYZ"にセット

〔表5〕SET コマンドの日時スタンプ・セット・オプション

SET [CREATE=ON]	ファイル生成日時スタンプをオン
SET [ACCESS=ON]	ファイル・アクセス日時スタンプをオン
SET [UPDATE=ON]	ファイル更新日時スタンプをオン
SET file[TIME]	日時スタンプ・オン以前に作られたファイルに対する日時スタンプをオン

は省略する。図14にその使用例を示す。

(6) SDIR コマンド

SDIR は DIR コマンドと STAT コマンドをあわせたような強力な機能をもつコマンドである。画面への出力形式には、フル・フォーマット、サイズ・フォーマット、ショート・フォーマットの3種類がある。フル・フォーマットでは図15に示すように、ファイルの属性、パスワード・プロテクトの有無、日時スタンプ情報などが表示される。また、最後の2行に表示したファイルの数、合計容量などの総括情報が表示されて便利である。サイズ・フォーマットは、ファイル名とファイル長(K バイト単位)を表示する。ショート・フォーマットの表示は DIR コマンドのそれと同一である。

SDIR コマンドには表3に示すような豊富なオプション



〔図17〕

SET コマンドの日時スタ  
ンプ・セット実行例

```

OB > SET (CREATE = ON)
.
.
.
ファイル ONE.TXT, TWO.TXT, THREE.TXT の生成
.
.
.
OB > SDIR

Directory For Drive B:

      Name      Bytes Recs  Attributes Prot   Update      Create
-----
ONE   .TEX    9K    71 Dir RW   None      08/03/81  10:56
THREE .TEX   12K   95 Dir RW   None      08/03/81  10:57
TWO   .TEX   10K   76 Dir RW   None      08/03/81  10:56
    
```

ョンが用意されており、意味的に矛盾しないものなら同時に指定可能である。

(7) SET コマンド

SET は、

- ファイル/ディレクトリ・ラベルのパスワードのセット
- ファイルの日時スタンプのセット
- ファイル/ドライブ属性のセット

などの機能を持ち、ユーザがコンカレント CP/M-86 のファイル・システムを利用する上で欠くことのできないコマンドである。

▷パスワード

表4 にパスワードに関連するオプションの一覧を示す。ファイル・レベルのパスワード・プロテクトは、ディレクトリ・ラベルにアクセスして、その属性を変更することにより簡単に破れるので、ファイルのプロテクトをおこなうときには、ディレクトリ・ラベルをパスワード・プロテクトしておくことが望ましい。図16 にディレクトリ・ラベルのパスワード設定の実行例を示す。

▷日時スタンプ

ファイルの日時スタンプの種類およびそのオン/オフ

〔表8〕 SPEED コマンドに指定可能なパラメータ▶

9600	} ボーレート
4800	
1200	
600	
300	
150	
110	} パリティ
PARITY-NONE	
PARITY-EVEN	
PARITY-ODD	} ストップ・ビット数
STOP-1	
STOP-2	} データ・ビット数
BITS-7	
BITS-8	

〔表6〕 SET コマンドのファイル/ドライブ属性セット・オプション

SET [MAKE=ON]	XFCB の自動生成をオン
SET [MAKE=OFF]	XFCB の自動生成をオフ
SET d: [RO]	ドライブを Read Only にセット
SET d: [RW]	ドライブを Read Write にセット
SET file[RO]	ファイルを RO 属性にセット
SET file[RW]	ファイルを RW 属性にセット
SET file[SYS]	ファイルを SYS 属性にセット
SET file[DIR]	ファイルを DIR 属性にセット
SET file[ARCHIVE=ON]	ファイルの ARCHIVE 属性をオン
SET file[ARCHIVE=OFF]	ファイルの ARCHIVE 属性をオフ
SET [NAME=×××]	ディレクトリ・ラベルの名前を"×××"にセット
SET [HELP]	ヘルプ機能

〔表7〕 SHOW コマンド・オプション

SHOW SPACE	ドライブ番号, RO/RW, 残りスペースの表示(CP/M の STAT と同じ)
SHOW DRIVES	ドライブの属性の表示(CP/M の STAT DSK : と同じ)
SHOW LABEL	ディレクトリ・ラベルの表示
SHOW USERS	カレント・ユーザ番号とデフォルト・ドライブにファイルをもつユーザ番号の表示
SHOW HELP	ヘルプ機能

〔図18〕  
SHOW コマンド実行例

```

1A>SHOW USERS

Active User: 1
Active Files: 0 2 3 4

(a) USERS オプション

OB>SHOW LABEL

Label for drive B:

Directory  Passwds  Make  Stamp  Stamp Label Created  Label Updated
Label      Reqd      XFCS  Create  Update
-----
TOMSDisk  on        on    on      on     07/04/81 10:30 07/08/81 09:30

(b) LABEL オプション
    
```

はドライブ単位(ディレクトリ・ラベル)に指定する。  
表5に日時スタンプ関係のオプション一覧を示す。図  
17にファイル生成の日時スタンプをオンにしたときの  
実行例を示す。  
▷ファイル/ドライブ属性など

ファイルやドライブの属性設定関係のオプションは  
表6に示すようなものがある。

(8) SHOW コマンド

SHOW は CP/M の STAT コマンドに似た機能をもつ  
コマンドである。表7にオプションの一覧を示し、

〔図19〕  
コンカレント CP/M-86  
の構成

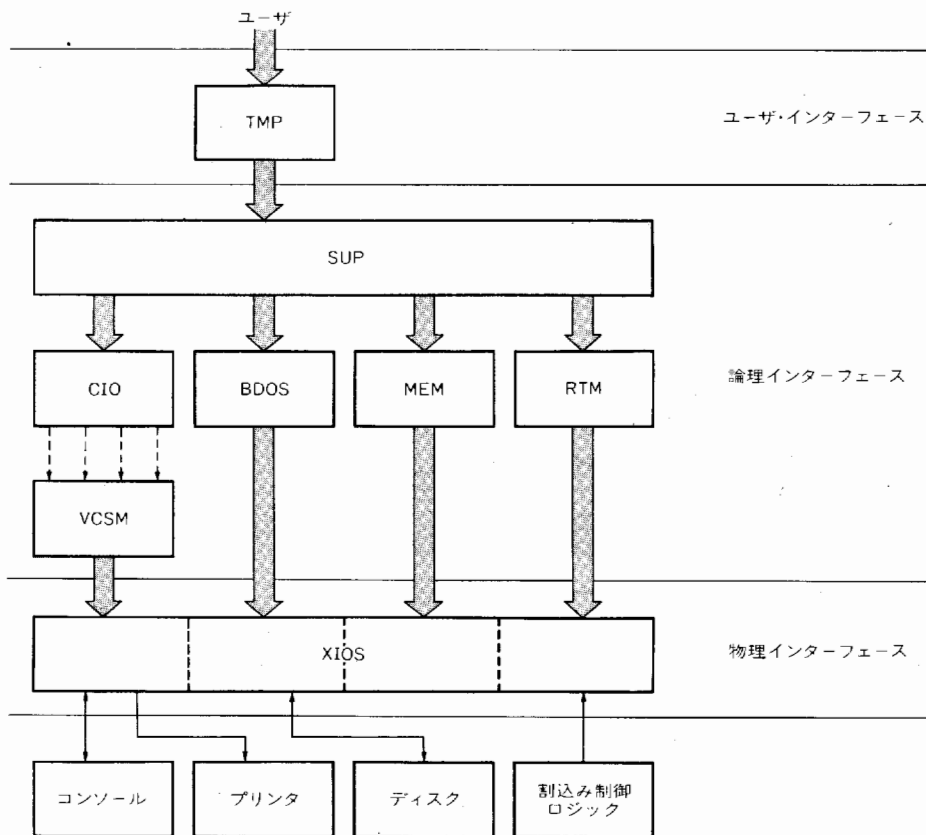


図18に USER, LABEL オプションの実行例を示す。

#### (9) SPEED コマンド

SPEED はシリアル・ポートをイニシャル・セットするためのコマンドで、表 8 に示すパラメータが指定可能である。

## 4 コンカレント CP/M-86 の実際

この章ではコンカレント CP/M-86 の内部について説明する。

### 4.1 コンカレント CP/M-86 のインターフェース

コンカレント CP/M-86 には図19に示すように 3 レベルのインターフェース・モジュールがある。

#### (1) ユーザ・インターフェース

標準的なユーザ・インターフェースは Terminal Message Process (TMP) と呼ばれる CP/M の CCP に相当するモジュールが提供する。TMP は通常 RSP で、システムに組み込まれている。TMP はユーザからのコマンドを受け付け、トランジェント・プロセスを起動したり、レジデント・プロセスにメッセージを送る役割をする。TMP は別のユーザ・インターフェースを実現するために置き替えることが可能で、たとえば UNIX 流のインターフェースにするために shell-like なもの)にのせかえてもよい。

#### (2) 論理インターフェース

コンカレント CP/M-86 の中のハードウェアから独立した部分である。ユーザ・インターフェースと物理インターフェースの中間に位置して、ファンクション・コールの形で OS の機能を実行中のプロセスに提供する。この部分はつぎに示す六つのモジュールから構成される。

- ① SUP(Supervisor)
- ② MEM(Memory Management Module)
- ③ CIO(Character I/O Module)
- ④ BDOS(Basic Disk Operating System)
- ⑤ RTM(Real Time Monitor)
- ⑥ VCSM(Virtual Console Session Manager)

#### (3) 物理インターフェース

〔表 9〕 プロセスの状態

状態名	プロセスの状態
running	CPU がプロセスに割り当てられている。
ready	プロセスはいつでも走れる状態にあり CPU の割当てを待っている。
blocked	I/O 動作の終了などのイベントを待っており CPU が割り当てられても走れない。

CP/M の BIOS にあたる部分で XIOS(Extended I/O System)と呼ばれる。XIOS を変更することによりコンカレント CP/M-86 を別の機種に移植することが可能である。

### 4.2 SUP(Supervisor)

SUP はすべてのファンクション・コールを管理するモジュールである。トランジェント・プロセスやシステム・モジュール内から呼ぶファンクション・コールは、いったん SUP を経由して、必要に応じていくつかに分解して下のモジュールにわたされる。

### 4.3 MEM(Memory Management Module)

MEM は、たとえばメモリの割当て状態を調べたり、メモリの割当て、解放などのメモリ管理に関するすべての機能をつかさどるモジュールである。

### 4.4 CIO(Character I/O Module)

CIO はコンソールやリスト・デバイスへの単純な文字入力をおこなうモジュールである。

### 4.5 BDOS(Basic Disk Operating System)

BDOS はファイル・システムを管理する部分で、CP/M-86 の BDOS と互換性をもっているが、複数のコンソールをサポートするための機能が追加されている。

### 4.6 RTM(Real Time Monitor)

RTM はコンカレント CP/M-86 の中心となるリアルタイム・マルチタスキング核である。RTM はスケジューリング、ディスパッチング、キュー管理、システム・タイミング管理、などをおこなう。

▷ ディスパッチング、スケジューリング

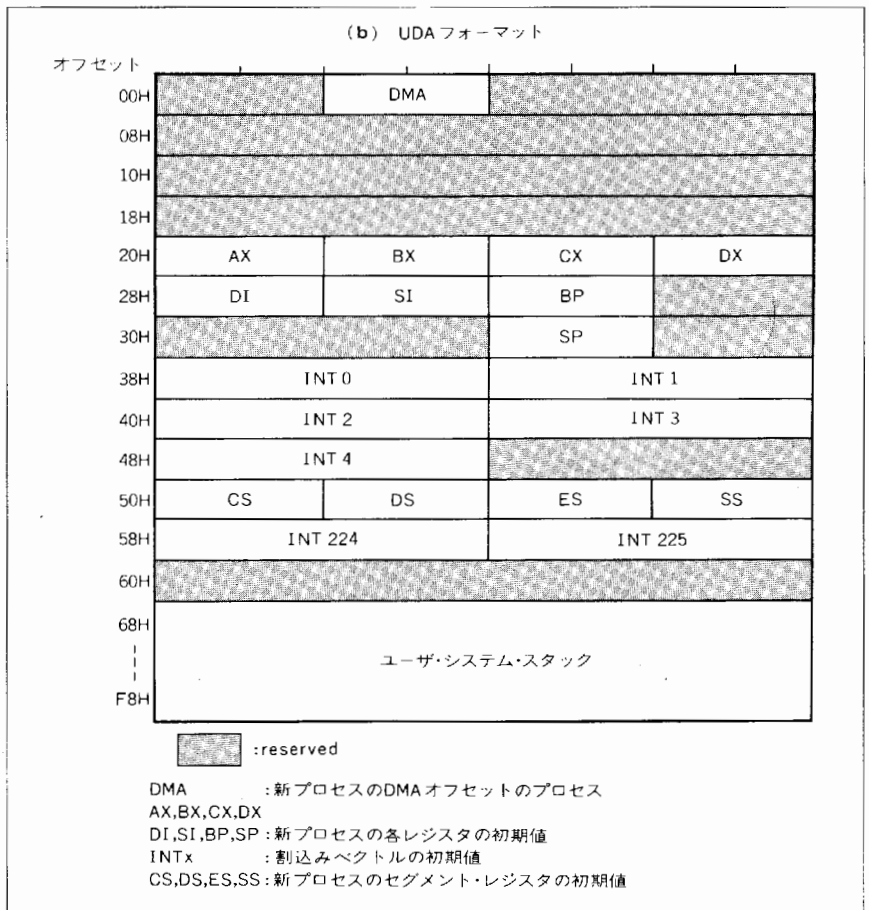
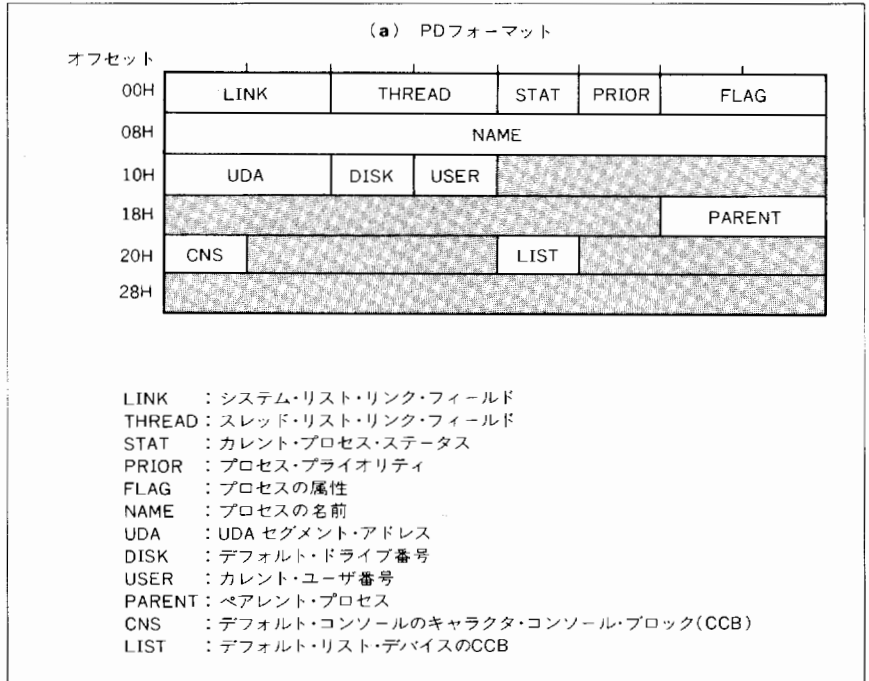
システム上のプロセスは表 9 に示すように running, ready, blocked のいずれかの状態にある。各プロセスにはプライオリティやステータスを保持するための、

〔表10〕 プロセス・プライオリティ

プライオリティ	プロセス
0~31	ハードウェア割込みハンドラ
32~63	システム・プロセス
64~149	未定義
150	初期化プロセス
151~197	未定義
198	TMP
199	未定義
200	ユーザ・プロセス(デフォルト・プライオリティ)
201~254	ユーザ・プロセス
255	IDLE プロセス

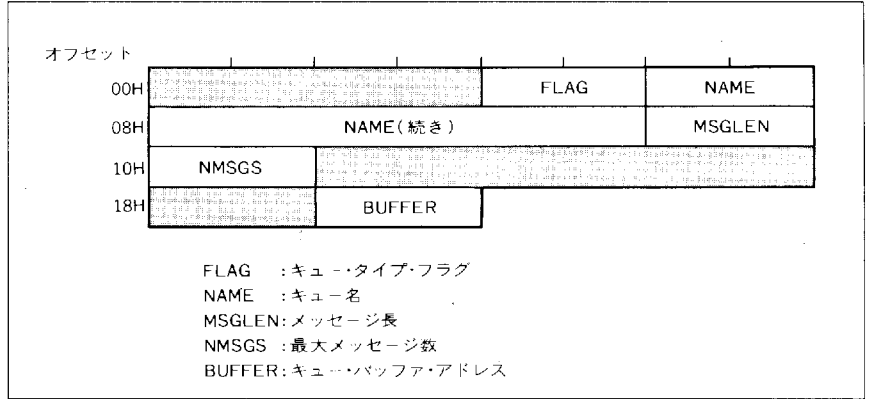
〔図20〕

PD, UAD フォーマット



〔図21〕

キュー・ディスクリプタ・  
フォーマット



PD(Process Descriptor)とUDA(User Data Area)の二つのデータ構造が与えられる(図20参照)。readyリスト上のプロセスに順々にCPUを割り当てるのがディスパッチャの仕事である。

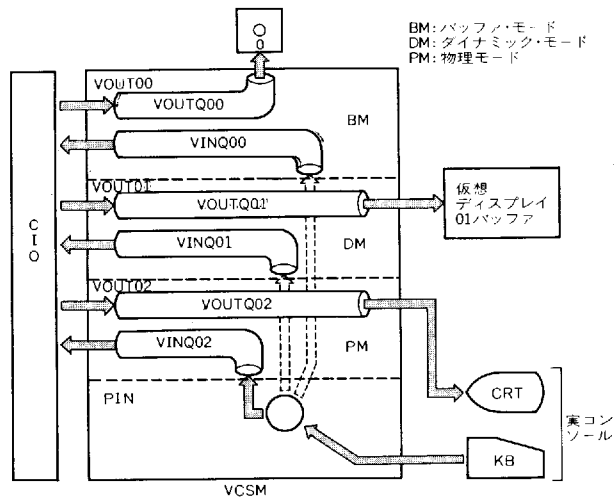
コンカレントCP/M-86のスケジューリングはプライオリティ・ベースでおこなわれる。システム内の各プロセスには表10に示すように0~255のプライオリティが与えられている。スケジューラは、つぎに走らせるプロセスとしてreadyリスト内の最高位プライオリティのプロセスを選ぶ。同じプライオリティをもつプロセスが複数あるときは、ラウンド・ロビンでスケジューリングされる。いったんCPUが割り当てられると、そのプロセスは、①I/O要求が発生する、②より高位プライオリティのプロセスがreadyになる、③割り当てられたタイム・スライスを使い切る、のいずれかになるまで走る。システム内につねにCPUを割り当てるプロセスがあることを保証するために、最低プライオリティをもつIDLEプロセスが存在し、つねにready状態にある。

▷キュー管理

プロセスが並行して走る場合は、おたがいが協調して走れるようにプロセスの同期、排他制御、プロセス間通信などをサポートする機構が必要になる。そのためにUNIXではパイプ、RMX-86ではメール・ボックスを用いているが、コンカレントCP/M-86ではキュー(queue)を使う。キューはキュー・ディスクリプタ(図21)とキュー・バッファで構成され、ファイルと同じプロトコル(open, read, write, closeなど)でアクセスされる。プロセスはキューに対してリード/ライトをおこない、プロセス間通信などを實現する。キューへのアクセスには条件付き(conditional)と無条件(unconditional)の2種類があり、条件付きアクセスの場合は、キュー・バッファがフルまたは空でリード/ライトができないときはエラーとして返ってくるが、無条件アクセスの場合は、他のプロセスが先にアクセスしてリード/ライトが可能になるまでそのプロセスは待たされる(blocked状態になる)。

名前がMXで始まるキューは、特別にバイナリ・セマフォとして實現され、システム資源の排他制御に用いられる。

〔図22〕 VCSMの動作概念



## ▷システム・タイミング管理

システム・タイミング機構は、ファイルの日時スタンプなどに利用されるシステム日時の管理と、プロセスを指定した時間だけ眠らせる DELAY ファンクションで用いられる。

## 4.7 VCSM

### (Virtual Console Session Manager)

VCSM は CIO と XIOS のインターフェースをとるモジュールである。VCSM はシステムに一つだけ存在する物理入力プロセス (PIN) と、仮想コンソールの数だけ存在する仮想出力プロセス (VOUT××) から構成され、それぞれは RSP である。

VCSM の動作概念を図22に示す。PIN は高プライオリティで走り、キーボードからの入力を遅れなしで受け取る。そこで仮想コンソールの切替えや CTRL-C などの処理を内部でおこない、残りの文字を、現在セレクトされている仮想コンソールの文字入力キュー (VINQ××) にわたす。CIO から文字入力要求があれば、VINQ×× の先頭の 1 文字をわたす。

CIO からの文字出力要求により、仮想コンソールの文字出力キュー (VOUTQ××) の最後尾に 1 文字書き込む。VCSM は仮想コンソールの状態により、この出力要求を XIOS にわたすか、内部的に処理するか決める。仮想コンソールが物理モードにあるときは、文字は VOUTQ にとどまらず、直接、実コンソールにわたされる。バッファ・モードにあるときは VOUTQ でバッファリングされ、いっぱいになればディスクへ書き出される。ダイナミック・モードの場合は、文字は直接 XIOS にわたされるが、XIOS 内のディスプレイ・バッファの内容を更新するだけで表示はされない。

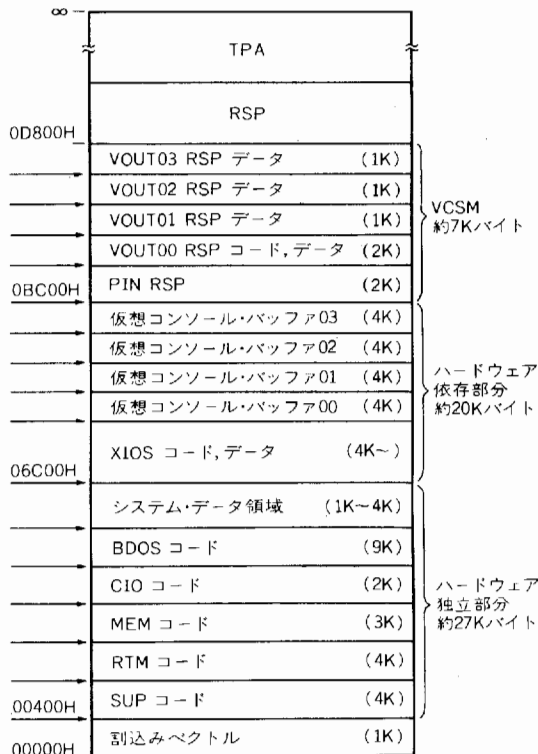
## 4.8 メモリ・マップ

図23に典型的なコンカレント CP/M-86 システムのメモリ・マップを示す。VCSM を除いたハードウェア独立部分 (SUP, RTM, MEM, CIO, BDOS, システム・データ領域) が 26 K バイト、VCSM では PIN が 2 K バイト、VOUT のコード部分が 1 K バイト、VOUT に関連するデータが仮想コンソールごとに 1 K バイト必要になる。ハードウェア依存部分では、XIOS が 4 K バイト以上、仮想スクリーン・バッファが仮想コンソールごとに 2~4 K バイト必要である。合計すると 4 仮想コンソールをサポートとした場合、コンカレント CP/M-86 は 54 K 必要であり、かなり大きなシステムになる。

## あとがき

コンカレント CP/M-86 は現在のところ IBM-PC バ

[図23] コンカレント CP/M-86 のメモリ・マップ



ーションだけしか発表されていないが、近々国産マシンに採用されるとの噂もある。またユーザ・コンフィギュラブルなバージョンもリリースされるようである。コンカレント CP/M-86 のシングルユーザ・マルチタスキングという従来のパソコンにはなかった新しい環境は、パソコン上でのソフトウェア開発を効率的に進めるための強力な武器になるだろう。ソフトウェア開発にたずさわる人間としては、このような OS がのったパソコンがもっと登場してほしいものだと感じる。いずれにしてもコンカレント CP/M-86 は、これからの 16 ビット・パソコンの OS 動向に大きな影響を与えることになるだろう。

## 参考文献

- 1) 石川 勝, 「インタビュー: CP/M の父 Gary Kildall 氏と語る」, 『information』 Vol. 1.2, 1982 年 pp. 33~41
- 2) Concurrent CP/M-86 Operating System Programmer's Guide, Digital Research Inc. (1982)
- 3) Concurrent CP/M-86 User's Guide, Digital Research Inc. (1982)
- 4) Horovitz, et al. "Concurrency: Key to 16-bit Operating System Efficiency", Computer Design, June 1982 pp. 183~192

たがや・てつや (株) エスクエラ