

資料

# インテリジェントディスクファイル管理ユニット 取扱い説明書

昭和 58 年 3 月



財団法人 日本情報処理開発協会

DEC

7

77

この資料は、日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて、昭和57年度に実施した「マイクロコンピュータの利用に関する共通的な技術開発」の一環としてとりまとめたものであります。

# 目 次

1. 概 要 .....	1
1.1 システムの概要 .....	1
1.2 システム構成 .....	2
2. 操作方法 .....	3
2.1 各部名称 .....	3
2.1.1 ユニット本体 .....	3
2.1.2 ハードディスク .....	5
2.2 設 置 .....	7
2.2.1 設置場所 .....	7
2.2.2 電源およびアース .....	8
2.2.3 設置手順 .....	8
2.3 起動、停止 .....	8
2.3.1 起 動 .....	8
2.3.2 停 止 .....	9
2.3.3 表示の説明 .....	9
2.3.4 スイッチ .....	10
3. システムの機能 .....	11
3.1 ファイル管理機能 .....	11
3.1.1 ファイルの種類 .....	11
3.1.2 ディスクのファイル構造 .....	15
3.1.3 ファイルのインデックス .....	16
3.1.4 ファイルアクセス .....	21
3.2 コマンドの多重処理 .....	24
3.3 障害(エラー)の検出および処理 .....	25
3.3.1 自己診断機能 .....	25

3.3.2	障害検出機能	28
3.4	停止	29
3.4.1	HALTスイッチによる停止	29
3.4.2	ホストインタフェースの障害による停止	29
3.4.3	重大な障害による停止	29
4.	動作シーケンス	30
5.	コマンドオペレーション	36
5.1	コマンドの構成	36
5.1.1	ヘッダ部	36
5.1.2	パラメータ部	37
5.1.3	データ部	38
5.2	処理結果の構成	39
5.2.1	ヘッダ部	39
5.2.2	ステータス部	39
5.2.3	データ部	40
5.3	各コマンドの機能と構成	40
5.3.1	コマンドコード	40
5.3.2	ステータスコード	40
5.3.3	ファイルに関するコマンド	48
5.3.4	本ユニットに関するコマンド	68
6.	システムジェネレーション	74
6.1	パラメータ	74
6.2	システムジェネレーションの方法	75
6.2.1	パラメータの設定	75
6.2.2	手順	77
7.	ホストとの接続例	78
7.1	構成	78
7.2	準備	78
7.3	制御手続	79

# 1. 概 要

## 1.1 システムの概要

本ユニットは、ファイル管理機能を備えたディスク統括管理システムである。

従来、ディスク装置は、ホストコンピュータ側のOS（ファイル管理機能）により制御されてきたが、本ユニットは独自のファイル管理機能を備えたハイレベルなコントローラによってディスク装置を制御しており、ホストコンピュータ側の処理を容易にし、処理効率の向上をはかっている。

本ユニットの主な特長は次のとおりである。

- 木構造にファイルを管理する。
- ハードディスク、フロッピーディスクを統括的に管理する。
- データバッファにより、入出力処理を高速化している。
- 複数のコマンドを並行処理し、処理効率の向上を図っている。
- コマンドにより容易なファイルアクセスを可能にしている。

## 1.2 システム構成

本装置のシステム構成を図1-1に示す。

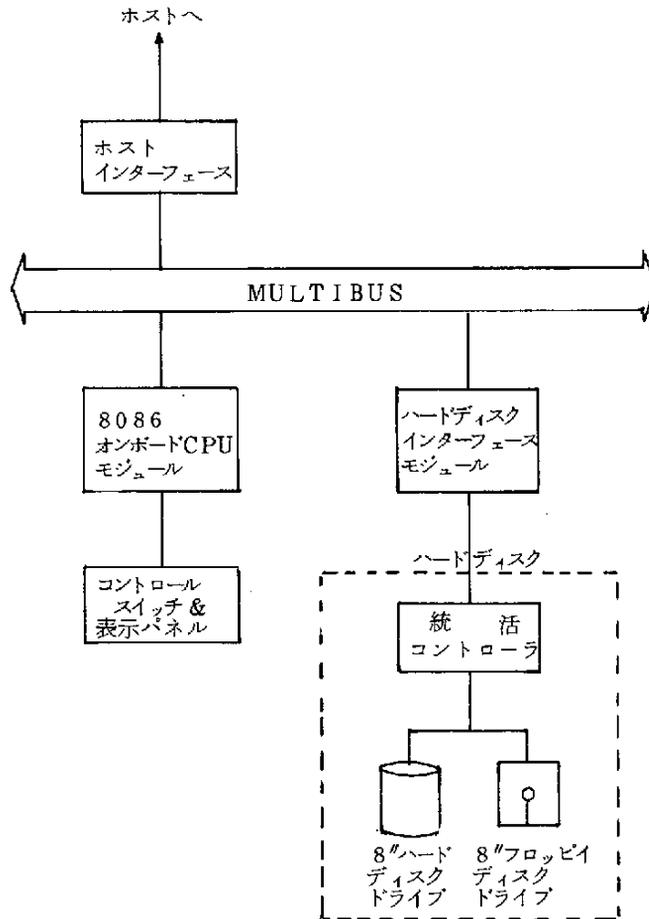


図1-1 システム構成

## 2. 操作方法

### 2.1 各部名称

#### 2.1.1 ユニット本体

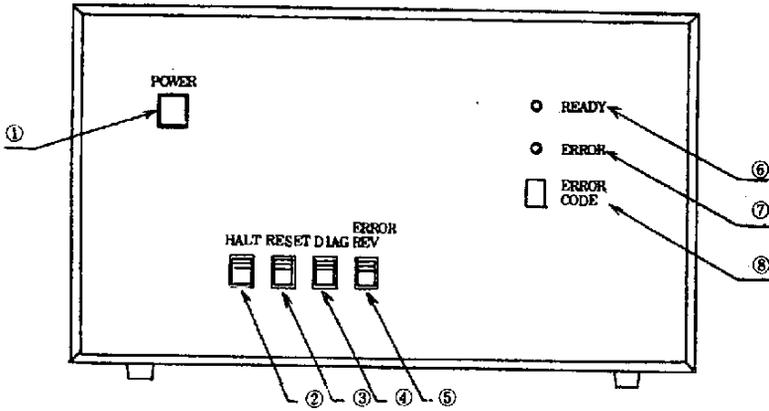


図 2-1 フロントパネル

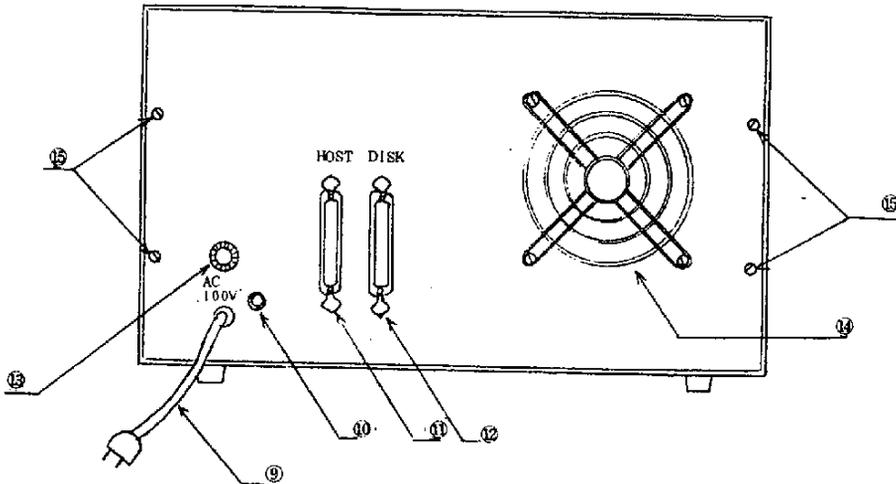


図 2-2 普通パネル

A) フロントパネル

- ① 電源スイッチ
- ② HALTスイッチ
- ③ RESETスイッチ
- ④ DIAGスイッチ
- ⑤ ERROR REV. スイッチ
- ⑥ READYランプ(LED)
- ⑦ ERRORランプ(LED)
- ⑧ ERROR CODE数字表示器

B) 背面パネル

- ⑨ 電源コード
- ⑩ GND端子
- ⑪ ホストコネクタ
- ⑫ DISKコネクタ
- ⑬ ヒューズホルダー
- ⑭ フィンガーガード
- ⑮ リアパネル取付ネジ

## 2.1.2 ハードディスク

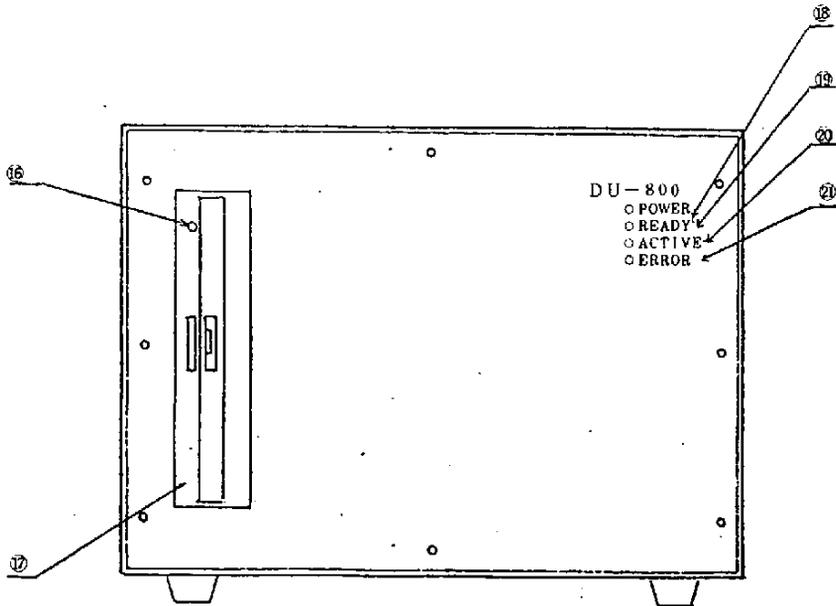


図 2-3 ハードディスクフロントパネル

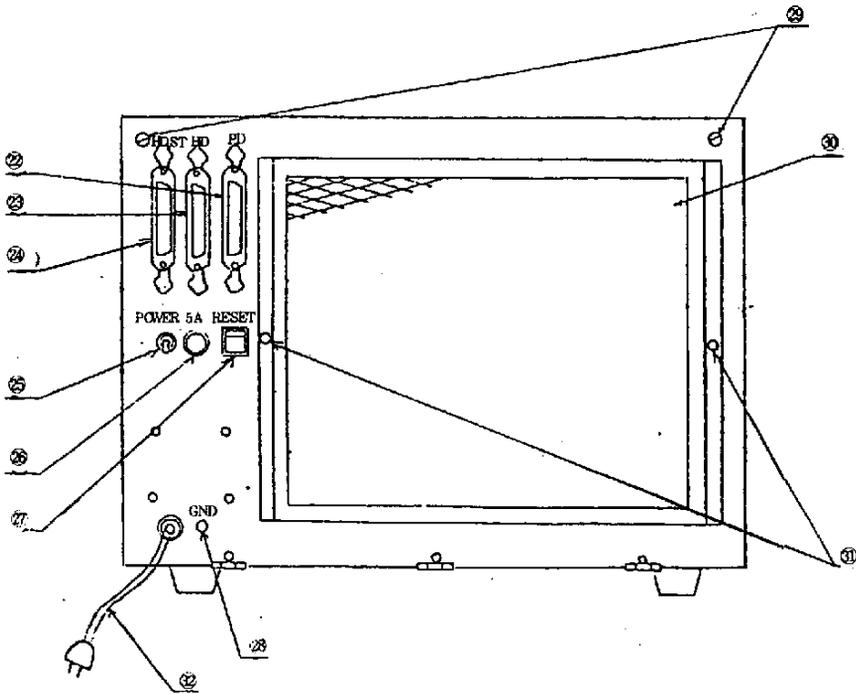


図 2-4 ハードディスク背面パネル

A) フロントパネル

- ⑩ フロッピーディスクドライブHEAD LOAD LED
- ⑪ フロッピーディスクドライブ
- ⑫ 電源表示用LED (POWER)
- ⑬ コントローラREADY表示LED (READY)
- ⑭ ハードディスクアクセス表示LED (ACTIVE)
- ⑮ コントローラエラー表示LED (ERROR)

B) 背面パネル

- ⑯ フロッピーディスクドライブ接続コネクタ (FD) (ディジーチェーン用)
- ⑰ ハードディスクドライブ接続コネクタ (HD) (ディジーチェーン用)
- ⑱ ホストコンピュータ接続コネクタ (HOST)
- ⑲ 電源スイッチ
- ⑳ ヒューズホルダー (5A)
- ㉑ リセットスイッチ
- ㉒ アース端子 (GND)
- ㉓ 背面パネル取付け用ローレットネジ
- ㉔ 防塵フィルタ  
フィルタ取付ネジ  
電源コード

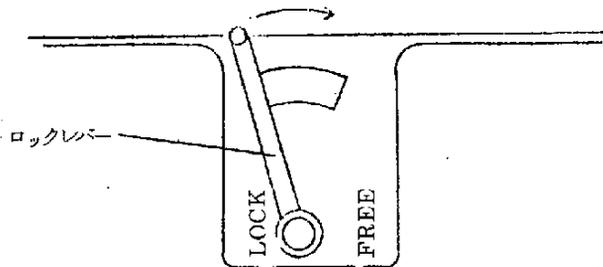


図 2.5 キャリッジの固定

## 2.2 設 置

### 2.2.1 設 置 場 所

本装置の設置場所は操作時の外的仕様に同じでなければならないが、操作上の信頼性を確保するために次の点を守る必要がある。

- (1) 塵やホコリが多量に発生する場所での使用を避け、できるだけ清潔な環境に設置すること。
- (2) 付近に、トランス、モーターなど強磁界を発生させる装置を置かないこと。
- (3) 腐食性ガスの濃厚な環境に設置しないこと。
- (4) 本装置は、温度上昇を防ぐために、ファン冷却を行なっているが、極端な高温下や低温下での使用は避けること。
- (5) 震動のある所に設置しないこと。

### 2.2.2 電 源 及 び アース

電源は必ずAC 100V (50Hz又は60Hz)を使用し、アースをとること。

### 2.2.3 設 置 手 順

- (1) ユニット本体及びハードディスクの電源プラグがコンセントに差し込まれていないことを確認する。
- (2) ハードディスクのフロントパネルを取り外し、中のディスクドライブのロックレバーをLOCK状態からFREE状態にする。このレバーにより、ディスクキャリッジの固定・解除を行なう。再び梱包、輸送する場合にはこのロックレバーをLOCK状態に戻すこと。
- (3) ユニット本体とハードディスクとの接続  
ユニット本体背面パネルにあるDISKコネクタ⑫と、ハードディスク背

面パネルにあるHOSTコンピュータ接続コネクタ⑭とを、付属のIFUケーブル①により接続する。

(4) ホストコンピュータとの接続

ホストコンピュータとの接続は、ユニット本体背面パネルにあるホストコネクタ⑭ (HOST) により行なう。

(5) 各電源プラグをコンセントに差し込む。

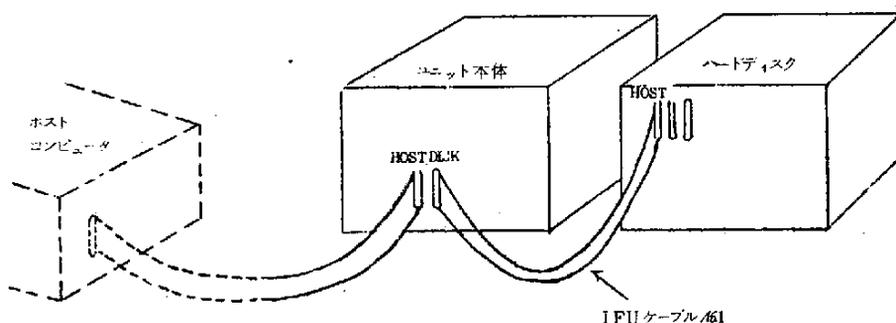


図 2.6 ユニット間の接続

## 2.3 起 動、停 止

### 2.3.1 起 動

本ユニットを起動するときは、まず、ディスク装置の電源スイッチ⑳をONにする。POWERランプ㉑が点灯し、しばらくしてREADYランプ㉒が点灯する。このときERRORランプ㉓が点灯した時は、電源スイッチ⑳をOFFにし、各部の接続を確かめてから、電源を投入する。

ディスク装置がREADY状態になってから、本ユニットの電源スイッチ①をONにする。POWERランプ①が点灯し、READYランプ⑥が点灯する。このとき、ERRORランプ⑦が点灯した時は、エラーコード⑧の内容を確かめ、本ユニットの電源スイッチ①をOFFにし、各部の接続および起動手順を確かめてから、電源を投入する。

本ユニットがREADY状態になってから、ホストコンピュータ側の電源を投入し、ホスト側のシステムを起動する。

起動時や次に述べる停止時は、フロッピーディスクはとりはずしておくこと。

### 2.3.2 停 止

本ユニットを停止する場合は、ホストコンピュータとの間で全ての処理（コマンドの処理、ファイルのクローズ、ディスマウント）が完了していることを確かめてから、本ユニットのHALTスイッチ②をおし下げる。READYランプが消灯してから、本ユニットの電源スイッチ①をOFFにする。

その後、ディスク装置の電源スイッチ③をOFFにする。

### 2.3.3 表示の説明

POWER — 点灯時、本ユニットに電源が供給されていることを示す。

READY — 点灯時、本ユニットがホストからのコマンドを受け付け可能な状態であることを示す。

ERROR — 点灯時、本ユニット、又はディスク装置にエラーが発生したことを示す。

ERROR CODE — ERROR点灯時、発生したエラーの種類を示す。

<エラーコードの意味>

コード (HEX)	意 味
0	正常 (エラーがない状態)
1	本ユニットのコントローラ上の致命的なエラー
2	ホストインタフェース上のエラー、ホストの順
{	(1~4)に対応
5	

6	入出力装置上のエラー
7	特殊ファイルリストの順に対応
D	
E	未 使 用
F	"

#### 2.3.4 スイッチ

- POWER — 本ユニットの電源をON-OFFする。
- RESET — 本ユニットをイニシャライズする。
- HALT — 本ユニットを停止する。
- DIAG — 本ユニットの自己診断を行なう。
- ERROR REV — 発生中の複数個のエラーコードを順に表示する。

### 3. システムの機能

#### 3.1 ファイル管理機能

本ユニットは、本ユニットに接続される入出力装置を統一的に制御し、木構造にファイルを管理する。

ホストコンピュータは、コマンドによってファイルをアクセスする。

##### 3.1.1 ファイルの種類

本ユニットが管理するファイルは下記の3つに分類される。

###### (1) 通常ファイル

テキストやデータを含むファイルである。

###### (2) ディレクトリファイル

通常ファイルや下位のディレクトリファイルを登録するためのファイルで、このファイルにより、木状のファイル構造が形成される。

ディレクトリファイルは、通常ファイルと同様の形式で作成するが、作成と同時に、\$\$（親ディレクトリを示すファイル）と\$（自分自身を示すファイル）が登録される。以後、ホストコンピュータからのコマンドにより、ディレクトリファイルへのファイルの登録、あるいは消去が行なわれる。

ディレクトリファイルを消去する場合は、上記の\$\$、\$以外のファイルが登録されていない。また、ディレクトリファイルは、下記の場合に限り、通常ファイルと同様の形式によるアクセスが可能である。

###### ① リード

ディレクトリファイルをオープンし、リードすることができる。ただし、ライトは許されない。これは次のような場合に有効である。

すなわち、非常に多くのファイルを1つのディレクトリファイルに登録

すると、ファイルリストのリードコマンドでは、全てのファイルのリストをリードできない(ホストとの間の最大転送データサイズにより制限されるため)場合である。

② コピー

ディレクトリファイルをコピーすると、ファイルの種類は通常ファイルとなる。ディレクトリファイルは、1レコードが16バイトのシーケンシャルなファイルで、各レコードは図3-1に示す構成である。

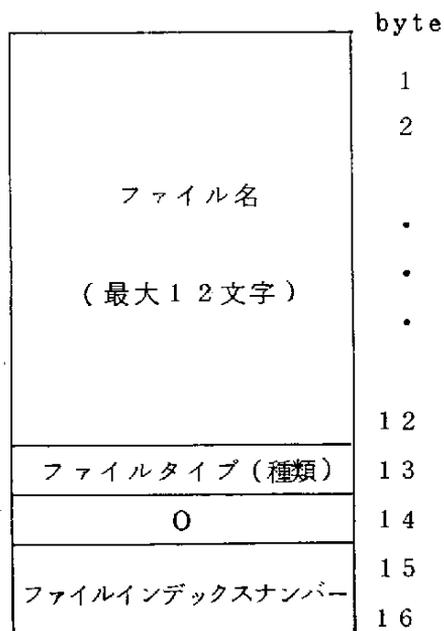


図3-1 ディレクトリファイルのレコード構成

なお、1つのディレクトリファイルに登録されるファイルの名前は、それらのファイルの中で全て唯一のものでなければならない。

(3) 特殊ファイル

本ユニットに接続される入出力装置の各ユニットに対応するファイルで、システムジェネレーションにより作成され、本ユニット起動時に登録される。

図3-2に登録された状態を示す。

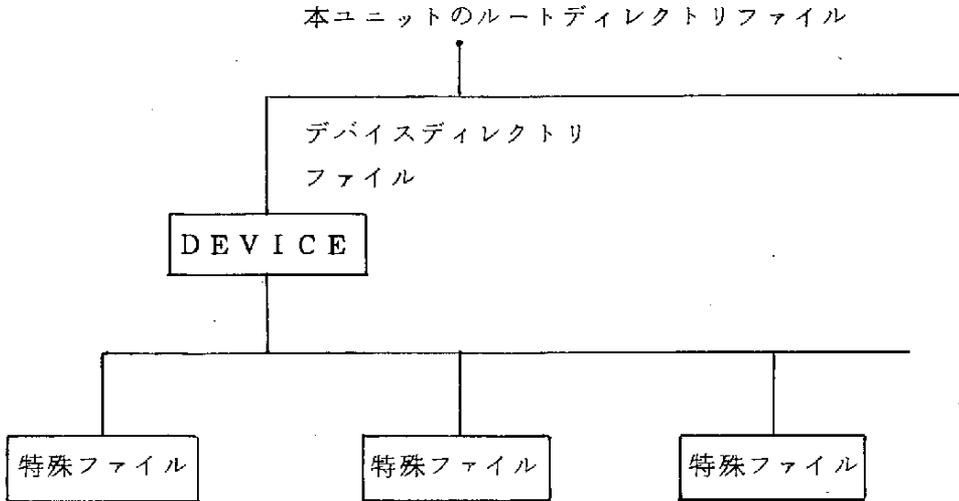


図3-2 本ユニット起動時のファイルの構成

特殊ファイルへのアクセスは下記の2とおりがある。

- ① ハードディスクやフロッピーディスクの場合に、その中にファイル構造を形成し、通常ファイルやディレクトリファイルを登録し、それらのファイルを個々にアクセスする。この場合のコマンドの手順を図3-3(i)に示す。

ディスクをマウントすると、初期化時に作成されたそのディスクのルートディレクトリファイルが、本ユニットのルートディレクトリファイルに登録され、そのディスクのファイル構造が本ユニットの管理の下におかれる。ディスマウントにより、切り離される。

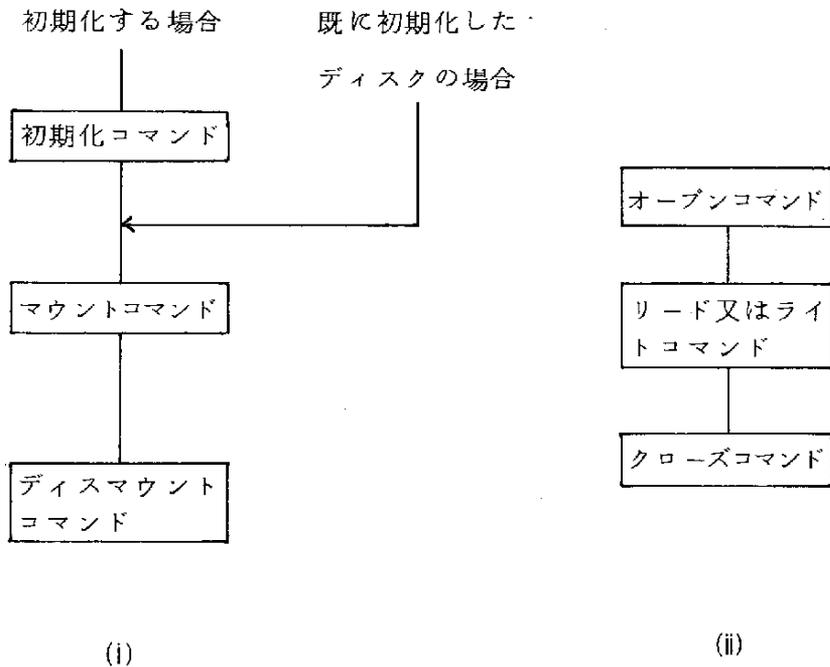


図 3-3 特殊ファイルへのアクセスの手順

② 特殊ファイル、すなわち対応する入出力装置への直接的な I/O を行う場合は、図 3-3 (ii) のような手順となる。

データのリード/ライトのコマンドにその入出力装置の論理的なアドレスを指定する。

特殊ファイルは消去することができない。また本ユニットが管理する特殊ファイルは、表 3-1 に示すように定められている。また、特殊ファイル名は次のように定める。

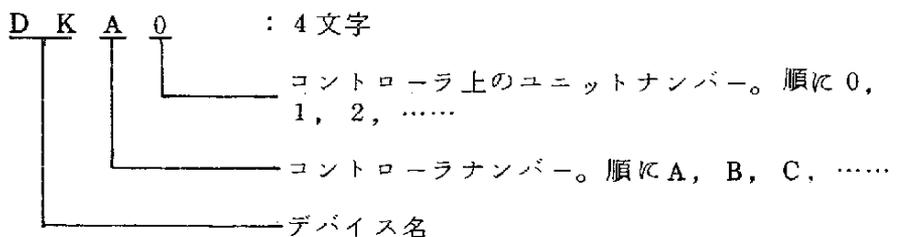


表 3-1 特殊ファイル

入出力装置の種類	ウィンチェスタ型 ハードディスク	両面倍密度フロッピー
デバイス名(2文字)	DK	FD
最大コントローラ数	1	1
最大ユニット数	4	4
1ユニット当りの容 量(ブロック数)*1	38400	1989 *2
最大登録ファイル数	4096	256

\*1 1ブロックは512バイト

\*2 フロッピーディスクは、下記のようなフォーマットで使用する。

ヘッダ数 = 2

シリンダ数 = 77

バイト/セクタ = 256

セクタ/トラック = 26

ただし、ヘッド=0のシリンダ0を使用しない。

フロッピーディスクを論理的にアクセスする場合は、ヘッド=0、シリンダ=0、セクタ=1から全てのセクタにアクセスできる。

### 3.1.2 ディスクのファイル構造

ディスクは初期化(コマンド)により、そのディスク固有のルートディレクトリファイルが作られ、その下にファイル構造が形成される。

図3-4に、初期化された状態を示す。

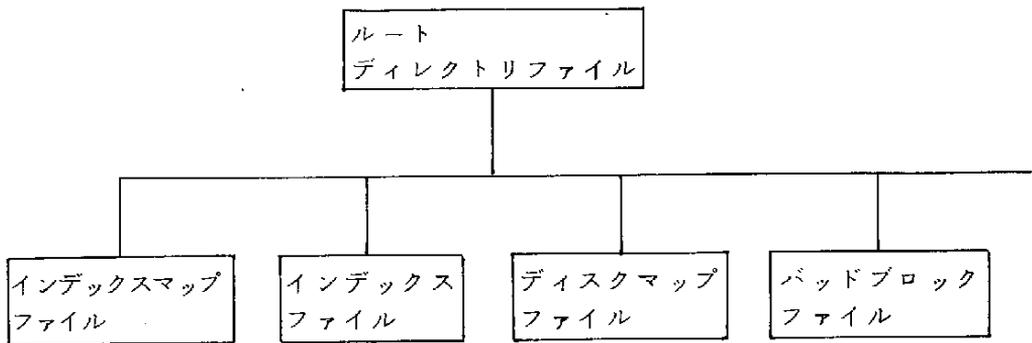


図 3-4 ディスク初期化時のファイル構成

- (1) インデックスマップファイル (ファイル名: INDEXM)  
インデックスファイルへのインデックスの登録状況を示す。
- (2) インデックスファイル (ファイル名: INDEXF)  
ディスクに登録される全てのファイルのインデックスを登録する。
- (3) ディスクマップファイル (ファイル名: BITMAP)  
ディスクの使用状況を示す。
- (4) バッドブロックファイル (ファイル名: BADBLK)  
ディスクのバッドブロックを登録する。

(1)~(4)の各ファイルは全て通常ファイルである。これらのファイルは、本ユニットが常時使用するものであり、ユーザが勝手にデータを書き込んだり、ファイルを消去すると、ディスクの中のファイルがアクセスできなくなるので、注意しなければならない。

ディスクの使用は、そのルートディレクトリファイルに、ユーザがファイルを登録することから開始される。

### 3.1.3 ファイルのインデックス

通常ファイルやディレクトリファイルは、インデックスを持ち、そのファイルに関する情報を登録する。

ファイルのインデックスは、ファイルの作成時に作られ、ファイルに変更が

ある毎に更新され、ファイルの消去とともにまっ消される。

図 3-5 にその構成を示す。

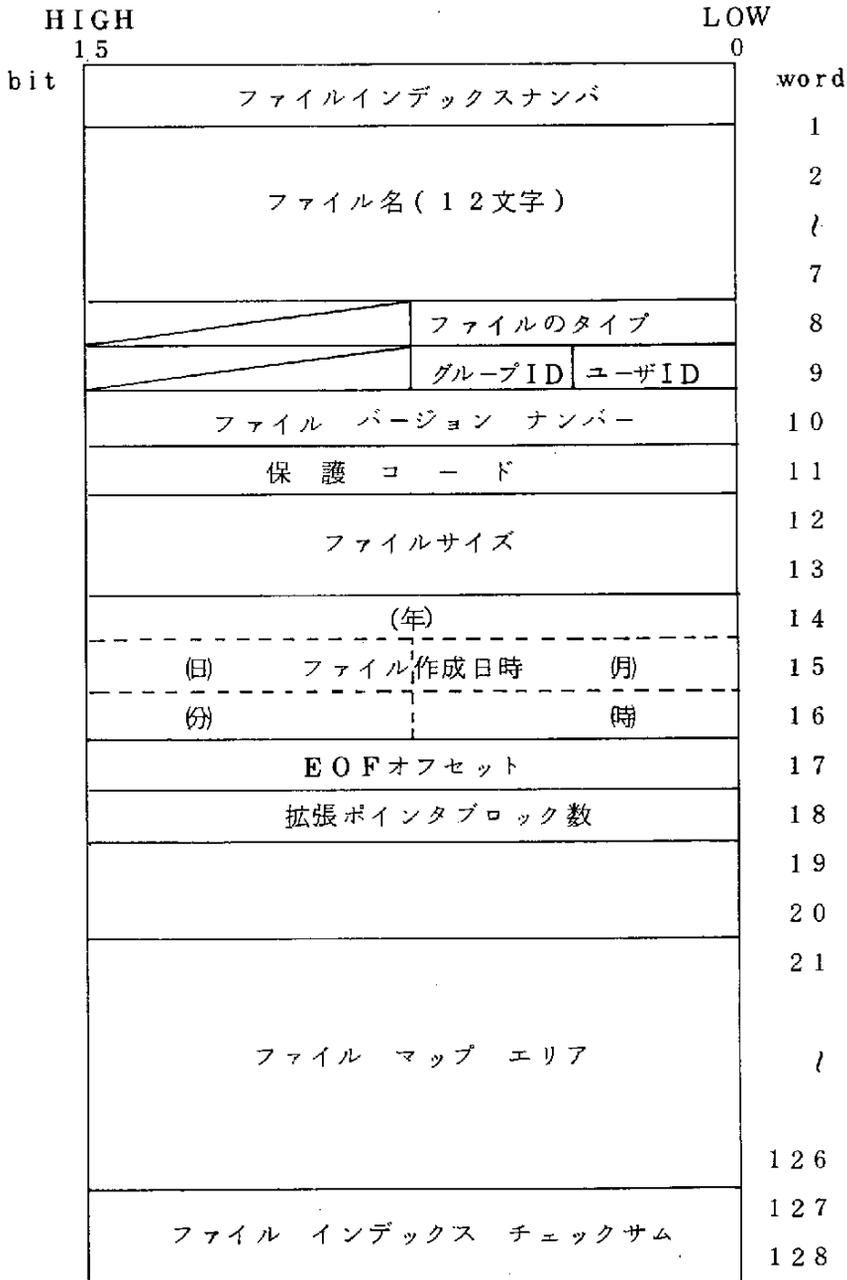


図 3-5 ファイルインデックスの構成

(1) ファイルインデックスナンバー

本ユニットが各ファイルに与えるファイル固有の値。インデックスファイルに登録されるインデックスの相対的な位置を示す。

(2) ファイル名

最大12文字のファイル名、使用できる文字は、英文字(A~Z、a~z)数字(0~9)で、他に/(slash)を除く英記号がある。'\$'と'\$ \$'の2つのファイル名は本ユニットが使用するため、指定できない。

(3) ファイルタイプ

通常ファイル、ディレクトリファイル、特殊ファイルの種類を示す。値はそれぞれ、0、1、2である。

(4) ユーザID、グループID

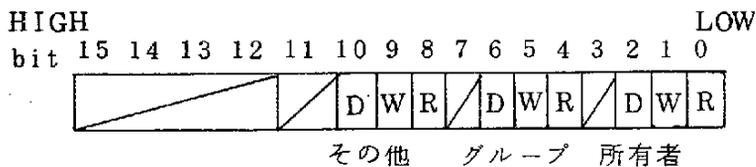
ファイルの所有者(作成者)を示すコードで、それぞれ1~15の値である。

(5) バージョンナンバー

ファイルの更新状態を示す。作成時に1がセットされ、通常ファイルの場合、以後書き込みが行なわれた後のクローズ時に1加算される。

(6) 保護コード

(4)のユーザID、グループIDにより、ユーザを所有者(同一のユーザIDグループIDを持つユーザ)、グループ(同一のグループIDを持つユーザ)その他のユーザの3つに分類し、それぞれに、リード(ファイルのリード)、ライト(ファイルへのライト)、デリート(ファイルの消去)の許可、禁止を示す。



ビットON=許可  
OFF=禁止

(R:リード、W:ライト、D:デリート)

保護コードはファイル作成時に指定されるが、指定されない場合は、次の値を使用する。

所有者 : R、W、D全て許可

グループ : R、W、D全て許可

その他 : Rのみ許可

(7) ファイル作成日時

ファイルが作成された時間を示す。年、月、日、時、分の値は、本ユニットが管理する時間の値が使用される。

(8) ファイルサイズ

ファイルのサイズを512バイトを1ブロックとしたブロック数で示す。

(9) EOFオフセット

ファイルの最後のブロックにおける最後のデータの相対アドレスを示す。0～511の値である。

(10) 拡張ポインタブロック数

(1)で述べるポインタブロックが拡張された場合に、その拡張ポインタブロックの数を示す。

(11) ファイルマップエリア

ファイルのデータを参照するためのエリア(ポインタブロック)である。

図3-6にその構成を示す。

ファイルのデータの参照は、各ポインターにより行なうが、ファイルのインデックス(マップエリア)の50ケのポインターで不足した場合は、ポインタブロックの拡張を行ない、125ケのポインターを設ける。以後も同様の拡張を行なう。

① 参照サイズ

おのおののポインタブロック内の各ポインタにより参照されるファイルデータの総ブロック数を示す。

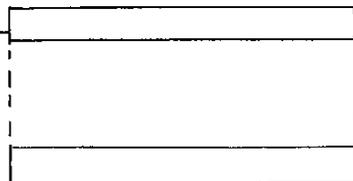
参照サイズ (LOW)	1
拡張フラグ (HIGH)	2
未使用ポインタナンバー	3
総ポインタ数 (50)	4
ポインタ - (1)	5
ポインタ - (2)	7
⋮	9
ポインタ - (50)	103
拡張ポインタブロックアドレス	105

ファイルマップエリア  
(インデックス内のポインタブロック)

参照サイズ (LOW)	1
拡張フラグ (HIGH)	2
未使用ポインタナンバー	3
総ポインタ数 (125)	4
ポインタ - (1)	5
ポインタ - (2)	7
⋮	9
ポインタ - (125)	253
拡張ポインタブロックアドレス	255

拡張ポインタブロック

先頭ブロックアドレス (LOW)	
ブロック数 (HIGH)	
ポインタ - (n)	



ブロック数

(各 512 バイトのブロック)

図 3-6 ファイルマップエリアの構成

- ② 拡張フラグ(0または1)、拡張ポインタブロックアドレス  
 拡張フラグが1のとき、ポインタブロックが拡張されていることを示し、その拡張ブロックのアドレスを示す。
  - ③ 未使用ポインタナンバー  
 最初の未使用ポインタを示す。
  - ④ 総ポインタ数  
 ポインタブロック内のポインタ数を示す。インデックス内のポインタブロックの場合50、拡張ポインタブロックの場合125である。
  - ⑤ ポインタ  
 連続したデータブロックの先頭のブロックのアドレスとそのサイズ(ブロック数)を示す。
- (12) チェック・サム  
 ファイルインデックスの word 1 ~ word 126 のチェックサムデータであり、ファイルのクローズ時に書き込まれ、オープン時にチェックされる。

### 3.1.4 ファイルアクセス

#### (1) ファイルへのアクセス手順

手順を図3-7に示す。

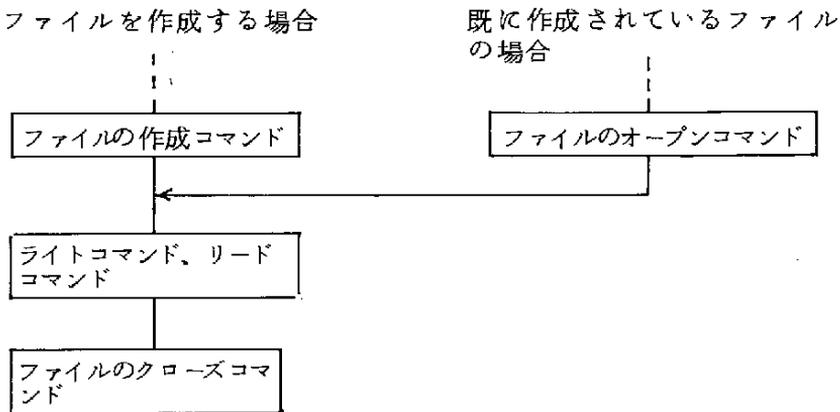


図3-7 ファイルのアクセス手順

## (2) ファイルの保護機能

3-1-3で述べたようにファイルはそのインデックスに所有者を示すユーザID、グループIDとファイルへのアクセスを定めた保護コードを持つ。これらの情報は、そのファイルの保護のために使用される。

### ① ファイルの消去

あるユーザID、グループIDを持つユーザがファイルを消去する場合、そのユーザがファイルの所有者を基準にして、所有者か、同じグループか、その他かに分類され、分類されたユーザに消去（デリート：1）が許されていれば、消去できる。

### ② ファイルへのリード/ライト

ファイルのオープン時に、そのファイルへのアクセスモードを指定する。アクセスモードは以下の値である。

リードのみ=1

ライトのみ=2

リード/ライト=3

ファイルの消去の場合と同様に指定されたアクセスモードが、そのユーザに、許可されているかがチェックされる。

また、このアクセスモードは、その後のファイルへのリード、ライトコマンドに対する保護に使用される。例えば、リードのみのモードでオープンしたファイルにライトコマンドは使用できない。

### ③ ファイルの専有、共有

ファイルのオープン時に、そのファイルへのアクセスタイプを指定する。アクセスタイプは以下の値である。

専有（他のユーザのアクセスを禁止する）=0

専有（クローズ時にファイルを消去する）=1

共有（他のユーザのアクセスを許す）=2

あるユーザが専有タイプでオープンしているファイルを他のユーザがオ

オープンすることや、共有タイプでオープンされているファイルを専有タイプでオープンすることはできない。

同じファイルを複数のユーザがオープンする場合は、全てのユーザが共有タイプでオープンしなければならない。

以上述べたように、ファイルの保護は、そのインデックスが持つ情報による静的な保護と、アクセスモード、アクセスタイプによる動的な保護に分類される。

### (3) ファイルアクセスバッファ

本ユニットは、ファイルへのリード/ライトのためにアクセスバッファを持っている。本ユニットはこのバッファを以下のように利用する。

- ① オープンされたファイルのインデックスをバッファに読み込んでおき、ファイルデータへのアクセスを容易にする。
- ② リードの場合は、バッファにデータを先読みしておき、このバッファ上にデータがあればそれをユーザに渡す。
- ③ ライトの場合はこのバッファ上にデータを書き込むことにより、ユーザに書き込み完了を通知し、一定の量になった時に、ディスク等へ書き込む。このような操作により、時間のかかる入出力装置へのアクセスの回数を減らし、処理を高速化することができる。

### (4) ファイル名の指定

ファイルは、ディレクトリファイルにより、木構造に管理されるので、ファイル名を指定する場合は、本ユニットのルートからの経路名（途中に介在するディレクトリファイル名）で指定する。

各ファイルは/（slash）で区切る。

例えば、ファイル“FILE”が本ユニットのルートから、各ディレクトリファイル“DIR1”、“DIR2”を経由しているならば、ファイルの指定は、

/DIR1/DIR2/FILE （DIR1はルートディレクトリファイル）

となる。本ユニットのルートからの経路を示す場合は、必ず“/”で始まらなければならない。

この指定を簡単にするために、次のような方法がある。すなわち、ユーザ毎にカレントディレクトリファイルの本ユニットに登録する方法である。ユーザはファイルを指定する場合、このカレントディレクトリファイルからの経路名を指定すれば良い。

上記の例で、ユーザのカレントディレクトリファイルを“DIR1”とすると、ファイルの指定は、

DIR2/FILE

となり、カレントディレクトリファイルを、DIR2とすると、

FILE

となる。

ただし、特殊ファイルの指定の場合は、使用するコマンドにより、多少異なる。

① 特殊ファイルのオープン

/DEVICE/特殊ファイル名

② 特殊ファイルのリストのリード

/DEVICE

③ 初期化、マウント、ディスマウント、使用状況のリード

特殊ファイル名

ここで、特殊ファイル名は4文字である。

## 3.2 コマンドの多重処理

ユーザは、先に発したコマンドの完了を待つことなく、新たにコマンドを本ユニットに送ることができる。

本ユニットに同時に書き込むことができるコマンド数は、システムジェネレ

ーションによって指定される。

本ユニットは、各コマンドの処理の進行により、他に実行可能なコマンドがあれば、その処理を開始する。例えば、入出力装置へのI/Oによりコマンドの処理が中断された場合である。

なお、ファイルへのリード/ライトコマンドは次のように処理する。

- (1) 同一のファイルへのリード/ライトは、複数のユーザが共有している場合も含めて、コマンドを受け付けた順に処理する。
- (2) 異なるファイルへのリード/ライトは、ファイル毎に並行して処理する。

このように、入出力装置へのI/Oとホスト間のデータ転送を独立して処理を行なうことにより効率の向上をはかっている。

### 3.3 障害(エラー)の検出および処理

#### 3.3.1 自己診断機能

本ユニットは障害を事前に検出するための自己診断機能を持つ。

表3-2に、診断タイミング、および処置を示す。

表3-2 診断の内容

対 象	タイミング			障害検出の処置	備 考
	起動時	コマンド	DIAG スイッチ		
CPU	○	○	○	エラー表示を行ない、停止する	
ROM	○	○	○	同 上	
RAM	○			同 上	
ディスクコントローラ およびハードディスク	○	○	○	エラー表示	
フロッピーディスク		○	○	同 上	診断用フロッピーディスクをセットする
表 示 パ ネ ル			○	—————	一定のパターンで点、消灯することを確認する。注1.
ホストインタフェース		○		—————	

注1. エラー表示は0, 1, 2, …… , Fを順に表示し、0に戻る。  
ランプは、点消灯をくり返す。

(1) 診断の結果

① エラー表示

障害を検出した場合は、ERRORランプの点灯、およびエラーコードの表示を行なう。表示されるエラーコードは表3-3に示す。複数の障害が発生した場合は、REVスイッチにより、順にエラーコードを見ることができる。

表3-3 エラーコード

エラーコード (HEX)	意 味
0	エラーがない状態
1	CPU、ROMあるいは、RAMに障害がある。 重大な障害である。
2	ホストインタフェース1に障害がある。
3	” 2 ”
4	” 3 ”
5	” 4 ”
6	デバイス1*のコントローラまたはドライブに障害がある。
7	デバイス2*の ” ” ”
8	6、7と同様
9	
D	
E	未 使 用
F	同 上

\* デバイスの順は、特殊ファイルリストの順に対応する。

## ② 診断結果の通知

診断コマンドの場合は、その処理結果として、診断の結果が、ホスト側に戻される。

以外の場合は、最新の診断結果を本ユニットが保持するため、コマンド（ステータスのリード）により得ることができる。

なお、重大な障害が検出された場合は、本ユニットは停止するので、通知されない。

### 3.3.2 障害検出機能

本ユニットは診断時以外においても、障害監視機能により障害を検出する。

#### ① ホストインタフェース

ERRORランプ表示、エラーコード表示を行なう。（表3-3）

#### ② 入出力装置（コントローラまたはドライブ）

ERRORランプ表示、エラーコード表示を行なう。（表3-3）

また、コマンドの処理中に障害が発生した場合は、コマンドの処理結果にエラーコードが戻される。（表5-2）

#### ③ コマンドのエラー

本ユニットが受け付けるコマンドには、処理する上で以下のようなエラーが考えられる。

a、コマンドコードエラーのような形式上のエラー

b、ファイルアクセスに関するエラー。ファイルがない、あるいは、保護エラー等である。

これらのエラーコードは、表5-2に示す。

## 3.4 停 止

本ユニットは、下記に述べる停止処理を行なう。

### 3.4.1 H A L Tスイッチによる停止

本ユニットは、H A L Tスイッチにより、次のような処理をおこなう。

- (1) ホスト側からのコマンドの受け付けを終了する。
- (2) 万一、未処理のコマンドがあれば、処理を完了し、オープン中のファイルはクローズする。なお処理結果はホスト側には戻されない。
- (3) R E A D Yランプを消灯し、停止する。

### 3.4.2 ホストインタフェースの障害による停止

いずれかのホストインタフェース上で障害を検出した場合は、そのホストユーザに関して、次のような処理をおこなう。

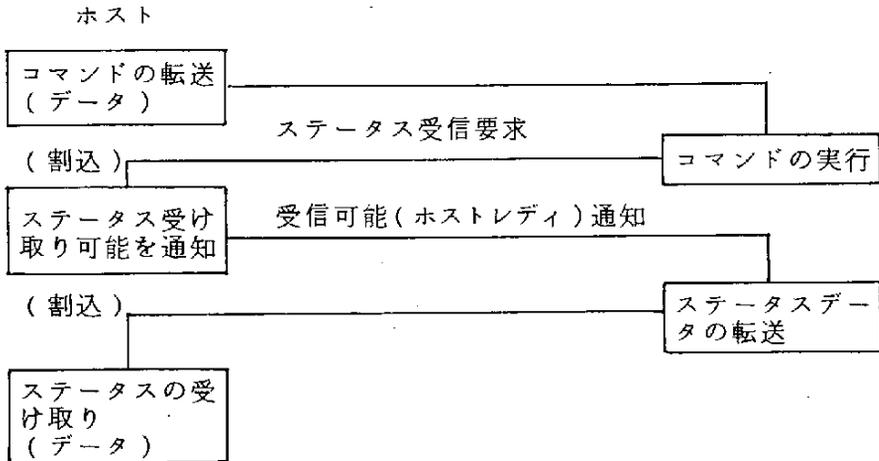
- (1) そのユーザが発したコマンドが未処理であれば、処理を完了する。ただし、処理結果は戻されない。
- (2) そのユーザがオープンしているファイルは全てクローズされ、マウントしている特殊ファイルはディスマウントされる。

### 3.4.3 重大な障害による停止

重大な障害が検出された場合は、ただちに停止する。

## 4. 動作シーケンス

ホストコンピュータが本装置に一つのコマンドを発行し、それを処理するシーケンスは次の様になる。



まずホストは本装置がレディである事を確認してコマンドおよび付随するパラメータを転送する。

本装置は与えられたコマンド固有の処理を実行した後、ステータスデータの受け取りをホストに要求する。

本装置はホストの受信可能通知を確認するとステータスデータを返送する。

### (1) 多重制御

ホストは一つのコマンド発行後、コマンドの完了を待つ事なく次のコマンドを発行する事が可能である。

本装置は複数のコマンド受信バッファを持っている為、一つのコマンドを受け取るとそのコマンドに対する処理を実行するとともに受信バッファに空があれば直ちにコマンド受け付け可能 (iFuレディ) を通知する。

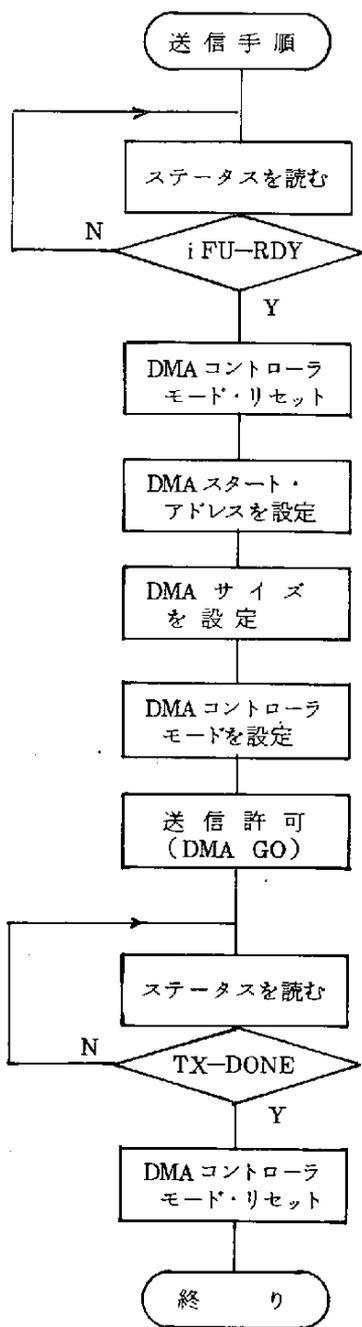
#1

注 1. iFu : 当装置の略称

## (2) DMA 転送

ホストからのコマンドデータ及び本装置からのステータスデータはDMAにより転送される。

ホストがコマンドを発行する時は、各コマンド固有のサイズ分だけDMA転送するが、本装置からのステータスデータを受信する時はコマンドによりデータサイズが異なる為、常に最大サイズのDMA受信準備をした後、ホストレディを発行する。



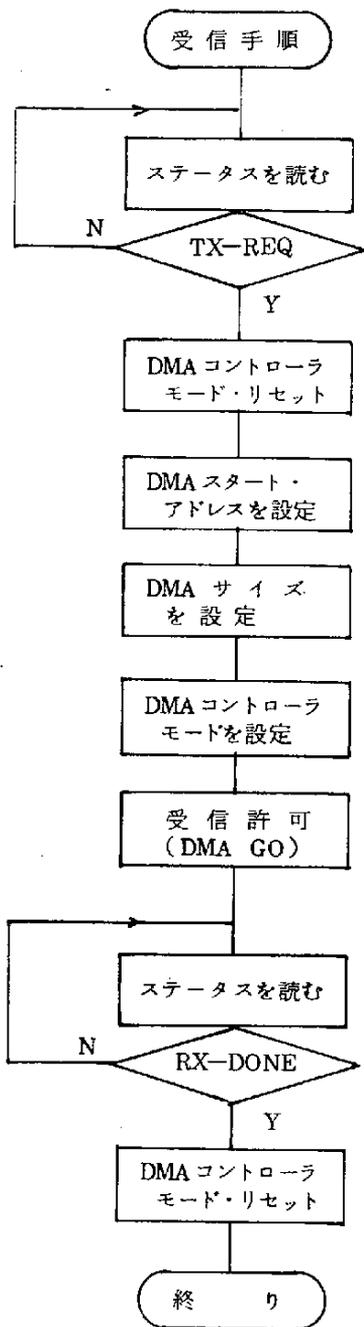
iFUシステムが受信可能状態となるのを待つ。  
 ※ iFU-RDY 割込で確認。

iFUとのDMA制御に用いているDMAチャンネルの動作を禁止する。

送信用チャンネルをENABLEとする。

DMAの終了を確認する。

ホストから iFU への送信手順

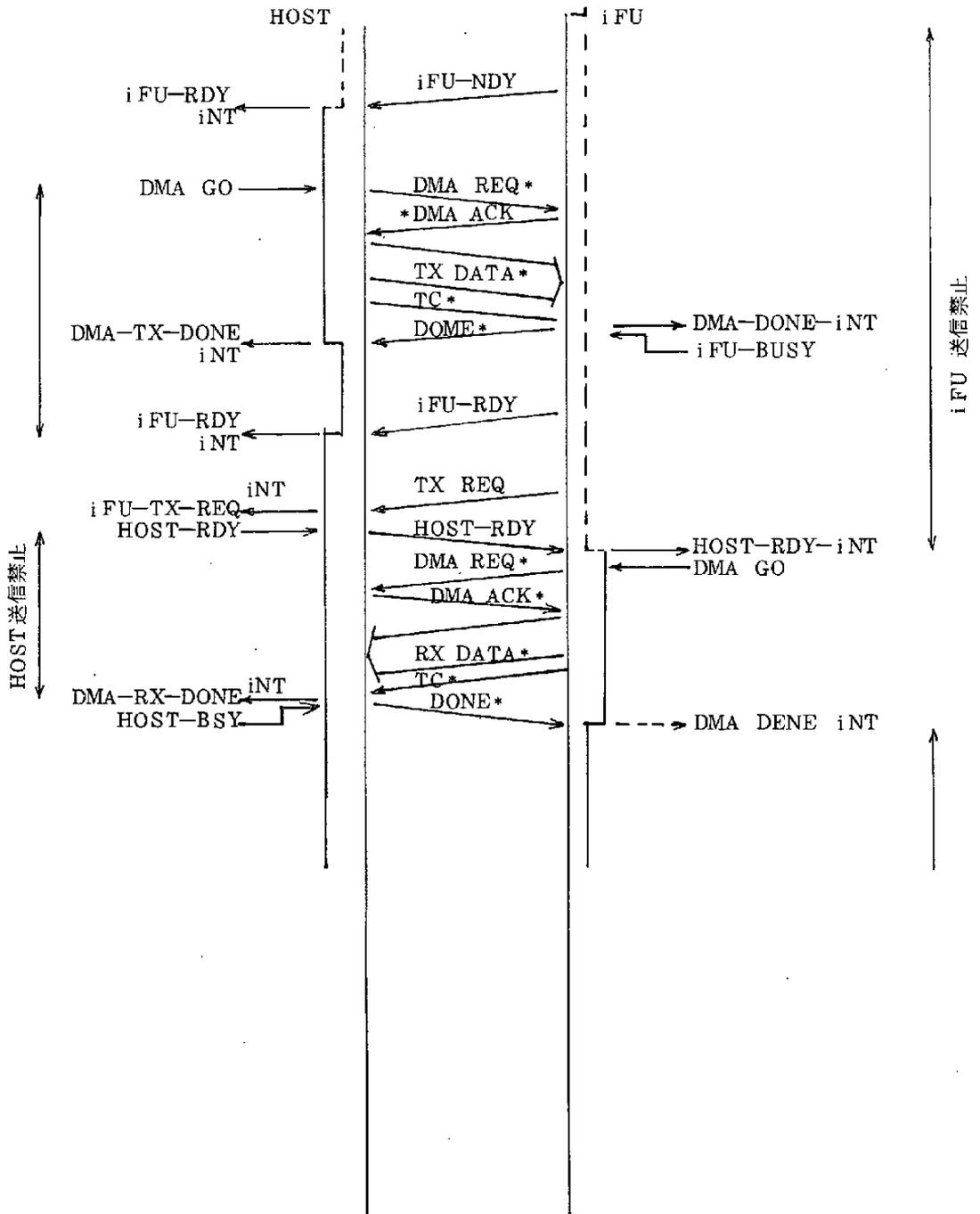


iFUシステムからの受信要求を待つ。

受信用チャンネルをENABLEとする。

DMAの終了を確認する。

ホストの iFUからの受信手順



DMA 制御手順



## 5. コマンドオペレーション

### 5.1 コマンドの構成

本ユニットが受け付けるコマンドの構成を図5-1に示す。

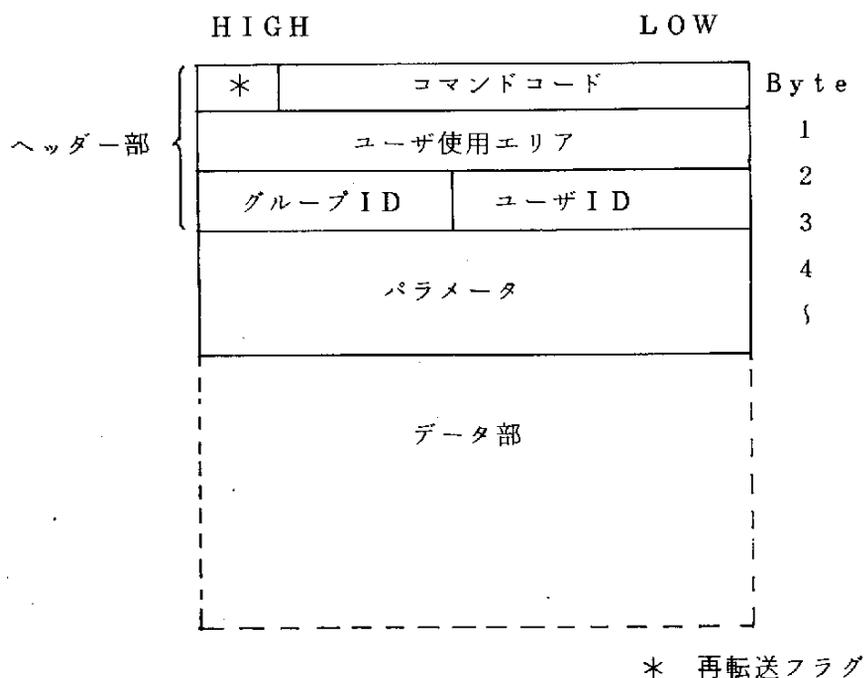


図5-1 コマンドの構成

#### 5.1.1 ヘッダー部

ヘッダー部はコマンドの識別、ユーザの識別のための情報を持ち、3バイトで構成する。

##### (1) コマンドコード

7ビットの各コマンドである。5-3項に、各コマンドのコードを示す。

##### (2) 再転送フラグ

ビットONの時、転送上のエラーによる再転送データであることを示す。

(3) ユーザ使用エリア

コマンドを多重に処理する場合、ユーザがコマンドと処理結果を対応させるために使用するエリアである。

(4) ユーザID、グループID

ホスト側のユーザを識別し、ファイルの所有者の識別、ファイルの保護に使用する。各4ビットで1～15の値。

(3)、(4)はユーザにより管理される。

### 5.1.2 パラメータ部

パラメータ部は、各コマンド毎に必要なパラメータを指定するエリアである。

(1) ファイルに関するコマンドのパラメータ

図5-2にその構成を示す。

HIGH		LOW	Byte
オープンファイルID			1
ファイルタイプ			2
アクセスタイプ	アクセスモード		3
保護コード	(LOW)		4
	(HIGH)		5

図5-2 パラメータ部の構成

① オープンファイルID

ファイルをオープン、または通常ファイルを作成すると、本ユニットはそのファイルに対応するオープンファイルIDを戻す。ユーザはこのコードを使用して、そのファイルをリード/ライト、クローズする。

② ファイルタイプ

対象のファイルのファイルタイプを指定する。

0…通常ファイル

1…ディレクトリファイル

2…特殊ファイル

③ アクセスモード

ファイルへのアクセスのモードを指定する。

1…READ only

2…WRITE only

3…READ/WRITE

④ アクセスタイプ

ファイルへのアクセスタイプを指定する。

0…専有

1…専有(クローズ時にファイルを消去する)

2…共有

⑤ 保護コード

ファイルの作成やコピーの時に、そのファイルの保護コードを指定する。

0の時、本ユニットは決められた値を使用する。(3.1.3(6))

(2) 本ユニットに関するコマンドのパラメータ

図5-3に示すように1バイトで構成し、0とする。

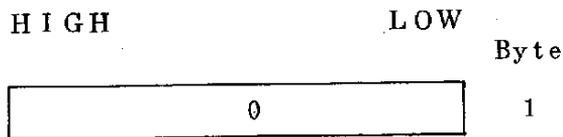


図5-3 パラメータ部の構成

### 5.1.3 データ部

コマンドによりデータが必要な場合に使用する。例えばライトコマンドのデータである。

このデータ部のサイズは、システムジェネレーションにより指定できる。

## 5.2 処理結果の構成

本ユニットがホスト側に戻す処理結果の構成を図5-4に示す。

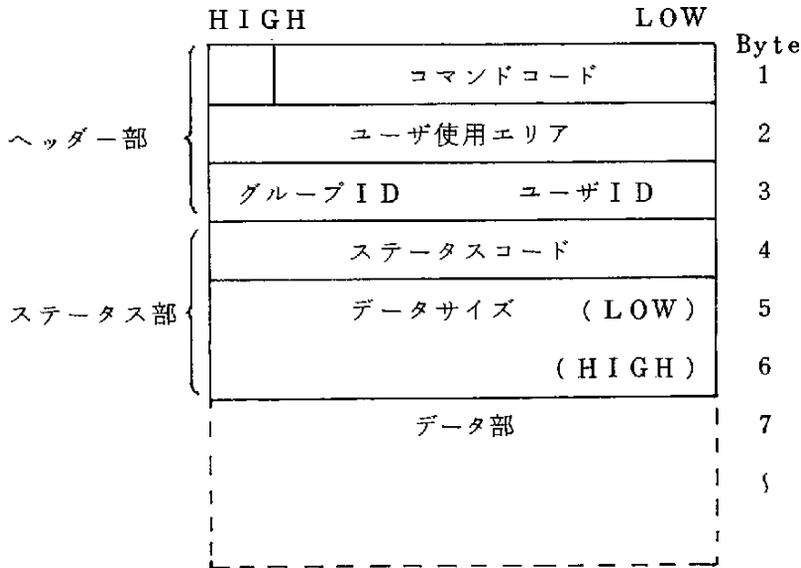


図5-4 処理結果の構成

### 5.2.1 ヘッダー部

コマンドに指定されたヘッダー部をそのままホストに戻す。構成は5.1.1と同じである。

### 5.2.2 ステータス部

#### (1) ステータスコード

コマンドの処理結果を示すコードである。5-3項に各コードを示す。

#### (2) データサイズ

コマンドにより、データがある場合、そのサイズを示す。0の時、データがないことを示す。単位はバイトである。ただし、ステータスコードがエラーを示す場合は0で、データ部はない。

### 5.2.3 データ部

リードコマンドのようにデータを戻す場合使用する。サイズは、システムジェネレーションにより指定できる。

## 5.3 各コマンドの機能と構成

### 5.3.1 コマンドコード

表 5-1 にコマンドコードを示す。

### 5.3.2 ステータスコード

表 5-2 にステータスコードを示す。

表 5 - 1 コマンドコード

コード (HEX)	コマンド名称
1	ファイルの作成
2	" 消去
3	" コピー
4	" オープン
5	" クローズ
6	ファイルへのライト
7	ファイルのリード
8	ボリュームの初期化
9	特殊ファイルのマウント
A	" ディスマウント
B	ファイル名リストのリード
C	ユーザの登録 (カレントディレクトリーのセット)
D	ユーザのカレント ディレクトリーのリード
E	ユーザ登録の消去
F	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
1A	
1B	
1C	
1D	
1E	
1F	

コード (HEX)	コマンド名称
20	
21	ファイルのインデックスのリード
22	ファイルのインデックスの変更
23	ディスク・ボリュームの使用状況のリード
24	
25	
26	
27	
28	
29	
2A	
2B	
2C	
2D	
2E	
2F	
40	診断要求
41	時間のセット
42	時間のリード
43	ユニットステータスのリード
44	
45	
46	
47	
48	
49	
4A	
4B	
4C	
4D	
4E	
4F	

表 5 - 2 ステータスとコード

Decimal	Hex-Decimal	Meaning
0	0 0 H	
1	0 1 H	Success
2	0 2 H	↑
		未使用
		↓
1 2 7	7 F H	

Decimal	Hex-Decimal	Meaning
- 1	0 FFH	Open File exist
- 2	0 FEH	Already used file name
- 3	0 FDH	Already file opened
- 4	0 FCH	No such file
- 5	0 FBH	File not opened
- 6	0 FAH	Already file closed
- 7	0 F9H	
- 8	0 F8H	
- 9	0 F7H	
- 10	0 F6H	
- 11	0 F5H	
- 12	0 F4H	
- 13	0 F3H	
- 14	0 F2H	
- 15	0 F1H	
- 16	0 F0H	
- 17	0 EFH	End of file
- 18	0 EEH	End of volume
- 19	0 EDH	End of index file
- 20	0 ECH	End of directory
- 21	0 EBH	
- 22	0 EAH	
- 23	0 E9H	
- 24	0 E8H	
- 25	0 E7H	
- 26	0 E6H	
- 27	0 E5H	

Decimal	Hex-Decimal	Meaning
- 2 8	0 E 4 H	
- 2 9	0 E 3 H	
- 3 0	0 E 2 H	
- 3 1	0 E 1 H	
- 3 2	0 E 0 H	
- 3 3	0 D F H	Device off-line
- 3 4	0 D E H	Bad block on device
- 3 5	0 D D H	Device already mounted
- 3 6	0 D C H	Device other user mounted
- 3 7	0 D B H	Device not mounted
- 3 8	0 D A H	
- 3 9	0 D 9 H	
- 4 0	0 D 8 H	
- 4 1	0 D 7 H	
- 4 2	0 D 6 H	
- 4 3	0 D 5 H	
- 4 4	0 D 4 H	
- 4 5	0 D 3 H	
- 4 6	0 D 2 H	
- 4 7	0 D 1 H	
- 4 8	0 D 0 H	
- 4 9	0 C F H	Bad command code
- 5 0	0 C E H	Bad owner code
- 5 1	0 C D H	Bad open file iD number
- 5 2	0 C C H	Bad file type
- 5 3	0 C B H	Bad access mode
- 5 4	0 C A H	Bad access type

Decimal	Hex-Decimal	Meaning
- 5 5	0 C 9 H	Bad protection code
- 5 6	0 C 8 H	Bad data address
- 5 7	0 C 7 H	Bad data size
- 5 8	0 C 6 H	Bad file name
- 5 9	0 C 5 H	Bad time parameter
- 6 0	0 C 4 H	Bad diag parameter
- 6 1	0 C 3 H	
- 6 2	0 C 2 H	
- 6 3	0 C 1 H	
- 6 4	0 C 0 H	
- 6 5	0 B F H	Locked by read protection
- 6 6	0 B E H	Locked by write protection
- 6 7	0 B D H	Locked by delete protection
- 6 8	0 B C H	Other file exist on directory
- 6 9	0 B B H	Current user not authorized
- 7 0	0 B A H	Cannot open for table full
- 7 1	0 B 9 H	Cannot close active file
- 7 2	0 B 8 H	Check sum error
- 7 3	0 B 7 H	
- 7 4	0 B 6 H	
- 7 5	0 B 5 H	
- 7 6	0 B 4 H	
- 7 7	0 B 3 H	
- 7 8	0 B 2 H	
- 7 9	0 B 1 H	
- 8 0	0 B 0 H	
- 8 1	0 A F H	

Decimal	Hex-Decimal	Meaning
- 8 2	0 A E H	
- 8 3	0 A D H	
- 8 4	0 A C H	
- 8 5	0 A B H	
- 8 6	0 A A H	
- 8 7	0 A 9 H	
- 8 8	0 A 8 H	
- 8 9	0 A 7 H	
- 9 0	0 A 6 H	
- 9 1	0 A 5 H	
- 9 2	0 A 4 H	
- 9 3	0 A 3 H	
- 9 4	0 A 2 H	
- 9 5	0 A 1 H	
- 9 6	0 A 0 H	
- 9 7	9 F H	
- 9 8	9 E H	
- 9 9	9 D H	
- 1 0 0	9 C H	
- 1 0 1	9 B H	
- 1 0 2	9 A H	Command error end
- 1 0 3	9 9 H	
- 1 0 4	9 8 H	Time out error( E. PH int. )
- 1 0 5	9 7 H	Too long time C. PH
- 1 0 6	9 6 H	Too long time R. PH
- 1 0 7	9 5 H	Time out error ( R. PH int. )
- 1 0 8	9 4 H	

Decimal	Hex-Decimal	Meaning
- 1 0 9	9 3 H	
- 1 1 0	9 2 H	
- 1 1 1	9 1 H	
- 1 1 2	9 0 H	Device not ready
- 1 1 3	8 F H	
- 1 1 4	8 E H	
- 1 1 5	8 D H	
- 1 1 6	8 C H	
- 1 1 7	8 B H	
- 1 1 8	8 A H	
- 1 1 9	8 9 H	
- 1 2 0	8 8 H	
- 1 2 1	8 7 H	
- 1 2 2	8 6 H	
- 1 2 3	8 5 H	Invalid block count
- 1 2 4	8 4 H	Invalid block No.
- 1 2 5	8 3 H	Invalid i/o function
- 1 2 6	8 2 H	
- 1 2 7	8 1 H	Undefined device No.
- 1 2 8	8 0 H	Undefined logical unit No.

### 5.3.3 ファイルに関するコマンド

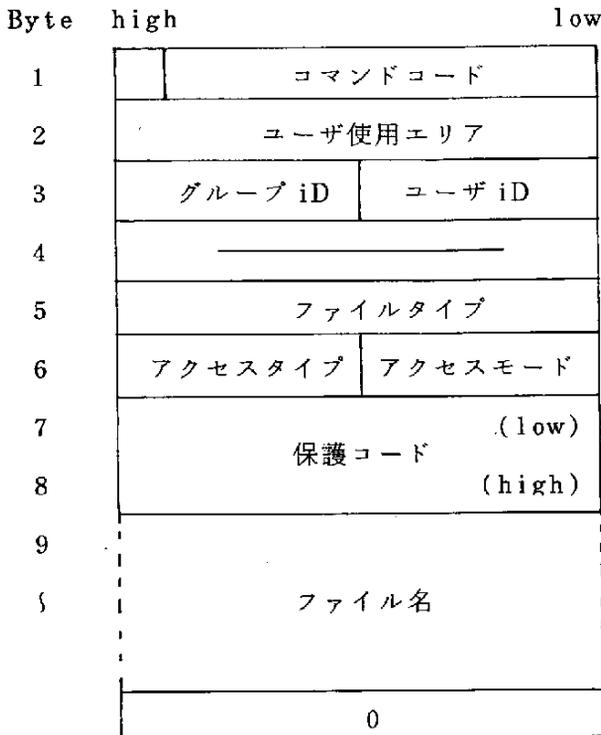
#### (1) ファイルの作成

##### ① 機能

ディレクトリ・ファイル、通常ファイルの作成を行なう。ディレクトリファイルの場合アクセスモード、アクセスタイプは無視する。ユーザにはステータスのみを返す。通常ファイルの場合にはファイルはオープン状態にする。ユーザにはオープンファイル iD が戻され、以後のライト又はリードコマンドに使用する。

ファイル名指定において、経路を示すディレクトリファイルは全て作成していなければならない。又既に指定ファイルが作成されている場合はエラーとなる。

##### ② 構成



③ 処理結果

Byte	high	low
1	コマンドコード	
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5	データサイズ ( 0 or 1 )	
6	" ( 0 )	
(7)	オープンファイル	

ディレクトリファイル作成の場合 6 Byte で構成する。

通常ファイル作成の場合 7 Byte で構成し、7 Byte 目にオープンファイル iD が入る。

(2) ファイルの消去

① 機能

指定するファイルを消去する。

ディレクトリファイルを消去する場合は、そのディレクトリファイルに \$ \$ ( 親ディレクトリファイル )、\$ ( 自分自身 ) 以外のファイルが登録されていない。

② 構成

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	_____	
5	ファイルタイプ	
6	_____	
7		
8		
9	_____	
{		
	ファイル名	
	_____	
	0	

③ 処理結果

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5	_____	
6		
	0	

(3) ファイルのコピー

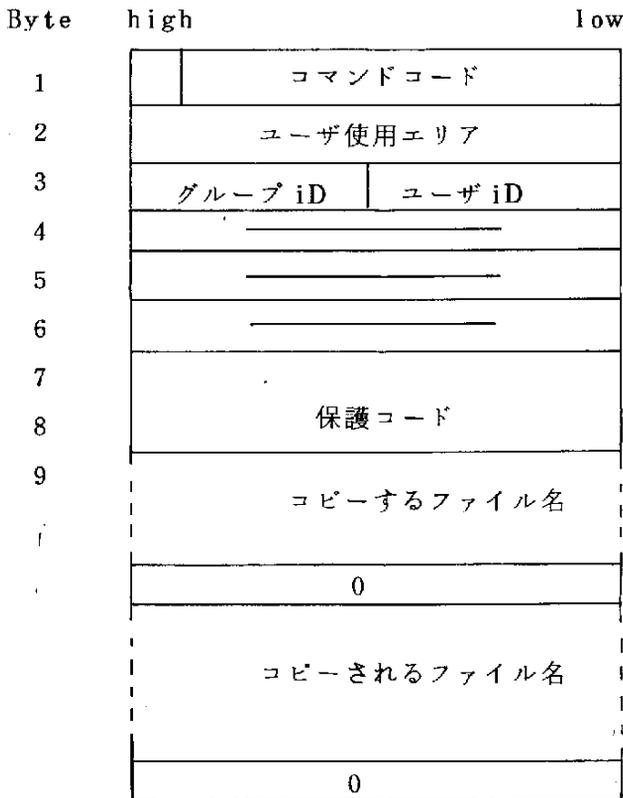
① 機能

ファイルを他のディレクトリファイル下へコピーする。

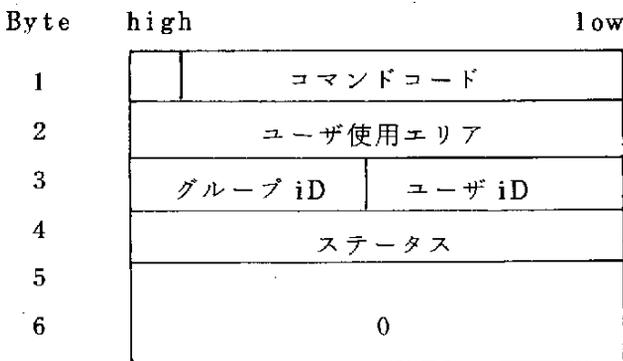
保護コードはコピーされるファイルに適用される。コピーされるファイルの指定は経路だけでなく最後のファイル名までとする。

ファイルは通常ファイルを対象とするが、ディレクトリファイルのコピーの場合、コピーされたファイルは通常ファイルとなる。

② 構成



③ 処理結果



(4) ファイルのオープン

① 機能

指定する通常ファイルをオープンし、リード/ライトの開始を宣言する。ディレクトリファイル、特殊ファイルも同様にオープンする。指定するファイルが複数のユーザにアクセスされる場合は、アクセスタイプが共有でなければならない。ファイルオープンエラー時には、オープンファイル iD が戻されない。

② 構成

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	_____	
5	ファイルタイプ	
6	アクセスタイプ	アクセスモード
7	_____	
8		
9	ファイル名	
	0	

③ 処理結果

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5	サイズ (1)	
6	オープンファイル iD	
7		

(5) ファイルのクローズ

① 機能

オープンされているファイルをクローズし、アクセスの終了を宣言する。  
通常ファイルの場合、作成又はオープン時にアクセスタイプが“1”の場  
合はファイルを消去する。

指定ファイルが同一ユーザによりオープンされていなければならない。

② 構成

Byte	high	low
1	コマンドコード	
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	オープンファイル iD	
5	_____	
6	_____	
7	_____	
8		

③ 処理結果

Byte	high	low
1	コマンドコード	
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5		
6	0	

(6) ファイルへのライト

① 機能

指定するファイルにデータを書き込む。

データが書かれるファイルのアドレスの指定は次のように扱う。

特殊ファイルの場合

アドレス=0 →ファイルの最後に書く。(カレント)

アドレス=1 →ファイルの先頭に書く。

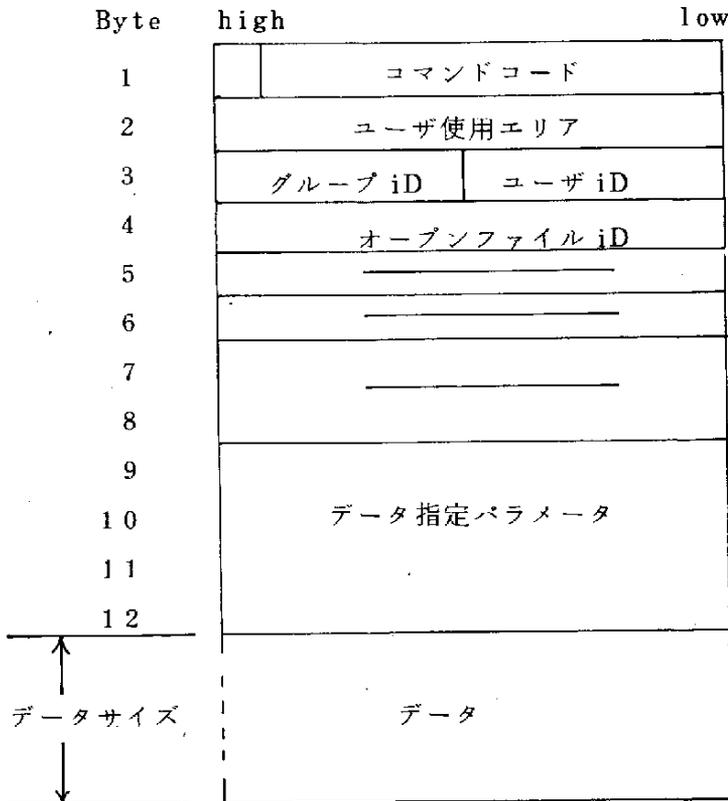
アドレス=n →ファイルのn番目のブロックに書く。

通常ファイルの場合

アドレス=0 →ファイルの最後に書く。

アドレス≠0 →アドレス指定を無視して“0”として扱う。

## ② 構成



③ 処理結果

Byte	high	low
1	コマンドコード	
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5		
6	0	

④ データ指定パラメータの構成

a) ファイルが特殊ファイル(ディスク、F/D)の場合

Byte	high	low
9	データアドレス	
10	(ブロック#)	
11	データサイズ(ブロック数)	
12		

アドレスは特殊ファイルのフィジカル・ブロック・ナンバー (ex. 1, 2, ……) サイズは 512 バイトのブロック数

b) ファイルが上記以外(ex. 通常ファイル)の場合

Byte	high	low
9	データアドレス	
10	(ブロック#)	
11	データサイズ	
12	(バイト数)	

そのファイルのブロック#(ex. 1, 2, ……) サイズはバイト数

(7) ファイルのリード

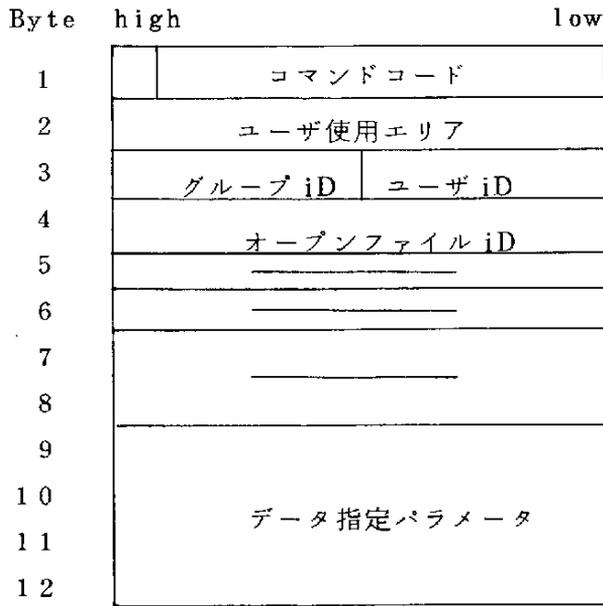
① 機能

指定するファイルのデータを読み出す。

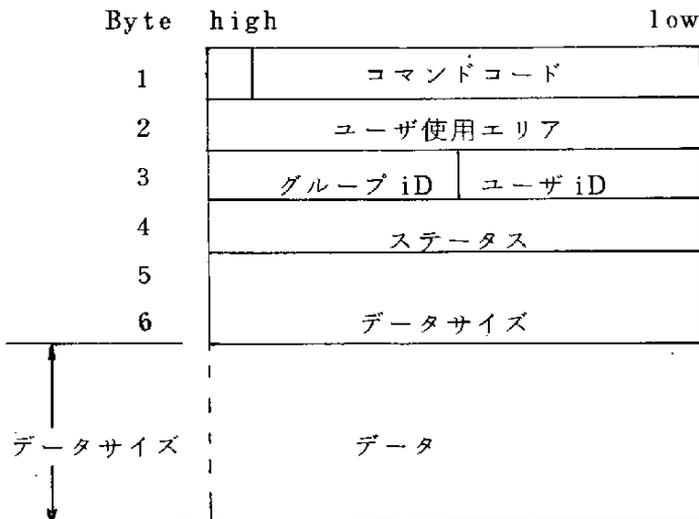
アドレス、サイズはライトコマンドと同じである。

アドレス=0は、ファイルの先頭から読み出し、以後読み出されたアドレスの次のアドレスから読んでいく。

② 構成



③ 処理結果



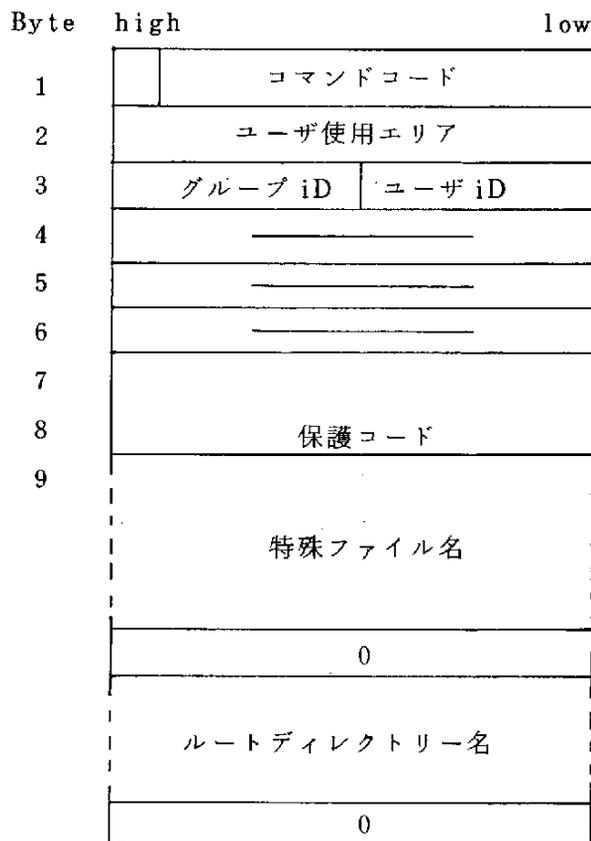
(8) ボリュームの初期化

① 機能

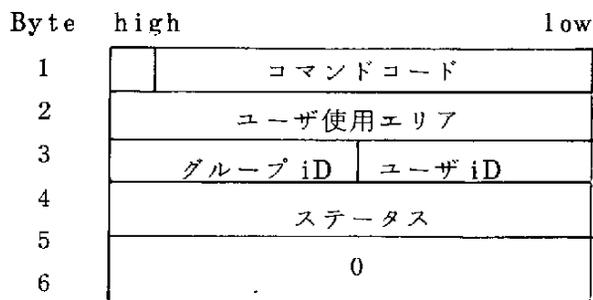
ディスクの初期化を行なう。

不良セクタのチェック、フォーマットを行った後、ルート・ディレクトリファイルの登録など、ファイル管理に必要なデータの書き込みを行う。

② 構成



③ 処理結果



(9) 特殊ファイルのマウント

① 機能

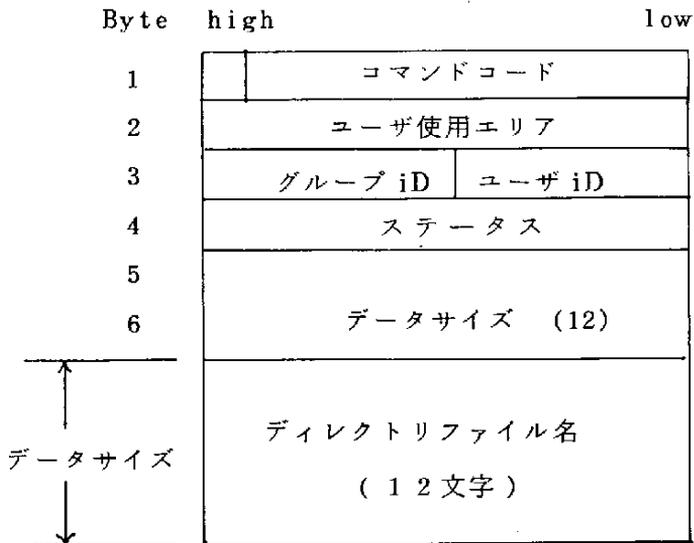
ファイル構造を持つボリュームを本ユニットのファイル管理機能の管理下に置く。

アクセスモード、タイプの指定によりディスクボリューム単位の専有、ライトに対する保護が可能である。ディスクボリュームを複数のユーザがアクセスする場合、アクセスモード、タイプは最初のユーザの指定が適用される。又、アクセスする全てのユーザが共有タイプでマウントしなければならない。

② 構成

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	_____	
5	ファイルタイプ (=特殊ファイル)	
6	アクセスタイプ	アクセスモード
7	_____	
8		
9	特殊ファイル名	
	0	

③ 処理結果



10 ディスマウント

① 機能

マウントされているボリュームを本ユニットのファイル管理機能から切離す。ディスマウントするボリュームのファイルが全てクローズされているなど、そのボリュームへのアクセスが全て完了している状態とする。尚、共有で複数のユーザがマウントしている場合各ユーザがそれぞれディスマウントする。

② 構成

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	_____	
5	_____	
6	_____	
7	_____	
8		
9	特殊ファイル名	
	0	

③ 処理結果

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5		
6	0	

(1) ファイル名リストのリード

① 機能

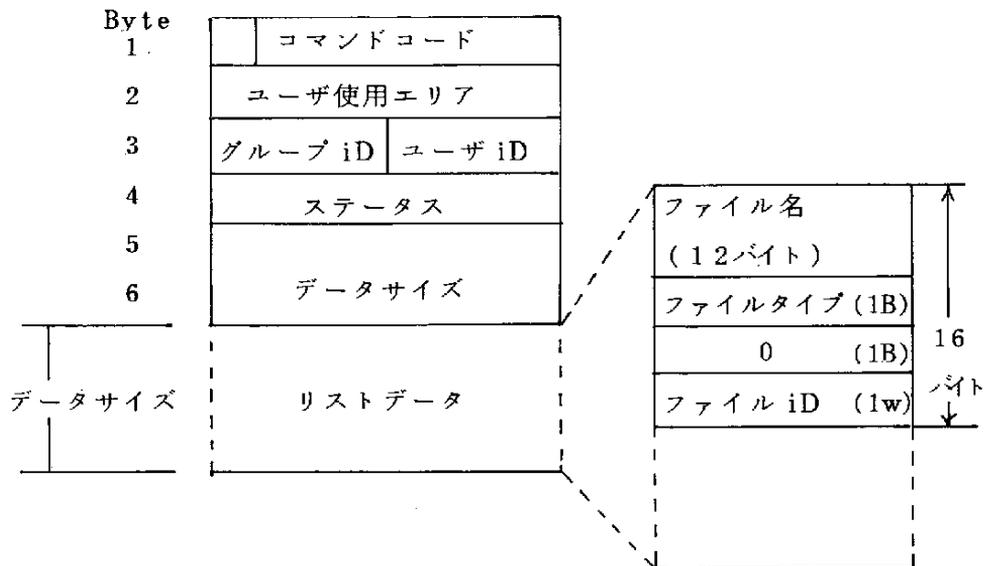
指定されたディレクトリファイルに登録されている全てのファイルのファイル名をリードする。

ディレクトリファイルへのリードが保護コードにより許可されているユーザは使用できる。

② 構成

Byte	high	low
1	コマンドコード	
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	_____	
5	_____	
6	_____	
7	_____	
8	_____	
9	ディレクトリファイル名	
	0	

③ 処理結果



(12) ユーザカレントディレクトリのセット

① 機能

ユーザの登録およびユーザのディレクトリファイルの指定をする。このユーザがファイルを指定する場合、経路名の指定がなければ、このディレクトリファイルに登録されているファイルが対象となる。

同一のユーザ iD、グループ iD を持つ複数のユーザが同時に存在してはならない。

② 構成

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	_____	
5	_____	
6	_____	
7	_____	
8		
9	ディレクトリファイル名	
	0	

③ 処理結果

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5		
6	0	

13 ユーザのディレクトリのリード

① 機能

ユーザの現在のディレクトリファイル名のリード

② 構成

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	_____	
5	_____	
6	_____	
7	_____	
8		

③ 処理結果

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5	サイズ (12)	
6	ディレクトリファイル名	
7		
18		

14 ユーザ登録の消去

① 機能

ユーザ登録をまったく消す。以後ユーザはファイルをルートディレクトリ

からの経路による指定で参照する。

② 構成

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	_____	
5	_____	
6	_____	
7	_____	
8		

③ 処理結果

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5		
6	0	

(15) ファイルのインデックスのリード

① 機能

指定するファイルのインデックスのデータを読み出す。ファイルへのリードが許可されているユーザは使用できる。

通常ファイル、ディレクトリファイルのみ有効である。

② 構成

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	_____	
5	_____	
6	_____	
7	_____	
8		
9	ファイル名	
	0	

③ 処理結果

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5		
6	サイズ ( 256 )	
	↑ サイズ ↓	
	インデックスデータ	

(19) ファイルのインデックスの変更

① 機能

指定するファイルの保護コード又はファイル名を変更する。保護コード

は“0”を指定する場合変更しない。ファイル名は変更するファイル名を指定する(最大12文字)ファイルへのライトが許可されるユーザが使用できる。通常ファイル、ディレクトリファイルのみ有効である。

② 構成

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4		
5		
6		
7		
8	保護コード	
9	ファイル名	
	0	
	変更するファイル名	
	0	

③ 処理結果

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5		
6	0	

(17) ディスクボリュームの使用状況のリード

① 機能

指定するボリュームの使用状況のリード。

次の様な情報を読み出す。

イ) ボリュームの総ブロック数

ロ) 未使用ブロック数

指定するディスクボリュームはマウントされていなければならない。

② 構成

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	_____	
5	_____	
6	_____	
7	_____	
8		
9	特殊ファイル名	
	0	

③ 処理結果

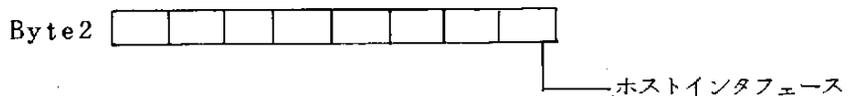
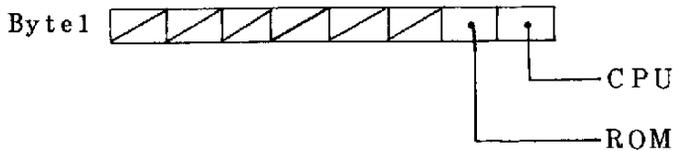
Byte	high	low
1	コマンドコード	
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5	総ブロック数	
6		
7		
8		
9	未使用ブロック数	
10		
11		
12		
13		
14		

5.3.4 本ユニットに関するコマンド

(1) 診断要求

① 機能

- 指定した装置の診断を行なう。
- 装置の指定は次のような各ビットに対応する。(4 Byteで構成)



Byte3、4は特殊ファイルの順でビット0から存在する数だけ指定。

- 本コマンドは一般にいつでも実行可能とする。

② 構成

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	0	
5	診 断 対 象	
6		
7		
8		

③ 処理結果

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5	サイズ (4)	
6		
7	診 断 結 果	
8		
9		
10		

診断結果 異常検出時はステータスにエラーを示し、機能①の診断対象と同じビットの対応で異常のあった対象の bit を ON にする。

(2) 時間のセット

① 機能

- 本ユニットの時間を更新する。
- 本ユニットは内部で時間の管理を行なう。従って、本ユニット起動時に

一度行ない、以後は停止まで本ユニットが更新する。

② 構成

Byte	high	low
1	コマンドコード	
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	0	
5	時間の値 (年)	
6	----- (月)	
7	----- (日)	
8	----- (時)	
9	----- (分)	
10		

③ 処理結果

Byte	high	low
1	コマンドコード	
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5		
6	0	

(3) 時間のリード

① 機能

- 本ユニットの時間（コマンド実行時）を読み出す。
- 本ユニット起動時の時間は、本ユニットが持つ初期値であるため、一度時間をセットしなければ、不正確な時間の値が読み出される。

② 構成

Byte	high	low
1	コマンドコード	
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	0	

③ 処理結果

Byte	high	low
1	コマンドコード	
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5	サイズ (6)	
6		
7	時間の値 (年)	
8		
9	(月)	
10	(日)	
11	(時)	
12	(分)	

(4) ユニット・ステータスのリード

① 機能

- ・本ユニットの内部状態を得る。
- ・状態とは次の様な各情報とする。

1. ホストからのコマンド数

2. ホストとの転送エラー、タイム・アウトエラーの数
3. 各入出力装置への i/o の数
4. 各入出力装置との i/o 時のエラー数
5. オープン中のファイル数
6. システム・ジェネレーション・パラメータ
  - ・最大オープン・ファイル数
  - ・最大転送データ・サイズ
  - ・最大コマンド登録数
7. 最新の診断結果

以上の項目中 1～4はユニット起動時からの総数とし、  
5～7は本コマンドの実行時の状態とする。

② 構成

Byte	high	low
1	コマンドコード	
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	0	

③ 処理結果

Byte	high	low
1		コマンドコード
2	ユーザ使用エリア	
3	グループ iD	ユーザ iD
4	ステータス	
5		
6	サイズ	
7		
8	最大転送データサイズ	
9	(バイトカウント)	
10		
11	最大オープンファイル数	
12	オープン中のファイル数	
13	最大コマンド登録数	
14	RAMサイズ (Kバイト)	
15		
16	診断結果の情報	
17		
18	ホストからのコマンド数	
19		
20	ホストとのエラー数	
21		
	特殊ファイル 1 に対応する装置への i/o 数	
	同上 i/o エラー数	
	特殊ファイル 2 に対応する装置への i/o 数	
	同上 i/o エラー数	



ホスト数分



特殊ファイル数分

## 6. システムジェネレーション

システムジェネレーションは、本ユニットの機能をその構成に応じて最適化するための機能である。

### 6.1 パラメータ

(1) ホストの数

本ユニットを接続するホストの数を定める。

(2) コマンド登録数

ホストが処理の完了を待つことなく、本ユニットに書き込むことができるコマンドである。

(3) 最大データサイズ

リードまたはライトコマンドで指定できるデータ部分の最大サイズである。この値は、またファイルリストのリードコマンドの処理結果におけるリストデータのサイズにも適用される。

(4) 同時オープンファイル数

本ユニットが同時に管理できるオープン中のファイル数である。同一のファイルを複数のユーザがオープンする場合は、おのおのを1つのオープン中のファイルとして扱われる。

(5) 入出力装置の数

本ユニットが管理する入出力装置の構成を指定する。

① 入出力装置名

② ユニット数

入出力装置は、表3-1で示されるものが有効である。

(6) メモリ ( R A M ) サイズ

本ユニットが使用できるメモリ ( R A M ) のサイズである。

本ユニットに必要なメモリのサイズは

本ユニット固有のワークエリア ( 1.4 K B )

+各パラメータによる拡張エリア

により定まる。

## 6.2 システムジェネレーションの方法

### 6.2.1 パラメータの設定

表 6 - 1 にパラメータを示す。

表 6-1 システムジェネレーションパラメータ

パラメータ	最小値	最大値	デフォルト	1単位当り必要なメモリサイズ	備 考
1. 最大転送データサイズ(P1)	512バイト	1024バイト	1024バイト		512又は1024バイトのみ設定可能。
2. ホスト数	1	4	1	46バイト+P1	
3. 最大コマンド登録数	1	~	8	32バイト+P1×2	最大値はメモリサイズにより定まる。
4. 入出力装置					
(1) DK(ハードディスク)					
コントローラ数	1	1	1	—————	
ユニット数/1コントローラ	1	4	1	208バイト	
(2) FD(フロッピーディスク)					
コントローラ数	1	1	1	—————	
ユニット数/1コントローラ	1	4	1	208バイト	
5. 最大オープンファイル数	1	10	8	—————	6のメモリサイズには影響しない。
6. メモリサイズ	32KB	64KB	32KB	—————	

## 6.2.2 手 順

システムジェネレーションの手順フローチャートを図6-1に示す。

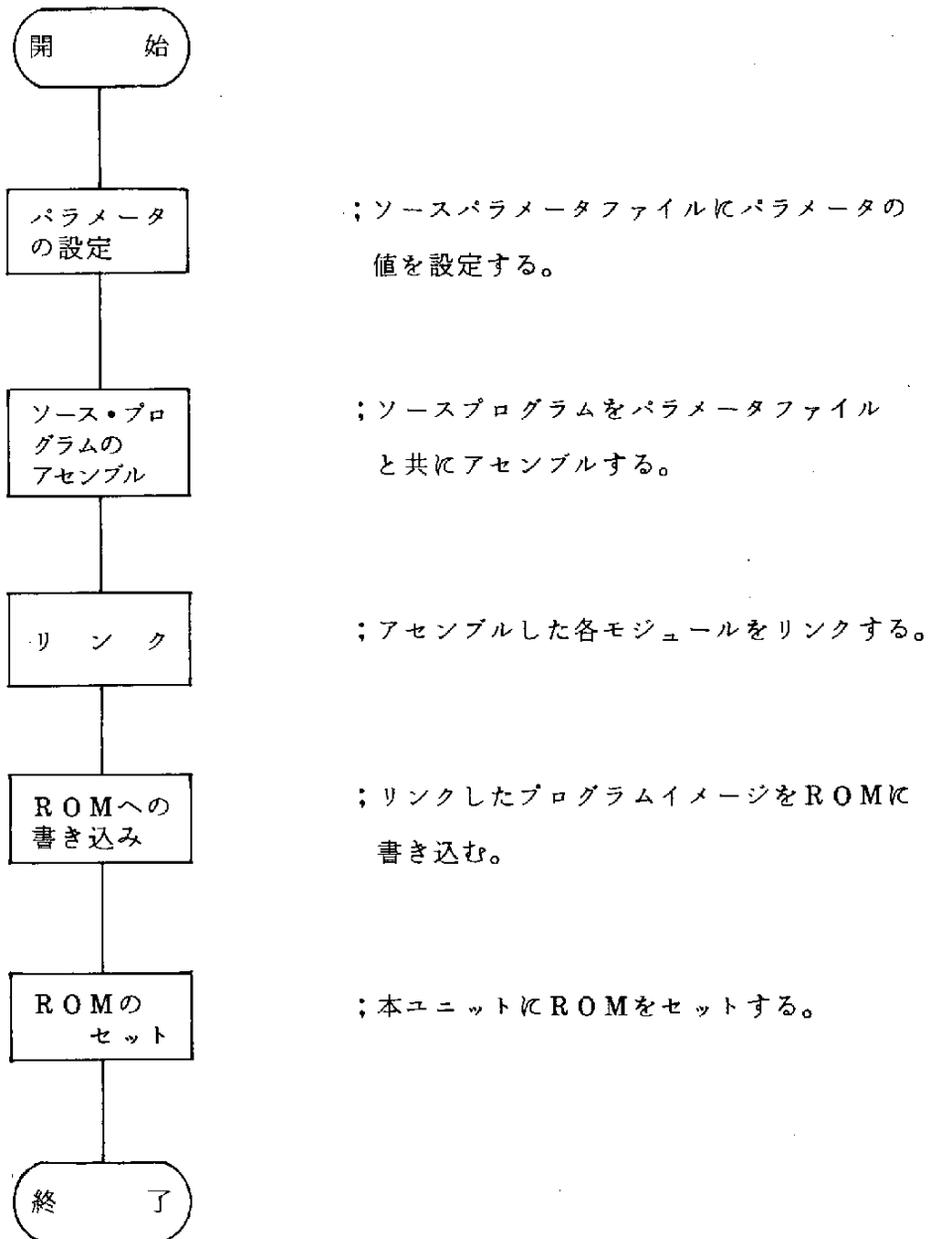


図6-1 システムジェネレーションの手順

## 7. ホストとの接続例

### 7.1 構成

パーソナルコンピュータPC-8800シリーズにインテリジェントディスクファイル管理ユニットを接続するために、専用アダプタを用い、図1に示すように構成する。すなわち、PC8801の拡張バススロットにアダプタを介して、インテリジェントディスクファイル管理ユニットを接続することによって、記憶容量1Mバイトのフロッピーディスクおよび記憶容量20Mバイトのハードディスクをサポートし、さらにこれらのファイル管理を可能にしている。

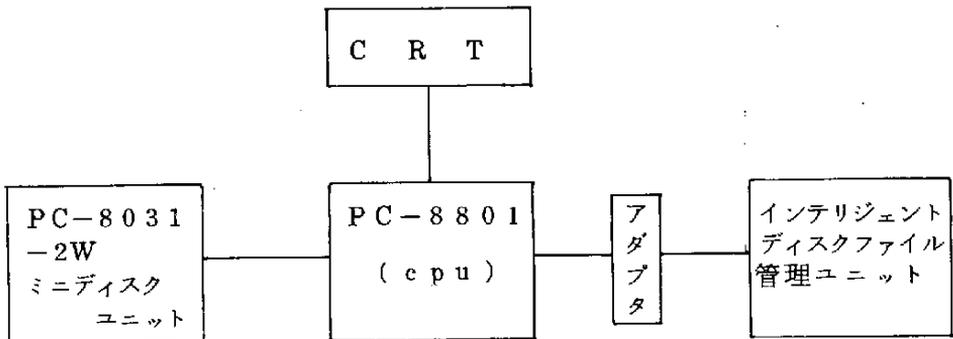


図7-1 PC-8800シリーズとインテリジェントディスクファイル管理ユニットの接続

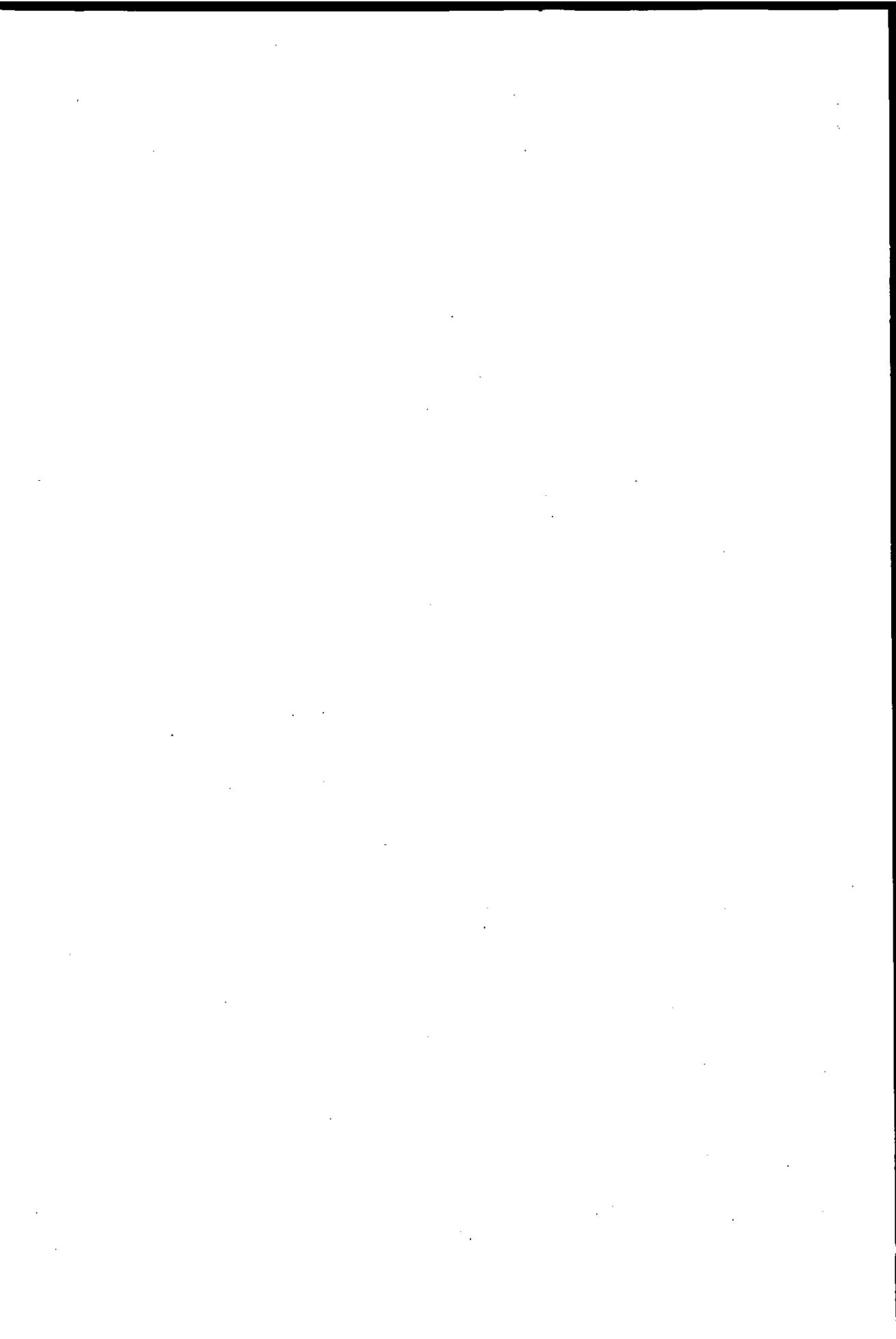
### 7.2 準備

- ① PC-8801を背面のディップスイッチによりN-BASICモードにする。
- ② PC-8801の拡張バススロットにアダプターボードをさしこむ。

- ③ インテリジェントディスクファイル管理ユニット背面パネルのHOST  
の位置へコネクタを接続する。
- ④ ③のケーブルのもう一方のコネクタをアダプターボード上のコネクタに  
さしこむ。

### 7.3 制 御 手 続

制御手順は4章を参照のこと。



禁無断転載

昭和58年3月発行

発行所 財団法人日本情報処理開発協会  
東京都港区芝公園3-5-8  
機械振興会館内

TEL(434)8211(代表)

印刷所 株式会社昌文社  
東京都港区芝5-26-30  
TEL(452)4931(代表)

