

《解説》

HITAC 8700/8800 OS7 のジョブ制御について

高須 昭輔* 大西 勲* 東常 修也**

1. はじめに

OS7 は HITAC 8700/8800 上で動作する仮想空間方式を採用した汎用オペレーティング・システムである。OS7 におけるジョブ管理（ジョブ制御）は次の特徴を持っている。

(1) 各処理形態の混在サービス

バッチ処理、会話処理 (TSS)、リモート・バッチ処理、実時間処理の各処理形態を1つのオペレーティング・システムでサポートしておりシステム資源の有効利用ができる。

(2) コマンド・システムの完備

OS7 ではユーザからシステムへのサービスの要求あるいは連絡の手段となるものをコマンドと呼んでおり、このコマンドには、バッチ・ジョブを記述するジョブ制御言語、会話処理における端末操作コマンド、センタ・オペレータの用いるオペレータ・コマンド、リモート・バッチ端末のオペレータが用いるリモート・バッチ・オペレータ・コマンドのすべてが含まれる。これらのコマンドは基本的には1つの概念で取扱うことができ、コマンド・システムが持つ数々の機能を用いることにより操作性が向上する。特にバッチ・ジョブで使用するコマンドのほとんどが会話ジョブでも使用できることは大きな特徴である。

(3) 資源の共用と排他的制御

OS7 ではシステムの利用者（ユーザ）を図-1のような階層構造で管理しており、個々のユーザには特権レベル（システム・プログラマ、特定ユーザ、一般ユーザ）が与えられる。階層構造、各個人の特権レベル、パスワード、課金情報（予算）等はユーザ登録簿と呼ばれるファイルの中に格納

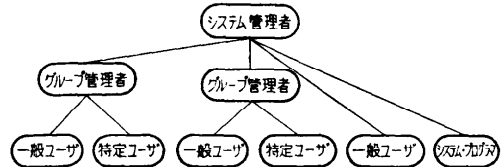


図-1 ユーザの階層の例

されており、これによりシステム資源のグループ単位の共用、あるいは特権レベルによる使用制限、さらには各個人の特権保護が充分行なえる。

2. コマンド体系

2.1 コマンドの文法と種類

OS7 のコマンドは一般的に図-2 の形式をしている。

△₀ は空白があってもなくてもよいことを示しており、△₁ は空白が1つ以上必要であることを示している。このような空白が入る場所には /* 文字列 */ の形式でコメントを挿入できる。

レコードの先頭の2文字はコマンド接頭記号と呼ばれるもので、このレコードがコマンドであることを示している。バッチ・ジョブの場合は // をレコードの先頭に記述する必要があるが、会話ジョブを実行中はシステムからプロンプティングとして //、1/ (クイット状態のとき) あるいは @/ (サブ・コマンドのとき) が出力される。センタ・オペレータ・コマンドの場合もシステムから // が出力される。リモート・バッチ・オペレータ・コマンドの場合は取扱うジョブと区別するために .. を用いている。

コマンド・ラベルおよびコマンド名は英字、カナ文字あるいは数字からなる8桁以内の記号である。コマンド・ラベルはジョブ名、ファイル定義名のように特

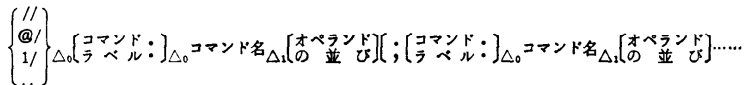


図-2 コマンドの形式

* (株)日立製作所ソフトウェア工場
** 日立ソフトウェア・エンジニアリング(株)

表-1 コマンドの分類

分 類	使用目的とコマンドの例
基本コマンド (ユーザ・コマンド)	バッチ・ジョブと会話ジョブに共通なコマンドで、例えば次のようなコマンドがある。 JOBジョブの先頭を示し、オペランドでユーザ登録名等を指定する。 DTFファイルを定義する。(4.を参照) JOBLIB.....プログラム・ライブラリ・ファイルを定義する。(3.4参照) RUNオペランドで指定したプログラムを実行する。 ENDジョブの終りを示す。
リーダ制御コマンド (ユーザ・コマンド)	バッチ・ジョブ専用のコマンドで入力リーダに指示を与える。例えば次のようなコマンドがある。 ADMIT.....JOB コマンドで表わせない情報をシステム(センタ・オウン・コーディング)に知らせる。 TRANS.....OS7以外のジョブ制御言語をOS7のコマンドに変換する。 ENTER.....定形的なジョブ・ストリームを取込む。
会話専用コマンド (ユーザ・コマンド)	会話ジョブでのみ使用できるコマンドで、例えば次のようなコマンドがある。 CONF確認モードの設定を行なう。 RESUME. TERM.....クイットの操作に関するコマンド。(3.2参照)
オペレータ・コマンド	センタ・オペレータが用いるコマンドで、例えば次のようなコマンドがある。 START.....入力リーダ、出力ライタ等を起動する。 STOP.....入力リーダ、出力ライタ等を停止させる。 CANCELジョブをキャンセルする。
リモート・バッチ・オペレータ・コマンド	リモート・バッチ・オペレータが用いるコマンドで、例えば次のようなコマンドがある。 RRDRリモート・バッチ端末からジョブを入力する。 ROUTPUT出力結果を端末に出力する。
コマンド定義制御コマンド	コマンドを定義するためのコマンドで、例えば次のようなコマンドがある。 BUILTIN.....組込みコマンドを定義する。 MCDEF.....マクロ・コマンドを定義する。(2.2参照)

別の意味を持つ場合もあるし(2.4参照)、条件によって特定の箇所までコマンドをスキップするのに用いられることもある(2.2参照)。

オペランドの並びは各オペランドをカンマ(,)で区切って並べたものであり、オペランドの個数、順序、およびそのオペランドがキーワードを持っているか否かはコマンドを定義した時に定まる。(コマンドの定義については2.2で述べる。)

例えばコマンド COM1のオペランドが順にキーワード KEY 1, KEY 2, KEY 3 を持っているとする、次のような記述ができてすべて同じ意味を示す。

- (i) //COM1 KEY 1=ABC, KEY 2=2, KEY 3=3

- (ii) //COM1 ABC, 2, KEY 3=3
- (iii) //COM1 ABC, 2, 3
- (iv) //COM1 KEY 3=3, KEY 1=ABC, KEY 2=2
- (v) //COM1 KEY 3=3, KEY 1=ABC, 2

OS7 のコマンドは使用目的別に表-1 のように分類されている。

2.2 コマンドの定義

OS7 におけるコマンドは組込みコマンドとマクロ・コマンドに分類される。組込みコマンドはこれ以上分解されないコマンドで、コマンド処理モジュールが対応している。マクロ・コマンドは複数個のコマンドを集めて1つのコマンドとしたもので、組込みコマンドおよびマクロ・コマンドの並びに分解される。

コマンドの定義形はコマンド・ライブラリと呼ばれるファイルの中に置かれ、シンボリック・ライブラリ保守と呼ばれるユーティリティ・プログラムで作成、修正できる。コマンド・ライブラリはシステム・コマンド・ライブラリとユーザ・コマンド・ライブラリからなり、システム・コマンド・ライブラリはユーザ・コマンド、オペレータ・コマンドおよびリモート・バッチ・オペレータ・コマンドの定義形を含んでいる。システム管理者はこのライブラリを取扱うことで各種のコマンドを変更、追加できる。

システム管理者以外のユーザはシステム・コマンド・ライブラリ中のユーザ・コマンドを共通的に使用できるが、さらに自分のコマンド・ライブラリを持つことにより独自のコマンドを作ることができる。

コマンド・ライブラリは図-3 の形式をしており、組込みコマンドは BUILTIN コマンドで定義し、マクロ・コマンドは MCDEF コマンドで始まるコマンド群で定義する。BUILTIN, MCDEF, PARAM はコマンド定義制御コマンドに含まれる。

- (1) BUILTIN コマンドのオペランドでは、この

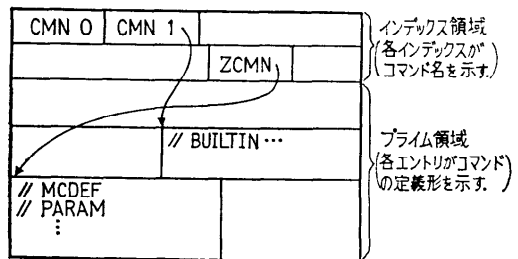


図-3 コマンドライブラリの形式

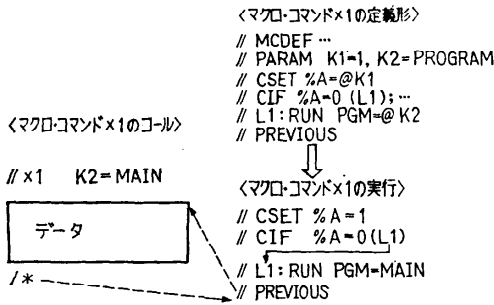


図-4 マクロ・コマンドの実行例

コマンドの属性 (例えばこのコマンドを使用できるユーザの特権レベル, 会話ジョブで使用できるか否か等), オペランドの個数, キーワード名, 順序およびコマンド処理モジュールの名称を指定する。

(2) MCDEF コマンドのオペランドでは BUILTIN コマンドと異なり, このコマンドの属性だけを指定する。MCDEF コマンドに続いて PARAM コマンドを置き, このマクロ・コマンドが持つべきキーワード名を指定する。

ここで指定されたキーワード名は, マクロ・コマンドを実行するときに, 後続するマクロ・コマンドの定義形 (展開形) を適当な値で置換する役割を持っている。PARAM コマンドの後には, すでに定義されているコマンドを並べるが, コマンド記号 (2.3 参照) と呼ばれる記号に値をセットする CSET コマンド, そのコマンド記号の値を判定して特定のコマンド・レベルまでコマンドを読み捨てる CIF コマンド, あるいはマクロ・コマンドの定義形の外側からデータを取る PREVIOUS コマンド等のコマンド定義制御コマンドが用意されている。これらのコマンドの作用を示すのが図-4 である。

2.3 プロファイル機能

コマンド・ライブラリと同様にシステム管理者およびその他のユーザがプロファイルと呼ばれるファイルを持つことができ, プロファイルの内容を変更することでコマンド体系を自分に適した形で使用できる。

プロファイルの中には次のものが格納されている。

(1) 同義語

ユーザはコマンド名やキーワード名に対して SYNONYM コマンドを用いて同義語を与えることができ, 与えた同義語でも元の名前でも記述できる。

(2) 標準値

ユーザはコマンドのキーワードに対して, DEFAULT

表-2 最も簡単な翻訳と実行の例

コマンド	説明
//COMPIL1: JOB UID=TARO, YAMA	ジョブの先頭を意味し, COMPIL1 がジョブ名, ユーザ登録名が TARO でパスワードが YAMA である。
//FG FORTRAN 原始プログラム	FORTRAN の翻訳と実行を行なうマクロ・コマンド。
/* FORTRAN 原始プログラム が実行時に必要とするデータ	データの終り (EOF) を示す。
//END	ジョブの終りを示す。

表-3 翻訳し, 指定したファイル上のオブジェクトと結合し, 実行する例

コマンド	説明
//COMPIL1: JOB UID=JIRO, KAWA	
//FT 01 F 001: DTF FN=B, DEV=MT,...	実行時に使用する磁気テープ・ファイルを定義するコマンド。
//SYSOBJ: DTF FN=OBJECT. FILE,...	結合すべきオブジェクトが入っているファイルを定義するコマンド。
//FLG FORTRAN 原始プログラム	FORTRAN の翻訳, 結合, 実行を行なうマクロ・コマンド。
/* INCLUDE SYSOBJ END	リンケージ・エディタに対するパラメータ
/* FORTRAN 原始プログラム が実行時に必要とするデータ	
//END	

LT コマンドを用いて標準値を与えることができる。コマンドをコールするときに標準値がとられてよい場合は, このオペランドを記述する必要はなくなる。

(3) コマンド記号

ユーザは CSET コマンドを用いてコマンド記号を定義したり, 値を変更できる。コマンド記号は%で始まる8桁以内の記号で, 特にシステムとの連絡のために用意されたシステム予約コマンド記号がある。例えば % COND はコマンドの実行結果のコードを表わす終了コードがセットされるし, % MSGFLT はどの程度のメッセージまでを出力するかをシステムに知らせるために用いられる。

2.4 FORTRAN の翻訳, 実行の例

バッチ・ジョブにおける FORTRAN の翻訳と実行の二つの例を表-2, 表-3 に示す。

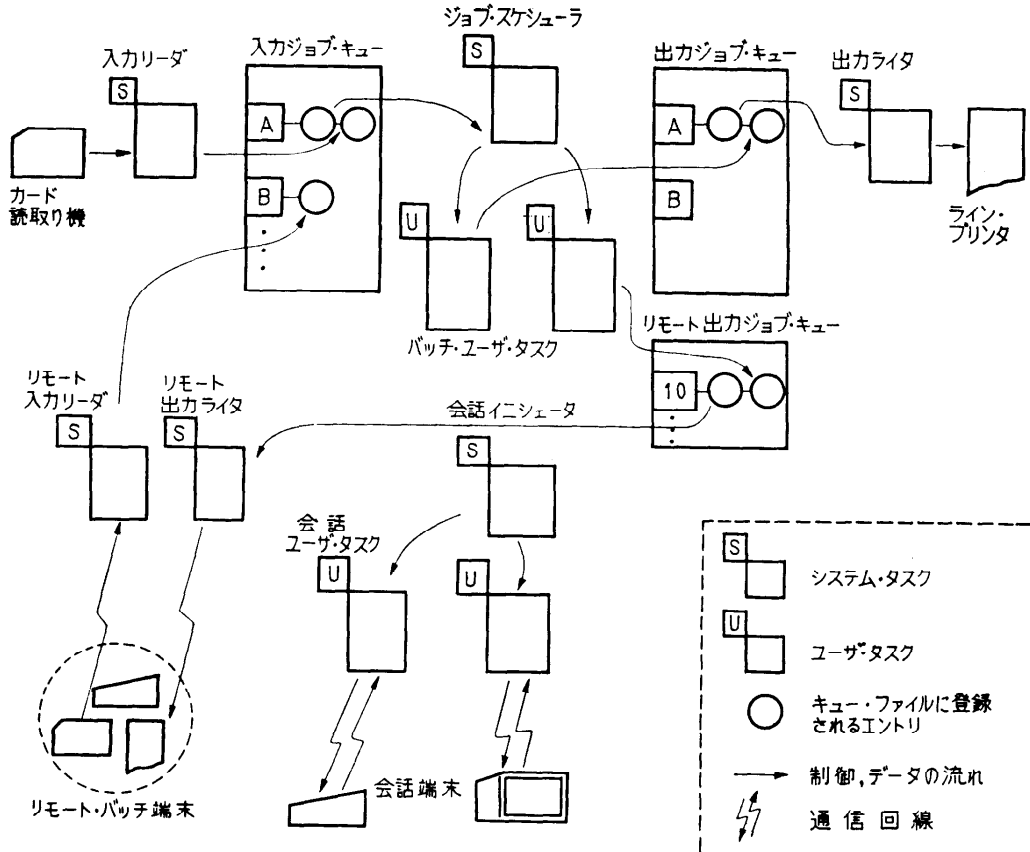


図-5 各処理形態でのジョブの流れ

3. ジョブの投入と起動

ジョブは各処理形態共通に JOB コマンドで始まり END コマンドで終るコマンド群とデータからなっており、センタの入力装置 (カード読取り機等), リモート・バッチ端末の入力装置, あるいは会話端末から投入され, 実行される。ジョブの出力結果はセンタの出力装置 (ライン・プリンタ等), リモート・バッチ端末の出力装置, あるいは会話端末に出力される。このような仕事をするためシステムの中には図-5 のように, システム・タスクとユーザ・タスクが存在する。システム・タスクは入力リーダーや出力ライターのようにシステムの仕事をするタスクで, オペレータ・コマンドによって発生される。一方ユーザ・タスクはジョブ・スケジューラと会話イニシエータと呼ばれるシステム・タスクにより発生される。ユーザ・タスクはユーザのジョブ対応に発生される会話ジョブ, バッチ

・ジョブにかかわらず, このタスク上でコマンドは解析, 実行される。ユーザ・タスクには1つの独立した仮想空間が与えられる。

3.1 バッチ処理

(1) ジョブの入力, 実行および結果の出力

センタの入力装置からのバッチ・ジョブの読み込みは入力リーダーによって行なわれ, ジョブを形成するコマンドおよびデータは一時的なファイルの中にジョブ単位に格納されると同時に, 入力ジョブ・キューに登録される。その後 (2) のようなスケジューリングを経て実行に移される。ジョブの出力結果は一時的なファイルの中に格納されると同時に出力ジョブ・キューに登録され, 出力ライターによってセンタの出力装置に出力される。入力ジョブ・キュー, 出力ジョブ・キューはそれぞれ A~Z のジョブ・クラス, 出力クラスに分れており, 業務内容別の分類等に用いられる。

(2) ジョブ・スケジューリング

ジョブ・スケジューリングは独立の1個のシステム・タスクであるジョブ・スケジューラによって行なわれる。ジョブ・スケジューリングの要素は次のものから成っている。

(a) ジョブ・クラス

各ジョブ・クラスにはシステム・ジュネレーション時に次の値を設定できる。

- ジョブ取出し比率
- スケジューリングの開始、終了時刻
- 各種打ち切り条件の標準値と最大値

ジョブ・スケジューラがどのジョブ・クラスから次のジョブを取出すかはジョブの取出し比率に従う。

システム・ジュネレーション時に各ジョブ・クラスに与えられた比率 r_0 と過去にそのジョブ・クラスから取出したジョブ r の比 r_0/r が最少のジョブ・クラスからジョブを取出す。

システムが標準的に持っているジョブ取出し比率によるジョブ・クラスの選択アルゴリズム以外に、センタ独自のアルゴリズムをシステム・ジュネレーション時に組込むことができる。

ジョブ・クラスは指定されたスケジューリングの開始時刻と終了時刻の間だけでスケジューリングの対象となる。

また緊急に実行したいジョブはジョブ・クラス Z を用いることができるし、一般的に同一クラス内でも複数ジョブが実行されるのに反して、シリアルにしかジョブを実行しないジョブ・クラスも設定できる。

(b) ジョブのプライオリティ

ユーザが JOB コマンドで 0~3 のプライオリティを指定できる。ジョブ・スケジューラは (a) のルールでジョブ・クラスを決めた後、そのジョブ・クラス内の最も優先順位の高い (3 が最も高い) ジョブを取出す。

(c) ジョブの多重度

バッチ・ジョブの多重度はシステム予約コマンド記号 (% SYSMABJ) の値で定まるが、オペレータ・コマンドによる変更もできるし、実メモリ (ページング・ドラムを含んだもの) の空き具合によってダイナミックに変化する。

(d) 同一ユーザ登録名を持つジョブ

OS 7 ではファイルの同時更新を防ぐため同一ユーザ登録名をもつジョブは同時には実行しない。

(e) ノン・セットアップ・ジョブの取出し

入出力資源が不足して待たされているジョブ数が一定量を越えると、セットアップ・ジョブ (私用ボリュームを使用するジョブ) の取出しを一時中止し、ノン・セットアップ・ジョブのみを取出す。

(3) オープン・バッチ処理

OS 7 のバッチ処理は (1) の標準的なクロズド・バッチ処理の他に、ジョブの入力、結果の出力をセンタ利用者に開放し、センタ業務の省力化およびそれによるターン・アラウンド・タイムの短縮を目的としたオープン・バッチ処理機能を備えている。

• ジョブの入力

センタ利用者が自分でカード読取り機からジョブを読ませる。ジョブ読み込み中に入力リーダーがユーザの記述誤まりを検出するとカード読取り機のエラー表示ランプを点灯しカード読取り機を停止させる。

• ジョブ実行の監視

入力したジョブの実行状態に関する情報は、センタ利用者に開放されたコンソール・ディスプレイまたは電光表示板にセンタ・OWN・コーディングを利用して表示できる。このとき OS 7 が持っているマルチ・コンソール機能が活用できる。

• 結果の出力

オープン・バッチ処理用の出力ライタにはトークン・カード読取り機が割当てられており、センタ利用者は自分の出力結果が欲しい時に自分のユーザ登録名が記録されたトークン・カードを読ませることによって結果を入手できる。

3.2 会話処理

会話端末から会話ジョブを起動するには、まず「割込みボタン」を押し、システムからのコマンド要求 (コマンド接頭記号 // が出力される) に対して JOB コマンドを投入し、自分のユーザ登録名等を指定する。正常にジョブが受け付けられれば、システムからの要求に対してコマンドやデータを投入してジョブを実行できる。

会話ジョブは一般に端末からの要求で即座に開始されるが、同時に実行するジョブの上限をシステム予約コマンド記号 (% SYSMATJ) の値で制限することもできる。この値はオペレータ・コマンドによる変更もできるし、実メモリ (ページング・ドラムを含めたもの) の空き具合によってダイナミックに変化する。

会話ジョブではバッチ・ジョブで使用されるコマンドと同じものを使用できるが、特に次の機能がある。

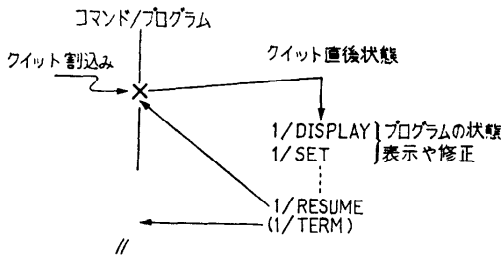


図-6 クイット

(1) 数字パスワードによる機密保護

JOB コマンド投入時にシステムから乱数が表示され、端末ユーザは自分の数字パスワード（マスク、乗数、加数から成り、ユーザ登録簿に格納されている）を用いて演算し、応答した結果によりユーザの正当性が確認される。

(2) バッチ・ジョブの発生

会話ジョブ実行中に、BATCH コマンドを用いて、あらかじめファイル上に用意されたバッチ・ジョブを入力し入力ジョブ・キューに登録できる。

(3) メッセージ・ハンドリング

一般的にシステムから出力されるメッセージに関して、標準形式か拡張形式かの選択、不要メッセージのフィルター、応答方法や説明文の問い合わせ等ができる。

(4) 確認モードの設定

2.3 で述べたようにコマンドのオペランドを省略すると標準値がとられるが、確認モードを設定しておくと、システムが標準値をとる場合は端末ユーザにその値が表示され、端末ユーザは標準値の確認ができ、さらに別の値に修正することができる。

(5) タイム・レポート機能

一定時間内に端末に対してデータの要求もメッセージの出力も行わず、内部演算を実行している場合はセンタとの通信が絶えていないことを確認するため端末のベルが鳴らされる。

(6) クイット機能

コマンドあるいは自分のプログラムを実行中に端末から「割込みボタン」を押して割込みをかけ、実行中のコマンド、プログラムを一時中断することができる。これをクイットと呼び、クイットが受けられると図-6 のようにクイット直後状態となり、コマンドを投入してクイットで中断されたプログラムの状態の参照あるいはプログラムの修正等ができる。

クイット直後状態で RESUME コマンドを投入すると、クイットにより中断されていたコマンド、プログラムは中断点から再実行される。また TERM コマンドを投入すると、クイットにより中断されていたコマンド、プログラムは異常終了する。

このようにクイット機能を用いることにより、ジョブの実行の監視および実行の変更が可能となる。

3.3 リモート・バッチ処理

リモート・バッチ端末は次の2種類に分類される。

(1) CCAM 端末 (Conversational Communication Access Method)

一般的にはタイプライタ、カード読取り機、ライン・プリンタを備えた複合端末で、リモート・バッチ処理を開始するには会話ジョブと同様に「割込みボタン」を押す。この割込みを受けると会話イニシエータはリモート・バッチ処理を開始するか、会話処理を開始するかをユーザに問合せる。

リモート・バッチ処理とユーザが答えた場合図-5 のようにそのリモート・バッチ端末に対して1組のリモート入力リーダーとリモート出力ライターを発生し、以後はこれらのタスクによりジョブの入力、結果の出力が行なわれる。

リモート・バッチ端末上での操作のために用意されているリモート・バッチ・オペレータ・コマンドは、次の2通りの方法で入力できる。

- 端末の「割込みボタン」を押し、出力されたコマンド接頭記号 (..) に対してコマンドを投入する。
- カード読取り機などから、ジョブ・ストリームと一緒に投入する。この場合レコードの先頭のコマンド接頭記号は (..) を指定する。

(2) RCAM 端末 (Remote Computer Communication Access Method)

一般には端末が小型のコンピュータである。起動は端末コンピュータから交信開始信号を送出することにより行なわれる。RCAM 端末に対しても CCAM 端末と同様のタスクが発生される。リモート・バッチ・オペレータ・コマンドの投入は、端末コンピュータのコンソールまたはカード読取り機から投入（ジョブ・ストリーム中）できる。

RCAM 端末の特徴的な機能は次のとおりである。

- 送受信データ中の連続した同一文字の圧縮機能
- 送受信データのブロッキング機能
- バイナリ・データの送受信機能

これらの機能によって処理能力の向上が図られてい

る。

CCAM 端末あるいは RCAM 端末から入力されたジョブはバッチ処理と同じ入力ジョブ・キューに登録され実行されるが、出力結果はバッチ処理と異なりリモート出力ジョブ・キューに登録される。リモート出力ジョブ・キューへ登録された出力結果は、即座にリモート・バッチ端末に出力することも、リモート・バッチ端末からのコマンドで指示があるまで置いておくこともでき、どちらにするかはジョブ入力時に指定できる。

3.4 プログラム・ライブラリの扱い

プログラム・ライブラリには次のものがある。

(1) システム・ライブラリ (SYSLIB)

言語プロセッサ、ユーティリティ・プログラム等の標準システム・プログラムを集めたものでシステムに1つ存在する。

(2) ユーザ・ライブラリ (USERLIB)

ユーザ対応に存在する私用ライブラリで、そのユーザのジョブ開始時にプログラム・ライブラリとして有効になる。USERLIB コマンドで定義する。

(3) ジョブ・ライブラリ (JOBLIB)

ユーザがジョブ実行中に必要なファイルを JOBLIB コマンドで指定したもので、そのジョブ中プログラム・ライブラリとして有効である。1ジョブ中に複数のジョブ・ライブラリを指定できる。

(4) 一時的ライブラリ

コンパイル・アンド・ゴーのオブジェクト・プログラムのライブラリのように一時的に作成、実行される。これらのライブラリ中のプログラムは RUN コマンド、あるいは組み込みコマンドで実行できる。

プログラムのユーザ間での共用はこれらのプログラム・ライブラリというファイル自身の共用で実現される。すなわちプログラム・ライブラリとなっているファイルの共用が許されれば、プログラムの共用が可能となる。

4. ファイルおよび入出力資源の使用法

プログラムから各種のデータを参照するにはファイルという論理的な単位で取扱う。プログラム実行時にそのファイルに対応した物理的なファイル実体の存在位置、大きさを指定するのが DTF コマンドで、図7のような関係になる。

図7のようにプログラム上ではファイルを定義するのに FCB という制御ブロックを用いる。この FCB

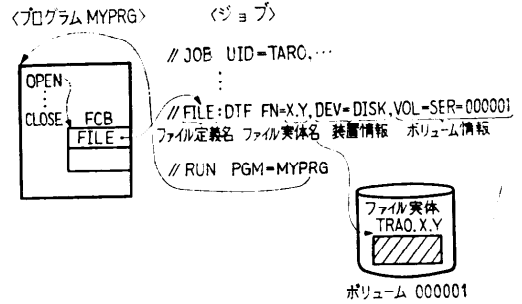


図-7 ファイルとファイル実体の関係

をオープン (OPEN) すると DTF コマンドを経由してファイル実対との対応がとられ、物理的なファイル実体へのアクセスが可能となる。ファイルの使用を終了するときは後始末のためクローズ (CLOSE) する必要があるが、ユーザーがクローズしなかった場合にはプログラムの終了時に管理プログラムが自動的にクローズする。

4.1 ファイル実体の扱い方

ファイル実体を扱う場合ジョブ・ステップが重要な役割を果たす。ジョブ・ステップは、ジョブを複数の作業単位に分割した単位で、STEP コマンドにより指定される。STEP コマンドのオペランドではそのステップに対する打切り条件 (JOB コマンドで指定したのと同種のもの) を設定できる。

DTF コマンドでファイル実体を定義する場合、DISP オペランドでファイル実体の初期処理方法およびジョブ・ステップ終了時の処理方法を指定する。ファイル実体の初期処理方法には、新規作成 (NEW)、追加更新 (MOD)、入力 (OLD)、全面更新 (RNW) があり、新規作成の場合は 4.2 で述べる方法で装置、ボリュームが割当てられた後ファイル実体が作成される。ファイル実体に必要なスペースは SPACE オペランド (あるいはセンタの定めた標準値) で指定する。DTF コマンドで定義したファイル実体は、一般的にそのジョブ・ステップ内だけで有効であり、ジョブ・ステップ終了時にはユーザの指定した終了処理方法に従って処理される。処理方法には、実体消去 (DELETE)、保存 (KEEP)、カタログ (CATLG)、カタログ消去 (UNCATLG) がある。またジョブ・ステップの終了時にこのような処理をしないで、次のジョブ・ステップにパス (PASS) することもできる。

ファイル実体は一般的に図7のように、ユーザが指定した名称の前にユーザ登録名を付けた形の名称が

付けられるので、一般に他人はそのファイル実体を参照できないがカタログされたファイル実体に関しては所有者が PERMIT コマンドで共用を許したユーザは参照できる。この場合、ファイル実体の同時更新が発生しないように DTF コマンドでチェックされる。

4.2 入出力資源の割当て方法

入出力資源は、装置（デバイス）、ボリューム、ファイル実体に分けて管理されており、DTF コマンドの要求に従って割当てられる。

各装置にはシステム・ジェネレーション時に装置記号名（例えば T 00, K 03 等）を付けることができ、さらにランダム・アクセス装置群、磁気テープ装置群のように複数の装置をグループとしてまとめ、これに装置グループ名（例えば、DISK, MT 等）を付けることができる。ユーザは DTF コマンドの DEVICE (DEV) オペランドに装置記号名または装置グループ名を指定できる。装置グループ名を指定するとシステムがグループ内の適当な装置を割当てる。

ボリュームは次のように分類され、ファイル実体がどのボリュームにとられるかは、DTF コマンドの VOLUME (VOL) オペランドの指定、あるいは FN-AME (FN) オペランドの指定で決められる。

(1) 公用ボリューム

ランダム・アクセス・ボリュームで、一般に常時マウントされており、各ユーザによって共用される。どのボリュームを公用ボリュームとするかは、システム管理者が公用ボリューム登録簿と呼ばれるファイルにボリューム通し番号をセットすることで定まり、システム開始時にそのボリュームがマウントされていることをシステムが確認する。またシステム運転中にオペレータがオペレータ・コマンドにより公用ボリュームを追加することもできる。

公用ボリュームは、さらに次の2種類に分けられ、新しくファイル実体を作成するときどのボリューム上に作成するかはファイル実体の属性によって決まる。

(a) 一時的ファイル用ボリューム

入力カーダがジョブを格納する領域、ジョブの実行時に使用される一時的なファイル実体（コンパイルの作業用ファイル等）の領域、およびジョブの出力結果を格納する領域が割当てられる。あるボリュームの中に上記3種類のどの領域とどの領域を割当て可能とするかは公用ボリューム登録簿またはオペレータ・コマンドでも指定される。これらの領域にとられたファイル実体はジョブ（ジョブ・ステッ

プ）終了時、あるいは結果出力終了時に消去される。

(b) 恒久的ファイル用ボリューム

ジョブ終了後も保存すべきファイル実体を格納する領域で、ファイル実体は常にカタログして保存される。

(2) 私用ボリューム

ユーザ固有のボリュームであり、DTF コマンド実行時にオペランドで指定したボリュームのマウント動作がオペレータに対して要求される。私用ボリューム上にとられるファイル実体はジョブ終了時に消去することも、ファイル実体を保存 (KEEP) することも、あるいはカタログすることもできる。

(3) 作業用ボリューム

センタで用意しておく磁気テープ・ボリュームで、DTF コマンドでボリューム通し番号を指定する必要はなく、センタ・オペレータが用意しておいたボリュームのうちから任意のものをマウントする。

作業用ボリューム上にとられたファイル実体は、ジョブ（ジョブ・ステップ）終了時に消去され、カタログすることはできない。

公用ボリューム以外はボリュームのマウント（取付け）が必要となるが、通常は DTF コマンド実行時にシステムからメッセージを出力し、オペレータがマウント動作を行なう必要がある。その他に、ボリュームのマウントを確認せずファイルがオープンされるまで先に進む方法や、自動ボリューム認識によるマウントがある。自動ボリューム認識とは、これから必要となるボリュームをオペレータがあらかじめ知っている場合に前もってボリュームをマウントしておき、そのボリュームを要求するジョブが開始されたとき、それに割当てる方法である。

公用ボリュームは（1）の（a）と（b）のように分類されるが、これらの属性を持ったボリュームが複数個存在する場合は、新たにファイル実体を作成するときに、各ボリュームの過去の I/O 回数に基づいて特定のボリュームに I/O が集中しないように割当てている。

参考文献

- 1) 大西助他: HITAC 8700/8800 オペレーティング・システム (OS 7), 情報処理, Vol. 14, No. 10, pp. 769~777 (1973).
- 2) 日立: OS 7 管理プログラム解説 (プログラムマニュアル, 8700-3-003-21 (1974)
(昭和 49 年 5 月 9 日受付)