



Title	ACOSのファイル
Author(s)	
Citation	大阪大学大型計算機センターニュース. 1977, 24, p. 33-63
Version Type	VoR
URL	https://hdl.handle.net/11094/65336
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

ACOSのファイル

0. 概 要

I. ファイルシステム

- 1) ファイル構造 (カタログ構造)
- 2) カタログ／ファイル
- 3) 管理保護と共同利用
- 4) ACCESSサブシステム

0. 概要

ACOS-6 システムでは、T.S.S. に限らず他の処理形態（ローカル・バッチ処理，リモートバッチ処理，会話型リモートバッチ処理，等）からでも，自由に作成およびアクセスできるファイル・システムが提供されます。ここでファイルとはデータの集合を入れておく論理的な媒体と考えていただきます。このファイルには，次の2種類のファイルに分けられます。ⅰ）パーマネントファイルとⅱ）テンポラリファイル：ⅰ）は利用者によってデータの出入れを行い，ⅱ）はシステムが管理するファイルです。以下の説明では，まずパーマネントファイルの説明を行い，後にテンポラリファイルの説明をT.S.S. を中心に行います。

I ファイル・システム（パーマネントファイル）

1）ファイル構造（カタログ構造）

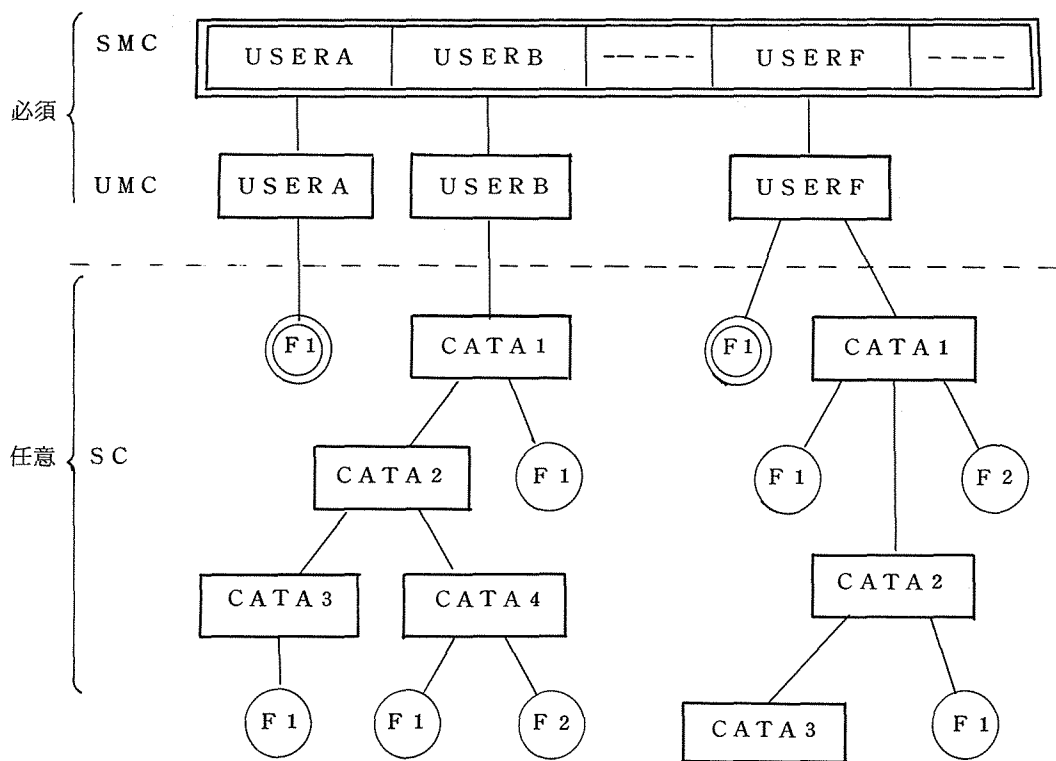
ACOS-6 のファイル・システムでのファイル構造はSMC（システム・マスタカタログ）を根とし，その下位レベルにUMC（ユーザ・マスタ・カタログ），SC（サブ・カタログ）そしてファイル記述，及びファイル・スペースの木構造を成しています。深さはUMCより数えて8レベルまで可能です。（図-1）

① システム・マスタ・カタログ（SMC）

木構造の最高位に位置しており，以下の内容です。

- SMC名（利用者の課題番号）
- UMCのアドレス
- 利用者の使用可能な最大ブロック数（Lリンク数）
- 利用者のログオン・パスワード
- 許されているサブシステム（T.S.S. 時）
- 許されているリソースの最大値（T.S.S. 時）
- リソースが使われた日付（T.S.S. 時）
- （スペース）×（時間）の量

利用者は課題申請を行いますと，センター側ではこのSMCをシステムに登録しておきます。もしバッチ処理でもパーマネントファイルを使用する場合（もちろんT.S.S. においても）はファイル利用申請書を添えて下さい。



SMC … システム マスターカタログ

UMC … ユーザ マスターカタログ

SC … サブ カタログ

□ … カタログ

◎ … クイック アクセス ファイル

○ … ファイル

(図 - 1)

② ユーザ・マスタ・カタログ (U M C)

S M C のすぐ下位の位置に存し、ユーザ・レベルの最高位にあります。下位のサブカタログやファイルを作成する時、自動的に作られます。

③ サブ・カタログ (S C)

U M C と次のレベルのカタログ又はファイル記述の中間に位置する。U M C も、その内容はカタログ名、パスワード・パーミッション等、カタログ属性の情報、そのカタログに従属するファイルと下位のカタログの情報が含まれている。

- パーミッション
- カタログ名 (U M C 名の時もある)
- 作成者の名前
- 作成者の日付
- 最後に変更された日付
- 従属するカタログ又はファイルの名前とアドレス

以上がその内容です。

④ ファイル記述

木構造の末端に位置し、ファイルシステムでのファイルとは、このファイル記述^{※)}を意味します。従って利用者は、ACCESS サブシステムやFILSYS アクティビティなどによって、ファイル記述を作成します。なお、サブ・カタログを経由しないで、U M C のすぐ下にファイルを作ることも可能である。このファイルは、利用者がファイルのSAVE を指定した場合、予め登録されていなくてもクイック・アクセス・ファイルとして登録されます。

- ファイル名
- 作成者の名前
- パスワード
- 作成の日付
- 最後に変更された日付
- 最後にアクセスされた日付
- パーミッション
- ファイルの最大容量
- S M C 名
- ファイルの状態
- ファイルのアドレス
- ファイルの容量

※) 利用者のデータを含んでいて、読み取り書込みが可能な物理的なスペースをもつファイルではない。ファイル (実際のプログラムやデータ) のスペースを確保する為のものと思って下さい。

2) カタログ／ファイル …… (…／…の“／”は“又は”を意味する)

① カタログ名／ファイル名

先のファイル構造でも簡単に述べましたが、カタログはカタログ名、パスワードおよびパーミッションを定義したもので構成され、また、ファイルはファイル名、ファイルのサイズ、パスワード、およびパーミッションなどの定義されたファイル記述であり、実際には読み込みおよび書き込みが行われるプログラムやデータの物理ファイルスペースを確保する為のものであった。それら個々のカタログ／ファイルは名前を付けて識別される(識別子と呼ぶ)。識別子の長さは12文字まで可能であるが、T.S.S. では8文字までに制限される。文字の種類は

- 大文字アルファベット
- 数字(但し全てゼロ(0)で構成される名前はだめです。)
- ピリオド “.”
- マイナス記号 “-”

が使用可能です。

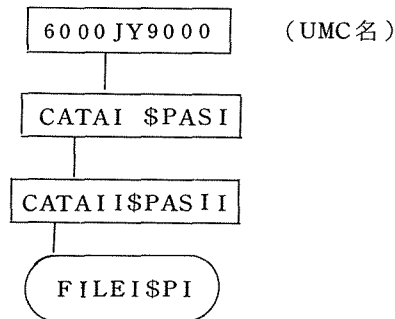
② 修飾名 (filedesc)

ACCESS または FiLSYS を用いてファイルシステムを利用する場合(パーマネントファイル指定する場合)木構造に沿って指定するカタログ／ファイル名に到達するまで、UMC以下のカタログ構造とパスワードを示します。すなわち、カタログを修飾するとい

い、修飾されたカタログ／ファイル名を修飾名と呼びます。

図のように修飾していきます。

ex. UMC名／カタログ名\$パスワード／……………／ファイル名\$パスワード
→ 6000 JY9000 / CATAI \$PAS I / CATAI I \$PAS I I / FILEI \$P I



3) 管理保護と共同利用

利用者ファイルは、各利用者1人1人に、SMCを与えられています。SMCは課題番号とパスワード(ログオンパスワード)を持っていますのである程度機密保護が出来ておりますが、ファイルシステムでは利用者ファイルを木構造とすることにより機密保護を高め、またそれぞれカタログ／ファイルの形態を知らせることにより、ファイルの共同利用が可能になります。そのカタログ／ファイルの形態を指定する為にパスワード及びパーミッションが設けられています。また、ファイルシステムにより、ファイルの必要なスペースは利用者がその大きさを指定するだけで自動的に確保され、割り当てられます。このときの割り当て単位はLリンク、またはリンクであり、1Lリンクは320語(JISコード; 9ビット/バイトなら1280バイト)で、1リンクは12Lリンク=3840語であります。

(注) バッチシステムではソースプログラム／データをファイルに格納する場合、通常BCDコードによるカードイメージ形式ですが、T.S.S.ではこれらがJISコードを用いられます。BCDコードでは12リンクは6ビット/字で1920字、カードで約21枚分、JISコードでは可変長。

① パスワード

カタログ／ファイルにパスワードを設けることにより、それにアクセス権を持たない利用者へのアクセスを防止できます。また逆に共同利用で複数の利用者間でパスワードを知らせることによりそのカタログ／ファイルにアクセスできます。従って、修飾カタログ名／修飾ファイル名の中のカタログ、および、最下位のファイル名に正しいパスワードを指定した時のみ、必要とするカタログ／ファイルに対して処理が可能である。

パスワードは12文字以内で

- 大文字アルファベット
- 数 字
- ピリオド“ . ”
- マイナス記号“ - ”

②- i パーミッション

作成者以外の利用者にもカタログ／ファイルのアクセス操作を可能にする為の機能です。このパーミッションには i) 一般パーミッション、と、 ii) 特定パーミッションがあります。 i) は全ての利用者に対する許可であり(但し、パスワードがあればパスワードを指定したのち)、 ii) は特定の利用者を対象とするものです。また i)、 ii) の排他的関係を示めず、EXCLUDEパーミッションがあります。これは一般パーミッションから特定利用者のみを除外したパーミッションです。なお、カタログ／ファイルの作成者にはそのカタログ／ファイルに対する、すべてのパーミッションが無条件に与えられる。但し、利用する時にはパスワードはすべて指定する必要がある。

(パスワードがカタログ／ファイルにあればの話)

②-ii パーミッションの種類

以下のものは特定パーミッションとしても一般パーミッションとしても指定できる。

READ ——— ファイルからの読み取りを許す。

WRITE ——— ファイルの読み取り／書き込を許す。

PURGE ——— カタログ／ファイルの抹消を許す。

MODIFY ——— カタログ／ファイルの変更を許す。

CREATE ——— カタログ／ファイルの作成を許す。

APPEND ——— 現時点で終端と識別されるファイルの部分に対して、書き込み、そのファイルへの追加を許す。

EXECUTE ——— コンパイラまたはローダに対してだけによるファイルの読み込み。

RECOVERY ——— 障害ファイルへの書き込み。

LOCK ——— カタログ／ファイルの保護ロックのセット／リセット

EXECLUDE ——— 特定利用者に与えられた全てのパーミッションを除去する。

DELETE ——— パーミッションの一部または全部を削除する。

(—— 下線の部分だけの入力でも可能)

これらのパーミッションの指定方法等はACCESE サブシステムで述べられています。

4) ACCESE サブシステム

これは、端末利用者に、ファイルシステムの機能を利用するためのインターフェイスを提供する。この機能により端末利用者は、次の処理を会話的に行うことができる。

- カタログ／ファイルの作成と削除
- カタログ／ファイルの属性の変更
- ファイルの AFT^{※)}への登録と除去
- カタログ／ファイルのリスト作成

※)AFT (アベイラブル・ファイル・テーブル)

T.S.S. 利用者に対して与えられT.S.E (タイム・シェアリング・エグゼクティブ：T.S.S. 用のオペレーティング・システムと考えて下さい)のUST (ユーザー・ステイタス・テーブル)領域にある。このテーブルに入るファイル名数は限られていて、ファイルに関する情報をもっている。(現在AFT内に常に残っているSY※※を含めて20にセットされている。)ファイルが最初にアクセスされたとき、そのファイル名がAFT内に入る。

- パスワードあるいは長いカタログ／ファイル記述を必要としているファイルでもそれが一度このテーブル内に記録されれば、ファイル名だけで参照することができる。
- たびたび繰り返して用いられるようなファイルの場合、一旦AFTに登録された後は常

に利用可能な状態のままになっている。従ってファイルをアクセスするたびににかかるオーバーヘッド時間およびコストの節約になる。

RUN, LIST, OLD, SAVE, RESAVE, GET, PRINT, LENGTH, PERM コマンドを用いることによって、指定されたパーマネントファイルが AFT 内に登録される。

機能の種類

ACCESS サブシステムからの最初の入力要求である FUNCTION? に対して次の機能のいずれかを指定する。

(短縮して下線で示されている部分だけでよい。)

<u>C</u> REATE <u>C</u> ATALOG	サブカタログを作成する。
<u>C</u> REATE <u>F</u> ILE	ユーザーマスターカタログまたはサブカタログの下にファイルを作成する。
<u>A</u> CC <u>E</u> SS <u>F</u> ILE	ファイルを AFT に登録する。
<u>D</u> E <u>A</u> CC <u>E</u> SS <u>F</u> ILE	ファイルを AFT から除去する。
<u>M</u> ODIFY <u>C</u> ATALOG	カタログの名前、パスワード、パーミッションの一部または全部を変更する。
<u>M</u> ODIFY <u>F</u> ILE	ファイルの名前、サイズ、パスワード、パーミッションの一部または全部を変更する。
<u>P</u> URGE <u>C</u> ATLOG	サブカタログを削除する。 それより下位のカタログおよびファイルも削除される。 それらのスペースはすべて消去(ゼロクリヤ)される。
<u>P</u> URGE <u>F</u> ILE	ファイルを削除する。スペースは消去(ゼロクリヤ)される。
<u>R</u> EL <u>E</u> ASE <u>C</u> ATALOG	サブカタログを削除する。 それより下位のカタログおよびファイルも削除される。 スペースは消去されない。
<u>R</u> EL <u>E</u> ASE <u>F</u> ILE	ファイルを削除する。スペースは消去されない。
<u>L</u> IST <u>C</u> ATALOG	カタログの下に属するサブカタログおよびファイル名をリストする。
<u>L</u> IST <u>S</u> PECIFIC	カタログまたはファイルの属性を詳細にリストする。

識別子

識別子には、カタログ名、ファイル名、利用者 ID およびパスワードがある。識別子はアルファベット、数字、ピリオドおよびマイナス記号が使用できる。

識別子の長さは 12 文字まで可能であるが Create File, Deaccess File のファイル名は 8 文字までに制限されている。

8文字を越えるパッチで作られたファイルをAFTに登録するときは8文字以内の代替名を指定する。またReleaseまたはPurgeで削除する場合は12文字までファイル名を指定することができる。

パーミッション

カタログやファイルの作成者以外のユーザにもカタログやファイルを操作させることを可能にするのがパーミッションの機能である。

・一般パーミッション

全てのユーザに操作を許可する。

指定方法： { 許可操作, 許可操作, ……., 許可操作
DELETE またはDELETE /GEN' L/ }

・特定パーミッション

特定のユーザに操作を許可する。

指定方法： { 許可操作 } /ユーザ名, ユーザ名, ……………, ユーザ名/
EXCLUDE
DELETE

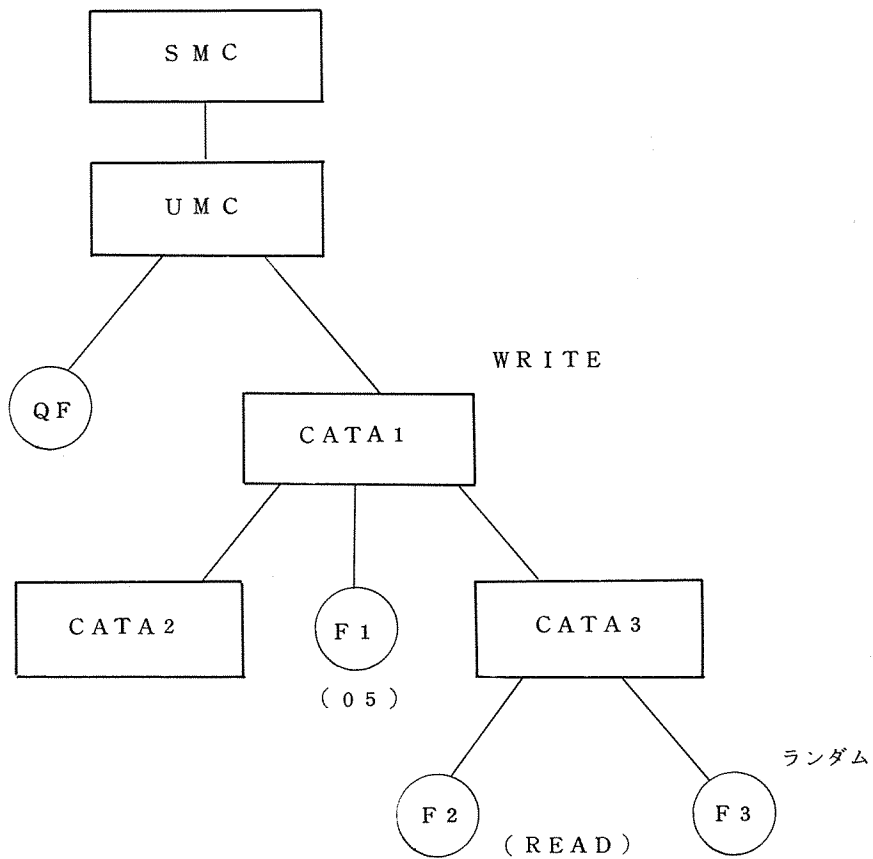
次の操作を指定することが可能である。指定するとき下線部分のみでもよい。

<u>R</u> EAD	ファイルからの読み取り。
<u>W</u> RITE	ファイルへの書き込み。
<u>A</u> PPEND	ファイルの終端へ書き込みそのファイルに追加する。
<u>E</u> XECUTE	実行するプログラムはカタログ化されたファイルに置かれていてコンパイラまたはローダによるファイルからの読み込み。
<u>P</u> URGE	カタログやファイルの抹消
<u>M</u> ODIFY	カタログやファイルの属性変更。
<u>L</u> OCK	カタログやファイルの保護ロックのセット／リセット
<u>C</u> REATE	カタログやファイルの作成
<u>R</u> ECOVERY	障害ファイルへの書き込み。
<u>E</u> XCLUDE	特定ユーザに与えられた全てのパーミッションを除去する。
<u>D</u> ELETE	パーミッションの一部または全部を削除する。

5) ACCESS使用例

ここでは、ACCESS サブシステムの使用例を使い、図のようなカタログ／ファイル構造を作成し説明を行ないます。

使用例



- (1) カタログの作成 (C A T A 1 , C A T A 2 , C A T A 3)
- (2) ファイルの作成 (Q F , F 1 , F 2 , F 3)
- (3) カタログのリスト (U M C 以下 , C A T A 1 以下 , C A T A 3 以下 , C A T A 2 以下)

- (4) カタログ (CATA 1) およびファイル (F 2 , F 3) の属性リスト
- (5) AFTへの登録 (QF , F 3 : 代替名 0 6)
- (6) AFTからの除去 (すべてのパーマネント・ファイル)
- (7) カタログの属性変更 (CATA 1 : 一般パーミッションの削除)
- (8) ファイルの属性変更 (F 1 : ファイル名 0 5 , F 2 : 一般パーミッション READ)
- (9) ファイルの削除 (QF , F 2)
- (10) カタログの削除 (CATA 1)

(注) はじめに一般の形を で示し、次にそれぞれの使用例を示します。使用例中の (Cr) は送信キーのみの送信を表わし、また、各段階での図で太線が今、作成したカタログ／ファイルを表わします。

I 会話型使用法

- (1) カタログの作成

```

FUNCTION? CC
CATALOG STRUCTURE TO WORKING LEVEL?
User-id/cat-name$Pass/.../cat-name$Pass
NEW CATALOG NAME? cat-name
PASSWORD ?
XXXXXXXXXXXXXXXXXXXX
GENERAL PERMISSION? access-type,...,access-type
SPECIFIC PERMISSION? access-type,...,access-type/
user-id/.../user-id

```

FUNCTION? CC

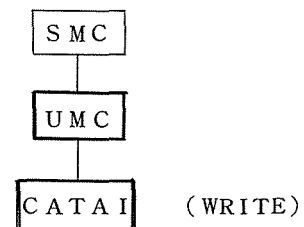
CATALOG STRUCTURE TO WORKING LEVEL? (Cr)

—— UMCのすぐ下に、次のCATA 1を作成する。 (Cr) キーのみ送信

```

NEW CATALOG NAME? CATA 1
PASSWORD?
OBOREFUMWXKZME ( Cr ) —— パスワードなし
GENERAL PERMISSION? WRITE
SPECIFIC PERMISSIONS? ( Cr )
SUCCESSFUL.

```



NEW CATALOG NAME? (Cr) ———— ここではCATA1と同一レベルのカタログを作らない為 (Cr) 送信し,
FUNCTION? CC FUNCTIONモードに戻す。

CATALOG STRUCTURE TO WORKING LEVEL? /CATA1
NEW CATALOG NAME? CATA2

```

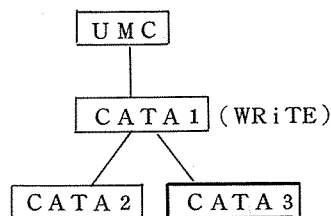
graph TD
    UMC[UMC] --> CATE1[CATE1]
    CATE1 --> CATA2[CATA2]
  
```

(WRITE)

PASSWORD?
XXXXXXXXXXXXXXXXXXXX (Cr)
GENERAL PERMISSION? (Cr)
SUCCESSFUL.

NEW CATALOG NAME? CATE3 — CATA2と同レベルに, すなわち CATA1の次のレベルにCATA3を作成する。

PASSWORD?
XXXXXXXXXXXXXXXXXXXX (Cr)
GENERAL PERMISSION? (Cr)
SPECIFIC PERMISSIONS? (Cr)
SUCCESSFUL.
NEW CATALOG NAME? (Cr)



ここまでで, 3レベル(UMCを含む)までのカタログができました。CATA1以下のレベルではREAD/WRITEが可能です。

(2) ファイルの作成

```

FUNCTION? CF
CATALOG STRUCTURE TO WORKING LEVEL?
Uses-id/cat-name$pass/.../cat-name$pass
FILE NAME, SIZE(IN L LINKS), MAX SIZE, MODE?
file name, initial size, max size { , L
                                     R }
PASSWORD?
XXXXXXXXXXXXXXXXXXXX
(LOGICAL RECORD SIZE? words)
GENERAL PERMISSION? access-type, ..., access-type
  
```

SPECIFIC PERMISSION? <u>access-type,...,access-type/</u>	
<u>user-id/user-id.../user-id</u>	
ACCESS FILE? $\left\{ \begin{array}{c} \text{YES} \\ \\ \text{Y} \end{array} \right\}$	または $\left\{ \begin{array}{c} \text{CLEAR} \\ \\ \text{C} \end{array} \right\}$

- ・ファイル名は8文字までに制限されている。
- ・max sizeを省略すると初期サイズと同じ値がとられる。
- ・また制限しない場合は“U”を指定する。
- ・連続したファイル領域を要求したいときはFile name, Initial size C, $\left\{ \begin{array}{c} \text{L} \\ \text{R} \end{array} \right\}$ と指定する。このとき最大サイズは指定できない。
- ・MODEに対してR(ランダム)を指定した場合には、カッコ内のLOGICAL RECORD SIZE?の質問に答える。
- ・ACCESS FILE?の質問にYESまたはYを指定すると作成されたファイルをAFTに登録する。またCLEARまたはCを指定するとファイルをAFTに登録後ファイルスペースをゼロクリアする。

FUNCTION? CF

CATALOG STRUCTURE TO WORKING LEVEL? (Cr)

UMCのすぐ下のレベルにクイックアクセスファイルとしてファイル名QFを作成

FILE NAME, SIZE(IN L LINKS), MAX SIZE, MODE?
QF, 1, 2

PASSWORD?

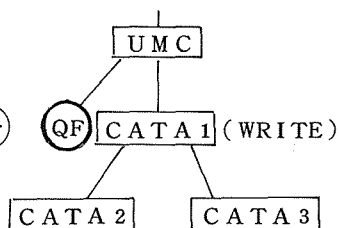
XXXXXXXXXXXX (Cr)

GENERAL PERMISSION? (Cr)

SPECIFIC PERMISSION? (Cr)

ACCESS FILE? (Cr)

SUCCESSFUL.



FILE NAME, SIZE(IN L LINKS), MAX SIZE, MODE?

(Cr) — UMCのすぐ下には“QF”だけを作成FUNCTIONモードに戻す。

FUNCTION? CF

CATALOG STRUCTURE TO WORKING LEVEL?

/CATA1

FILE NAME, SIZE(IN L LINKS), MAX SIZE, MODE?

F1, 2, 3

PASSWORD?

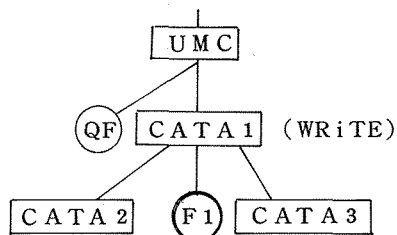
XXXXXXXXXXXXXXXXX (Cr)

GENERAL PERMISSION? (Cr)

SPECIFIC PERMISSION? (Cr)

ACCESS FILE? (Cr)

SUCCESSFUL.



FILE NAME, SIZE(IN LLINKS) MAX SIZE? (Cr)

FUNCTION? CF

CATALOG STRUCTURE TO WORKING LEVEL?

/CATA1/CATA3

FILE NAME, SIZE(IN LLINKS) MAX SIZE, MODE?

F2, 1, 3

PASSWORD?

XXXXXXXXXXXXXXXXX (Cr)

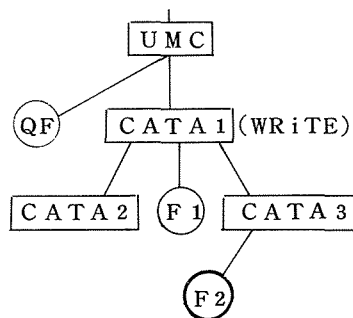
GENERAL PERMISSIONS? (Cr)

SPECIFIC PERMISSION? (Cr)

ACCESS FILE? YES—ファイル

が作成された時点で、そのファイルのアクセスが
可能になる。

SUCCESSFUL.



FILE NAME, SIZE(IN LLINKS), MAM SIZE, MODE?

F3, 2, R

PASSWORD?

XXXXXXXXXXXXXXXXX (Cr)

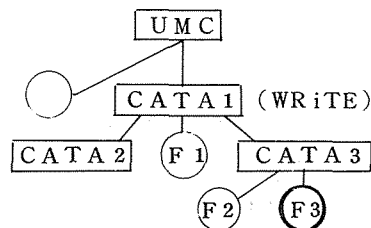
LOGICAL RECORD SIZE? (Cr)—論理レコードサイズの属性が必要と
されない。*)

GENERAL PERMISSION? (Cr)

SPECIFIC PERMISSION? (Cr)

ACCESS FILE? (Cr)

SUCCESSFUL.



FILE NAME, SIZE(IN LLINKS), MAX SIZE, MODE? (Cr)

※) FORTRANのランダム入出力ファイルの場合には論理レコードサイズの属性をもた
せることができる。

これで4つのファイル名の作成ができました。“QF”はクイックアクセスファイル

F1の修飾は UMC名/CATA1/F1

F2 “ UMC名/CATA1/CATA3/F2

F3 “ UMC名/CATA1/CATA3/F3

(UMC名は作成者本人であれば省略可その場合 “/CATA1/
CATA3/F3”と修飾する)

(3) カタログ／ファイルのリスト

```
FUNCTION? LC
CATALOG STRUCTURE INCLUDING CATALOG TO
BE LISTED?
user-id/cat-name,...,cat-name,x(mm-dd-yy,n,R),
FIRST/name/
```

- ・カタログ構造にパスワードの指定はいらない。端末利用者は自分自身のカタログ、ライブラリカタログ(#LIB)あるいはコマンドライブラリカタログ(#CMD)のみをリストできる。

- ・指定したカタログの下に属するカタログおよびファイルのリストが出力される。

- ・リストするカタログあるいはファイルを限定するとき

x(mm-dd-yy,n,R)またはFIRST/name/を指定する。

x;x=C:作成日による限定

x=A:最終アクセス日付による限定

x=L:最終修正日付による限定

mm-dd-yy;xに対応する日付、この日付以降のものを古い順に出力する。

n;リストすべきファイル数

} 指定の順番は任意

R;出力の順番を逆(新しい順)にする。

FIRST/name/によってカタログ／ファイル名で指定したところからリスト出力する。

FUNCTION? LC

CATALOG STRUCTURE INCLUDING CATALOG TO

BE LISTED? (Cr) — UMC以下すぐ下のレベルのカタログ名／ファイル名を

リストアップするために (Cr) キー送信

LIST OF CATALOG-UMC ——— UMC名: UMC

CATALOG

FILES

CATA1

QF

FUNCTION? LC

CATALOG STRUCTURE INCLUDING CATALOG TO

BE LISTED? /CATA1 ——— CATA1名のカタログの下にぶらさがっている
カタログ名のリストアップ

LIST OF CATALOG-CATA1

CATALOGS

FILES

CATA2

CATA3

F1

FUNCTION? LC

CATALOG STRUCTURE INCLUDING CATALOG TO
BE LISTED?

/CATA1/CATA3 ——— CATA3はCATA1の下にある為、必ずCATA1よ
り修飾して行く。

LIST OF CATALOG-CATA3

CATALOG

FILES

F2

F3

FUNCTION? LC

CATALOG STRUCTURE INCLUDING CATALOG TO
BE LISTED?

/CATA1/CATA2

CATALOG 'CATA2' IS EMPTY

FUNCTION? (Cr)

(4) カタログまたはファイルの属性をリスト

FUNCTION? LS

CATALOG STRUCTURE TO WORKING LEVEL?

user-id/cat-name,...,cat-name

CATALOG OR FILE TO BE LISTED? $\left\{ \begin{array}{l} \text{cat-name} \\ \text{file-name} \end{array} \right\}$

- ・パスワードを与える必要はない。端末利用者は自分自身のカタログやファイルあるいはライブラリ(#LIB), コマンドローダ(#CMD)のカタログおよびファイルのみをリストできる。

(注)・ 入力行がデリミッタ(スラッシュ, コンマ, ドル記号)で終わっている場合は, 端末利用者に対してMORE? が出され, 入力行の続きが要求される。

- ・ CATALOG STRUCTURE TO WORKING LEVEL? の質問に対してUMCに位置づける方法は3通りある。

① 何にも応答しない。(キャリッジリタンのみ)

② UMC名を応答する。

③ / を応答する。

- ・ SPECIFIC PERMISSION? の質問に対しては1回の応答につき1セットのパーミッションしか応答することはできない。応答するたびにこの質問が繰り返されキャリッジリタンのみを与えることにより以降の質問に移る。

SPECIFIC PERMISSION? READ/UEKI/KATO

SPECIFIC PERMISSIONS? WRITE/NARISAWA/
KATO

SPECIFIC PERMISSIONS? (Cr)

- ・ カタログやファイルの属性変更(MC, MF)またはカタログやファイルの削除(PC, PF, RC, RF)においてパスワードのついているカタログやファイルを操作するときに操作するカタログやファイルの名前とともにパスワードを指定しなければならない。パスワードを指定しないと会話における最後に

PASSWORD - $\left\{ \begin{array}{l} \text{カタログ名} \\ \text{または} \\ \text{ファイル名} \end{array} \right\} ?$

XXXXXXXXXXXXXXXX

と質問してくるので操作前のパスワードを答える必要がある。

FUNCTION? LS

CATALOG STRUCTURE TO WORKING LEVEL? (Cr)

CATALOG OR FILE TO BE LISTED? CATA1

CATALOG NAME-CATA1	—— カタログ名
ORIGINATOR-KYOIKU	—— 製作者
DATE CREATED-012376	—— 製作日
DIVICE-DP4	—— 入っている装置
GENERAL PERMISSION-R,W,A,E	—— 一般パーミッション
SPECIFIC PERMISSION-NONE	—— 特定パーミッション

CATALOG OR FILE TO BE LISTED? (Cr)

FUNCTION? LS

CATALOG STRUCTURE TO WORKING LEVEL?

/CATA1/CATA3

CATALOG OR FILE TO BE LISTED? F2

FILE NAME-F2	—— ファイル名
ORIGINATOR-KYOIKU	—— 製作者
DATE CREATED-012376	—— 製作日
DATE CHANGED-012376(14.645)	—— 変更された日
LAST DATE ACCESSED-012376	—— 最近使われた日
NUMBER OF ACCESSES-1	—— 使われた回数
MAX FILE SIZE-3 LLINKS	—— 最大サイズ
CURRENT FILE SIZE-1 LLINK	—— 現在使われている大きさ
FILE TYPE-LINKED	—— ファイルタイプ
DEVICE-DP4	—— 入っている装置
GENERAL PERMISSION-NONE	—— 一般パーミッション
SPECIFIC PERMISSION-NONE	—— 特定パーミッション

CATALOG OR FILE TO BE LISTED? F3

FILE NAME-F3
ORIGINATOR-KYOIKU
DATE CREATED-012376
DATE CHANGED-012376(14.659)
LAST DATE ACCESSED-012376
MAX FILE SIZE-2 LLINKS
CURRENT FILE SIZE-2 LLINKS
FILE TYPE-RANDOM
DEVICE-DP4
GENERAL PERMISSION-NONE
SPECIFIC PERMISSION-NONE

CATALOG OR FILE TO BE LISTED? (Cr)

(5) AFTへの登録

FUNCTION? AF
CATALOG STRUCTURE WORKING LEVEL?
user-id/cat-name\$pass/.../cat-name\$pass
FILE NAME? file name\$pass “代替名”(,R)
PERMISSION DESIRED? access-type,...access-type

・代替名は次のような目的で使用する。

- (1) バッチで作られたファイルのファイル名が8文字を越えていてそのファイルを利用するとき、
- ・ファイル名が8文字以内ではあるが、他のファイル名でAFTに登録したいとき。
- ・FILE NAME?の質問に、シーケンシャル ファイルをランダム ファイルとして扱いたいときはRを指定する。

FUNCTION? AF

CATALOG STRUCTURE TO WORKING LEVEL? (Cr)

FILE NAME? QF

PERMISSIONS DESIRED? (Cr)

SUCCESSFUL.

FILE NAME? (Cr)

FUNCTION? AF

CATALOG STRUCTURE TO WORKING LEVEL?

/CATA1/CATA3

FILE NAME? F3 "06"—ファイル名“F3”の代替名を“06”として登録

PERMISSION DESIRED? (Cr)

SUCCESSFUL.

FILE NAME? (Cr)

(6) AFTからの除去

FUNCTION?	<u>DF</u>					
FILE NAME?	<table border="0"><tr><td rowspan="4">{</td><td>ファイル名</td></tr><tr><td>CLEARFILES</td></tr><tr><td>PERMFILES</td></tr><tr><td><u>TEMPFILES</u></td></tr></table>	{	ファイル名	CLEARFILES	PERMFILES	<u>TEMPFILES</u>
{	ファイル名					
	CLEARFILES					
	PERMFILES					
	<u>TEMPFILES</u>					

- ・ FILE NAME? の質問に対して CLEARFILES を指定するとすべてのファイルを取り除く。
- ・ どれを指定してもカレントファイル (*SRC), コレクタファイル (SY**) を取り除くことはできない。

FUNCTION? DF

FILE NAME? R@PERMFILES @; 文字Rを省くために用いられる。

SUCCESSFUL.

FUNCTION? (Cr)

(7) カタログの変更

```

FUNCTION? MC
CATALOG STRUCTURE TO WORKING LEVEL?
user-id/cat-name$pass/.../cat-name$pass
CATALOG TO BE MODIFIED? cat-name$pass
NEW NAME? new cat-name$pass
PASSWORD?

XXXXXXXXXXXXXXXX { new pass }
                  { DELETE }

GENERAL PERMISSIONS? { access-type,...,access-type }
                     { DELETEまたはDELETE/GEN' L/ }

SPECIFIC PERMISSION?
{ access-type,...,access-type/user-id...user-id }
{ DELETE/user-id...../user-id }

```

- ・現在のカタログ名、パスワードの変更および属性の追加、削除などの変更を行う。
- ・変更しない部分は、キャリッジリタンのみを応答する。

```

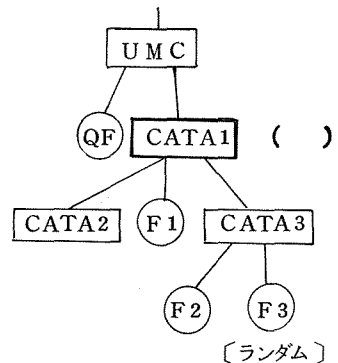
FUNCTION? MC
CATALOG STRUCTURE TO
WORKING LEVEL? (Cr)

CATALOG TO BE MODIFIED?
CATA1

NEW NAME? (Cr)

NEW PASSWORD?
XXXXXXXXXXXXXXXX (Cr)

```



```

GENERAL PERMISSIONS? DELETE ——— すべてのパーミッシ
ョンを削除

```

SPECIFIC PERMISSIONS? (Cr)
 SUCCESSFUL
 CATALOG TO BE MODIFIED? (Cr)

(8) ファイルの属性変更

```

FUNCTION? MF
CATALOG STRUCTURE TO WORKING LEVEL?
user-id/cat-name$pass/.../cat-name$pass/file-name$pass
FILE TO BE MODIFIED? file-name$pass
NEW MAX SIZE IN LLINKS? new-max-size
NEW PASSWORD?
XXXXXXXXXXXXXXXX { new pass }
                  { DELETE }
GENERAL PERMISSIONS? { access-type,...,access-type }
                     { DELETEまたはDELETE/GEN' L/ }
SPECIFIC PERMISSIONS?
{ access-type,...,access-type/user-id,.../user-id }
{ DELETE/user-id/...user-id }

```

- ・現在のファイル名、パスワード、サイズ (LLINK) の変更および属性の追加、削除などの変更を行う。
- ・ファイル名は8文字までに制限されている。
- ・変更しない部分はキャリッジリタンのみを応答する。

```

FUNCTION? MF
CATALOG STRUCTURE TO WORKING LEVEL?
/CATA2@1 —— @は文字2の打ちまちがいを文字1に打ち直す。
FILE TO BE MODIFIED? F1
NEW NAME? 05 —— ファイル名“F1”をファイル名“05”に変える。

```

NEW MAX SIZE IN LLINKS? (Cr)

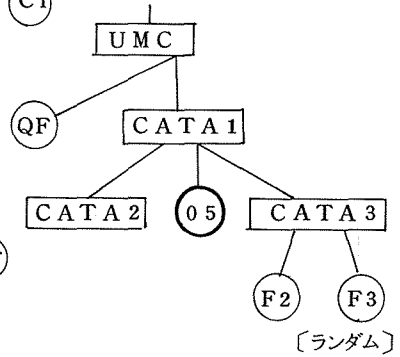
NEW PASSWORD?

XXXXXXXXXXXXXXXXXXXX (Cr)

GENERAL PERMISSIONS? (Cr)

SPECIFIC PERMISSIONS? (Cr)

SUCCESSFUL.



FILE TO BE MODIFIED? (Cr)

FUNCTION? MF

CATALOG STRUCTURE TO WORKING LEVEL?

/CATA1/CATA3

FILE TO BE MODIFIED F2

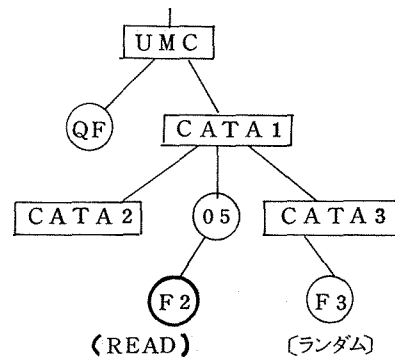
NEW NAME? (Cr)

NEW MAX SIZE IN LLINKS? 7 — MAX SIZE 3LLi

NKを7LLINKに変更

NEW PASSWORD?

XXXXXXXXXXXXXXXXXXXX (Cr)



GENERAL PERMISSIONS? READ

SPECIFIC PERMISSIONS? (Cr)

SUCCESSFUL.

FILE TO BE MODIFIED? (Cr)

(9) ファイルの削除 (リリース ファイル)

FUNCTION? RF

CATALOG STRUCTURE TO WORKING LEVEL?

user-id/cat-name\$pass/.../cat-name\$pass

FILE TO BE RELEASED? file-name\$pass

- ・ 削除されたスペースはゼロクリヤされない。
- ・ ファイル内容の保護を要求しない場合は通常この方法が用いられる。

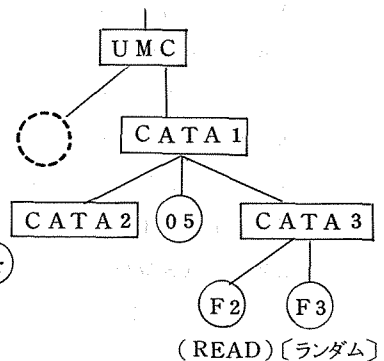
FUNCTION? RF

CATALOG STRUCTURE TO
WORKING LEVEL? (Cr)

FILE TO RELEASED? QF

SUCCESSFUL.

FILE TO BE RELEASED? (Cr)



FUNCTION? RF

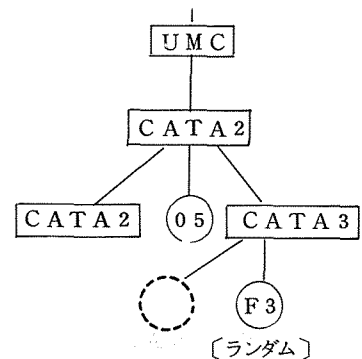
CATALOG STRUCTURE TO
WORKING LEVEL?

/CATA1/CATA3

FILE TO BE RELEASED? F2

SUCCESSFUL.

FILE TO BE RELEASED? (Cr)



ファイルの削除 (ページ ファイル)

FUNCTION? PF

CATALOG STRUCTURE WORKING LEVEL?

user-id/cat-name\$pass/.../cat-name\$pass

FILE TO BE PURGED? file name\$pass

- ・ 削除されたペースはゼロクリヤされる。

(10) カタログの削除 (リリース カタログ)

```

FUNCTION? RC
CATALOG STRUCTURE TO WORKING LEVEL?
user-id/cat-name$pass/.../cat-name$pass
CATALOG TO BE RELEASED? cat-name$pass

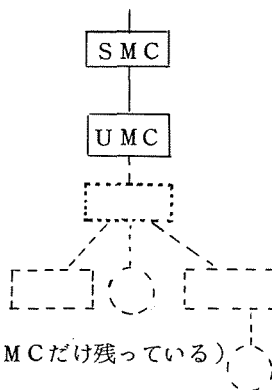
```

- ・サブカタログおよびその下位にあるすべてのサブカタログおよびファイルを削除する。
- ・削除されたスペースはゼロクリアされない。
- ・ファイル内容の保護を要求しない場合は通常この方法が用いられる。

```

FUNCTION? RC
CATALOG STRUCTURE TO
WORKING LEVEL? (Cr)
CATALOG TO BE RELEASED?
CATA1
SUCCESSFUL.
CATALOG TO BE RELEASED? (Cr)

```



これですべてのカタログ／ファイルがなくなりました。(UMCだけ残っている)

カタログの削除 (パージ サブカタログ)

```

FUNCTION? PC
CATALOG STRUCTURE TO WORKING LEVEL?
user-id/cat-name$pass/.../cat-name$pass
CATALOG TO BE PURGED? cat-name$pass

```

- ・サブカタログおよびその下位にあるすべてのサブカタログおよびファイル削除する。
- ・削除されたスペースはゼロクリアされる。

今まで会話型使用を説明してきましたが、次に非会話型の例を説明します。

Ⅱ 非会話型の使用法

一 般 型

SYSTEM?ACCE△ } ディレクティブ, { 修飾カタログ名 }
 FUNCTION? } { 修飾ファイル名 }, オプション

- (1) 非会話型短縮形入力はSYSTEM?レベルまたはFUNCTION?レベル入力することができる。

例)

SYSTEM? ACCE CF/ CAT\$PASS/FILE, B/1, 3/, W
 FUNCTION? CF, /CATA\$PASS/FILE, B/1, 3/, W

- (2) ある機能(ディレクティブ)に対する複数の処理を単一行(セミコロンで区切って)で指定することができる。

例)

CF, /CATA1/FILEA, B/1, 5/, R; /CATA1/FILEB, B
 /1, 2/

- (3) 適当なところでデリミッタ(スラッシュ, コンマあるいはドル記号それにキャリッジリターン)を与えて行を終りにし, 入力が次の行に続くことを示せば複数行にわたってもよい, その時端末利用者に対してMORE?が出力される。

- (4) オプションの指定方法は以下の通りである。

形	式	機 能
PASSWORDまたはPASS	password/	パスワード指定
access-type		一般パーツション
assess-type	user-id, ..., user-id/	特定パーツション
EXCLUDE	user-id, ..., user-id/	パーツションの削除, 変更
DELETE	user-id, ..., user-id/	
DELETE		
NEWNAME	{ new-catalog-name } / { new-file-name } /	カタログ名またはファイル名の変更
{ BLOCKまたは LINKまたはL SIZE LINKS }	/x, y/	作成時のサイズ指定 x: 初期サイズ y: 最終サイズ

$\left\{ \begin{array}{l} \text{BLOCKS または B} \\ \text{LINKS または L} \\ \text{SIZE} \\ \text{LINKS} \end{array} \right\} / y /$

最大サイズの変更

$\text{MODE} / \left\{ \begin{array}{l} \text{LINKED または L} \\ \text{SEQ または S} \\ \text{RAND または R} \end{array} \right\} /$

モードの指定

$\text{LRS} / \text{size} /$

論理レコードサイズ

$\text{DEVICE} / \text{装置名} /$

固有の装置名を指定する。

DKU 410

DKU 411

AF または OPEN

ファイルを作成した後にアクセスする。

CLEAR

ファイルを作成しアクセスした後にフ

ァイルスペースをゼロクリヤする。

さらに CF または MF ディレクティブにおいて会話形では扱うことのできない次のオプションを使用することができる。以下のオプションの詳細については「ファイルマネージメントスーパーバイザ (FMS) 説明書」を参照して下さい。

$\text{ABORT} / \left\{ \begin{array}{l} \text{LOCK} \\ \text{DELAY} \\ \text{ROLLBACK} \\ \text{NONE} \end{array} \right\} /$

書き込んだジョブがアボートしたとき、
何をするか?

$\text{ACCESS} / \left\{ \begin{array}{l} \text{NORMAL} \\ \text{RWW} \\ \text{CONCURRENT} \\ \text{MONITOR} \end{array} \right\}$

要求したファイルが他で書き込まれて
いるとき、何をするか?

$\text{VERIFY} / \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$

Write を Write-Verify 命令に
換えるべきか?

$\text{AUDIT} / \left\{ \begin{array}{l} \text{ALL} \\ \text{DENIED} \\ \text{NONE} \end{array} \right\}$

実績記録

$\text{INCRSAVE} / \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} /$

最後にセーブしたものから内容が変更
されたら、そのファイルをセーブする
か?

PAGESIZE/#Words/

40≤#word≤3840

RDERR/DUP

読み取りエラーを回復させるために何をするか？

注 ・ { } 内の任意のいずれかを選択する。

・ 下線はそのオプションにおける省略値である。

2. 非会話型による使用例

(1) カタログの作成

```
FUNCTION? CC, /CATA1, WRITE;/CATA1/CATA2;  
/CATA1/CATA3  
SUCCESSFUL.
```

(2) ファイルの作成

```
FUNCTION? CF, QF, B/1, 2//;/CATA1/F1, B/2, 3//;  
MORE?/CATA1/CATA3/F2, B/1, 3//, AF;/CATA3/  
F3, B/2, 2//, MODE/RAND/  
SUCCESSFUL.
```

(3) カタログのリスト

```
FUNCTION? LC, KYOIKU;/CATA1;/CATA1/CATA3;  
/CATA1/CATA2  
LIST OF CATALOG-KYOIKU  
CATALOG  
FILES  
CATA1  
QF  
LIST OF CATALOG-CATA1  
CATALOGS  
FILES  
CATA2  
CATA3  
F1  
LIST OF CATALOG-CATA3  
CATALOGS  
FILES  
F2  
F3  
CATALOG 'CATA2' IS EMPTY
```

(4) カタログおよびファイルの属性のリスト

FUNCTION? LS, /CATA1; /CATA1/CATA3/F2; /
CATA3/F3

CATALOG NAME-CATA1
ORIGINATOR-KYOIKU
DATE CREATED-012976
DEVICE-PK1
GENERAL PERMISSION-R, W, A, E
SPECIFIC PERMISSIONS-NONE

FILE NAME-F2
ORIGINATOR-KYOIKU
DATE CREATED-012976
DATE CHANGED-012976(17.221)
LAST DATE ACCESSED-012976
MAX FILE-3 LLINKS
CURRENT FILE SIZE-1 LLINK
FILE TYPE-LINKED
DEVICE-PK7
GENERAL PERMISSIONS-NONE
SPECIFIC PERMISSIONS-NONE

FILE NAME-F3
ORIGINATOR-KYOIKU
DATE CREATED-012976
DATE CHANGED-012976(17.221)
LAST DATE ACCESSED-012976
MAX FILE SIZE-2 LLINKS
CURRENT FILE SIZE-2 LLINKS
FILE TYPE-RANDOM
DEVISE-PK7
GENERAL PERMISSIONS-NONE
SPECIFIC PERMISSIONS-NONE

- (5) AFTへの登録
 FUNCTION? AF, QF; /CATA1/CATA3/F3 "06"
 SUCCESSFUL.
- (6) AFTからの除去
 FUNCTION? DF, PERMFILES
 SUCCESSFUL.
- (7) カタログの属性変更
 FUNCTION? MC, /CATA1, DELETE
 SUCCESSFUL.
- (8) ファイルの属性変更
 FUNCTION? MF, /CATA1/F1, NEWNAME/05;/ /CATA1
/CATA3/F2, B/7/, READ
 SUCCESSFUL.
- (9) ファイルの削除
 FUNCTION? RF, QF
 SUCCESSFUL.
- (10) カタログの削除
 FUNCTION? RC, /CATA1
 SUCCESSFUL.

Ⅲ アスタリスクの効果

ACCESSは、アスタリスクを与えられる時点までの指示情報をもとに処理を行う。もちろんACCESSには、処理を行うに十分な情報が与えられていなければならない。

1. 会話型における効果

アスタリスクを与えられると以降の質問はスキップされて処理が行われる。

(1) 質問に応答した後に1つのアスタリスクを与えた場合。

ACCESSは、処理後に同じカタログレベルの最初の質問に戻る。

例)

```

FUNCTION? CF
CATALOG STRUCTURE TO WORKING LEVEL?
/GMAP
FILE NAME, SIZE(IN LLINKS), MAX SIZE,
MODE? F1, 1, 3
PASSWORD?
XXXXXXXXXXXXXXXXXXXX
GENERAL PERMISSION? R*

```

SUCCESSFUL.

FILE NAME, SIZE (IN LLINKS), MAX SIZE, MODE?

- (2) 質問に応答した後に2つのアスタリスクを与えた場合。

ACCESSは、処理後にFUNCTION?の質問に戻る。

例)

FUNCTION? CF

CATALOG STRUCTURE TO WORKING LEVEL?

/GMAP

FILE NAME, SIZE (IN LLINKS), MAX SIZE, MODE?

F 2, 3, 3 ※ ※

SUCCESSFUL.

FUNCTION?

2. 非会話型における効果

- (1) アクセス要求が1つのアスタリスクで終わっている場合。

ACCESSは、処理後に、会話型に切り換え非会話型で要求した同じカタログレベルの最初の質問を出力する。

例)

FUNCTION? CF, /GMAP/F 4, B / 1, 2 / ※

SUCCESSFUL

FILE NAME, SIZE (IN LLINKS), MAX SIZE, MODE?

- (2) アクセス要求が2つのアスタリスクで終わっている場合。

ACCESSは、処理後にFUNCTION?の質問を出力する。

例)

FUNCTION? CF, F 3, B / 1, 2 / ※ ※

SUCCESSFUL

FUNCTION?

注 SYSTEM?レベルで使用しても同じ結果です。

なお、ファイルの使い方については次回以降に説明します。