九州大学学術情報リポジトリ
Kyushu University Institutional Repository

[01_01]情報処理教育広報表紙奥付等

https://hdl.handle.net/2324/6766030

出版情報：情報処理教育広報. 1 (1), 1978-03-20. Educational Center For Information Processing, Kyushu University
バージョン：
権利関係：

KYUSHU UNIVERSITY

資　料

センター利用の案内

ここでは，センターを利用する上で知っておいて頂きたい事項についてその概要を述べます。

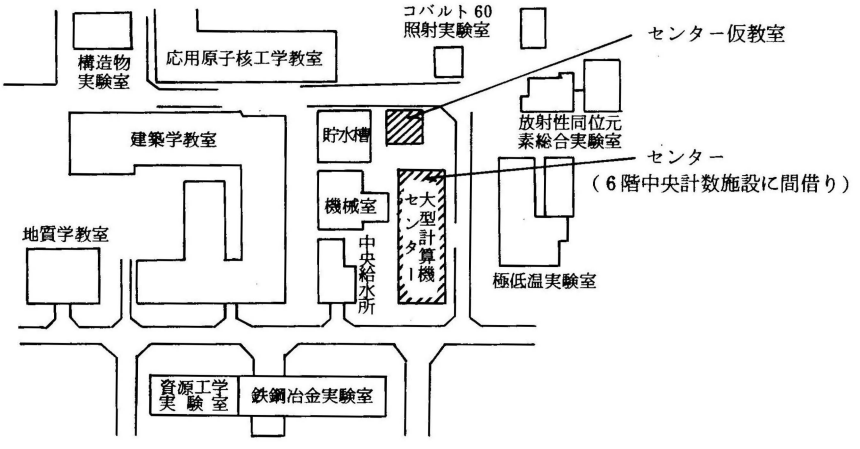
1　センターの位置と名称

(1) 所在地　（箱崎地区）〒812 福岡市東区箱崎6丁目10番1号

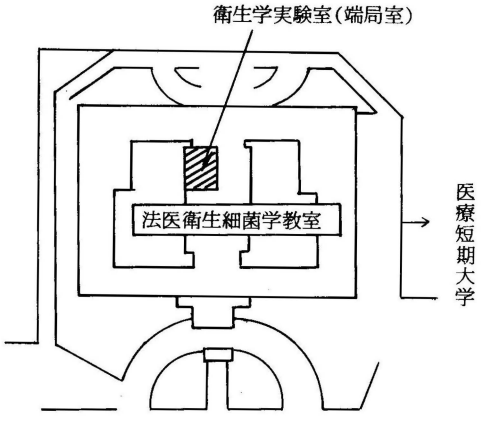
(2) 名　称　九州大学情報処理教育センター

Educational Center for Information Processing, Kyushu University

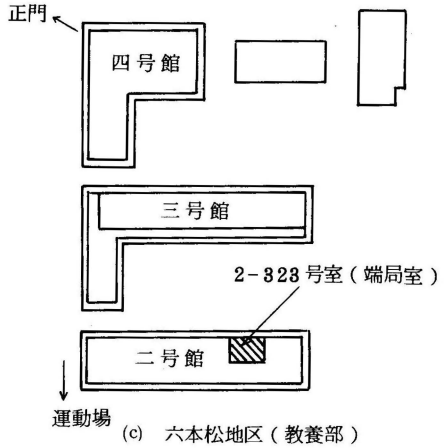
(3) 位　置　図1.1に示すように，センター（箱崎地区）は大型計算機センターが設置されている場所に仮設されており，端局は，堅粕地区（医歯薬学部）と六本松地区（教養部）に設置されています。



(a)　箱崎地区



(b)　堅粕地区（医歯薬学部）

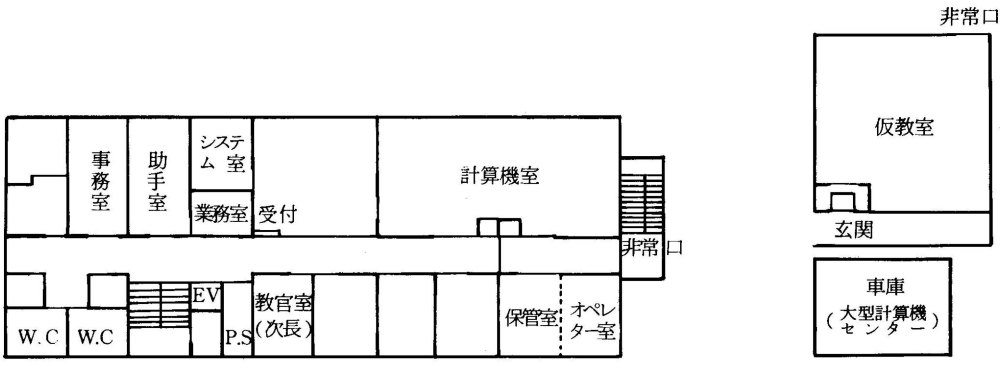


(c)　六本松地区（教養部）

図1.1　センターの位置図

2. センター（箱崎地区本部）の建物

センター建物新設の予算が，52年度は残念ながら認められませんでしたので，下記のよう講義・演習用の教室(センターでは情報処理教室と呼んでいます）が仮設され，その他の施設は中央計数施設に間借りしています。



中央計数施設（大型計算機センター6階）平面図

図2.1　センター平面図

3. センターの性格

センター規則第2条（→規則）で謳っていますように，センターは情報処理教育の用に主として供するための学内共同利用施設であります。従って，センターの職員が情報処理教育に関した授業を直接担当することはありません。

4. センターの組織（→組織）

センターは事務組織をもたないため，センターに関する事務は当分の間工学部事務部において処理して頂いております。また，センター職員の数も十分でないため，センターの利用に関しては利用者の方々の積極的な支援と協力をお願いすることになると思われます。

なお，センターには運営に関してセンター長の諮問に応ずるために，各部局の代表者から構成される運営委員会が置かれています。

5. センターの設備

センターの設備は

(1) 日本電気㈱製　ACOSシリーズ77 NEACシステム600電子計算機システム一式

(2) ソニー・テクトロニクス㈱製　グラフィク・ディスプレイ一式

から構成され，月間賃借料予算（52年度実績）は5,778千円です。表5.1に教育設備の概要を示します。

表5.1　センター教育設備

キャンバス名	六本松（教養）	堅　粕（医歯薬）	箱　崎（文科系・理農工）	
			情報処理教室	計　算　機　室
配　置　機　器	• ミニコンピュータ • データ・ステーション （2,400ビット／秒）	データ・ステーション （2,400ビット／秒）	• CRTディスプレイ　40台 • ハイプリンタ　3台 • グラフィク・ディプレイ　6台 • ハード・コピー1台	• FACOM 230-45 S 注1) • ACOS NEAC 600 • 漢字端末
利　用　形　態	• バッチ • リモート・バッチ	• バッチ • リモート・バッチ	会話型	バッチ 注2)
入　力　方　法	マーク・カード （ハード的にはパンチ・カードも読取り可）	マーク・カード （ハード的にはパンチ・カードも読取り可）	キーボード	• マーク・カード • パンチ・カード
利用可能言語	アルゴリズム教育用 ＊ FORTRAN ＋ COBOL ＋ PL／I ＊ ALGOL ▫ BASIC ▫ ABACUS ＋ GMAP（アセンブラ） 図形処理教育用 ＊ FORTRAN { X−Y plotter { Graphic Display シミュレーション教育用 ＋ GPSS（離散型） ＋ DYNAMO（　〃　） ＋ CSPL（連続型） データ・ベース教育用 ＋ IDS ＋ ADBS（CODASYL仕様） ＊ INQ	ただし ＊ バッチ，リモート・バッチ，会話型 ＋ バッチ，リモート・バッチ ▫ 会話型 ○ バッチ で利用可能であることを示す。 LISP（53年4月） SNOBOL（予定） PASCAL（予定）		○ FORTRAN ○ COBOL ○ PL／I ○ ALGOL ○ LISP ○ PASCAL ○ SNOBOL ○ FASP（アセンブラ） ○ GMAP（アセンブラ） ○ FORTRAN X−Y plotter

注1）バッチ処理には，今までどおり中央計数施設のFACOM 230-45 Sが使用できます。

注2）ACOS NEAC 600によるバッチは，学生教育用にはほとんど利用できない可能性もあります。

6．利用の際の諸注意 注)

6.1 利用者の資格

センターを利用することができる者は，次の(1)～(6)に掲げる方に限ります。

(1) センターの計算機を使用する授業科目を履修する本学の学部学生及び教養部学生（以下「学部学生」と略）。

(2) センターの計算機を使用する授業科目を履修する本学の大学院学生

(3) センターの計算機を使用する授業に携わる本学の教官（以下「授業担当教官」と略）及びこれを補佐する本学の職員（以下「授業補佐職員」と略）

(4) センターで開催される講習会等に参加する本学の職員（以下「講習会参加者」と略）

(5) 本学の学生及び職員で，利用の成果がセンターの運営・開発研究に寄与するとセンター長が認めた者（以下「開発研究者」と略），ただし学生の場合には指導教官の許可が必要

(6) その他特にセンター長が適当と認めた者

以上述べた方が有資格者です。特に注意して頂きたいことは，センターは情報処理教育の用に主として供するため学内共同利用施設であるため，一般の研究用としては利用成果がセンターに寄与するものに限ってしか認めない点です。

6.2 利用の手続

センターを利用しようとする際には，あらかじめ次の手続きが必要です。

(1) 授業の場合

① 授業計画書の提出（→様式１）

毎年12月までに，次年度の授業計画書を各部局を通じてセンターに提出して下さい。

② 授業（利用）時間割の通知

センターでは，これらの計画書に基づいて授業（利用）時間割を編成し，その結果を各部局あて通知します。演習のみに利用する場合には，原則として全学部共同利用時間帯で使用して頂くことになりますが，この場合においても利用時間帯をセンターが特に指定することがあります。

③ 利用申請書の提出（→様式2.1～2.2）

授業担当教官は，授業補佐職員と受講学生の分を取りまとめて，必要事項の記載された利用申請書を遅くとも利用開始日の一週間前までに一括してセンターに提出して下さい。なお，受講学生の利用申請書はマーク・カード。（→様式2.2）

④ 利用のための登録番号の通知

センター長は，この利用申請を承認すると登録番号を発行し，授業担当教官と補佐職員については利用申請書に登録番号を付して，受講学生については登録番号を一覧表にして授業担当教官に通知します。

(2) 講習会などの場合

① 講習会計画書の提出（→様式１）

センターには教室が一つしかありませんので，いつでも講習会のための利用を認めることはできません。そのため，講習会を行おうとする場合には，講習会開催責任者は，あらかじめ開催時期をセンターと協議した上で講習会計画書をセンターに提出して下さい。

② 講習会利用時間帯の通知

センターでは，計画書に基づいて講習会のための利用時間帯を設定して，その結果を講習会開催責任者あて通知します。

③ 利用申請書の提出

講習会開催責任者は，講習会の講師と受講者の分を取りまとめて，必要事項の記載された利用申請書を遅くとも開催日の一週間前までに一括してセンターに提出して下さい。なお，受講者の申請書はマーク・カードです。（→様式2.2）

注）この章で述べる利用原則などは，運営委員会の最終的な承認を受けておりません。大綱は了承済みのものですが，細かい点については多少変更があることをお含みおき下さい。

④ 利用のための登録番号の通知

センター長は，この利用申請を承認すると登録番号を発行し，講師については利用申請書に登録番号を付して，受講者については登録番号を一覧表にして講習会開催責任者に通知します。

(3) その他の場合

① 利用申請書の提出

利用申請書に必要事項を記入してセンターに提出して下さい。その際，開発研究者については，その課題がセンターに寄与する理由を詳しく具体的に記して下さい。

② 利用申請に対する通知

センター長は，この利用申請を承認した場合には登録番号を，承認しなかった場合にはその理由を利用申請書に付して，その旨申請者に通知します。

6.3 利用の期間

利用の期間は次のとおりです。

(1) 学部学生並びに大学院学生については，当該授業科目の期間

(2) 授業担当教官及び授業補佐職員については，次学期の授業が始まる半年前より一年以内の期間

(3) 講習会参加者については，当該講習会の期間

(4) その他の利用者については，当該年度内で利用者が申請した期間

6.4 利用内容の変更

利用の期間が規定の期間を超える場合などを含め，利用内容が利用申請書に記載された事項と異なる事情が生じた場合には，改めて利用申請書を提出して下さい。

6.5 利用の制限

センターは，その機能が著しく低下する恐れがある場合または利用者の公平なサービスが損なわれる恐れがある場合には，各利用者に対して利用全体および1回当たりの利用で使用できる計算機使用量を制限することがあります。

6.6 目的外使用の禁止

言うまでもありませんが，利用者は，登録番号を当該課題以外のために使用し，または第3者に使用させてはなりません。

6.7 利用の報告

授業に関しては授業担当教官，講習会に関しては講習会開催責任者，その他においては各利用者は，センター長の求めに応じて当該課題に係る利用の報告書を提出して下さい。

6.8 経費の負担

センターの運営費が予算を超過する場合には，運営委員会において差額の負担方法を決定することになります。センターでは，53年度は負担金を徴収しない方向で検討中ですが，利用申請の際には，支払責任者登録手続きも行って下さい。

なお，端局の経費は，一部教育用計算機消耗品を除き端局側の負担とします。

6.9 利用の形態

センターに設置されているNEAC 600計算機システムのオペレーティング・システムACOS－6には，センター・バッチ，リモート・バッチ，会話型，会話型リモート・バッチの処理形態が用意されています。センターでは，計算機の処理能力や運用方法などを検討した結果，利用形態としては

(1) 情報処理教室では会話型

(2) 端局ではリモート・バッチ

のオープン利用方式を原則とします。これ以外の形態でどうしても利用したい場合には，利用申請時にその旨記して下さい。

なお，多人数の学生教育をセンター・バッチで行いたい場合には，従来どおり中央計数施設のFACOM 230－45 Sが利用できます。

6.10　52年度の利用から

53年1月から3月までを試験運用期間として，センターの情報処理教室を講義・演習用に開放しましたが，特に演習に関して幾つかの問題点が上っています。センターではこれらの解決に鋭意努力を続ける所存ですが，授業を担当される先生方にも御協力を頂きたいと考えています。

(1) 演習態度について

演習の基本的態度の身についていない学生が中にはいるようです。演習を行う前に心得なども御指導下さい。

(2) 登録番号とパスワードについて

学生の中には，登録番号とパスワードに対する認識の浅い者がいるようですので，この点を厳しく御指導下さるようお願いします。もちろん，これについてはシステムの側で考えなければならないことも多いようです。

(3) 端末の占有について

端末の数は，学期当たりに換算しますと15人～20人に1台位の割合でしか割当てられません。その結果，1人が，長時間，端末を占有することになると全体の公平なサービスが期待できなくなります。センターでは，これに対処するため次のような対策を講じて行く計画です。

① m分端末，n時間端末の設置

すなわち，端末を2つのグループに分け，m分しか使用できないものとn時間まで使用できるものを設け，午後の混雑時に適用する。m，nの値は検討中。

② マーク・カードなどによるプログラムの初期投入

端末を占有しなければならない原因の一つは，プログラムの投入に時間がかかることである。プログラムの修正や実行のみであれば，もう少し占有時間が短かくなると予想されるので，プログラムの初期投入をマーク・カードなどによっても行える運用体制を検討中である。

③ 計算機による使用時間の管理

1人の利用者が長時間端末を占有できないように，計算機の中で1日当たり，1週間当たりに使用できる時間（端末接続時間）を設定し管理する。

(4) 会話型処理と論理的思考

会話型処理ですぐに応答が返ってくることになると，とにかく先ずやってみることが先に立ち，じっくりとアルゴリズムを考えることなど論理的思考の習慣がおろそかになるのではないかと少しばかり心配になりました。

(5) 演習の問題とレポートの出力

レポート出力は教室のプリンタからも行えるように準備しています。演習で学生に計算機を使わせる場合には，問題の選択，演習の方法など今まで以上の配慮が望まれます。

以上，気のついた主な事柄を取上げてみましたが，この施設を十分に使いこなすためには，施設を提供する側はもちろんですが利用する側にも十分な準備が必要だと思われます。

7. 問　合　せ

センターの利用，運営に関しては，下記あてお問合せ下さい。

(1) 利用に関して　　センター受付　　内線2268

(2) 運営に関して　　センター次長　　内線4564

なお，電話によるプログラムの相談は御遠慮下さい。

8　教育講習会について

センターは，必要がある場合には授業担当教官および授業補佐職員向けに教育講習会を開催します。

原則として，各学期の始まる前一度ずつを目度に開催する計画を立てています。

9. 利用者への資料

9.1 センター発行の資料

(1) 広　　報　　年2回，3月と9月に発行予定

(2) 利用の手引　　発行を検討中

(3) ニュース　　必要に応じて発行予定

(4) センター案内　3月に発行予定

広報の配布を希望される方は，

- 氏　　名
- 職　　業
- 所　　属
- 広報発送先
- 電話番号

を記して

〒812　福岡市東区箱崎6－10－1

九州大学情報処理教育センター

までお申込み下さい。

9.2 利用者マニュアル購入方法

利用者マニュアルは，日本電気株式会社が作成したものですが，

〒812　福岡市東区箱崎6－10－1

九州大学生協書籍部（内線6117）

にて購入することができます（→マニ）。

センターの計算機システム概説

1. ACOS 600 ハードウェア・システム概説 …… 14
   1.1 ハードウェア概説 …… 14
   1.2 データ表現形式 …… 23
2. ACOS 600 ソフトウェア・システム概説 …… 27
   2.1 基本ソフトウェアの構成 …… 27
   2.2 基本ソフトウェアの概要 …… 28
   2.3 処理形態と処理システム …… 29
   2.4 プログラミング言語の概要 …… 35
   2.5 ファイル・システムの概要 …… 36
   2.6 プログラムの形式とファイル …… 37
   2.7 データ・ベース・システムの概要 …… 38
   2.8 応用ソフトウェアの概要 …… 38

センターが導入したACOSシリーズ77 NEACシステム600（以下「ACOS 600」と略称）のハードウェアとこのオペレーティング・システムACOS-6について概要を紹介する（→マニ1）。センター計算機システムに関する性能の比較，機器構成などについては付録を参照されたい（→付表6，付図1.1）。

1. ACOS 600 ハードウェア・システム概要

1.1 ハードウェア概要

ACOS 600 は下記の各装置によって構成される。

(1) システム制御装置　　SCU (System Control Unit)
(2) 主記憶装置　　MMU (Main Memory Unit)
(3) 中央処理装置　　CPU (Central Processing Unit)
(4) 統合処理装置　　ICU (Integrated Control Unit)
(5) 周辺装置
(6) 通信処理装置　　FNP (Front-end Network Processor)
(7) データ通信機器
(8) 電源制御装置　　PSQ (Power Sequencer)

以下これらのうちセンターに設置されている主な装置についてその特徴，機能，仕様および性能などを簡単に説明してゆく。

1.1.1 システム制御装置

システム制御装置 (SCU)はACOS 600 の中枢であり，主記憶装置と，中央処理装置および統合処理装置などの能動モジュールとを接続し，これらの間のデータ読書きやデータ転送などを制御する。データは 8 バイト単位で転送される (→表 1.1)。

表 1.1　システム制御装置の構成・性能

型　　番	N 7161-50
システムポート数	3
能力モジュール数	3 個のシステムポートはCPU，ICU，EMP (Emulator Processor) が割当てられる。
メモリポート数	最大　4
接続可能主記憶容量	最大 2,048 K バイト

1.1.2 主記憶装置

主記憶装置 (MMU) はMOS-LSI 素子を使用した高速記憶装置であり，データ読取りの際，1 ビットの誤りは自動訂正し，2 ビットの誤りは完全に検出することができる (→表 1.2)。

表 1.2　主記憶装置の仕様・性能

型　　番	N 7161
サイクルタイム	0.7 マイクロ秒／8 バイト
最小記憶容量	512 K バイト
最小記憶容量	2,048 K バイト
素　　子	MOS-LSI
誤り検出方式	ECC (データビット　72 ビット / 冗長ビット　8 ビット)

1.1.3 中央処理装置

中央処理装置 (CPU) は，200 種を超える命令群を処理し，命令の実行に当たってはキャッシュ・メモリ(容量 8 K バイト，サイクル・タイム 400 ナノ秒) やパイプライン制御の採用により高速演算処理ができる (→表 1.3)。

CPUが取扱う命令形式を表1.4に，CPUがもっているレジスターの一覧を表1.5に示す。

表1.3　中央処理装置の仕様・性能

型　　番	N 7061
最大システムポート数	1
キャッシュメモリ	あり（8Kバイト）
語　構　成	4バイト（1バイト＝9ビット） 36ビット／語
命　令　形　式	基本命令（1語） 複数語命令（3～4語）
アドレス形式	0，1，2，3アドレス方式
アドレス修飾	直接，間接，相対，インデックス，タリー
仮　想　記　憶	ページング／セグメンテーション

表1.4　命令形式

命令形式	命令の仕様		取扱うデータの形式	
	長　さ	アドレス方式	長　さ	表　現
基本命令	1語 （36ビット）	1アドレス	固定長	①2進数（固定／浮動） ②文字（6ビット）
複数語命令	3語 または 4語	2アドレス または 3アドレス	可変長	①10進数 （固定／浮動） （パック／アンパック） ②文字 （8ビット／6ビット） ③ビットストリング

表1.5　レジスター覧

レジスタ名称	略　称	レジスタビット長
アキュムレータレジスタ	A Q	72ビット
インデックスレジスタ（8個）	Xn	各18ビット
指数レジスタ	E	8ビット
ベースアドレスレジスタ	BAR	18ビット
インディケータレジスタ	IR	18ビット
タイマレジスタ	TR	27ビット
インストラクションカウンタ	IC	18ビット
アドレスレジスタ（8個）	ARn	各24ビット
ベース拡張レジスタ	BER	6ビット
マスタモード　BAR　A	MBA	15ビット
マスタモード　BAR　B	MBB	15ビット

1.1.4　統合処理装置

統合処理装置（ICU）はすべての入出力に関する処理を一つの装置に統合することにより実装上のコンパクト化を図ったものである。この入出力処理部は，最高24本の論理チャネルを通じて，周辺装置や操作卓を

を制御する（→表 1.6）。これらのチャネルには次の 4 種類の入出力チャネルがある。

(1) MPC（Microprogrammed Peripheral Controller）用入出力チャネル

磁気ディスク処理部 ―――――― 磁気ディスク装置

磁気テープ処理装置 ―――――― 磁気テープ装置

ユニット・レコード処理部 ―――――― カード読取装置・ラインプリンタなど

(2) PTC（Paper Tape Controller）用入出力チャネル

(3) 操作卓用入出力チャネル

システム操作卓，ステータス・ディスプレイ

(4) FNP用入出力チャネル

通信処理装置

表 1.6　統合処理装置の仕様・性能

性能 ＼ 型番	N 7 2 2 3 － 0 1
データチャネル数	最大24チャネル
総合転送能力	4.4 Mバイト／秒
データチャネルの転送速度（データ転送中）	① MPC用入出力チャネル　1.6 Mバイト／秒（8ビット） ② PTC用入出力チャネル　650 Kキャラクタ／秒（6ビット） ③ 操作卓用入出力チャネル　9,600 ビット／秒（1ビット） ④ FNP用入出力チャネル　1 Mバイト／秒（36 ビット）
接続周辺装置	• N 7441，N7445 磁気ディスク装置　4台 • N 7333，N7334，N 7337，N 7338，N 7341 ラインプリンタ　4台 } 合計4台 • N 7436，N 7445 カード読取装置　2台 } 合計4台 • N 7471 カードせん孔装置　2台 } 合計4台

1.1.5 周辺装置

ACOS 600 周辺装置の中で，センターに設置されている装置には次のものがある。

(1) システム操作卓　SCC（System Control Console）

(2) 磁気ディスク装置　DKU（Disk Unit）

(3) 磁気テープ装置　MTU（Magnetic Tape Unit）

(4) カード読取装置　CR（Card Reader）

(5) ラインプリンタ　LP（Line Printer）

1.1.5.1 システム操作卓

システム操作卓（SCC）により，システムの制御，システムとの会話，稼動状況の表示，ハードウェアの監視などシステム全体を集中的に管理できる（→表1.7）。

表1.7 システムの操作卓の仕様・性能

種類	名称	型番	仕様・性能
基本部	システム操作卓	N 7208-21	•操作パネル 各種スイッチ，表示ランプ，メータ •キーボード グラフィック・キー，ファンクション・キー，コントロール・キー •ディスプレイ 14インチカラーCRT，表示文字数80桁×24行＝1920字 •フロッピー・ディスク 転送速度 31Kバイト／秒
付加機構	ステータス・ディスプレイ	N 9208-30	基本部ディスプレイに同じ
付加機構	シリアル・プリンタ	N 9208-15	印字速度 180字／秒，印字文字種 96種，1行印字数80字／行，操作卓ディスプレイまたはステータス・ディスプレイの内容のハード・コピーをとる

1.1.5.2 磁気ディスク装置

センターに設置されている磁気ディスク装置（DKU）N 7745の仕様・性能を表1.8に示す。

1.1.5.3 磁気テープ装置

センターに設置されている磁気テープ装置（MTU）N 7621の仕様・性能を表1.9に示す。

表1.8 磁気ディスク装置の仕様・性能

項目	仕様・性能
記憶容量／スピンドル（1レコード／トラックの場合）	200Mバイト
記憶容量（フォーマット済）	
バイト数／セクタ	288
バイト数／トラック	11,520
バイト数／シリンダ	218,880
バイト数／パック	177,511,680
ビット数／バイト	8
セクタ数／トラック	40
トラック数／シリンダ	19
シリング数／パック	808
呼び出し時間（ミリ秒）	
シーク時間（隣接トラック）	10
シーク時間（平均）	30
シーク時間（最大）	55
平均回転待ち時間	8.5
データ転送速度（Kバイト／秒）	806

表1.9 磁気テープ装置の仕様・性能

項目	仕様・性能
トラック数	9
テープ走行速度（cm／秒）	191
記録密度（列／2.54cm）	1,600／800
データ転送速度（Kバイト／秒）	120／60
ブロック間隔（mm）	15

表1.10 カード・リーダの仕様

項目	仕様・性能
使用カード	80欄カード
読取り速度	600枚／分
読取り方式	光電式
ホッパ容量	1,000枚
スタッカ容量	1,000枚

1.1.5.4 カード読取装置

センターは主として教育用であるためマーク・カード読取機能をもつカード・リーダ（CR）N 7436-51を

設置している。この仕様・性能を表1.10に示す。

1.1.5.5 ラインプリンタ

センターに設置されているラインプリンタ（LP）N 7333-51の仕様・性能を表1.11に示す。

表1.11 ラインプリンタの仕様・性能

項目		仕様・性能		
データ形式	印字密度	（横方向）10字／25.4㎜，（縦方向）6または8行／25.4㎜		
	印字字数	144字／行		
	字形	OCR-Bフォント，OCR-Kフォント		
印字速度		字種	印字ドラム	印字速度（行／分）
		111	標準ドラム	1,000（17字）～500（111字）
改行速度		1,270 ㎜／秒		
印字用紙		長さ355㎜，幅127～457.2㎜，複写枚数6枚（オリジナルを含む）		
動作モード	Edit mode	指定された印刷様式に従って印刷（ブランク，紙送り制御）		
	Not Edit mode	コンピューターの内容をそのまま印刷		

1.1.6 通信処理装置

通信処理装置（FNP）は前置プロセッサ（Front-end Processor）方式を採用し，中央処理装置とは独立に通信処理を一手に引受けて行い，中央処理装置の負荷を軽減し，システム全体の効率向上を図っている。FNPはNPS（Network Processing Supervisor）と称すプログラムによって動作する。表1.12にセンター設置のFNPとこの回線接続機構すなわちベーシック回線接続機構A（BLA-A：Basic Line Adaptor-A）の仕様・性能を示している。

表1.12 通信処理装置と回線接続機構の仕様・性能

装置	仕様・性能		
通信処理装置（N7294-01）	処理機能	制御方式	マイクロプログラム制御によるストアードプログラム方式
		命令数	105
		命令アドレス方式	1アドレス方式
		処理データの単位	6，9，18または36ビットいずれも可
		主記憶容量	基本 64Kバイト　最大 512Kバイト
		メモリアクセスタイム	0.5マイクロ秒／2バイト
		入出力チャネル数	最大 16
		入出力データ転送能力	最大 2Mバイト／秒
		割込みレベル	16 × 16 ＝ 256
	通信機能	通信速度	50 ～ 48,000 ビット／秒
		通信方式	全二重／半二重
		同期方式	調歩式／SYN同期／フレーム同期
		伝送符号単位	5～8単位
		通信回数	特定回線注)／公衆回線
		収容回線数	最大 512回線
		制御符号の検出	可
ベーシック回線接続機構A	通信機能	通信速度	50 ～ 9,600 ビット／秒
		通信方式	半二重／全二重
		収容回線数	4回線／台

注）特定回線の中には電々公社回線を通さない専用回線を含む。

1.1.7 データ通信機器

センターに設置されているデータ通信機器には次のものがある。

(1) キーボード・ディスプレイ　　DSP (Display unit)
(2) 高速シリアル・プリンタ　　HSP (High speed Serial Printer)
(3) ミニプリンタ　　MP (Miniprinter)
(4) グラフィック・ディスプレイ　　GDU (Graphic Display Unit)
(5) 漢字端末装置
(6) データ・ステーション
(7) 変復調装置，網制御装置　　M (Modem)，NCU (Network Control Unit)

1.1.7.1 文字ディスプレイとプリンタ

センター情報処理教室に配置してあるキーボード・ディスプレイ (通称N 6300／20 N) とそのハード・コピー装置として使用する高速シリアル・プリンタ (HSP) N 6323－10，およびミニプリンタ (MP) NM 3031－1 Jの仕様・性能を表 1.13，表 1.14 に示す。

表1.13　キーボード・ディスプレイの仕様・性能

項目		仕様・性能
通信機能	通信速度	1,200，2,400，4,800ビット／秒
	通信方式	半二重
	接続制御方式	ポーリング／セレクティング
	同期方式	調歩式 (1,200)　同期式 (2,400，4,800)
	使用符号	JIS 7単位符号
	適用回線	特定回線4線式　D－1，D－5 (1,200ビット／秒) 特定回線4線式　D1，D7，D9 (2,400，4,800ビット／秒)
表示部	画面の大きさ	13インチ
	有効表示面積	250(横) ×130(縦) mm
	表示文字数	80字×24行＝1,920字
	表示文字種	129種
	文字の大きさ	2.4(横) × 3.1(縦) mm
	文字間隔	0.7 mm
	行間隔	2.4 mm
	表示色	緑
	文字発生方式	7×9ドット・マトリクス
	表示方式	ラスター・スキャン
	画面の再生	62回／秒
	輝度	強／弱切換
キーボード部	キーの種類	データ・キー：128種 (英，数，カナ，記号)，テンキー：13種 (0～9,000，•，－)，ファンクション・キー：42種，シフト・キー：4種
	キー配列	JIS (C－6237) に準拠

表1.14　高速シリアル・プリンタ，ミニプリンタの仕様・性能

項目		仕様・性能	
		HSP(H6323-10)	MP(NM3031-1J)
通信機能	通信速度	1,200，2,400ビット／秒	300ビット／秒以下
	通信方式	半二重，全二重	半二重
	接続制御方式	無手順	無手順(注)
	同期方式	調歩式，同期式	調歩式
	使用符号	JIS単位符号	JIS7単位符号
	適用回線	特定回線(D−1, D−5, D−7)	公衆回線(直接接続可)
印字機能	印字速度	180字／秒	最高30字／秒
	文字種類	128種	128字
	最大印字数	132字／行	80字／行
	印字方式	7×9ドット・マトリックス方式	5×7ドット・マトリックス方式
	複写能力	5枚	3枚
	紙送り方式	トラクタ・フィード方式	
	騒音	印字時55ホーン以下	印字時60ホーン以下
キーボード	キー配列		JIS標準キー配列

注）無手順　○スタート，ストップビット 各1，データ 8ビット
○水平，垂直パリティ・チェックなし

1.1.7.2 グラフィック・ディスプレイ

センター情報処理教室に配置してあるグラフィック・ディスプレイ(GDU)は，ソニー・テクトロニクス社(Sony Tektronics)が開発した蓄積管方式の4006，4014−1型GDUで，これにハード・コピー(HC)装置4631型が接続されている。これらの仕様・性能を表1.15，表1.16に示す。

表1.15　グラフィック・ディスプレイの仕様・性能

項目		仕様・性能	
		4006　型	4014−1　型
表示部		11インチ，直視型，ストレージ管使用	19インチ，スクリーン直視型ストレージ管使用
表示面積		19.05(幅)×14.22(高) cm	38.1(幅)×27.94(高) cm
アルファニューメリック・モード	容量	74文字／行×35行＝2,590文字	74文字／行×35行から133文字／行×64行まで4段階
	文字種類	63種(ASCⅡコード)	フルASCⅡ文字セット
	文字形状	5×7ドット・マトリクス	7×9ドット・マトリクス
	カーソル	8×8ドット・カーソル	7×9ドット・カーソル
グラフィック・モード	ベクトル書込み速度	3.6±0.2ミリ秒	5,000インチ／秒
	座標点	X軸1,024×Y軸1,024	
	可視点	X軸1,024×Y軸780	
通信速度		75〜4,800ビット／秒	
接続制御装置		無手順	
適用回線		特定回線，公衆回線	

表1.16　ハードコピー（グラフィック用）の仕様・性能

項　　目	仕　　様　・　性　　能
ペーパ・サイズ	21.59 × 27.94 cm
コピー・サイズ	ポジションⅠ（管面を左から右に走査）：22.48 × 17.02 cm ポジションⅡ（管面を下から上に走査）：18.03 × 13.72 cm
コピー時間	最初のコピーに18秒，続いてとる場合8秒ごとに1枚
ウォーム・アップ時間	10分間
現象方式	熱現象乾式
コピー用紙	3M社タイプ777ドライ・シルバ用紙，$8\frac{1}{2}$インチ×500フィート

1.1.7.3　漢字端末装置

センターには，ACOS600計算機本体と9,600ビット／秒の専用回線で接続された漢字端末装置が設置されている。これは，漢字端末制御装置（KTC），漢字パターン発生装置（KPG），漢字ディスプレイ（KDP）N7872－11，漢字キーボード（KKB）N7876－21および漢字ハード・コピー装置（HC）N7852－11から構成されている。これらの仕様・性能を表1.17～表1.20にまとめている。

表1.17　漢字端末制御装置と漢字パターン発生装置の仕様・性能

装　置　名	型　番	項　　目	仕　様　・　性　能
漢字端末制御装置	N 7271-11 N 9271-11	CPU　部	16ビットマイクロ・コンピュータのマルチ・プロセッサ3台構成
		メモリ容量	プロセッサ当たり　64Kバイト 共通メモリ　64Kバイト
		バス方式	ユニバス方式
		通信速度	2,400～9,600ビット／秒
		通信方式	半二重
		誤まり検出方式	VRCチェック
		フロッピー・ディスク	2台，256KB／台，平均アクセス・タイム　260 ms
漢字パターン発生装置	N 7864-24	文字発生方式	ドット・マトリックス
		文字記憶容量	24×24　4.5K字，9K字までの増設可
		文字記憶素子	LSIメモリ

表1.18　漢字ディスプレイの仕様・性能

項　　目	仕　　様　・　性　　能
表示文字数	32字／行×15行＝480字
文字構成	(1)　漢字（全角）モード　24×24　ドット・マトリクス (2)　ANK（半角）モード　11×24　ドット・マトリクス
表示管	12インチ高解像度型CRT，　螢光体　P 39（緑）
表示方式	標準TVスキャンニング方式（1／2インタレース），像数30枚／秒
付加機能	プリンキング指定，　ライト・ペン入力
表示文字の大きさ	24×24　約5.3 m／m□

表1.19　漢字キーボードの仕様・性能

項　　目	仕　様　・　性　能
入　力　方　式	電磁結合方式による指示ペン入力
文　字　盤　面	64 × 56 ＝ 3584 字 1セグメントの大きさ 5.08*m*/*m*×5.08*m*/*m*
盤　面　文　字　種	JIS第1水準文字3306 字 その他記号　45 字 その他　79 字 ファンクション文字 ユーザ・エリア　) 154 字
盤　面　文　字　配　列	NEAC標準配列
入　力　確　認　音	入力時発生
ファンクション・キー	40 種

表1.20　漢字ハード・コピー装置の仕様・性能

項　　目	仕　様　・　性　能
記　録　方　式	WALK UP 電子写真方式
露　光　方　式	全面同時露光方式
現　像　方　式	液体現像，ローラ現像方式
定　着　方　式	熱風乾燥方式
用　　紙	B 6 版（125 *m*/*m*×182*m*/*m* ） ロール紙，内蔵カッター切断
コ　ピ　ー　速　度	1枚のみ　12 秒／枚 リピート2枚目　8 秒／枚 最大リピート　10 枚
記　録　形　態	陽／陰画両記録可能 中間調記録可能

1.1.7.4　データ・ステーション

堅粕地区（医歯薬学部）と六本松地区（教養部）には，リモート・バッチ処理に適したN 6240 データ・ステーションが設置されている。表1.21にその仕様・性能を示す。

表1.21　データ・ステーションの仕様・性能

項　　目		仕　様　・　性　能
通信機能	通　信　速　度	2,400 ～ 9,600 ビット／秒
	通　信　方　式	半二重，全二重
	同　期　方　式	同期式
	適　用　回　線	特定回線（D 1，D 7，D 9 ）
制御部	キーボード・ディスプレイ	40 字× 16 行　JIS標準配列
	フロッピー・ディスク装置	2 台，243 Kバイト／台
入出力装置	ラインプリンタ装置	
	印 字 速 度	200 行／分
	印　字　数	132 字／行
	印 字 文 字 種	111 種
	カ ー ド 読 取 装 置	マーク・カード読取り可
	読 取 速 度	300 枚／分
	使 用 コ ー ド	ホレリス／バイナリ
	シリアル・プリンタ	
	印 字 速 度	40 字／秒
	印　字　数	132 字／行
	印 字 文 字 種	127 種

1.1.7.5 変復調装置と網制御装置

端末と通信回線，通信回線と通信処理装置の間には，通常，変復調装置（M）が必要であり，電話を使用し公衆回線を経由して計算機を利用するためには，変復調装置の先に更に網制御装置（NCU）が必要である。センターで使用しているこれらの装置の規格・性能を表1.22，表1.23に示す。

表1.22　変復装置の規格・性能

項目＼型番	N 1200 A	N 1200 B	N 2400 B	S 9600
通信速度	1,200ビット／秒以下		1,200／2,400 ビット／秒	1,200／2,400／4,800 ／9,600 ビット／秒
適用回線	公衆回線（2線式／4線式）	特定回線（2線式／4線式）	特定回線（2線式／4線式）	特定回線（4線式）
通信方式	半2重（2線式） 全2重（4線式）		半2重（2線式） 全2重（4線式）	全2重
伝送方式	非同期		同期	同期

表1.23　MA-102型網制御装置の規格

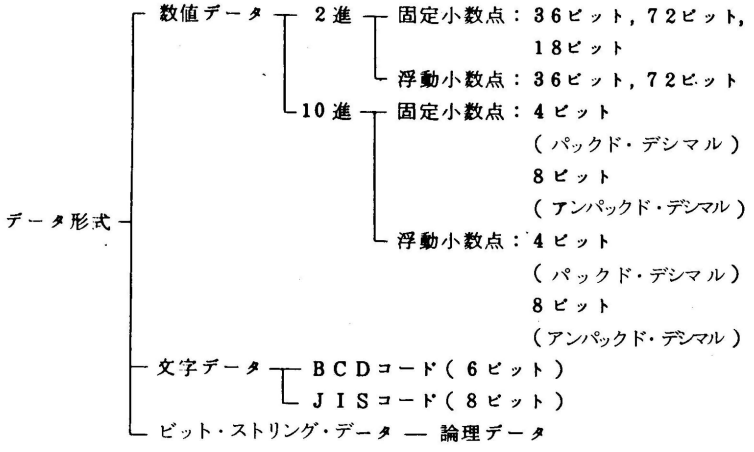
項目	規格
適用回線	公衆回線
形式	MA（Manual-Auto）形
適用電話	20 P.P.S.回転ダイヤル用

1.2 データ表現形式

1.2.1 データ形式

ACOS 600では，種々のデータ形式を扱うことができる。すなわち，数値データ，文字データ，およびビット・ストリング・データを取扱うことができる。これらを表に示すと次のようになる。

表1.24　データ形式



1.2.2 数値データ

(1) 2 進 数

(a) 固定小数点

2 進固定小数点データ形式には，①36 ビット，②72 ビット，および③18 ビットの 3 種類がある。これらの形式を図 1.1 に示す。

2 進固定小数点データの最初の 1 ビット（ビット 0 ）は，符号指定に用いられる。また，負数のデータは 2 の補数表示で表す。

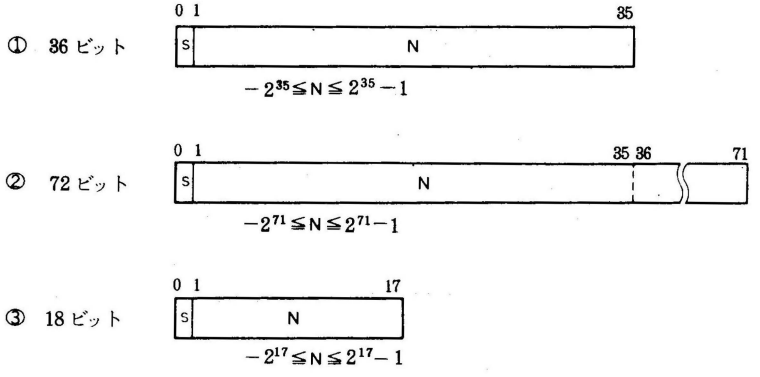


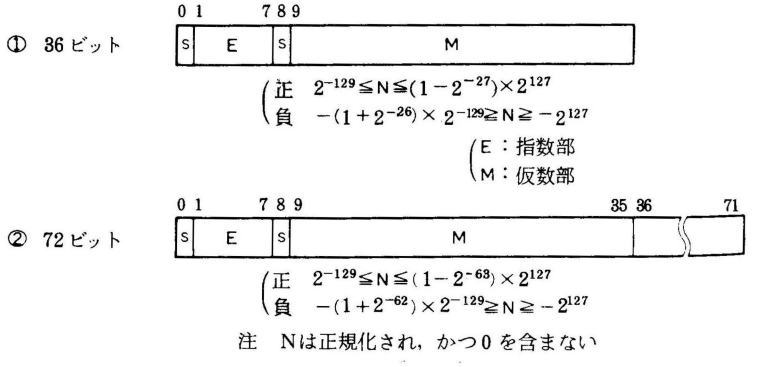
図 1.1　2 進固定小数点データ形式

(b) 浮動小数点

2 進浮動小数点データ形式には，①36 ビット，②72 ビットの 2 種類がある。これらの形式を図 1.2 に示す。

2 進浮動小数点データは，8 ビットの指数部と，①28 ビット，②64 ビットの仮数部とから成り，いずれも最初の 1 ビットは符号指定に用いられる。また，指数部・仮数部とも，負数のデータは 2 の補数で表す。

指数部は 2 のべき乗を表す。



注　N は正規化され，かつ 0 を含まない

図 1.2　2 進浮動小数点データ形式

(2) 10 進 数

(a) 固定小数点

10 進固定小数点データ形式には，

① 4 ビット（パックド・デシマル）

② 8 ビット（アンパックド・デシマル）

の2種類がある。

これらの形式を図1.3に示す。

10進固定小数点データの符号は，下記の3種類の形式を，ディスクリプタの指定により選択して，使用することが可能である。

(i) 数値データの後に置く。（後置符号）

(ii) 数値データの前に置く。（前置符号）

(iii) 使用しない。

上記(i)，(ii)の場合には，符号は数値データ1桁と同じビット数（パックド・デシマルで4ビット，アンパックド・デシマルでは8ビット）を占める。

数値データの桁数は，符号桁を含めて最大63桁まで指定することができる。さらに10進固定小数点データは，記述語の中の指定（位取り因子：スケール・ファクタ）により，小数点の位置を動かすことができる。

位取り因子は2進6ビットで表され，最初の1ビットは符号指定に用いられる。負数は2の補数で表す。

位取り因子は10のべき乗を表し，10^{-32}～10^{31}の範囲をとることができる。

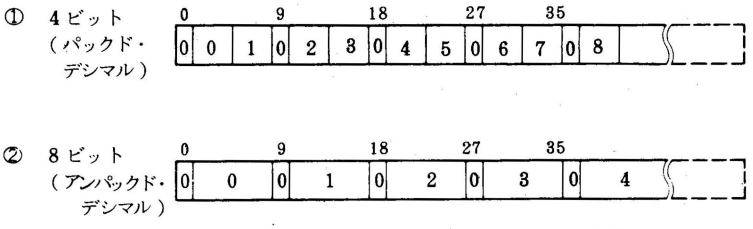


図1.3　10進固定小数点データ形式

(b) 浮動小数点

10進浮動小数点データ形式には，

① 4ビット（パックド・デシマル）

② 8ビット（アンパックド・デシマル）

の2種類がある。

これらの形式を図1.4に示す。

10進浮動小数点データは，符号桁を含めて最大61桁（4ビット）または62桁（8ビット）の仮数部と，8ビットの指数部から成る。

指数部は2進8ビットで10のべき乗を表し，10^{-128}から10^{127}の範囲をとることができる。

指数部の最初の1ビットは符号指定に用いられ，負数は2の補数で表す。指数部は図1.4に示したとおり，仮数部の後に続いて置かれる。

仮数部の符号は，常に，数値データの前に置かれ，数（データの1桁と同じビット数（①4ビット，②8ビット）を占める。

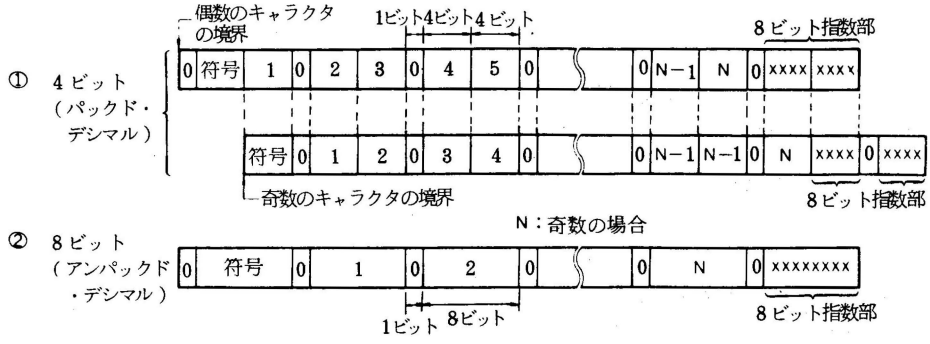


図1.4　10進浮動小数点データ形式

1.2.3 文字データ

(1) BCDコード（6ビット）注)

1字6ビットで表される文字データで，BCD（Binary Coded Decimal）コードと呼ばれる。この形式を図1.5に示す。

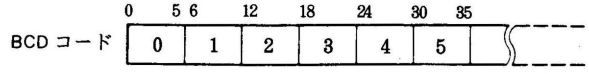


図1.5 BCDコード文字データ形式

(2) JISコード（8ビット）注)

1字8ビットで表される文字データでJIS（Japanese Industrial Standard）コードと呼ばれる。このデータ形式を図1.6に示す。

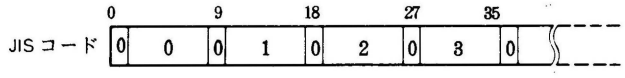


図1.6 JISコード文字データ形式

注) ACOS 600システムにおいては，BCDコードは主としてバッチ・システムにおけるシステム入力，システム出力およびファイルの標準コードとして用いられ，JISコードは主として言語処理プログラム，タイムシェアリング・システムにおけるファイルおよびデータ通信の標準コードとして用いられる。

従って，ソース形式のファイルについては，そのコード系が何であるかを注意しておく必要がある。

1.2.4 ビット・ストリング・データ―論理データ

ビット・ストリング・データは，独立したビットの集まりとして，各ビットごとに指定して処理することができる。

このデータ形式を図1.7に示す。

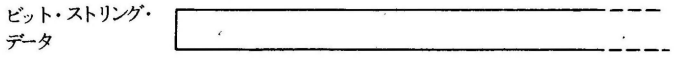


図1.7 ビット・ストリング・データ形式

2. ACOS 600 ソフトウェア・システム概説

2.1 ソフトウェア・システムの構成

オペレーティング・システムACOS-6は，基本ソフトウェアと応用ソフトウェアとに大きく区分される。基本ソフトウェアはすべての利用者によって使用されるもので，計算機システムの機能を発揮させるためのプログラム全体である。一方，応用ソフトウェアは，個々の具体的問題を計算機システムを利用して解決するために用意されたプログラムの集りである。

ACOS-6のソフトウェア体系を機能的に分類したものが図2.1である。

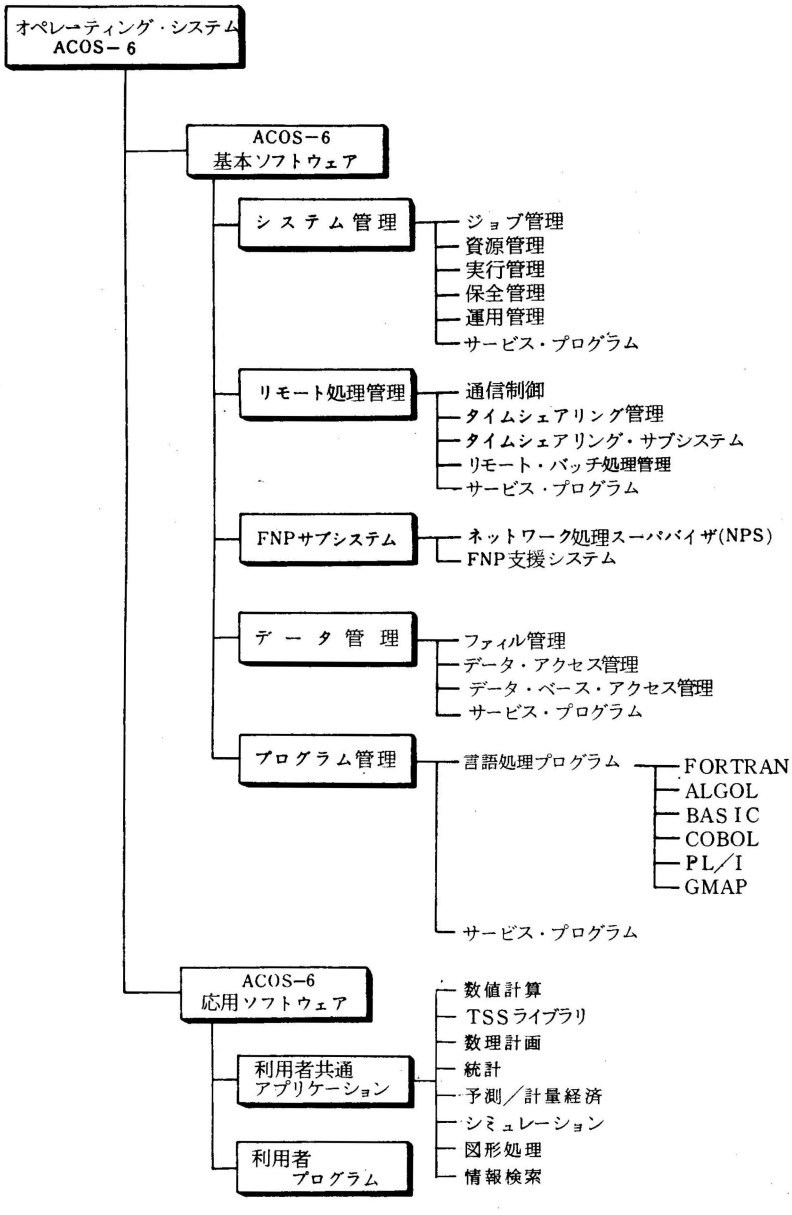


図2.1　ACOS-6のソフトウェア体系

2.2 基本ソフトウェアの概要（→マニ）

2.2.1 システム管理

システム管理は，多重プログラミングや多次元処理形態の下で，システム資源の有効利用を図り，システム全体の処理効率を高めるACOS-6の中枢機能である。更に，信頼性を高めるため各種障害に対処する機能，課金情報の処理，システム全体を円滑に動かすための機能などももっている。

システム管理の各部分と主な機能は次のとおりである。

(1) ジョブ管理

ジョブの受付，実行および結果の出力

(2) 資源管理

ジョブのアクティビティ（→2.3.2）に対する資源の割付け，解放，およびスワッピング処理

(3) 実行管理

実行優先度に基づくCPUの割当て，入出力の制御，割込み処理や利用者プログラム・モード（スレーブ・モード）では直接処理できない種々のシステム・サービス（マスタ・モードでの処理）

(4) 保全管理

ハードウェア・システムのエラー監視と回復

(5) 運用管理

システム運用開始時のスタート・アップ（立ち上げ），操作員との交信，課金情報の処理

システムに投入されるジョブは，ジョブ管理，資源管理および実行管理の連携動作によって処理される。

2.2.2 リモート処理管理

リモート処理管理は，通信処理装置を介して入出力するデータを取扱い，リモート処理形態を効果的に管理する機能である。

リモート処理管理の各部分と主な機能は次のとおりである。

(1) 通信制御

中央装置（CPU，メモリなど）上で働き，中央装置と通信処理装置とのインターフェース，各処理形態管理機能とNPS（Netwok Processing Supervisu）とのインターフェースを制御する。

(2) タイムシェアリング管理

タイムシェアリング・システムを管理する管理プログラムである。

(3) タイムシェアリング・サブシステム

タイムシェアリング・システムを構成する要素で，利用者が端末を通して使用する会話型言語または会話型処理プログラムである。

(4) リモート・バッチ処理管理

リモート・バッチ処理システムを機能させる管理プログラムである。

これらの機能の上に，会話型処理，リモート・バッチ処理，会話型リモート・バッチ処理が可能になっている。

2.2.3 FNP（Front-end Network Processor）サブシステム

FNPサブシステムは，オンライン・システムの中でデータ通信機能を受持ち，ネットワーク処理スーパバイザNPSとこれの作成，保守および運用のためのFNP支援システムから成っている。NPSは，種々の端末装置がFNPを通して中央装置にある情報処理システムと通信できるようにする。

2.2.4 データ管理

データ管理は，システムで入出力されるデータ，およびシステムに蓄積されるデータを制御し管理する機能をもち，多様なアプリケーションや処理形態を統合化する中心的な役割を果たす。

(1) ファイル管理

多種多様なアプリケーションによって使用されるシステム内に蓄積されたデータをファイル・システムとして階層化し，集中的かつ統一的に管理する。

(2) データ・アクセス管理

プログラムからファイル内の個々のデータをアクセスする際，その手続を容易にしその制御を利用者に代って行うサービス機能の集りである。

(3) データ・ベース・アクセス管理

プログラムからデータ・ベースをアクセスする際，その手続を容易にしその制御を利用者に代って行うサービス機能の集りである。

2.2.5 プログラム管理

プログラム管理は，利用者がプログラムを作成し，実行させ，意図とする結果を得るまでの過程において，プログラミング言語の翻訳処理，オブジェクト・プログラムのロードおよび実行，プログラムのデバックや保守などを容易にする機能を提供する。

プログラム管理の各部分とその機能は次のとおりである。

(1) 言語処理プログラム

FORTRAN，ALGOL，BASIC，COBOL，PL／I，GMAP などがプログラミング言語として利用できる。

(2) サービス・プログラム

ローダは，各種言語処理プログラムなどによって作られたオブジェクト・プログラム群を結合し実行させる機能を提供する。またプログラムのデバックやオーバーレイを行う機能ももっている。

その他，プログラムを，ソース・プログラム，オブジェクト・プログラムあるいは実行形式プログラムの各形式でファイルに登録したり，更新したりするサービス・プログラムも提供されている。（→バ利）

2.3 処理形態と処理システム

2.3.1 処理形態

計算機システム内で実行される利用者の計算そのものについては，その形式はシステムから見た場合には本質的な差違はないが，計算のシステムへの受付けと結果の呈示方法には種々の形態が考えられる。この形態を，システムから見た場合に処理形態，利用者から見た場合には利用形態と呼ぶことにする。ACOS－6 には以下に示す 4 つの利用形態があり，利用者は必要に応じてどの処理形態でも選択することができ，しかも複数利用者が選択したすべての処理形態が同時に，多様な組合せで利用できる。

(1) ローカル・バッチ（センター・バッチ）処理

ローカル・バッチ処理は，中央装置が設置されている施設内で，通常，中央装置に直接接続されているカード読取り装置から計算を入力させ，計算結果とか計算過程を示す情報を，中央装置に直接接続されているラインプリンタに出力する形態である。この処理形態は，利用者が大量のプログラムやデータを初めてシステムに投入する場合などに便利である。

(2) リモート・バッチ処理

ローカル・バッチ処理との相違は，計算の入力と結果の出力を中央装置と通信回線で結ばれた遠隔地にある入出力装置で行うことのみである。

(3) タイムシェアリング処理

タイムシェアリング処理は，多数の利用者が端末装置を用いてセンターにある中央装置を，別々の場所から，同時にかつ独立にあたかも自分が占有しているかのように利用できるようにするものである。その結果，端末利用者と計算機システムとの相互の応答を連続して行うことができるため，計算機と人間の共同作業による問題の解決を可能にする。

これらを具体化したものが会話型処理で，言語処理やファイル操作などを行うための種々のサブシステムが用意されており，これらはほかの処理形態の場合と同様に共通な方法で利用することができる。

(4) 会話型リモート・バッチ処理

タイムシェアリング用端末装置（TSS端末と略）からバッチ処理を行うための形態が会話型リモートバッチ処理である。TSS端末は中央装置を会話形式で使用できる利点はあるが，通信回線や端末装置などの

制約から，通常，多量のデータの入出力には不向きである。そこでTSS端末からファイルに直接ジョブ・イメージ（一連のジョブ制御文）をタイプインし，そのジョブのバッチ処理を依頼する方法をとる。計算結果については，すべての出力の中から必要な部分，例えばコンパイル時のエラー・メッセージ部分を選択出力させる機能をもっている。

2.3.2 バッチ処理システム

ジョブとは，利用者がシステムに仕事を依頼する単位であり，これは更にシステムの中では，コンパイル，実行などのアクティビティと呼ばれる単位に分けられて処理される。利用者がシステムに仕事を与えるときにはプログラムやデータと共にジョブ制御言語で書いたジョブ記述をシステムに入力する。ジョブ記述は，利用者がどのようなサービスや環境を要求しているかを細く規定するもので，いわば利用者とシステムとのソンターフェースであり，システムはこの記述に基づいてジョブの処理を行うことになる。

ジョブ内のアクティビティは原則としてジョブ記述中に書かれた順に逐次実行される。システム資源の割当てはアクティビティ単位に行われ，アクティビティに割当てられた周辺装置やファイル領域は通常それが終了すれば解放されるが，指定により後続のアクティビティに渡すこともできる。

代表的なジョブ記述の例を図2.2に示す。（→バ利）

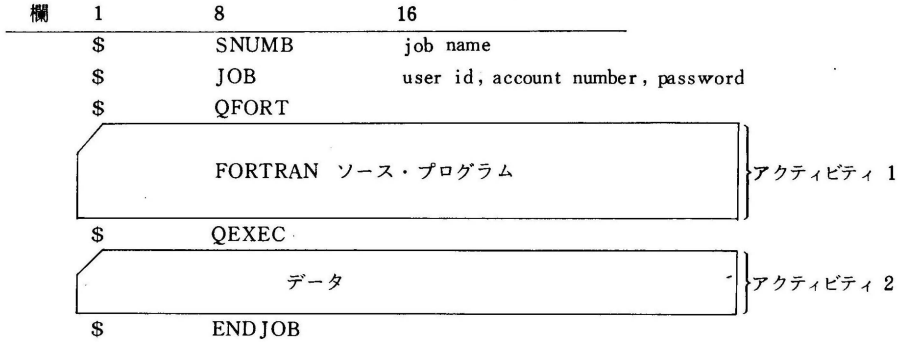


図2.2 ジョブ記述の例

ローカル・バッチ処理，リモート・バッチ処理および会話型リモート・バッチ処理において，システムに投入されたジョブは図2.3に示す過程を経て処理される。（→マニ2）

(1) ジョブの入力

ジョブ入力はジョブ管理のモジュールであるシステム・インプット（SYSIN）によって制御される。

利用者はカード・デック，磁気テープ，または磁気ディスク上のファイル中に記述されているジョブを，システム・インプットを通してシステムに投入する。ジョブ投入は，ローカル・バッチ処理だけでなくリモート・バッチ処理や会話型リモード・バッチ処理で行うことができる。

(2) ジョブのスケジューリング

ジョブのスケジューリングは，ジョブ管理のモジュールであるシステム・スケジューラにより行われる。

ジョブをスケジュールするというのは，ジョブの実行順序を計画することである。ACOS-6システムでは，システム全体が十分にその役割を果たすことを目指して，次の観点からジョブ・スケジューリングを行っている。

① ジョブの固有な要求に応じた処理応答時間を相対的に保証すること

② システム資源を効果的に使用して処理量を最大にすること

(3) アクティビティに対する資源の割当て

資源の割当ては，資源管理のモジュールすなわちペリフェラル・アロケータおよびメモリ・アロケータによって行れる。

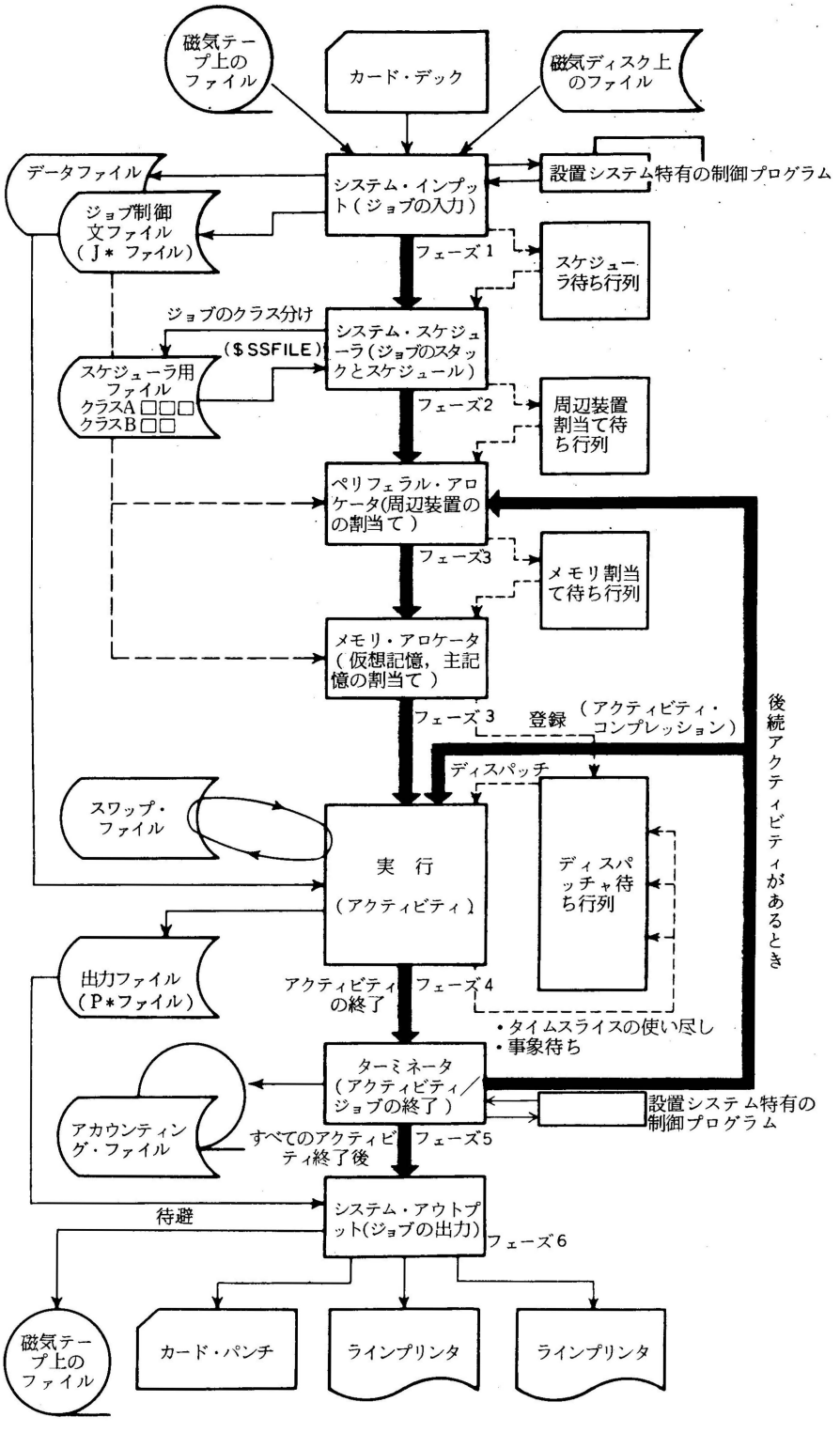


図2.3　ジョブ処理の流れ図（概念図）

アクティビティがシステム資源割当てのフェーズを通るのは，次の二つの場合である。

- 新しいジョブが割当て待ち行列に登録され，割当ての対象になったとき
- 一つのアクティビティが終了し，次のアクティビティが資源の割当ての対象になったとき

システム資源の割当ては，次の二つのサブフェーズにより行われる。

① 周辺装置の割当て（ペリフェラル・アロケータ）

ペリフェラル・アロケータは，待ち行列に並んだ順番に各ジョブを調べて，そのジョブのアクティビティに対して周辺装置の割当てを行う。

② 主記憶の割当て（メモリ・アロケータ）

アクティビティに割当てられる主記憶は連続している。アクティビティが終了すれば，その領域は他のアクティビティに割当てるためにシステムに返却される。また，必要に応じて，メモリ・コンパクション（アクティビティの占める領域の移動）が行われ，非連続領域を連続にして，他のアクティビティに割当てるべき領域を確保する。（→図2.4）

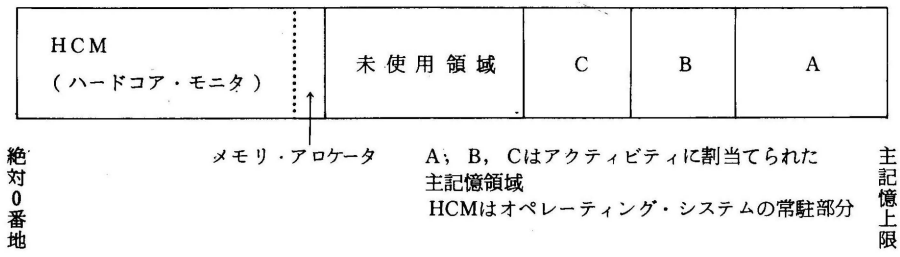


図2.4　主記憶レイアウト

(4) アクティビティの実行

アクティビティが実行状態になるのは，必要な周辺装置および主記憶装置が割当てられた後，中央処理装置（CPU）を割当てられることによってである。

CPUの割当ては実行管理のディスパッチャが行う。ディスパッチャはディスパッチャ待ち行列の中から，要求度の高い順にCPUを一定時間づつ割当ててゆく。

(5) アクティビティ／ジョブの終了

アクティビティやジョブの終了処理は，ジョブ管理のモジュールであるターミネータによって行われる。

実行されているアクティビティは，正常な動作終了後，プログラムなどの誤り発生時，あるいは一定時間経過後に終了する。終了時の処理は，それぞれのタイプにより異なる。

① 正常な終了

ジョブが正常な状態で終了すると次のアクティビティが調べられる。以降のアクティビティに不必要な資源はシステムに戻し，必要な資源はそのまま保存される。そのアクティビティで使用した資源は，アカウンティング・ファイルに記録が収集され，使用した資源の情報は，項目別にまとめられてシステム出力ファイルに記録される。

ジョブの最後のアクティビティであれば，そのジョブはジョブ・スタックから外されて，使用したシステム資源はすべてシステムに返却される。

② 異常な終了

プログラムが誤った処理を行ったときには，異常終了として打切られる。以降にあるアクティビティの実行は通常行われない。しかしその場合にも，ジョブ制御言語で指定することにより以降のアクティビティを実行することも可能である。

(6) ジョブの出力

実行中のプログラムからの出力（印字やせん孔など）は，システム出力ファイルに集められる。システム出

力ファイルの内容は，システム・アウトプット（ SYSOUT）によって，ラインプリンタやカードせん孔装置あるいは端末装置に出力される（→図2.5）。

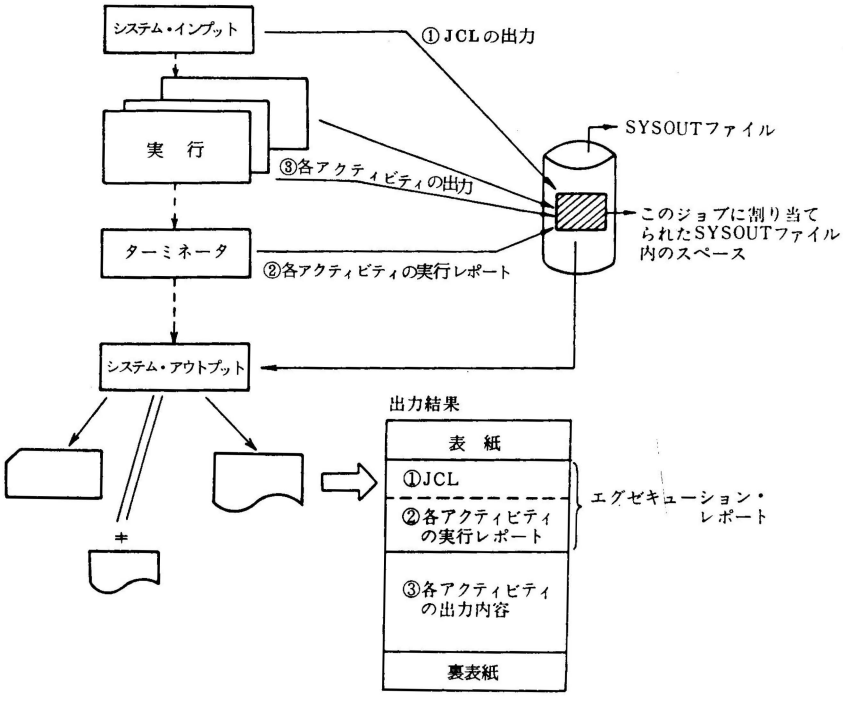


図2.5　ジ　ョ　ブ　出　力

2.3.3. リモート・バッチ処理システム

リモート・バッチ・ジョブは，通信処理装置との仲介をとるためのリモート用入力／出力ルーチンを通して，システム入力／出力が行われる（→図2.6）。その他の点は，システム内ではローカル・バッチ・ジョブと同じように処理される。リモート・バッチとしてシステム入力されたジョブの出力情報は，次の処理法の中から選択することができる（→マニ24，リ利）。

- 入力した端末へ返す。
- センターに出力する。
- ファイル・システムに保存する。

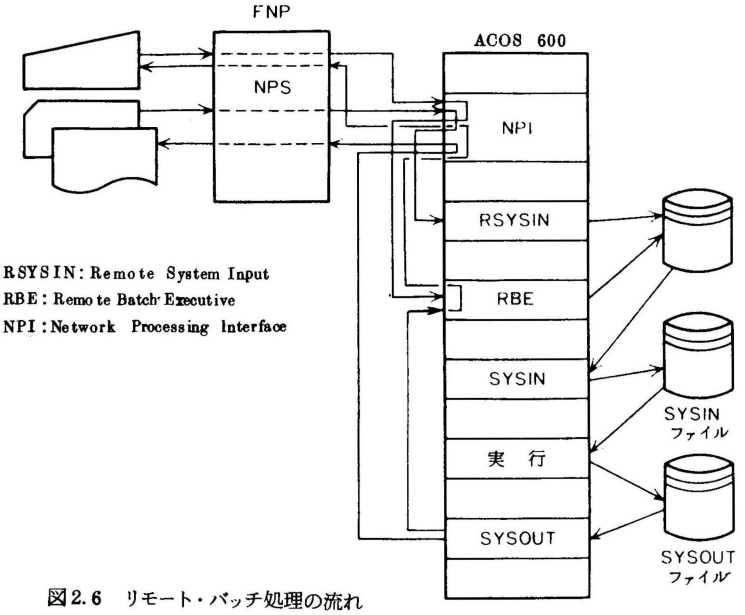


図2.6 リモート・バッチ処理の流れ

2.3.4 タイムシェアリング・システム

タイムシェアリング・システムは，タイムシェアリング・エグゼクティブ（TSE）とタイムシェアリング（TS）サブシステムから成る（→マニ18）。

(1) タイムシェアリング・エグゼクティブ（TSE）

タイムシェアリング・エグゼクティブは，タイムシェアリング・サブシステムに対して必要とする主記憶装置やファイルなどシステム資源の割当てを行い，また，端末装置との入出力操作なども行う。タイムシェアリング・エグゼクティブは，利用者が存在しない場合には，主記憶装置の外部に存在しており，利用者が端末装置から交信を要求すると，主記憶装置内にロードされる。

(2) タイムシェアリング（TS）サブシステム

タイムシェアリング・システム利用者に対しての実際のサービスは，タイムシェアリング・エグゼクティブ（TSE）の制御の下に，次のサブシステムによって行われる。利用者はこれらのサブシステムをコマンドによって使用し，会話型処理あるいは会話型リモートバッチ処理のサービスを受けることができる。（→タ利）

(a) 言語処理サブシステム

① BASIC …… 比較的簡単な数値処理を行う利用者のための言語で，BASICのプログラミングから実行までを行うサブシステムである（→マニ17～18）。

② FORTRAN…… バッチ処理システムのFORTRANと共通なFORTRAN言語によるプログラミングから実行までを行うサブシステムである（→マニ18，マニ7～9）。

③ ALGOL …… ALGOL言語によるプログラミングから実行までを行うサブシステムである（→マニ18，マニ10～12）。

④ ABACUS …… 卓上計算機的機能をもつサブシステムである。関数計算，変数の記憶機能をもち，加算演算に特徴をもつ。（→マニ18）

(b) ファイル処理サブシステム

① ACCESS…… ファイル・システムのカタログやファイルの作成，削除，変更を行うサブシステムである。バッチ処理システムにおけるFILSYSプログラムに対応するもので，ファイルの領域を取扱うサブシステムである。（→マニ18）

② EDITOR…… テキスト・ファイルの作成，保守，参照をするためのサブシステムである（→マニ20）。

③ RUNOFF…… テキスト・ファイルを編集して印字するサブシステムである。1ページ当たりの行数，各行の長さ，ページの付け方，その他編集のための位置づけなどを指示することができる。（→マニ20）

(c) 会話型リモート・バッチ処理サブシステム

会話型リモート・バッチ処理をサービスする機能として次のサブシステムがある（→マニ19）。

① CARDIN …… タイムシェアリング・システムを利用している端末からバッチ・ジョブを投入するためのサブシステムである。

② SCAN …… タイムシェアリング・システムを利用している端末から，バッチ・ジョブの出力を調べるサブシステムである。ただし，ジョブの結果はパーマネント・ファイル上になければならない。

③ JOUT …… タイムシェアリング・システムを利用している端末から，バッチ・ジョブの出力を操作するサブシステムである。

2.4 プログラミング言語の概要

ACOS-6オペレーティング・システムにおいて，各処理形態で使用可能なプログラミング言語を表2.1に示す（→マニ8～16）。

表2.1 処理形態とプログラミング言語

処理形態 / 言語処理プログラム	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	会話型処理	コード
COBOL	○	×	JIS, BCD
FORTRAN	○	○	JIS
PL/I	○	×	JIS
BASIC	×	○	JIS
ALGOL	○	○	JIS
GMAP	○	×	JIS, BCD

2.4.1 FORTRAN

FORTRAN（FORmula TRANslator）は，主として数値計算に適したプログラミング言語である。ACOS-6 FORTRANは，JIS FORTRAN水準7000を包含し，更に幾つかの機能が付加されている。主な特徴は次のとおりである。

(1) 1記憶単位は36ビットで，データの精度は，実数型で10進8桁，倍精度実数型で10進18桁である。指数は，−38～38の範囲である。また，整数型データは35ビットで表現される。

(2) ソース・プログラムはもちろん，オブジェクト・プログラムのレベルで，バッチ，タイムシェアリング両処

理形態での互換性がある。（→マニ7～9，F文）

2.4.2 ALGOL

ALGOL（ALGOrithmic Language）は，算法記述を目的として作られた言語で，数値計算に適したプログラミング言語である。ACOS-6 ALGOLはJIS ALGOL水準5050に準拠し，更に，それを拡張したものである（→マニ10～12）。主な特徴は次のとおりである。

(1) データの精度は，実数型で10進8桁，倍精度実数型で10進18桁である。指数は－38～38の範囲である。また，整数型は35ビット，倍精度整数型は71ビットで表現される。

(2) 手続きの回帰的呼び出し，占有型（own）宣言，入出力に関する設定手続きは使用できない。これらは，JISのより上の水準の機能である。

2.4.3 BASIC

BASIC（Beginner's All purpose Symbolic Instruction Code）は，数値計算を目的とした会話型の言語である。ACOS-6 BASICには，次のような特徴がある。（→マニ17）

(1) 数の精度は10進8桁，指数は－38～38の範囲である。

(2) 配列は2次元までである。また，配列の計算，代入，入出力を行なう機能をもっている。

2.4.4 COBOL

COBOL（COmmon Business Oriented Language）は，事務用データ処理を目的とした言語である。ACOS-6 COBOLには，COBOLとCOBOL(B)とがある。

COBOL(B)は，JIS COBOL 72の最高水準を満している。COBOLは，JIS COBOL 72から報告書機能を除いたものの最高水準を満している。（→マニ3～6）

2.4.5 PL/I

PL/I（Programming Language/I）は，事務用データ処理，数値計算およびシステム・プログラム開発用に適用することを目的とした多目的言語である。ACOS-6 PL/Iは国際標準化が進められているPL/Iの言語仕様BASIS/Iに準拠している。（→マニ13～15）

2.4.6 GMAP

GMAPはACOS-6のアセンブラであり，全機械命令のほか多くの疑似命令やマクロ命令が用意されている（→マニ16）。

2.5 ファイル・システムの概要

ファイル・システムの概要を以下に説明する。ファイルとその利用法の詳しい説明は本号「ファイル・システムの利用法」を参照して頂きたい。（→マニ18，マニ32）

2.5.1 ファイルの種類

ACOS-6が取扱うファイルには，大きく分けて次の2種類のファイルがある。

(1) パーマネント・ファイル

ジョブやセッション（session：タイムシェアリング処理において端末とシステムが論理的に結合関係にある期間）終了後も必要とする情報（プログラムやデータ）を格納する保存用ファイル

(2) テンポラリ・ファイル

ジョブやセッション内でのみ一時的に発生する情報を格納する一時ファイル

この中でパーマネント・ファイルのみFMS（File Management Supervisor）が管理する。

2.5.2 ファイル・システムの特徴

ACOS-6ファイル・システムには次のような特徴がある。

(1) ファイルの共用

バッチ処理，タイムシェアリング処理のどの処理形態からも共通に使用できる。

(2) カタログ／ファイルの階層構造

図2.7に示されるように，ファイルはシステム・マスタ・カタログ（SMC）を頂点とする階層構造を成しており，これにより統一的に管理される。

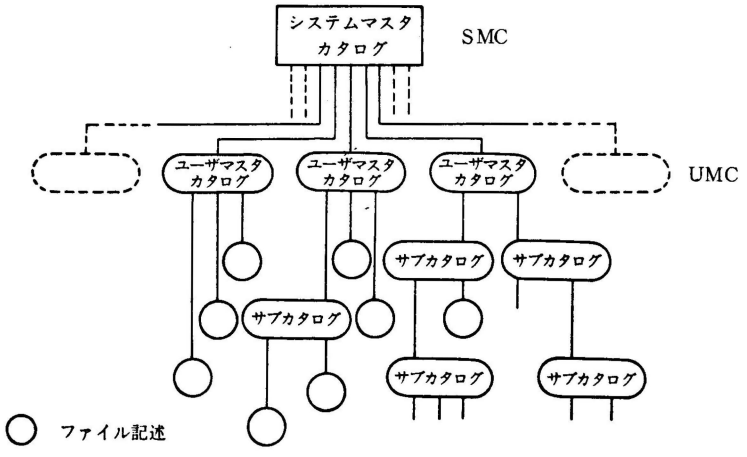


図2.7　ファイルの階層構造

(3) 機密の保護

ファイルを共同で使用する場合に機密の保護ができるように，利用者識別名，パスワード（合い言葉），パーミッション（許可操作）が用意されている。

(4) 多数ジョブの同一ファイル同時利用機能

複数ジョブが同一ファイルに対して同時にアクセスすることができる

(5) ファイルの取扱いの簡単さ

カタログやファイルは，その名前を指定するだけでハードウェアを意識することなく簡単にアクセスすることができる。

2.5.3 カタログ／ファイルの操作

カタログやファイルの作成・更新・削除を行うために，タイムシェアリング・システムではACCESS サブシステム，バッチ・システムではFILSYSプログラムが用意される。

2.5.4. ファイルの編成

ACOS-6 で扱えるファイル編成には次の 3 種類がある（→マニ 32 ）。

(1) 順編成ファイル

データが処理される順序と各レコードの物理的順序が一致するファイル編成である。種々の媒体が利用できる。

(2) 直接編成ファイル

ファイルを 64 語単位のブロックに区切り，ファイルの始点を 0 とした相対アドレスで当該ブロックにアクセスする。媒体はディスクに限る。

(3) 索引順編成ファイル

索引順編成ファイルはインデックス（索引）・ファイルとデータ・ファイルにより構成される。データ・ファイル内のレコードはキーの昇順に並んでおり，インデックス・ファイル内の索引はキー値とデータ・アドレスから成る。

2.5.5 ファイル領域の確保と解放

ファイルに必要な領域はLリンク（ 320 語）という単位で割当てられる。ファイル作成時に，ファイル領域の初期値と最大値を指定できる。ファイルを削除すればそのファイルの領域が，カタログを削除すればそのカタログに属するすべてのファイルの領域が解放される。

2.6 プログラムの形式とファイル

ACOS-6 プログラム管理で取扱われるプログラムの形式とファイルについて述べる（→マニ 30 ）。

(1) ソース・プログラム／データはサービス・プログラム＊Cエデイタやシステム・インプットにより，この形式

の順編成ファイルの作成（更新）をすることができる。コンパイラの入力となるソース・プログラム・ファイルをＳ＊ファイルと呼ぶ。

(2) オブジェクト・プログラム

コンパイラやアセンブラが出力するオブジェクト・プログラムをバイナリ・デック形式の順編成ファイル（Ｃ＊ファイルと呼ぶ）として登録できる。このファイルを利用者個人のシーケンシャル・ライブラリとして利用できる。

(3) サブルーチン・ライブラリ

(2)で述べたシーケンシャル・ライブラリを，サービス・プログラムのランダム・ライブラリ・ゼネレータ（RANLIB）を用いて，直接編成ファイル形式のランダム・サブルーチン・ライブラリに変換し，利用者ライブラリとして使用できる（→バ利 1.2.5 ）。

(4) 実行形式プログラム

ローダの出力する直接編成ファイル（Ｈ＊ファイルと呼ぶ）である。ファイルに登録された実行形式プログラムはタイムシェアリング・システムではRUNコマンドにより，バッチ・システムではジョブ制御文PROGRAM文により呼び出して実行することができる（→タ利3.3.3，バ利 1.2.3 ）。

(5) その他　　プログラムが出力するシステム出力用の順編成ファイルをＰ＊ファイルと呼ぶ。

2.7 データ・ベース・システムの概要

データ・ベースは，利用者プログラムから独立して蓄積，管理されているデータの集まりで，異なる利用者の異なるプログラムで使用することを可能にしているものである。データ・ベース・システムは，データ・ベースを統一的に管理するためのもので，ACOS－6には次のようなデータ・ベース・システムが用意されている。

(1) IDS（ Integrated Data Store ）

(2) IDS（ IDS Data Query ）

(3) INQ（ Information Query ）

(4) ADBS（ Advanced Data Base System ）

2.7.1 IDS

IDSは種々の利用で発生する情報を統括し，ディスク上でデータの蓄積・検索・変更を行うことにより汎用ファイルを編成するシステムである。IDSの親言語はCOBOLである。（→マニ 36～38 ）

2.7.2 IDSデータ・キュアリ

IDSデータ・キュアリはIDSファイルをデータ・ベースとして端末からデータ検索を行うことを容易にするシステムである（→マニ 22～23 ）。

2.7.3 INQ

INQは，インバーテッド・ファイル型を主体としたデータ・ベース・マネジメント・システム（DBMS）でデータ・ベースにおける検索の即応性，変化に対する柔軟性及び適応性に富む汎用性の高いシステムである（→マニ 46～49 ）。

2.7.4 ADBS

CODASYL（The Conference on Data System Language ）仕様に基づく総合データ・ベース・システムであり，COBOLを親言語とする（→マニ 41～44 ）。

2.8 応用ソフトウェアの概要

応用ソフトウェアは，利用者それぞれがもつ種々の問題を解決させるためのプログラムの集まりであり，ACOS－6には次のものが用意されている（→マニ 50 ）。

(1) 数値計算

数値計算ライブラリとして，数値計算，統計計算のサブルーチン群からなるMATHLIB－6（ Mathematical Library ）が準備されている（→マニ 51～53 ）。

(2) TSS

タイムシェアリング処理のために準備されたアプリケーション・パッケージ群TSS／LIB－6（Time Sharing

System Application Library ）がある。その主な内容は，数値計算，統計計算，マネージメント・サイエンスなどである。（→マニ54～55 ）

(3) 数理計画

線形計画法を中心とした各種計画問題の最適化を図るための数理計画システムMPS-6（ Mathematical Programming System ）が準備されている（→マニ56～60 ）。

(4) 統　計

統計処理システムSTATPAC-6（ Statistical Package ）が準備されている（→マニ61～62 ）。

(5) 予測／計量経済

予測／計量経済システムFORES-6（ Forecasting System ）と会話型予測／計量経済システムFORES／TSS-6（ Forecasting System on Time Sharing System ）が準備されている（→マニ64～67 ）。

(6) シミュレーション

シュミレーションシステムとして，つぎの三つが準備されている（→マニ68～70 ）。

① 離散型シミュレーション言語GPSS／V-6（ General Purpose Simulation System／v ）

② ダイナミックモデルのシミュレーション・システムDYNAMO／F-6（ Dynamic Model／F ）

③ 連続型シミュレーション言語CSPL-6（ Continuous System Programming Language ）

(7) 図形処理

図形処理システムGDSP-6（ Graphic Display Subroutine Package-6 ）として，基本システム（GDCP-6／PLOT ），三次元処理サブシステム（GDSP-6／3 D ）およびデータ管理サブシステム（CDS-6／DMG）の三つが用意されている（→マニ71～73 ）。

(8) 情報検索

情報検索システムIRS-6（ Information Retrieval System ）と会話型情報検索システムIRS／TSS-6（ Information Retrieval System on Time Sharing System ）が準備されている（→マニ74～76 ）。

タイムシェアリング・システム利用法

1. タイムシェアリング・システム利用法 …… 41
1.1 手 順 例 …… 41
1.2 手順の説明 …… 42
2. タイムシェアリング・システムにおけるファイルとコマンド …… 44
2.1 テンポラリ・ファイル …… 44
2.2 パーマネント・ファイル …… 44
2.3 ファイルとコマンドの関係 …… 46
3. タイムシェアリング・システム用コマンド …… 48
3.1 コマンド一覧 …… 48
3.2 会話の開始・終了コマンド …… 49
3.3 サブシステム選択・実行コマンド …… 49
3.4 アベイラブル・ファイル・テーブルに対する登録・削除コマンド …… 52
3.5 カレント・ファイル指示コマンド …… 52
3.6 ファイル内容の操作・表示コマンド …… 55
3.7 ファイルの登録状況表示・複写・消去コマンド …… 58
3.8 エラー・メッセージ説明コマンド …… 60
3.9 状態表示コマンド …… 60
4. タイムシェアリング・システム使用中のエラー・メッセージ …… 61
4.1 TSSからのエラー・メッセージ …… 61
4.2 NPSからのメッセージ …… 62
5. タイムシェアリング・システム用端末機器の操作法 …… 62
5.1 一般的操作法 …… 62
5.2 N6300／20Nキーボード・ディスプレイ操作法 …… 63
5.3 グラフィック・ディスプレイ（4006）操作法 …… 68

端末からのセンターの計算機を会話的に利用する形態であるタイムシェアリング・システム（TSS）の利用法について説明する。ここではタイムシェアリング・システムで使用するコマンドやファイルを中心に述べる。（マニ18）

1. タイムシェアリング・システム利用例

タイムシェアリング・システムにおけるFORTRANサブシステムを例にとって，システムとの接続，会話の開始，テキストの入出力，会話の終了などの諸手順について説明してゆく。

1.1 手順例

① 端末の電源投入 …… 電源投入

② [$$$ CON, TSS] キー押下または $$$ CON, TSS (SD) 注1) …… TSSとの接続

③ A COS－b TSS(R3.3) ON 03/10/78 AT 10.750 CHANNEL 端末識別名 …… 接続メッセージ

④ USER ID －△QZUTIMURA999；123456(SD) …… 登録番号（利用者識別名），支払責任者登録番号入力
（登録番号（user id）　支払責任者登録番号（account number））

⑤ PASSWORD－－△
XXXXXXXXXXXX(SD) …… パスワード入力

⑥ SYSTEM ？△FORTRAN(SD) …… サブシステム名入力

⑦ OLD OR NEW－△NEW(SD) …… カレント・ファイルの作成

⑧ *△AUTO(SD) …… オートマティック・モード

⑨ *010 ␣△READ, M, N(SD)

⑩ *020 ␣△PRINT ␣10, M, N(SD)

⑪ *030 ␣△WA＝M＋N(SD)

⑫ *040 ␣△SA＝M－N(SD)

⑬ *050 ␣△SEKI＝M*N(SD)

⑭ *060 ␣△SYO＝M／N(SD)

⑮ *070 ␣△AMARI＝M+N*SYO(SD) …… テキストの入力

⑯ *080 ␣△PRINT ␣10, M, N, WA, SA, SEKI, SYO, AMARI(SD)

⑰ *090 ␣△10 ␣FORMAT(1×,"M=", I5,"␣N=", I5,"␣SA=",(SD)

⑱ *100 ␣△&␣I5,"␣SEKI=", I5,"␣SYO=", I5,"␣AMARI=", I5)(SD)

⑲ *110 ␣△STOP(SD)

⑳ *120 ␣△END(SD)

㉑ *130 ␣△(SD) …… オートマティック・モードの終了

㉒ *△002 C␣EXAMPLE OF ARITHMETIC EXPRESSIONS(SD) …… テキストの追加

㉓ *△004 ␣INTEGER␣WA, SA, SEKI, SYO, AMARI(SD)

㉔ *△SAVE ␣QDAFL01(SD) …… ファイル名QDAFL01のファイルを作成してこのテキストを保存する。

㉕ DATA SAVED－－QDAFL01 …… 正しく保存されたというメッセージ

㉖ *△LIST␣2－10(SD) …… ライン番号2から10の間の全行リスト

㉗ 002 C␣EXAMPLE OF ARITHMETIC EXPRESSIONS

㉘ 004 ␣INTEGER␣WA, SA, SEKI, SYO, AMARI

注1) (SD)は1テキスト入力終了信号（ETX）に対応するもので機器によって異なるが，キーボード・ディスプレイ（N6300／20N）では [送信] キーである。

注2) ＿部分はシステムから出力される部分を区別するときに用いる。ただしこれが明らかな場合や下線を引くことによって見ずらくなる場合には省略することがある。

注3) △，␣　␣は空白文字を表す，△はディスプレイ上のアトリビュート文字で␣とは異なるが，表示は空白と同じに見える，しかしテキスト上ではこの位置に文字はない。△が出力されるディスプレイとされないディスプレイがある。

㉙ 010 ␣READ, M, N

㉚ *△070 ␣AMARI＝M－N*SYO(SD) ……ライン番号70の行訂正

㉛ *△030 (SD) ……ライン番号30の行削除

㉜ *△RUN(SD) ……FORTRANサブシステムの実行

㉝ ＝△100, 15, ……M, Nの入力要求

㉞ M＝␣␣100␣N＝␣␣␣15␣SA＝␣␣␣85␣SEKI＝␣1500␣SYO＝␣␣␣␣6␣AMARI＝␣␣␣10
……結果の出力

㉟ *△RESAVE␣QDAFL01(SD) ……ファイルQDAFL01への再保存

㊱ DATA SAVED －－ QDAFL01

㊲ *△PRINT(SD) ……プリンタへの出力

㊳ ACTION?△100(SD) ……行数の最大値100行

㊴ USERID?△QZUTIMURA999;123456(SD) ……登録番号，支払責任者番号（学生はクラス名）

㊵ RASSWORD?△XXXXXXXXXX(SD) ……パスワード入力

㊶ FILE STRING?△QZUTIMURA999/QDAFL01(SD) ……ファイル名の全名前入力

㊷ YOUR LIST WILL BE PRINTED OUT TO Pi ……プリンタPiへ出力するというメッセージ

㊸ *△BYE ……TSSとの会話終了

㊹ ** COST： ¥100 TO DATE： ¥9000＝10％ ……課金情報出力

㊺ ** ON AT 10.750－OFF AT 11：250 ON 12/10/77

㊻ 端末の電源切断

1.2 手順の説明

①，② 電源投入後，端末の「通信可」および「けん盤入力可」のランプが点灯していることを確認の上，[$$$ CON, TSS] キー（もしあれば）の押下または$$$ CON, TSS(SD)を入力する。

③ 正しく接続されると，このメッセージが出力される。

R3.3 ……システムのバージョン

ON mm/dd/yy …… 19yy年mm月dd日

AT hh.hhh ……時刻は時間単位に小数第3位まで表される。
10.750は10時45分を意味する。

④～⑦ ④～⑦が会話開始の手順（ログオン・シーケンス）を示す。これによりセッションが開かれる。

④ USER IDーに引続いて，センターから認可された登録番号（user id）を入力する。正しく入力された場合のみ⑤のパスワードの要求が行われる。

⑤ PASSWORDーに続いて（他の利用者からパスワードが使用されないようにするためのストライク・オーバーマスク（strike over mask）が出力される端末もある。そのときはこの上に）利用者はパスワードを入力してゆく。正しく入力されたときのみ⑥のサブシステム名の入力要求が行われる。④と⑤を2度間違うとシステムとの接続が切られる。

⑥ SYSTEM?に続けて，使用するサブシステム名（この例ではFORTRANサブシステム）を入力する。この時点でタイムシェアリング処理をやめたい場合は [BYE] と打けんする。

⑦ FORTRANサブシステムは，利用者が新しくソース・テキストの入力開始（NEW）を望むのか，このセッションの中で作成した同一のソーステキストの使用（SAME）を望むのか，あるいは既に保存しているソース・テキスト・ファイルの呼出し（OLD）を望むのかを問合せてくる（NEW OR OLDーー）。これに対して

{ NEW / OLD␣ファイル名 / SAME }注) (SD)

注）{ } はパラメータのどれかを選ぶこと
〔 〕は省略可のパラメータであること
）を表すものとする。

これらの応答の結果，テキスト用のファイル（カレント・ファイルという（→2.1））の設定が終了する。

OLD OR NEWに対する応答の後，

*△

のメッセージを出力後，システムはビルド・モード(build mode：タイムシェリング用コマンドまたは各サブシステムへの入力が可能な状態をいう)になる。

⑧ *に続いてコマンドAUTOを入力する。これ以後，システムはオートマティック・モードとなり，ライン番号を自動的に生成する。オートマティック・モードは，ライン番号の出力に引続いて㊿ を入力することにより終了する。オートマティック・モード中は，コマンドの入力はできない。AUTOコマンドの形式は

```
AUTO〔X〕␣〔m〕〔,n〕
```

m：開始ライン番号，　n：ライン番号の増分

（省略時のm，nの値は10である）

AUTOXとすると，ライン番号の直後のスペース(␣)は付加されない。

⑨～⑳ ライン番号に続いてテキストを入力してゆく。ソース・テキストにはライン番号が付けられ，この番号によって編集されてカレント・ファイルに蓄積されてゆく。ライン番号に続く文字には次の働きがある。

(a) ␣（スペース記号）	ライン番号の直後にスペースがあり，以降に"＆"以外のスペースでない文字が現れると，この文字は新しい文の始めを示す。この文字が数字のときには，文番号とみなされる。
(b) ＆（アンパー・サンド）	ライン番号に続くスペースでない最初の文字が"＆"の場合は継続行とみなす。
(c) ♯（ナンバー・サイン）	ライン番号直後の文字が"♯"のときは，"♯"直後の文字が第1桁目となる。
(d) C	ライン番号直後の文字が"C"であれば注釈行とみなされる。

㉑ マニュアル・モードになり，コマンドの入力またはライン番号付テキストの入力が可能となる。

㉒，㉓ マニュアル・モードで，ライン番号002，004のソース・テキストをライン番号010の前に追加する。

㉔，㉕ この時点までのカレント・ファイルの内容を，SAVEコマンドにより新たに開設したパーマネント・ファイル（ファイル名QDAFL01）に保存する。このファイルが作成済の場合は，㉟のRESAVEコマンドを使用する。

㉖～㉙ カレント・ファイルにおいて，ライン番号が2から10までの間にある行全部を端末上に表示する。

LISTコマンドの形式は

```
LIST␣〔a, b, c－d, …, g－h〕
```

オペランド省略時：全行表示

a，b，f：ライン番号a，b，fの行表示

c－d，g－h，…：ライン番号cからd，gからhの行表示

㉚ ライン番号70の内容が間違っていたので，正しいものと置き換える。

㉛ ライン番号30の行を削除する。削除は，このほかDELETEコマンドによっても行うことができる。

DELETEコマンドの形式は

```
DELETE␣{ ; *
         a, b, c－d, …, g－h }
```

; * 場合：全行削除

a，b，f，…：ライン番号a，b，fの行削除

c－d，g－h，…：ライン番号cからd，gからhの行削除

㉜ カレント・ファイルの内容を翻訳後実行する。

㉝ 文READ，M，Nの実行により，FORTRANサブシステムは変数M，Nに対するデータ入力を要求してくるので，=に引続きその値をコンマ(,)で区切って入力する。

㉞ PRINT文の実行によって，それぞれの変数の値がFORMAT文の指定に従って出力される。

㉟ 入力したソース・プログラムが正しく動作したので，この内容を㉔で作成済のパーマネント・ファイルに再度保存する。

㊲〜㊷ ファイルQDAFL01の内容をシリアル・プリンタ上に印刷する。

PRINTERがビジー（busy）の場合，他の利用者の印刷が終了するのを確認後再度繰返す。

㊸ BYEコマンドによって会話を終了させる（ログオフ・シーケンス）。これによりセッションが閉じる。

㊹ このセッションでの計算に要した費用（a），今までの総使用額（b）と予算額に対する総使用額の%表示（c）が次の形式で出力される。

** COST　　¥a　TO　DATE：　　¥b＝c%

㊺ 利用開始時刻（$h_1h_2\cdot i_1i_1i_1$），利用終了時刻（$h_2h_2\cdot i_2i_2i_2$），利用年月日（m′m′／dd／yy）が以下の形式で出力される。

ただし年月日は19yy年m′m′月dd日とし，時刻をhh時mm分ss秒とするとき，iiiは（mm×60＋ss）／3600の小数以下第3位までを表すものとする。

** ON　AT　$h_1h_1\cdot i_1i_1i_1$ −OFF　AT　$h_2h_2\cdot i_2i_2i_2$　ON　m′m′／dd／yy

㊻ ㊺のメッセージ出力後，端末はタイムシェアリング・システムとの接続が切断される。

2. タイムシェアリング・システムにおけるファイルとコマンド

システムには2種類のファイルがある。第1は，会話中（セッション）のみ有効で端末の切断とともに消滅するテンポラリ・ファイル（一時ファイル）で，第2は，センターまたは利用者が作成し，消去の指示がない限り消滅することはないパーマネント・ファイル（保存ファイル）である。

2.1 テンポラリ・ファイル

テンポラリ・ファイルにはシステムから与えられるものと利用者が作成するものとがあるが，ここでは前者についてのみ説明する。

(1) コレクタ・ファイル（ファイル名SY**）

コレクタ・ファイル（collector file）は，利用者の会話開始時点で各端末ごとに割当てられ，ビルド・モード（*に続いてコマンドまたはライン番号付データの入力可能な状態）の間，ライン番号が先行する入力データはすべてこのファイルに集められる。コレクタ・ファイルが一杯になったとき，またはある種のコマンドを受取ったときに，ファイルの内容がカレント・ファイルに併合される。

(2) カレント・ファイル（ファイル名 *SRC）

カレント・ファイル（current file）は，サブシステムからのOLD OR NEW 要求に対する応答の時点で初めて端末利用者ごとに割当てられる。このファイル上に，ニュー・ファイル（空のファイル）が作成されたり指定された作成済のパーマネント・ファイルの内容が複写されたりする。カレント・ファイルには，OLD OR NEW応答あるいはNEW，OLDコマンドで選択されたファイルの内容にその後の修正内容が付加されて入っている。カレント・ファイルの内容は，SAVEあるいはRESAVEコマンドによってパーマネント・ファイルに転写される。

2.2 パーマネント・ファイル

パーマネント・ファイルは，SAVE，PERMコマンドあるいはACCESSサブシステムのCREATEコマンド（→フ利5.3.1）によって開設される。

2.2.1 パーマネント・ファイルの指定形式と説明

(1) 指定形式一覧

① ファイル名

② ファイル名$パスワード

③ 〔user id〕／カタログ名〔$パスワード〕／…／ファイル名〔$パスワード〕

以上①〜③の形式を修飾ファイル名と略す。

④ 修飾ファイル名 "代替名" 注)

⑤ 修飾ファイル名，オプション1，オプション2，…

⑥ 修飾ファイル名 "代替名"，オプション1，オプション2，

⑦ 修飾ファイル名（ i－j ）

⑧ 修飾ファイル名 "代替名"（ i－j ）

⑨ 修飾ファイル名 "代替名" オプション1，オプション2，…（ i－j ）

注) " は引用符を表す。

(2) 説　　明

形式① UMC（User Master Catalog）直下にあるファイル（クイック・アクセル・ファイル），もしくは，一度形式②③⑤⑦でアクセスしたファイルを指定する場合。

形式② UMC直下にあるパスワード付ファイルを指定する場合

形式③ サブカタログ下にあるファイルを指定する場合で，パスワードの付いてないものはパスワードを省略できる。user id は指定ファイルが利用者自身のものである場合は省略できる。

形式④ タイムシェアリング・システムでは，9文字以上の名前のついたファイルはアクセスできないので，8文字以内の代替名を引用符（"）で囲んで与える。以後のアクセスはこの代替名のみで行える。

形式⑤ 形式⑥ } ファイル指定の際にオプションを付加できる。次のようなオプションが指定できる。

- R]EAD 注)　パーミッションの指定
- W]RITE
- MODE／{ RAND / SEQ }／アクセス・モードの指定
- S]IZEn　ファイルのサイズ指定，nはファイルの大きさでLリンク（1Lリンク＝320語）単位

注) a]xxx は axxx を省略して a と書けることを意味する。

形式⑦ 形式⑧ 形式⑨ } この指定によりファイルの内容の必要な部分（ファイル・セグメント）のみを指定することができる。i，j は，それぞれ対象とするファイル内容の開始ライン番号，終了ライン番号を表す。ファイル名（ i－j ）を以後ファイル・セグメント名と略す。

2.2.2 パーマネント・ファイルの構造と指定例

(1) 構　　造

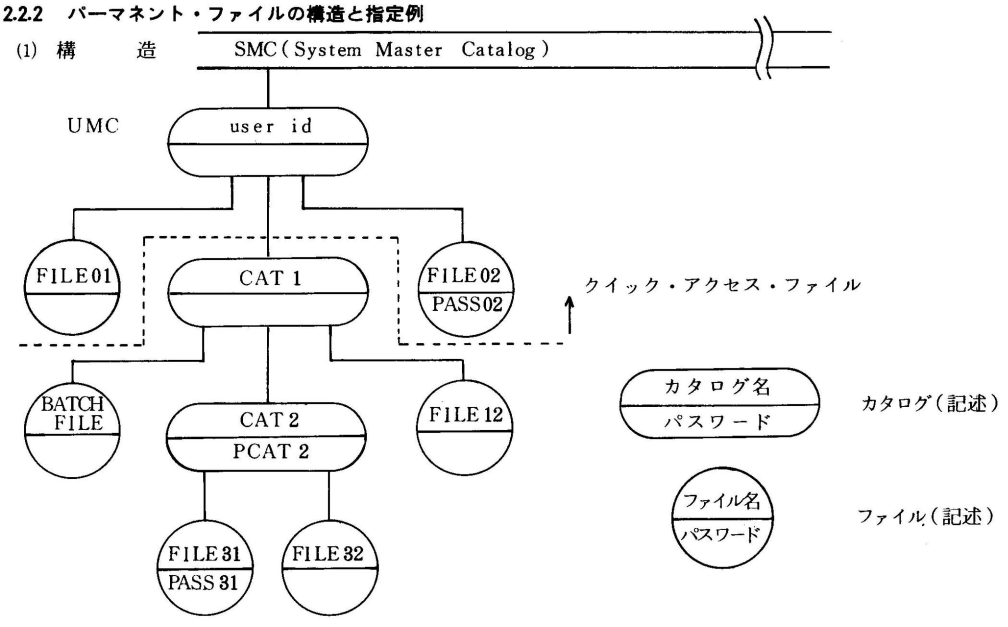


図2.1　ファイルの構造

(2) ファイル指定例

コマンド名	ファイル名	形式
OLD	FILE 10	①
OLD	FILE 02 $PASS 02	②
RESAVE	/CAT 1/CAT 2$PCAT 2/FILE 31$ PASS 31	③
RESAVE	user id/CAT 1/CAT 2$PCAT 2/FILE 31$ PASS 31	③
GET	/CAT 1/BATCHFILE"BFILE"	④
GET	user id/CAT 1/FILE12, MODE/RAND/	⑤
GET	/CAT 1/BATCHFILE"BFILE",MODE/SEQ/	⑥
OLD	FILE 01 (50－100)	⑦
LIST	user id/CAT 1/BATCHFILE"BFILE"(10－300)	⑧

2.2.3 パーマネント・ファイルの略式指定法とアベイラブル・ファイル・テーブル

アベイラブル・ファイル・テーブル(AFT)は会話を開始した時点で端末利用者ごとに割当てられ，会話が終了するまで有効なテーブルで，下記のタイムシェアリング・コマンドを使用した際にそのファイル名が登録される。一旦AFTに登録されると，以後形式①のファイル名のみ，あるいは代替名のみでアクセスでき，複雑なカタログ名やパスワードを省略することができる。

(1) AFTに登録できるコマンド(→3)

GET
LENGTH
LIST
OLD
PERM
RESAVE
RUN
SAVE

(2) AFTから取除くコマンド(→3.4.2)

REMOVE

(3) AFTの内容を表示するコマンド(→3.9)

STATUS

AFTには，コレクタ・ファイルおよびカレント・ファイルを含めて計20個までのファイルを登録できる。

2.3 ファイルとコマンドの関係

図2.2にタイムシェアリング・システムで使用するファイルとコマンドの関係を示している。

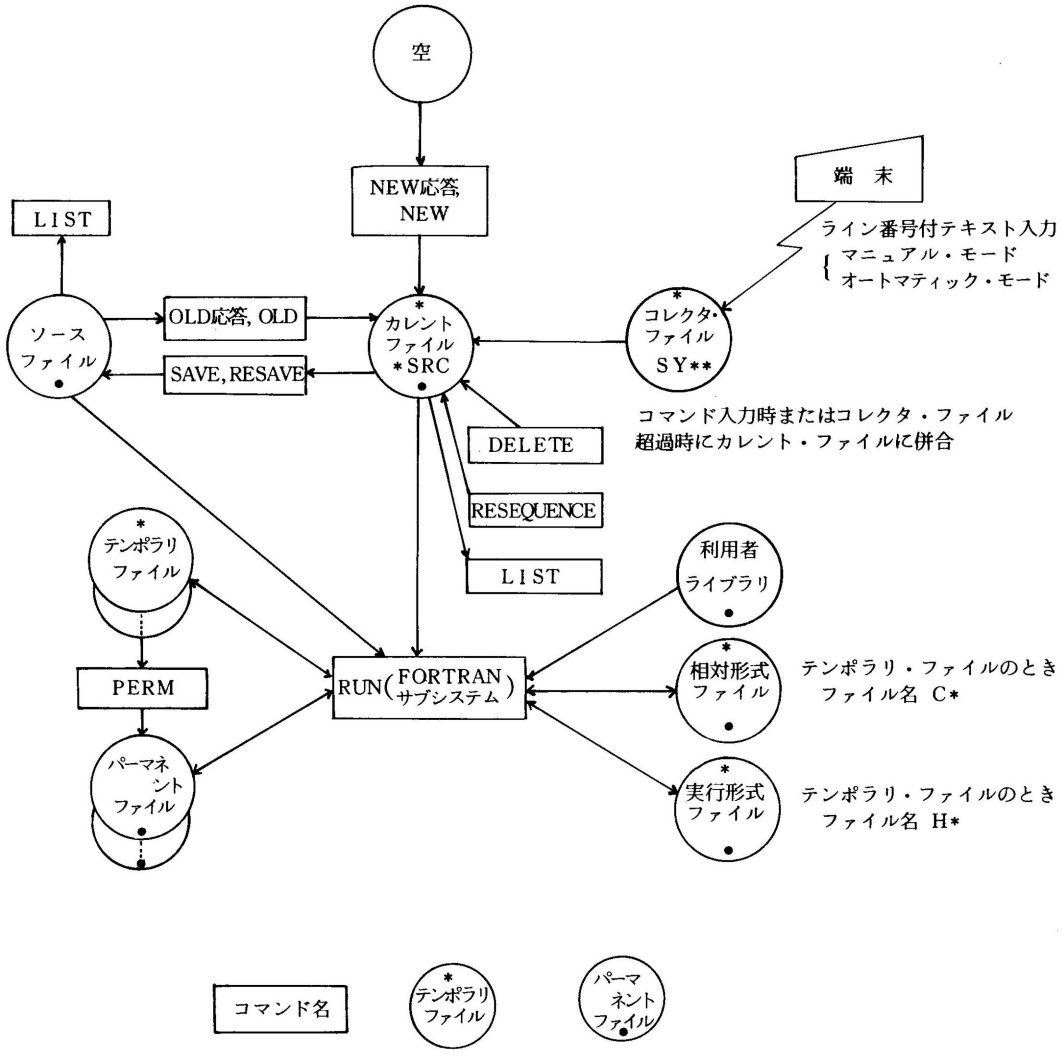


図 2.2　ファイルとコマンドの関係

3. タイムシェアリング・システム用コマンド

3.1 コマンド一覧

記号 ␣ 略号を表す
{ } この中からどれかを選ぶことを示す
〔 〕 省略可のパラメータであることを示す

機能	コマンド名	パラメータ	意味	記述箇所
会話の開始・終了	NEWUSER	な し	新しいログオン手順（会話開始処理）の実行	49
	BYE	な し	ログオフ手順（会話終了処理）の実行	49
サブシステムの選択・実行	SYSTEM	サブシステム名	サブシステムの新たな選択	49
	DONE	な し	現在使用中サブシステムの終了	50
	RUN (H)	各 種	現在選択中サブシステムの実行	50
アベイラブル・ファイル・テーブル（AFT）の制御	GET	修飾ファイル名〔,オプション〕	AFTへのファイル名の登録	52
	REMOVE	{ ファイル名1;…;ファイル名n / CLEARFILES / PERMFILES / TEMPFILES / STARFILES }	AFTからオペランドで指定された形式のファイルを取除く	52
カレント・ファイルに対する指示	NEW NEWP NEWP#	ファイル名 ファイル名	指定された形式のファイルを新たにカレント・ファイルとする	52
	OLD	{ ffs / ffs_1 { : } … { : } ffs_n }	ffs：ファイル名またはファイル・セグメント名 指定された形式のファイルの内容がカレント・ファイル上に複写される	53
	OLDP OLDP#	ファイル名 ファイル名		
	SAVE	ファイル名〔,オプション〕	カレント・ファイルの内容をパーマネントファイルに保存する	54
	RESAVE	ファイル名		55
カレント・ファイルまたはパーマネント・ファイルの内容に対する指示	AUTO (X)	〔m〕〔,n〕	ライン番号を自動的に発生する	55
	DELETE	{ a,b,…,c−d,…,e,…,f−g / ; * }	指定されたライン番号の行を削除する	55
	RESEQUENCE	〔〔m〕〔,n〔,i−j〕〕〕	ライン番号の再順序付けを行う	56
	LIST LISTH LISTEn LISTI LIRTL	{ 〔a,b,…,c−d,…,e,…,f−g〕 / ファイル名かファイル・セグメント名 / ファイル名(a,b,…,c−d,…,e,…,f−g) } n	カレント・ファイルまたはパーマネント・ファイルの内容を端末上に表示する	56
	PRINT		ファイルの内容をシリアル・プリンタに印刷する	58
	LENGTH	〔ファイル名〕	ファイルの大きさを端末上に表示する	58
	FORM	〔ファイル名〕	ファイルの行形式をFORMとし，段落付けも行う	58
ファイルの制御	CATALOG	各 種	カタログ／ファイルの登録状況を表示する	58
	PERM	テンポラリ・ファイル名；パーマネント・ファイル名	テンポラリ・ファイルの内容をパーマネント・ファイルに複写する	59
	ERASE	ファイル名1;…;ファイル名n	ファイルの内容をゼロ・クリアする	60
	RELEASE		ファイルを消去する	60
	PURGE		ERASE後 RELEASEする	60
エラー問合せ	SYSTEM	HELP␣エラー番号	エラー・メッセージに関する説明を求める	60
状態表示	STATUS	〔FILES〕	システム利用記録やAFTの内容を表示する	60

3.2 会話の開始・終了コマンド

3.2.1 [NEWU] SERコマンド

(1) 機　　能　利用者のシステム使用料と会話の開始・終了時刻および年月日が表示され，新しいログオン手順（会話開始処理）が始まる。このコマンドはあくまで会話中に新しいログオン手順を始めるもので，端末からの最初のログオン手順またはログオフ（会話終了，BYEコマンドによる）後のログオン手順については，別に述べる（→5.1）。

(2) 形　　式　NEWU

```
* NEWU
** COST :  ¥ 100  TO  DATE      ¥ 1000 = 10 %
** ON  AT  15.045 - OFF  AT   15.445  ON  12/1/77
ACOS-6  TSS(R3.3)  ON  12/1/77  AT  15.445  CHANNEL  2300
```

（→3.2.2）

3.2.2 BYEコマンド

(1) 機　　能　利用者のシステム使用料を計算し，会話を終了させ，端末装置をシステムから切断する。その際，最後に選択したサブシステムに利用者のテンポラリ・ファイルが作成されていれば，テンポラリ・ファイルの数に関するメッセージが出され，その処理に対する質問をしてくる。質問と応答形式は以下の表に示しています。

質　　問	応　　答	意　　味
ファイル・コード ？	(SD) のみ	このファイルを消去し，次のファイルへ移る
	NONE	このファイルに続くファイルをすべて消去する
	SAVE␣ファイル名	テンポラリ・ファイルの内容をファイル名で指定したパーマネント・ファイルに保存する

(2) 形　　式　BYE

① テンポラリ・ファイルが作成されていない場合

```
                使用料          今までの使用料の合計  予算額に対する割合
                  ↑                    ↑              ↑
* BYE
** COST:        ¥ 50   TO  DATE      ¥ 100 =        1 %
** ON  AT    10.500 - OFF  AT    10.555    ON     12/1/77
              ↓                   ↓               ↓
開始時刻の時間単位表示          終了時刻          年月日
（10時30分）          （10時33分18秒）  （1977年12月1日）
```

② テンポラリ・ファイルが作成されている場合

```
* BYE
    2   TEMPORARY  FILED  CREATED ?
01          ?    (SD)
02          ?    SAVE␣QDAFL 1
** COST
** ON
```

①と同じ形式

注）サブシステム選択レベルでBYEを打けんするとBYEコマンドと同等になる。

SYSTEM ?　BYE

3.3 サブシステム選択・実行コマンド

3.3.1 [SYST] EMコマンド

(1) 機　　能　現在のサブシステムが終了し，指定されたサブシステムが新たに選択される。サブシステム名が省略されると，制御はサブシステムの選択レベル（→形式②）に戻る。

(2) 形　　式

① SYST　サブシステム名

```
* SYST␣FORTRAN␣NEW
      ↑        ↑
*   サブシステム名  カレント・ファイル種別
```

左記の例はログオン手順で

SYSTEM ?　FORTRAN

OLD OR NEW - NEW

とした場合と同等である。

② SYST

* SYST 注)

SYSTEM? FORTRAN␣NEW

注) SYSTEM? BYE

とするとこの時点で会話は終了する。

3.3.2 DONEコマンド

(1) 機　　能　現在のサブシステムを終了させ，サブシステム選択レベルに制御を移す。

(2) 形　　式　DONE

* DONE

SYSTEM? サブシステム名

3.3.3 RUNコマンド

(1) 機　　能　現在選択されているサブシステムを実行する。

(2) 形　　式

① RUN パラメータ または RUNH パラメータ

サブシステムを実行する。RUNHは，サブシステムの実行に先立ち日付と時刻を出力する。

パラメータは各サブシステムにより意味が異なる。

② RUN－n パラメータ または RUNH－n パラメータ

サブシステムの実行をn秒間行う。nは $1 \leqq n \leqq 900$ の整数

(3) FORTRANサブシステムの例

(a) 形　　式　RUN 〔fs〕=〔fh〕〔;fc〕〔(opt)〕〔u lib〕〔#fe〕 注)

(b) 説　　明

① fs　コンパイラまたはローダに対する入力ファイルで次のもののファイル名を指定できる。

- カレント・ファイル
- 相対形式プログラム・ファイル（C＊形式ファイル）
- 実行形式プログラム・ファイル（H＊形式ファイル）
- 原始プログラム・ファイル

無指定の場合はカレント・ファイルが入力となり，複数のファイルを指定する場合はセミコロン(;)で区切って指定する。

② fh　致命的なコンパイル・エラーがない場合に，ローダによって作成される実行形式プログラムを格納するファイルで，fhによるファイル名の指定がない場合はテンポラリ・ファイルは作成されず，代わりにローダはオブジェクト・プログラムが実行できるコア・イメージを作り，直接実行に入る。指定されたファイルが存在しないときは，36 Lリンクのパーマネント直接編成ファイルが作成される。

③ fc　コンパイル結果の相対形式プログラムを格納するファイルである。指定されたファイルが存在しないときは，初期値3 Lリンク，最大値20 Lリンクのパーマネント順編成ファイルが作成される。

④ opt　コンパイラまたはローダに対するオプション指定で，次のようなものがありコンマ(,) で区切って並べる。

FORM, NLNO　原始プログラムの形式で，ライン番号を付けない通常パンチされたカード形式がFORM，NLNO形式である。標準はNFORM，LNO形式。

OPTZ　コンパイルの最適化指定，標準はNOPTZ

NOGO　コンパイルのみ。標準はコンパイルと実行(GO)。

注) 区切り記号＝は，fh, fc, opt, ulibをすべて省略したときのみ省略可であり，かつ省略しなければならない。

ULIB　　利用者ライブラリを使用する。

⑤ ulib　optのパラメータでULIBオプションを選択したときに，利用者のライブラリ・ファイル名を指定する。最大9個まで指定でき，ファイル名の間はセミコロン(;)で区切る。

⑥ fe　実行時に必要とするファイルを指定する。

指定の仕方は次のうちのどれかまたはその組合せである。

- ファイル・コード；…
- ファイル名"ファイル名の代替ファイル・コード"；…
- "ファイル・コード"；…

ファイル・コードのみ指定の場合には，利用者カタログ下の同一名のクイック・アクセス・ファイルが探され，あればそのファイルが選択され，なければテンポラリ・ファイルが作成される。ファイル名"ファイル・コード"は，左側に指定したファイル名のファイルは以後""に指定されたファイル・コードの代替名が付けられ，このファイル・コードで識別されることを意味する。"ファイル・コード"は，このファイル・コードを端末に割当てることを指示する。通常は05, 06が端末との入出力をするものとして 暗黙のうちに指示されている。

(c) 使 用 例

① カレント・ファイルの原始プログラムを実行

```
*RUN
```

② ファイル名SFILE 0

```
*RUN  SFILE 0
```

③ カレント・ファイルとSFILE 0の原始プログラムを実行

```
*RUN      *(注1) ;SFILE 0
```

④ カレント・ファイルの原始プログラムを実行し，データ・ファイルINPUTを使用

```
*RUN  # INPUT " 01"  (注2)
```

FORTRANプログラム

```
    READ(1, 100)…
100 FORMAT(……)
```

⑤ カレント・ファイルの原始プログラムを実行し，入出力機番03, 04を端末とする。

```
*RUN  # "03" ; "04"
```

FORTRANプログラム

```
    READ(3,100)  …端末からの入力
    WRITE(4,200) …端末への出力
```

⑥ カレント・ファイルの原始プログラムを実行し，テンポラリ・ファイルを使用

```
*RUN  #  01,02
```

FORTRANプログラム

```
    WRITE(1,200)  ……
    WRITE(2,210)  ……
```

⑦ カレント・ファイルをコンパイルし，相対形式プログラム・ファイルRFILE 1を作成(注3)

```
*RUN  = ; RFILE 1(NOGO)
```

⑧ カレント・ファイルの原始プログラムと相対形式ファイルRFILE 1のプログラムを実行

```
*RUN    * ; RFILE 1
```

⑨ ファイルSFILE 0とカレント・ファイルとの原始プログラム，および相対形式ファイルRFILE 1のプログラムを実行

```
*RUN  SFILE 0 ; * ; RFILE 1
```

注1) カレント・ファイルを陽に指定しなければならないときは*と書く。

注2) 再度RUNするときにはREMOVE␣01を実行しておかないといけない。

注3) 指定されたファイルがないときには3Lリンクの順編成パーマネント・ファイルが作成される。ファイル領域は最大20リンクまで拡張される。

⑩ カレント・ファイルと利用者ライブラリ・ファイルLFILE 1, LFILE 2を使用して実行。

＊RUN　=(ULIB) LFILE 1 ; LFILE 2

⑪ カレント・ファイルの原始プログラムを実行し，実行形式ファイルEFILE 2を作成 注4)

＊RUN　=EFILE 2

⑫ 実行形式ファイルEFILE 2を実行

＊RUN EFILE 2

注4) 指定されたファイルがないときには，36Lリンクのパーマネント直接編成ファイルが作成される。

3.4 アベイラブル・ファイル・テーブルに対する登録・削除コマンド

3.4.1 GETコマンド

(1) 機　　能　カタログやパスワードの記述を必要とするファイル(修飾ファイル)をファイル名だけで参照できるようにするため，ファイル名をアベイラブル・ファイル・テーブル(AFT) に登録する。

(2) 使 用 例　＊GET　user id／CAT1$PCAT1／FILE 1 $ PASS 1

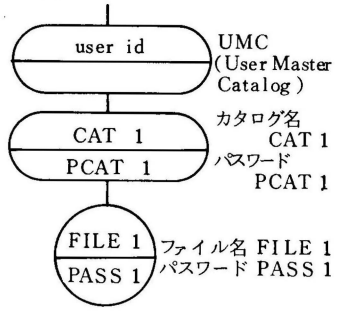
修飾ファイル名

右図のファイル名FILE 1を上記のように修飾ファイル名で指定してAFTに登録すれば，以後

OLD　FILE 1

LIST　FILE 1

のようにファイル名だけで参照できる。



3.4.2 REMOVEコマンド

(1) 機　　能　AFTに入るファイル名の数は限られている(コレクタ・ファイルSY＊＊，カレント・ファイル＊SRCを含めて，その数は20に設定されている)ので，AFTがあふれた場合にこのコマンドによって指定されたファイルをAFTから取除く。

(2) 形　　式

① REMO　ファイル名1 ; …… ; ファイル名n

ファイル名1,……，ファイル名nをAFTから取除く。

② REMO CLEARFILES

AFTに登録されている＊SRC，SY＊＊以外のすべてのファイルを取除く。

③ REMO PERMFILES

AFTに登録されているパーマネント・ファイルをすべて取除く。

④ REMO TEMPFILES

AFTに登録されている＊SRC，SY＊＊以外のすべてのテンポラリ・ファイルを取除く。

⑤ REMO STARFILES

AFTに登録されているワーク・ファイルのみを取除く。

3.5 カレント・ファイル指示コマンド

3.5.1 NEWコマンド

(1) 機　　能　新しいファイルをカレント・ファイル上に作成する。機能によって，NEW, NEWP, NEWP＃の3種類のコマンドがある。(→図2.3)

(2) 形　　式

① NEW　ニュー・ファイル(空のカレント・ファイル)が開始する。その際，カレント・ファイルが存在しなければ新たにテンポラリ・ファイルが作成され，既にNEWP＃，OLDP＃コマン

ドで作成済であればその内容がクリア（帰零）される。

② NEWP ファイル名

指定された名前をもつファイルがカレント・ファイルとして，新しく作成される。従って同一名のファイルが既に存在する場合にはエラーとなる。このファイルは，他のOLDあるいはNEWコマンド（OLD, OLDP, OLDP#, NEW, NEWP, NEWP#）が与えられるまで，利用者のカレント・ファイルの役割を果たす（*SRCという代替名でオープンされる）。

③ NEWP# ファイル名

指定された名前をもつパーマネント・ファイルが新しく作成され，会話終了まで，あるいは，別のOLDP, OLDP#, NEWP, NEWP#コマンドが与えられるまで，カレント・ファイルとして使用される（*SRCという代替名でオープンされる）。従って，普通のNEWコマンドが与えられたときには以前の内容がクリアされ，普通のOLDコマンドが与えられるとそのファイル内容が，NEWP#で指定されたファイル上に複写される。

3.5.2 OLDコマンド

(1) 機　　能　パーマネント・ファイルの内容をカレント・ファイル上に複写する。

機能によって，OLD, OLDP, OLDP#の3種類のコマンドがある（→図2,3）。

(2) 形　　式

① OLD ファイル名またはファイル・セグメント名

ファイル・セグメント名はファイル名（i, j）として表現し，ファイルのライン番号iからjの内容を意味する。

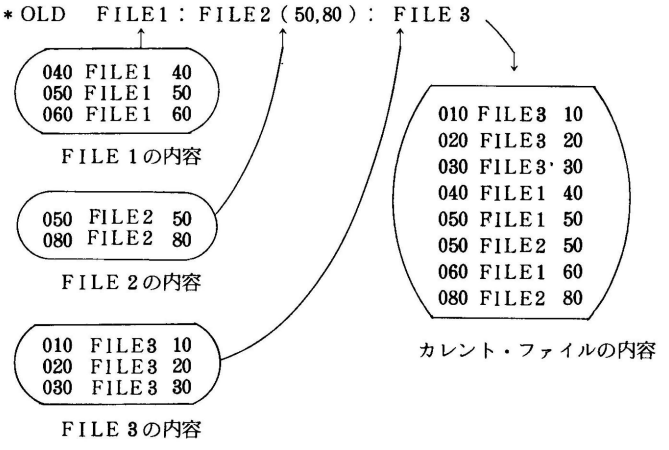
ファイル名またはファイル・セグメント名で指定したファイルの内容がカレント・ファイル上に複写される。

② OLD ファイル名1またはファイル・セグメント名1；…；ファイル名nまたはファイル・セグメント名n

セミコロン(;)で区切られたn個のファイルまたはファイル・セグメントが，指定された順に接合されて，カレント・ファイル上に複写される。

③ OLD ファイル名1またはファイル・セグメント名1：…：ファイル名nまたはファイル・セグメント名n

コロン(:) で区切られたn個のファイルまたはファイル・セグメントの内容がライン番号でソートされて，カレント・ファイル上に併合される。



④ OLD ファイル名1またはファイル・セグメント名1；…：…；…：ファイル名nまたはファイル・セグメント名

形式②と形式③を組合せたもので，指定されたファイルまたはファイル・セグメントのリストが，接合あるいは併合されてカレント・ファイル上に転写される。

⑤ OLDP ファイル名

指定されたパーマネント・ファイルが，他の形式のOLDあるいはNEW コマンド（OLD, OLDP, OLDP#, NEW, NEWP, NEWP#）が与えられるまで，利用者のカレント・ファイルとしての役割を果たす（＊SRCという代替名でオープンされる）。

⑥ OLDR# ファイル名

指定されたパーマネント・ファイルが，会話終了まで，あるいは，別のOLDP, OLDP#, NEWP, NEWP#コマンドが与えられるまで，カレント・ファイルとして使用される（＊SRCという代替名でオープンされる）。従って，普通のNEWコマンドが与えられたときには以前の内容がクリアされ，普通のOLDコマンドが与えられるとその ファイルの内容がOLDP#で指定されたファイル上に複写される。

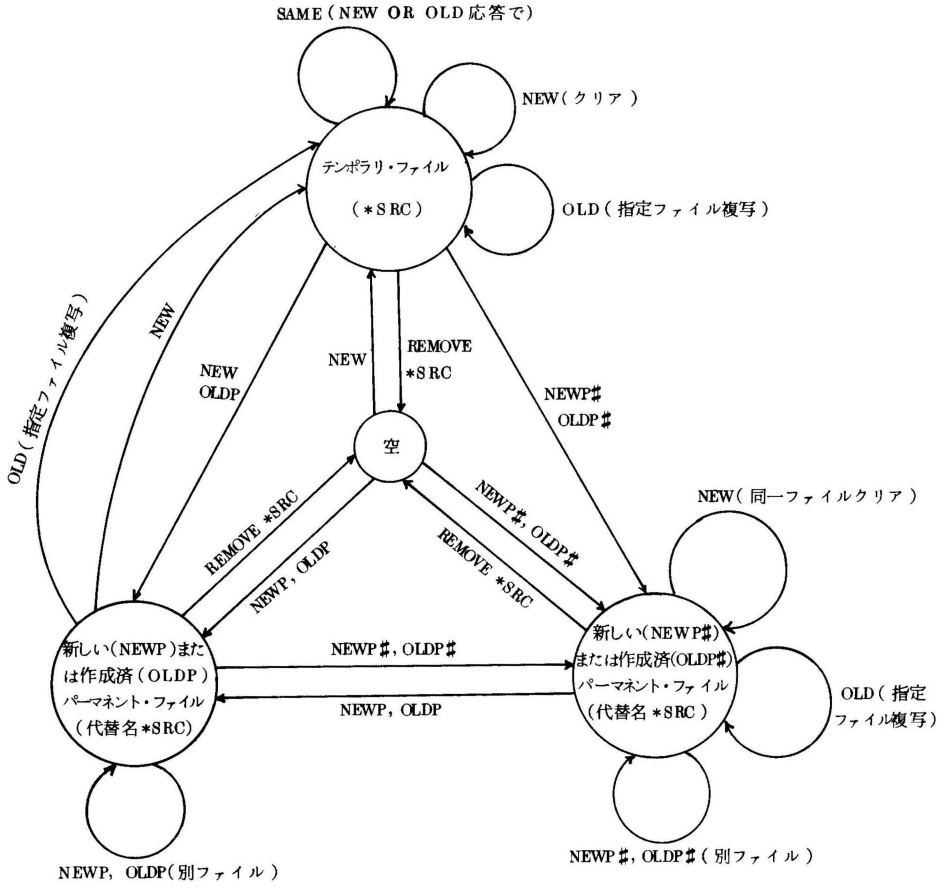


図2.3　カレント・ファイル状態遷移図

3.5.3　SAVEコマンド

(1) 機　　能　指定されたファイル名をもつファイルを新たに作成し，そのファイルにカレント・ファイルの内容を保存する。このコマンド実行後もカレント・ファイルの内容は変化しない。

(2) 形　　式　SAVE ファイル名1〔,オプション1〕〔,サイズ1〕;…;ファイル名n〔,オプションn〕〔,サイズn〕 注)

＊SAVE QDAFL01, SIZE 10 または S 10 DATA SAVED－QDAFL01

10 L リンクの大きさからなるファイル名QDAFL01のファイルにカレント・ファイルの

内容を保存する。

注) サイズはSIZEmまたはSmと書く。mはLリンク（1Lリンク＝320語）を単位とする。サイズ省略時は，初期値として必要な大きさの領域が確保され，最大値はその20倍の大きさがとられる。

3.5.4 RESAVEコマンド

(1) 機　　能　指定されたファイル名をもち，既に存在するパーマネント・ファイル上にカレント・ファイルの内容を保存し，前の内容と取替える。

(2) 形　　式　RESA ファイル名1；…；ファイル名n

```
* RESA  QDAFL01
DATA  SAVED－Q DAFL01
```

3.6 ファイル内容の操作・表示コマンド

3.6.1 AUTOMATICコマンド

(1) 機　　能　ライン番号を自動的に生成する。オートマティック・モードは(SD)のみの入力で終了する。オートマティック・モードにある間，コマンドの入力はできない。

(2) 形　　式

① AUTO 〔m〕〔,n〕

ライン番号をmから始めてnずつ増やしてゆく。省略時のnの値は10がとられ，省略時のmの値は，会話開始後の最初のAUTOコマンドの場合は10が，2回目以後の場合は前回の最終ライン番号がそれぞれとられる。

```
* AUTO                注1)
*010 ␣△A =  10.0
*020 ␣△B =  20.0
*030 ␣△C =  30.0
*040 ␣△(SD)
*
```

```
*AUTO␣ 100, 20
*100 ␣△A =  100.0
*120 ␣△B =  120.0
*140 ␣△C =  140.0
```

```
*AUTO␣, 100
*0010000 ␣△A＝ 1.0
*0010100 ␣△B＝ 2.0
*0010200 ␣△C＝ 3.0
```

ライン番号1000以上は7桁で出力する。この例では，前回の最終ライン番号が0010000であった。

② AUTOX 〔m〕〔,n〕

AUTOコマンドとの違いは，ライン番号の次にスペースがこないことだけである。

```
                    注2)
*AUTOX
*010 △A＝  10.0
*020 △B＝  20.0
*030 △C＝  30.0
```

注1) △はアトリビュート文字でディスプレイ上では空白に見えるが，空白(␣)コードがある訳でない。

注2) AUTOでは注釈文は入力できない，AUTOXでは空白を1個以上入力しないと数字はライン番号の続きに，Cで始まる文はすべて注釈文に見なされる。

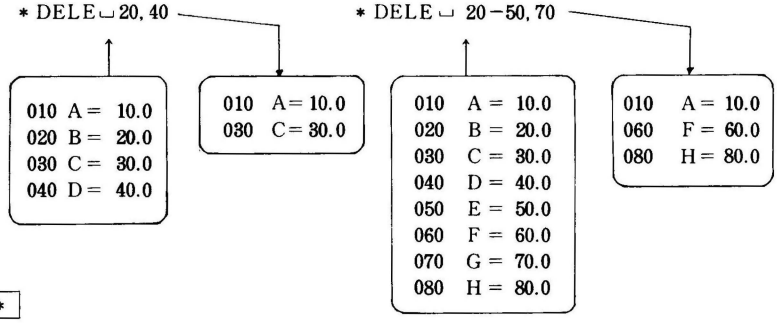
3.6.2 DELETEコマンド

(1) 機　　能　カレント・ファイルの内容を，部分的にあるいは全面的に削除する。

(2) 形　　式

① DELE a, b, …, c－d, …, e, …, f－g

ライン番号a，b，eの行を削除する。　ライン番号cからd，fからgの間にあるすべての行を削除する。



② DELE␣;*

カレント・ファイル内の行全部を削除する。

3.6.3 RESEQUENCEコマンド

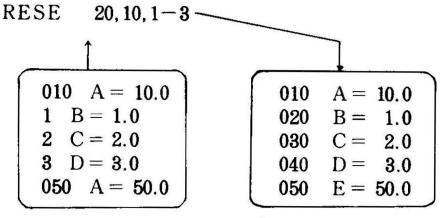
(1) 機　　能　カレント・ファイルのライン番号の再順序付けを行う。

(2) 形　　式

① RESE　　カレント・ファイルのライン番号を，10から始め10ずつ増えるように付け直す。

② RESE 〔m〕〔,n〔,i−〔j〕〕〕または〔m〕〔,n〔,〔i〕−j〕〕

カレント・ファイルのライン番号を，mから始めnずつ増えるように付け直す。その際，i−jのパラメータが指定されていれば，再順序付けの対象となるライン番号の起点がi，終点がjになることを意味する。このとき，iが空（すなわち−j）の場合は起点が最初の行を，jが空（すなわちi−）の場合は終点が最終行を指す。m，nは省略すると10がとられる。



③ RESEX 〔m〕〔,n〕

元のファイルのライン番号の有無に関係なく，カレント・ファイルの各行にmから始まりnずつ増加するライン番号を挿入する。元の行の最初の文字が数字のときは，ライン番号の後にスペース(␣)が1個挿入される。省略時のm，nの値は10である。

3.6.4 LISTコマンド

(1) 機　　能　カレント・ファイルまたは利用者のパーマネント・ファイルの内容を端末に出力する。

(2) 形　　式

① LIST

カレント・ファイルの内容を端末上に出力する。

② LIST a, b, …, c−d, …, e, …, f−g

カレント・ファイルのライン番号がa，b，…，eの行，cからd，…，fからgの間にある行を出力する。

③ LISTI n

カレント・ファイルの最初の行からn番目おきの各行を出力する。

④ LISTL

カレント・ファイルの最終行のみを出力する。LIST 99999999と同じ機能である。

⑤ LIST ファイル名またはファイル・セグメント名

ファイル・セグメント名はファイル名（i，j）で表現し，ファイルのライン番号iからjの内容を意味する。

ファイル名またはファイル・セグメント名で指定されたファイルの内容を端末上に出力する。

⑥ LIST ファイル名(a, b, …, c−d, …, e, …, f−g)

カレント・ファイルがファイル名で指定されたファイルとなる点を除けば，形式②と同じである。

⑦ LISTH 形式①,②,⑤のオペランド

ファイルの内容の出力に先立って，ヘッダ(日付と時刻)を出力する。

形式①，②および形式⑤に適用される。

⑧ LISTE n 形式①,②,⑤のオペランド

ファイルの各行をn桁で分割し，分割したものを端末上に1行ずつ出力する。nの指定がない場合は 72 とみなす。

```
*LISTE 10
```

```
010 ␣ 123456789012345678901
020 ␣ 23456789
```

```
010␣123456
7890123456
7890
020␣234567
89
```

(3) 使 用 例

① *LIST 10−20, 40, 60−70, 110

```
010  INTEGER WA, SA, SEKI, SYO, AMARI
020  READ, M, N
040  SA = M−N
060  SYO = M/N
070  AMARI = M−N*SYO
110  END
```

② *LIST QDAFL 01(30,60)

```
030  WA = M + N
040  SA = M−N
050  SEKI = M*N
060  SYO = M/N
```

○ カレント・ファイル
○ QDAFL 01

```
010  INTEGER WA, SA, SEKI, SYO, AMARI
020  READ, M, N
030  WA = M + N
040  SA = M−N
050  SEKI = M*N
060  SYO = M/N
070  AMARI = M−N*SYO
080  PRINT 10, M, N, WA, SA, SEKI, SYO, AMARI
090  10 FORMAT(7I5)
100  STOP
110  END
```

③ *LISTH

```
12/31/1977   13.30
   ↳日付        ↳時刻
```

```
右に同じ
```

④ *LIST QDAFL 01(20−50,70,100−110)

```
020  READ, M, N
030  WA = M + N
040  SA = M − N
050  SEKI = M*N
070  AMARI = M−N*SYO
100  STOP
110  END
```

⑤ *LISTI 4

```
010  INTEGER WA, SA, SEKI, SYO, AMARI
050  SEKI = M*N
090  10 FORMAT(7I5)
```

⑥ ＊LISTLまたは ＊LIST 99999999

```
110  END
```

3.6.5 PRINTコマンド

(1) 機　　能　ファイルの内容を，端末上ではなく，シリアル・プリンタ上に印刷する。あるいは，既に印刷中の出力をとりやめる。

(2) 形　　式

① プリント出力をする場合

```
＊PRINT
ACTION?  n ·····  n は印刷行数の最大値，省略すると 200 となる。
〔PRINTER IS BUSY〕 ·····  プリンタが全部使用中。空いているのを確認して最初から。
USERID?  user id; account number…user id と account number を入力する。
PASSWORD?  ////////  ·····  パスワードの入力
FILE STRING?  修飾ファイル名  ····  ファイル名の入力
YOUR LIST WILL BE PRINTED OUT TO Pi  ·····  機番Piのプリンタに出力される。
```

user id ： 登録番号

account number ： 支払責任者番号 またはクラス名（学生）

② プリント出力の取消を行う場合

```
＊PRINT
ACTION?  CANCEL  ·····  取消の指示
CANCELED SUCCESSFULLY  ·····  取消されたことの表示
```

3.6.6 LENGTHコマンド

(1) 機　　能　ファイルの大きさ（LLINK単位；1LLINK= 320 語）などを表示する。

(2) 形　　式

① LENG

カレント・ファイルの現在の大きさを表示する。

```
＊LENG
TSS  CURRENT  FILE  3  L LINK
```

② LENG ファイル名

ファイル名で指定したテンポラリ・ファイルまたはパーマネント・ファイルの大きさを表示する。

```
＊LENG  QDAFL 01
TSS    LINKED    PERM    FILE      10 LLINKS    - -USED    1  LLINK
```

順編成ファイル　パーマネント・ファイル　確保された大きさ　使用中の大きさ

3.6.7 FORMコマンド（東北大学大型計算機センター提供）

(1) 機　　能　ソース・プログラムをファイル上でFORM形式（→3.3.3.）に変換し，更にINDENTATION（段落付け）を行って端末に表示する。

(2) 形　　式　FORM　〔ファイル名〕

ファイル名を省略するとカレント・ファイルが対象となる。

3.7 ファイルの登録状況表示・複写・消去コマンド

3.7.1 CATALOGコマンド

(1) 機　　能　利用者自身のUMC（User Master Catalog）下に属するカタログ名およびファイル名などを端末上に表示する。

(2) 形　　式

① CATA

利用者自身のUMC下に属するカタログ名およびファイル名をすべて表示する。

② CATA　x（date, n〔, R〕），FIRST／カタログ名／

指定した名前のカタログより下位のカタログおよびファイルに対して，指定した条件に合致するものを選

んで端末上に表示する。

x，date，n，Rの意味は次のとおりである。

x：{ C　作成日付 / A　最終アクセス日付 / L　最終修正日付

date：xに対する日付をmm－dd－yy（例えば12－31－77）で指定する。

n：表示すべきファイルの数（最新のものから始めて古いものへ）。

R：指定すると，出力の順番を古いものから表示する。

③ CATA␣／カタログ名1／……／カタログ名i

指定された名前のカタログ（カタログ名i）下に属するカタログ名およびファイル名を端末上に表示する。カタログ名1，……，カタログ名i－1はカタログiに達するまでの上位のカタログ名を上から並べたものである。

④ CATA␣／カタログ名1／……／カタログ名i＊

指定された名前のカタログ（カタログ名i）の属性を端末に表示する。

カタログ名1,…，カタログ名i－1の説明は形式③と同じである。

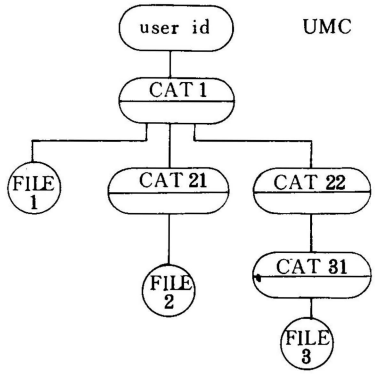
⑤ CATA␣〔／カタログ名1／……／カタログ名i／〕ファイル名

指定されたファイルの属性を端末に表示する。

(3) 使　用　例

① ＊CATA

```
LIST OF CATALOG－user id
CATALOGS
   FILES
CAT 1
   FILE 1
CAT 21
   FILE 2
CAT 22
CAT 31
   FILE 3
```



② ＊CATA␣／CAT 1／CAT 22／CAT 31

```
LIST OF CATALOG－CAT 31
CATALOGS
   FILES

   FILE 3
```

③ ＊CATA␣／CAT 1／CAT 21／FILE 2

```
FILE NAME－FILE 2
ORIGINATOR－user id
DATE CREATED－123177
DATE CHANGED－010578
LAST DATE ACCESSED－010578
NUMBER OF ACCESSES－5
MAX FILE SIZE－50 LLINKS                    （1 LLINK＝320語）
CURRENT FILE SIZE－5 LLINKS
```

FILE TYPE－LINKED

DEVICE－DP 3

GENERAL PERMISSIONS－NONE

SPECIFIC PERMISSIONS－NONE

3.7.2 PERMコマンド

(1) 機　　能　テンポラリ・ファイルの内容をパーマネント・ファイルに複写する。

(2) 形　　式　PERM テンポラリ・ファイル名；パーマネント・ファイル名

パーマネント・ファイルがまだ存在していないときには，指定した形式のファイルを新たに作成後複写する。このコマンド実行後テンポラリ・ファイルは消滅する。

3.7.3 ERASEコマンド

(1) 機　　能　指定されたファイルの内容をゼロ・クリア（帰零）する。

(2) 形　　式　ERAS ファイル名１；…；ファイル名ｎ

ファイル名で指定されたｎ個のファイルの内容がゼロ・クリアされる。

3.7.4 RELEASEコマンド

(1) 機　　能　指定されたファイルを消去する。

(2) 形　　式　RELE ファイル名１；…；ファイル名ｎ

3.7.5 PURGEコマンド

(1) 機　　能　指定されたファイルに対して，その内容をゼロ・クリア後消去する。

(2) 形　　式　PURG ファイル名１；…；ファイル名ｎ

このコマンドは ERASE　ファイル名１；…；ファイル名ｎ

RELEASEファイル名１；…；ファイル名ｎ　を続けて入力したものと同じである。

3.8 エラー・メッセージ説明コマンド（HELP）

(1) 機　　能　HELP サブシステムを呼出して，エラー・メッセージに関する説明を得る。

(2) 使 用 例

```
009－SYSTEM UNKNOWN          出力されたエラー・メッセージ
* SYSTEM␣HELP␣009           HELP サブシステムの呼出し，エラー番号 009 も同
  エラーメッセージの説明      時に入力
* SYSTEM HELP
PLEASE ENTER MESSAGE NUMBER－009
  エラー・メッセージの説明
```

3.9 状態表示コマンド（STATUS）

(1) 機　　能　現在までに使用したCPU時間，ファイル入出力回数，端末に出力した文字数およびアベイラブル・ファイル・テーブル（AFT）に登録されているファイル名を出力する。

(2) 形　　式

① STAT

```
* STAT
          端末番号
          ↑
CHANNEL 0203                                    会話開始時刻
                                                ↑
USER STATUS ON DEC 1, 1977 AT 10:45:20 LOG－ON 10:30:10
PROC TIME USED 1.50 SEC., 100 FILE I/O 2000 CHAR KEY I/O
               └CPU時間  └ファイルI/O数  └端末出力文字数
LIST OF OPEN FILES: FILE 1      FILE 2
                    ↑           ↑
                    └───────────┘
                                AFT上にあるファイル名
```

② [STAT FILES] または [STATF]

AFTに登録されているファイル名を出力する。

＊STATF

LIST OF OPEN FILES：FILE 3

4. タイムシェアリング・システム実行中のエラー・メッセージ （→マニ 18）

4.1 TSSからのエラー・メッセージ

4.1.1 TSS管理プログラムからのエラー・メッセージ

009－SYSTEM UNKNOWN

意 味 ： 要求したサブシステムがない。

対 策 ： サブシステム名のつづりをチェックすること。

4.1.2 種々のサブシステムからのエラー・メッセージ

〈50〉 a … a b … b

a … a
- FILE ファイル名
- CURRENT FILE …… カレント・ファイル
- COLLECTOR FILE …… コレクター・ファイル
- WORK FILE

b … b

① STATU 01

意味 ： 指定されたマスタ・カタログがない。

対策 ： 登録番号（user id）をチェックする。

② I／O ERROR, STATUS 02

意味 ： ディスク・ファイルの入出力エラー

対策 ： 再度試行してみる。

③ NO PERMISSION, STATUS 03

意味 ： 要求したパーミッションが許可されていない。

対策 ： 指定パーミッションの間違いか，ファイル開設時のパーミッションの誤まり。

④ FILE PRESENTLY BUSY, STATUS 04

意味 ： 要求したパーミッションを論理的に排除するようなアクセス・モード・パーミッションで，別の利用者がこのファイルをアクセス中である。

対策 ： 他の利用者が使用し終ってから利用する。

⑤ NON－EXISTENT FILE, STATUS 05

意味 ： 指定されたファイルがない。

対策 ： ファイル名の間違いかファイルそのものが作成されていない。

⑥ NON－UNIQUE NAME, STATUS 11

意味 ： 指定した新しいカタログ名またはファイル名が既に存在する。

⑦ NO FILE SPACE, STATUS 13

意味 ： ファイル作成用の領域がない。

対策 ： 不要なファイルを消去する。

⑧ INVALID PASSWORD, STATUS 14

意味 ： パスワードの間違いかパスワードが全然与えられていないかのどちらかである。

⑨ ILLEGAL CHAR, STATUS 34

意味 ： カタログ名，ファイル名あるいはパスワードに英数字，ピリオドあるいはマイナス記号以外の文字が含まれている。

⑩ FILE TABLE FULL, STATUS 36

意味 ： AFTに既に20個のファイルが登録されている。

対策 ： STATUS FILESコマンドによりAFTの内容を表示させ，REMOVEコマンドを使って不要なファイルをAFTから取除く。

⑪ DUPLICATE NAME, STATUS 37

意味 ： 指定されたファイル名はAFTに既に存在する。

対策 ： 別の代替名を与える。

〈51〉 aaaa I/O STATUS yy

意味 ： ファイルの入出力エラー

対策 ： パーマネント・ファイルの場合は，センターに連絡する。カレント・ファイル等テンポラリ・ファイルの場合には，一旦会話を終了させ，再度最初からやり直す。

〈52〉 CURRENT FILE NOT DEFINED

意味 ： カレント・ファイルがない。

対策 ： ログオフして再度やり直す。

〈59〉 FILE ファイル名 NOT IN TSS FORMAT

意味 ： TSSで作成したファイルでない。

〈60〉 NO DATE ON FILE ファイル名

意味 ： ファイルにデータが入っていない。

4.2 NPS（Network Processing Supervisor）からのメッセージ

(1) 形　　式　$ N1 nnn hh:mm:ss メッセージ本体

nnn ：メッセージ番号

hh:mm:ss ：発生時刻 hh時mm分ss秒

メッセージ本体 ：メッセージ内容

(2) メッセージ

① $N1005 hh:mm:ss ALREADY CONNECTED

意味 ： 既に回線が接続されている。

② $N1006 hh:mm:ss NOT CONNECTED

意味 ： 回線が接続されていない。

③ $N1007 hh:mm:ss UNACCEPTABLE NOW

意味 ： TSSのサービスを行っていない。

④ $N1021 hh:mm:ss NOT ACCEPT－NO RCV

意味 ： 入力要求がないのに打鍵した。

⑤ $N1022 NOT ACCEPT－TOO LONG

意味 ： 指定された値より長いメッセージを入力しようとした。

⑥ $N1042 hh:mm:ss DIS－TIME OUT

意味 ： 端末の入出力が17分以内に終了しなかったので，その端末を切断する。

5 タイムシェアリング・システム用端末機器操作法

5.1 一般的操作法

5.1.1 会話の開始（ログオン）

① モデムおよび端末の電源を入れ，システムとオンライン状態になっていることを確認する。

② $$$ CON, TSS ㊐　（㊐は1行分の送信を行うキーを表す）

接続（コネクト）処理は，各端末によって簡易型で行えるものもある（簡易コネクト）。

③ ACOS−6 TSS（R3.3） ON 03／10／78 AT 10.750 CHANNEL 2101

USER ID− 登録番号（user id）；支払責任者登録番号（account number）㊈　　TSS開始メッセージ

④ USERID−に引続いてuser idを入力する。正しく入力するとログオン・パスワードの要求がある。

PASSWORD−
xxxxxxxxxxxxxx㊈　　注）2度，利用者識別名とパスワードを間違うと端末は切断される。

⑤ パスワードが正しく入力されると，次のようなメッセージが出力されてセッションが開かれシステム選択レベルとなる。利用者は使用するサブシステム名を入力する。

SYSTEM？　サブシステム名㊈

5.1.2 会話の終了（ログオフ）

システム選択レベル（SYSTEM？）またはビルド・モード（→1.2）のいずれかの時点でBYEコマンド（→3.2.2）を入力することにより行う。

5.1.3 端末の自動切断

システムからの入力要求あるいは質問に対して17分以内に応答しない場合には，次のメッセージが出力され，その端末は切断される。

SYSTEM？　　17分以内に応答しない場合
$ N1042　　11：10：35　　DIS−TIME OUT

5.1.4 実行の中断

実行中のコマンドを中断したい場合は㊆（BREAK）キーを押下することにより行う。㊆キーは端末によって異なる。ファイル作成中に㊆キーを押下すると，コレクタ・ファイルにある入力データが消滅するので注意しなければならない。

＊RUN㊈　　RUNコマンド実行によりプログラムがループしたとき。
㊆　←
＊

5.1.5 回線障害時などの再接続法（再コネクト）

BYEコマンドによらないで，端末操作のミスや回線障害によりセッションが途中終了してしまった場合には，2分以内にログオンをし直し，最初のシステム選択レベルで次のようにする。

⋮
SYSTEM？△ CONNECT␣ xxxx㊈
　　└端末識別名の8進表示，今まで使用していたセッションのログオン・メッセージの末尾に表示されたCHANNEL xxxxと同一のものを指定する。

これにより以前使っていたセッションを引き続き処理することができる。

5.2 N6300／20Nキーボードディスプレイ操作法（→図5.1，図5.2）

(1) 電源投入

図5.1に示すつまみを手前に引くと電源が入る（ただし主電源を投入していること）。

(2) 輝度調節

図5.1に示すつまみを回して画面上の文字の輝度を調節する（右に回すと濃くなる）。

(3) 初期設定

操作盤上の「通信可」のランプ（→図5.1）が点灯しているのを確認して，[INITIAL]，[RESET]キーの順で押し装置の初期設定をする。

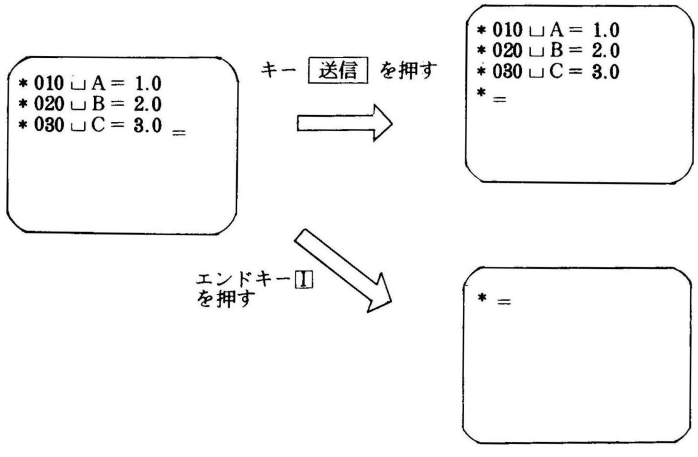
(4) システムへの入力操作

操作盤上の「けん盤可」のランプ（→図5.1）が点灯していることを確認の上，カーソールの示す位置へデータを入力する（カーソルは打鍵操作のたびに自動的に次の入力位置に移動する）。システムへの入力は行単位で行なわれ，1行分のデータを入力するには[送信]キーあるいはエンド・キー[I]を押す。

① 送信 キーを押した場合 …… 送信後ディスプレイ画面はクリアされず，引き続いて次の行の入力が可能となる。

② エンドキー I （消去送信 キー）を押した場合…送信後のディスプレイ画面はクリアされ，次の入力は画面の最初の行へ戻る。

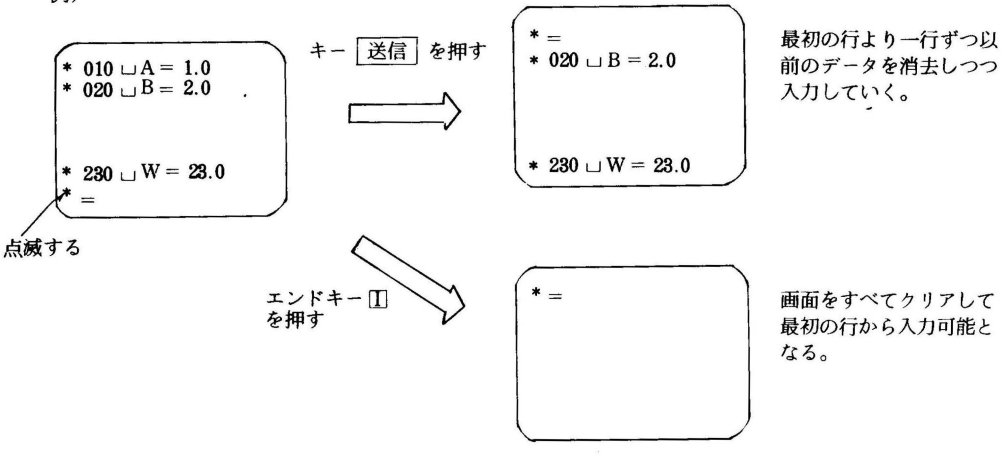
例）



注意 1　1行に入力できる最大文字数は 79 文字である。

2　画面上の入力できる最後の行になった場合，システムから点滅するアスタリスク（*）が表示されるのでこのとき 送信 キーあるいはエンドキー I を押す。*が点滅している行での 送信 キーあるいはエンド・キー I 以外の入力は無視される。

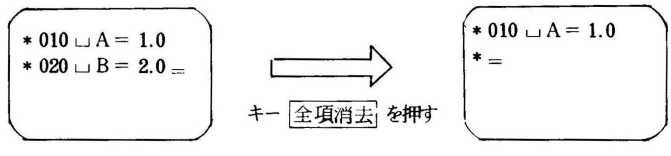
例）



(5) 編集操作

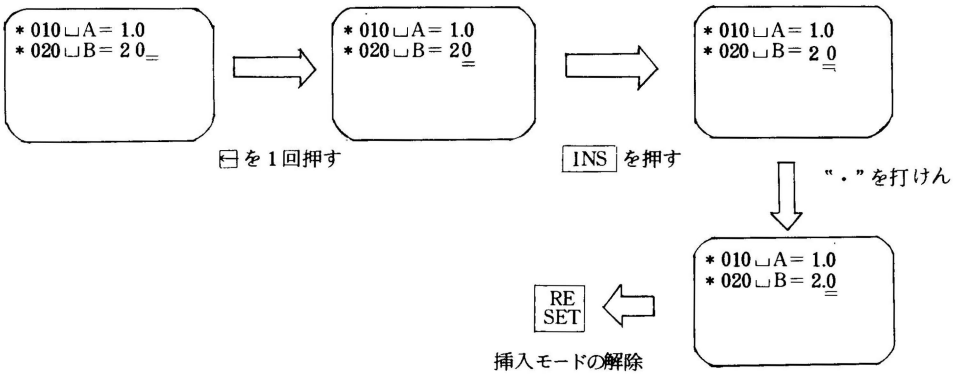
① 行全体抹消…… 全項消去 キーを使う。その行はクリアされカーソルはその行の先頭位置に戻り，一字目からの再入力が可能となる。

例）



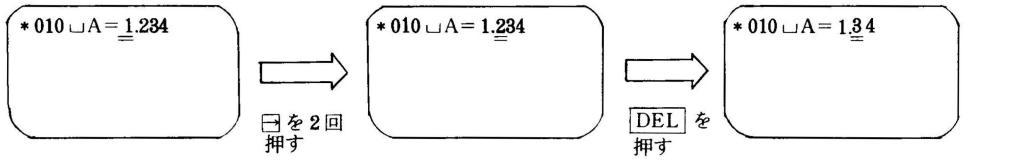
② 行の途中へ文字を挿入……カーソルムーブキー（図5.1で←→など）を使ってカーソルを挿入したい位置まで移動させ，INS キーを押し挿入したい文字を打けんする。

例）次図でB＝20をB＝2.0に変えたいとき



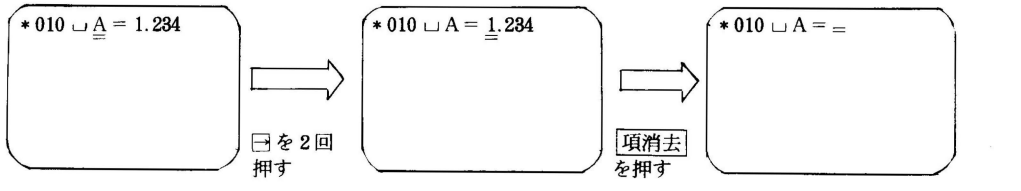
③　行の途中の文字を削除……カーソルムーブキーを使い削除したい文字の位置までカーソルを移動し，DELキーを押す。

例）A＝1.234をA＝1.34に変えたいとき。



④　行の途中以降の文字列を削除……カーソルを消去したい文字列の先頭位置まで移動し，項消去キーを押すとカーソルの示す位置以降の文字列が消去される。

例）A＝1.234の式で右辺を消去したいとき。



注意　送信キーを押した後でのその行の編集操作はできない。

(6)　システムから一画面以上のファイルを出力させるときの操作

データの表示が画面の最終行以上になるときには，システムから最終行において点滅するアスタリスク（＊）が表示されるので，利用者は次の二つの方法をとる。

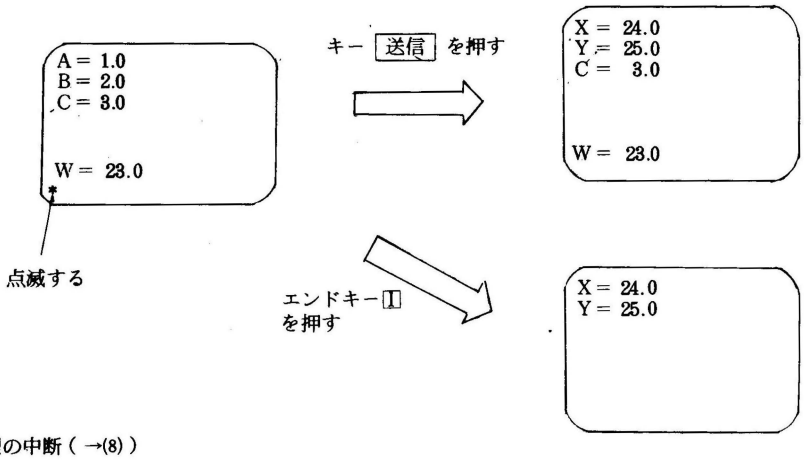
①　送信キーを押す……次の画面は最初の行より一行ずつ以前のデータを消去した後へ出力される。

②　エンドキー［I］を押す[注1)]……画面上の表示がすべてクリアされたのち，最初の行から順次データが画面上に出力される。

例）次のデータを画面上に縦列に出力するとする。

A＝1.0，　B＝2.0，　C＝3.0，……W＝23.0，　X＝24.0，　Y＝25.0

注1）　エンドキー［I］を［消去送信］キーと呼ぶこともある。



(7) 処理の中断 (→(8))

(8) ファンクション・キーの使用法 (図5.2)

① PF 1 を押す…… $$$ CON , TSS ㊎を打けんしたのと同じでありTSSとの接続処理が開始される。

② PF 2 を押す…… BREAK キーとして機能し，処理を中断する機能すなわち出力表示の中止， 実行の打切りなどに用いる。

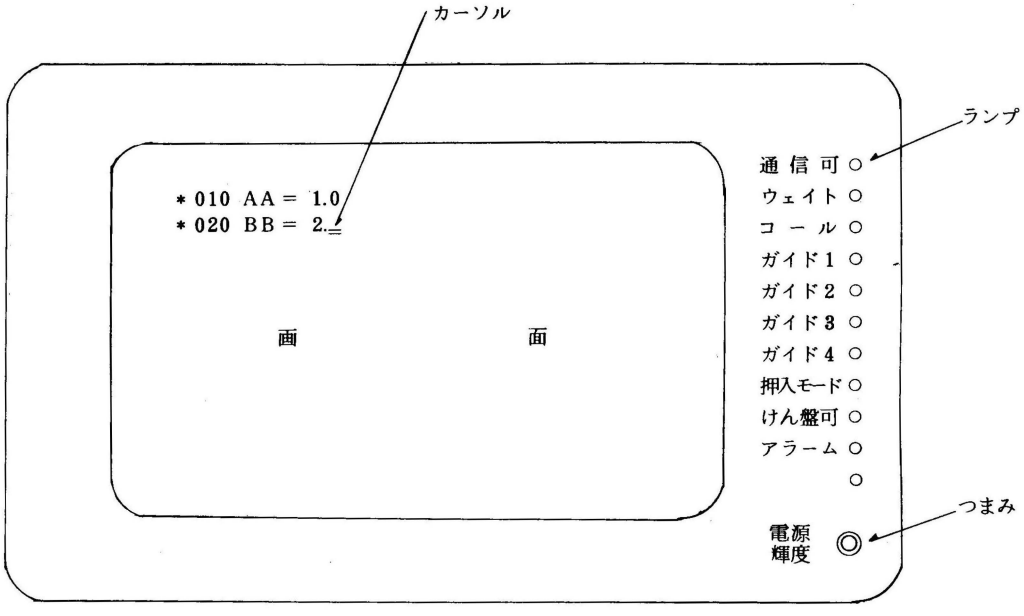


図 5.1 ディスプレイ・ターミナル

注意 ウエイト・ランプが長い間点灯したまま消えない場合は，システム障害の可能性もあるので係員にその旨申し出ること。

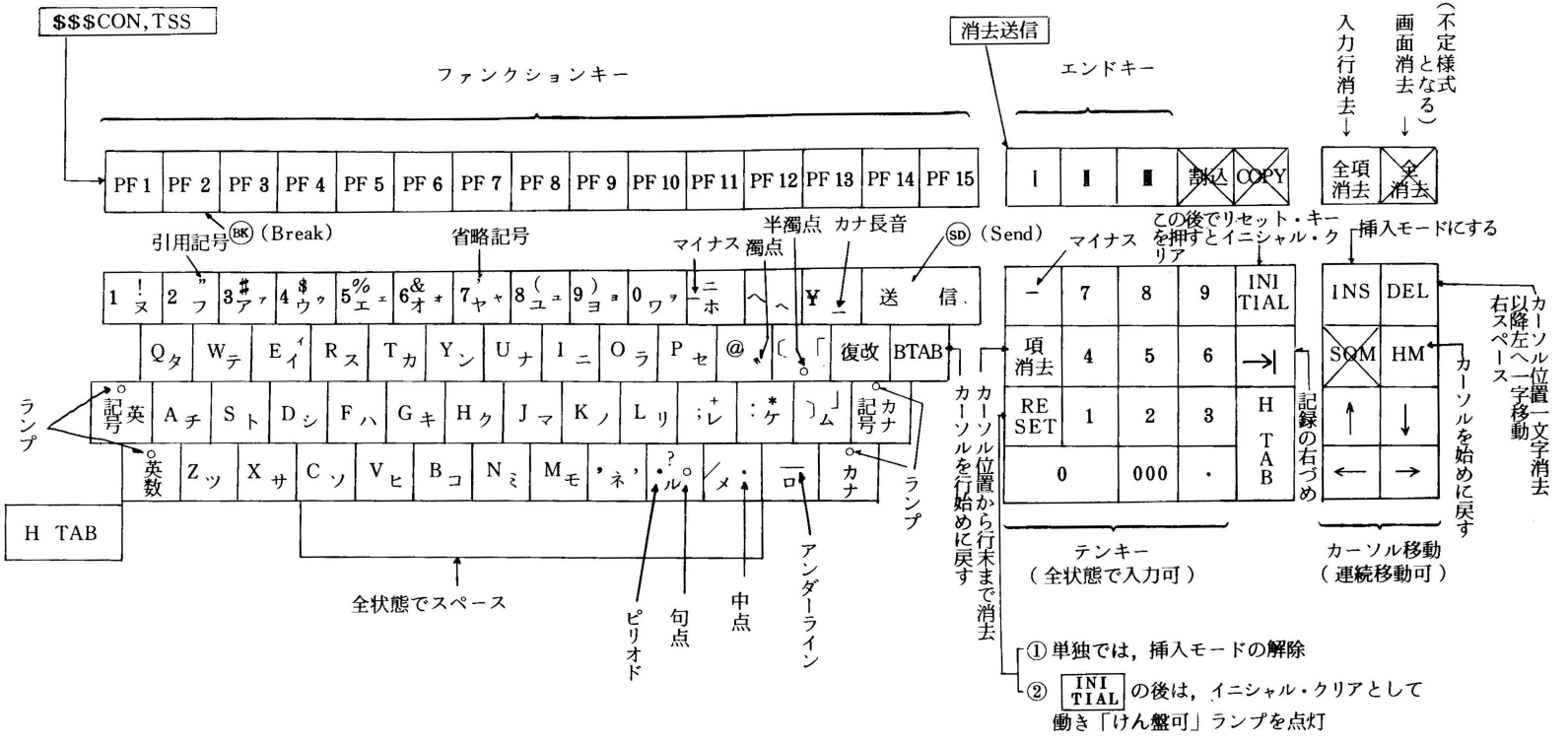


図5.2　キーボード配列図

5.3 グラフィック・ディスプレイ（4006）操作法

(1) 電源投入

① モデム（変復調装置）の電源を入れる。

② ディスプレイ装置の電源を入れる。

(2) 管面消去

入力開始前に PAGE キーを押下して，管面を消去する。

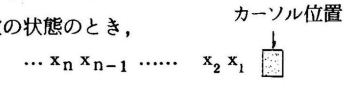
(3) システムへの入力操作

① データの送信

1行分の入力データを送信したい場合には，入力データの後に RETURN を押す。

② 文字列の修正

次の状態のとき，



を y_n y_{n-1} …… y_2 y_1 で置き換えたい場合には，SHIFT を押したまま

@ …… @ を打けん後 y_n y_n …… y_2 y_1

とすればよい。

(4) 画面の再生

このグラフィック・ディスプレイは蓄積管型なので，入出力後約 90 秒経過すると表示が消えてゆく。画面を再生するためには SHIFT キーを押せばよい。

(5) 処理の中断

出力表示や実行などシステムに対する処理の中断は BREAK キーによる。

バッチ・システム利用法

1. センター・バッチ・システム利用例 …… 70
   1.1 ファイルを使用しないジョブの例 …… 70
   1.2 ファイルを使用するジョブの例 …… 71
   1.3 出力リストの説明 …… 80
2. センター・バッチ・システム用制御文 …… 81
   2.1 ジョブ制御文一覧 …… 81
   2.2 ジョブ制御文の一般規則 …… 82
   2.3 各ジョブ制御文の説明 …… 82

ジョブ制御文一覧

機能	制御文	記述箇所	機能	制御文	記述箇所
基本ジョブ制御文	SNUMB	82	システムコール文	FORTRAN	85
	JOB	82		ALGOL	86
	QFORT	82		COBOL,CBL74	86
	QEXEC	83		PL/I	87
	SELECT	83		GMAP	88
	ETC	83		PROGRAM	89
	JIS	83		FILSYS	89
	ENX	84	ファイル定義文	DATA	89
	ENDJOB	84		ENDCOPY	89
	***EOF	84		FILE	89
ローダ制御文	OPTION	84		REMOTE	90
	DUMP	85		SYSOUT	90
	USE	85		PRMFL	90
	LIBRARY	85			

1 センター・バッチ・システム利用法

1.1 ファイルを使用しないジョブの例

使用例	使用言語	ソース・リスト	オブジェクト・リスト	実行打切時間指定	LP 打切行数	メモリ・マップ	クロス・リファレンス・リスト	数値計算ライブラリ使用
①	FORTRAN	○		(30分)	1000			
②	〃	○	○	(30分)	1000	○	○	
③	〃			45分	2000			
④	〃	○						○

使 用 例 ①

```
欄   1          8          16
     $          SNUMB      nnnnn                    ……ジョブ名を定義する
                           └ジョブ名(5桁以内の英数字)
     $          JOB        cddddddddmmm,アカウント番号,パスワード(注)   ……利用者の資格を識別する
                           └登録番号(user id)  └支払責任者登録番号
     $          QFORT                               ……FORTRANコンパイラの呼出し
                ┌──────────────────┐
                │ FORTRAN ソース・プログラム │
                └──────────────────┘
     $          QEXEC                               ……プログラムを実行する
                ┌──────────┐
                │ デ ー タ │
                └──────────┘
     $          ENDJOB                              ……ジョブの終り
〔***EOF〕  センター・バッチのときのみ必要              ……カードデックの終り
```

注) 登録番号，支払責任者番号，パスワードは利用申請したものを使用する。

使 用 例 ②

```
欄   1          8          16
     $          SNUMB      nnnnn
     $          JOB        cddddddddmmm, アカウント番号, パスワード
     $          QFORT      LSTOU, MAP, XREF
                           ↓オブジェクト ↓メモリマップ ↓クロスリファレンス
                ┌──────────────────┐
                │ FORTRAN ソース・プログラム │
                └──────────────────┘
     $          QEXEC
                ┌──────────┐
                │ デ ー タ │
                └──────────┘
     $          ENDJOB
```

使 用 例 ③

欄	1	8	16
	$	SNUMB	nnnnn
	$	JOB	cddddddddmmm, アカウント番号, パスワード, LINE＝2000, TIME＝75 注)
	$	QFORT	NLSTIN

└ソース・リストを出力しない

```
FORTRAN ソース・プログラム
```

	$	QEXEC	

```
デ ー タ
```

	$	ENDJOB	

注) 単位は 1／100 時間

使 用 例 ④

欄	1	8	16
	$	SNUMB	nnnnn
	$	JOB	cddddddddmmm, アカウント番号, パスワード
	$	LIBRARY	L1
	$	QFORT	

```
FORTRAN ソース・プログラム
```

	$	QEXEC	
	$	PRMFL	L1,R,R,NAPLIB／MATHLIB

数値計算ライブラリ・ファイル名

```
デ ー タ
```

	$	ENDJOB	

1.2 ファイルを使用するジョブの例

1.2.1 ソース・プログラム／データの登録，更新およびリスト

サービス・プログラム＊Cエディタを用いてソース・プログラム／データをファイルに登録したり更新したりすることができる。

(1) ソース・プログラム／データの登録

欄	1	8	16
	$	SNUMB	……
	$	JOB	……
	$	PROGRAM	SCED …… ＊Cエディタを呼出し実行する
	$	PRMFL	OT,W,S,修飾ファイル名 ……登録するための出力ファイルを割当てる

└(→フ利 1.4)

欄	1	8	16	
	$	DATA	IN,,COPY	……登録すべき入力ファイルを作成し，$ ENDJOB 文までのソース・プログラム／データをファイルに書込む
			登録するソース・プログラム／データ	
	$	ENDCOPY		
	$	ENDJOB		

注意 1　ソース・プログラム／データを登録する出力ファイルはFILSYSアクティビティ（TSSの場合，ACCESSサブシステム）により前もって作成しておく必要がある（→フ利 5 ）。

2　登録後，出力ファイルのイメージにオルタ番号（TSSにおけるライン番号に対応）が付与されたリストを出力する。

(2) ソース・プログラム／データの更新

欄	1	8	16	
	$	SNUMB	……	
	$	JOB	……	
	$	PROGRAM	SCED	……＊Cエディタを呼出し実行する
	$	PRMFL	IN,R,S,修飾ファイル名 1	……更新すべきソース・プログラム／データの登録されているファイルを＊Cエディタの入力ファイルに割当てる
	$	PRMFL	OT,W,S,修飾ファイル名 2	……更新後の出力ファイルを割当てる
	$	DATA	A＊,,COPY	……更新情報を$DATA文と$ENDJOB文の間に置く
更新情報	$	CHANGE	3	……オルタ番号3の前に，次に続く1枚のカードを挿入する
		READ(1, 10)……		
	$	CHANGE	20, 25	……オルタ番号20から25までを削除する
	$	CHANGE	30, 30	……オルタ番号30を削除し，次に続く2枚のカードと置き換える。
		WRITE(6, 100)……		
		100 FORMAT(…		
	$	ENDCOPY		
	$	ENDJOB		

注意 1　更新後の出力ファイルは前もって作成しておく必要がある。

2　更新情報リストとリナンバリングされたオルタ番号リストがプリントされる。

備考　更新情報は$ CHANGE文を用いて作成することができるが，次にその使用法を簡単に述べる。

(a) 挿　　入

欄	1	8	16	
	$	CHANGE	m	……オルタ番号mの直前に，この文に続くカードを挿入する
		挿入すべきカード（1枚以上）		

(b) 削　　除

欄 1	8	16	
$	CHANGE	m, n	……オルタ番号mからnまでを削除する

(c) 置　　換

欄 1	8	16	
$	CHANGE	m,n	……オルタ番号mからnまでを削除し，この文に続くカードと置き換える。

置換すべきカード（1枚以上）

(3) 登録しているソース・プログラム／データのリスティング

登録しているソース・プログラム／データのファイル

欄 1	8	16	
$	SNUMB	……	
$	JOB	……	
$	PROGRAM	SCED	……＊Cエディタを呼出し実行する
$	PRMFL	IN,R,S, 修飾ファイル名	……リスティングすべきファイルを入力ファイルに割当てる。
$	FILE	OT,NULL	……出力ファイルとして，NULLファイル（実際には存在せず単なる手続きの上だけで存在するダミーファイル）を割当てる。
$	ENDJOB		

注意1　出力ファイルとして，NULLファイルの代わりに，$PRMFL 文を用いて利用者のファイルを割当てると，ファイルのコピーを同時に行なうことができる。

(4) 登録されているソース・プログラムのコンパイル実行

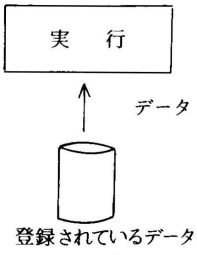
ソース・プログラム
の登録ファイル

欄 1	8	16	
$	SNUMB	……	
$	JOB	……	
$	QFORT		
$	PRMFL	S＊,R,S,修飾ファイル名	……コンパイラ入力ファイルにソース・プログラムの登録されているファイルを割当てる。
$	QEXEC		

デ　ー　タ

$	ENDJOB		

(5) 登録されているデータを実行時に読取る。

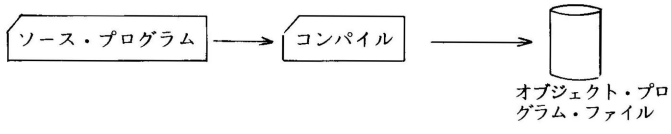



```
欄  1          8           16
    $          SNUMB       ……
    $          JOB         ……
    $          QFORT
                  ソース・プログラム
                    ⋮
                  READ ( 1,  100 ) ……
                    ⋮
                  READ ( 2,  200 ) ……
                    ⋮
    $          QEXEC
    $          PRMFL       01, R , S , 修飾ファイル名   ……READ文の装置番号01をファイル
                                                        コードとしてデータの登録されてい
                                                        るファイルを割当てる。
    $          DATA        02
                  デ　ー　タ                           ……ファイル・コード02のテンポ
                                                        ラリ・ファイルに書込まれる
                                                        データ
    $          ENDJOB
```

1.2.2 オブジェクト・プログラムの登録，更新

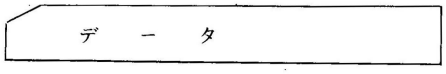
コンパイラの出力するオブジェクト・プログラムをバイナリ・デックの形式でファイルに登録したり，更新したりする方法について述べる。

(1) オブジェクト・プログラムの登録




```
欄  1          8           16
    $          SNUMB       ……
    $          JOB         ……
    $          OPTION      FORTRAN
    $          FORTRAN     DECK                     ……オブジェクト・プログラムをバイナ
                                                        リ・デック形式で出力する。
    $          PRMFL       C*,W, S, 修飾ファイル名   ……バイナリ・デック・ファイルとして
                                                        オブジェクト・プログラムを登録す
                                                        るファイルを割当てる。
                  FORTRAN ソース・プログラム           …オブジェクト・プログラムとして登
                                                        録されるソース・プログラム
```

```
欄  1          8          16
    $          QEXEC
```

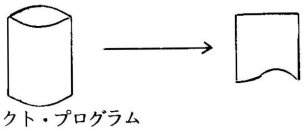



```
    $          ENDJOB
```

注意 1　各オブジェクト・プログラムは，コンパイルされた順序でファイルに登録されている。

2　オブジェクト・プログラムを登録するファイルは，前もって作成しておく必要がある。

(2) 登録されているオブジェクト・プログラムのリスティング




```
欄  1          8          16
    $          SNUMB      ……
    $          JOB        ……
    $          PROGRAM    SCED          ……*Cエディタを呼出し実行する
    $          PRMFL      IN,R,S,修飾ファイル名   ……オブジェクト・プログラムの登録
                                                  されているファイルを *Cエディ
                                                  タの入力ファイルに割当てる。
    $          FILE       OT,NULL       ……NULLファイルを*Cエディタの
                                                  出力ファイルに割当てる。
    $          ENDJOB
```

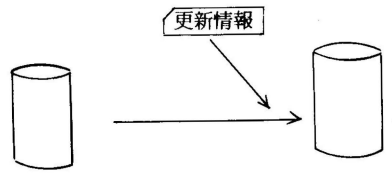
リスト例)

```
 1        ¥  OBJECT   **MAIN**  ┐
 2           (BINARY DECK)      │ 一つのサブプログラム(一つのメイン・ルーチンあるいは一
 9           (END)              │ つのサブルーチン)のオブジェクト・プログラム
10        ¥  DKEND   CONTINUE   ┘
11        ¥  OBJECT             ┐
12           (BINARY DECK)      │ 一つのサブプログラムのオブジェクト・プログラム
16           (END)              │
17        ¥  DKEND              ┘
18  ******   END OF FILE ON OUTPUT(OT).  ******
```

└→ オルタ番号

注意　メインルーチンまたはサブルーチンの先頭が注釈文であれば，そのカラム 8 ～ 50 にパンチされている内容が$ OBJECT文のカラム16 ～ 58 のコメント・フィールドに移される。
（上の例では「** MAIN **」）

(3) 登録されているオブジェクトプログラムの更新



欄	1	8	16	
	$	SNUMB	……	
	$	JOB	……	
	$	PROGRAM	SCED	…… *Cエディタを呼び出し実行する
	$	PRMFL	IN,R,S,修飾ファイル名1	…… *Cエディタの入力ファイルに更新すべきオブジェクト・プログラムのファイルを割当てる。
	$	PRMFUL	OT,W,S,修飾ファイル名2	…… *Cエディタの出力ファイルに更新後のオブジェクト・プログラムのファイルを割当てる。
	$	DATA	A*,,COPY	……$ DATA文と$ ENDCOPY 文の間に更新情報を置く。
更新情報	$	CHANGE	1, 10	……オルタ番号1～10までの オブジェクト・プログラムを削除する。
更新情報	$	CHANGE	18	……オルタ番号18の次に，修飾ファイル名3で示されるファイルにバイナリ形式で登録されているオブジェクト・プログラムを追加する。
更新情報	$	SELECTD	修飾ファイル名3	
	$	ENDCOPY		
	$	ENDJOB		

注意　オルタ番号リストがリナンバリングされて出力される。

備考　オブジェクト・プログラムの更新情報は，$ CHANGE文と$ SELECTD文を用いて作成するが次にその機能を述べておく。（mは$ OBJECT文，nは$ DKEND文のオルタ番号を指す。）

(a) 挿　　入

欄	1	8	16	
	$	CHANGE	m	……オルタ番号mで始まるオブジェクト・プログラムの直前に，修飾ファイル名が示すファイルに登録されている一つ以上のオブジェクト・プログラムを挿入する。
	$	SELECTD	修飾ファイル名	

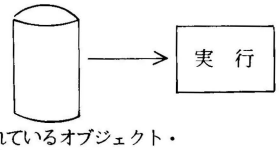
(b) 削　　除

欄	1	8	16	
	$	CHANGE	m,n	……オルタ番号mからnまでの一つあるいは複数個のオブジェクト・プログラムを削除する。

(c) 置　　換

欄	1	8	16	
	$	CHANGE	m,n	……オルタ番号mからnまでの一つあるいは複数個のオブジェクト・プログラムを削除し，修飾ファイル名が示すファイルに登録されている一つ以上のオブジェクト・プログラムと置き換える。
	$	SELECTD	修飾ファイル名	

(4) 登録されているオブジェクト・プログラムの実行



登録されているオブジェクト・プログラム・ファイル

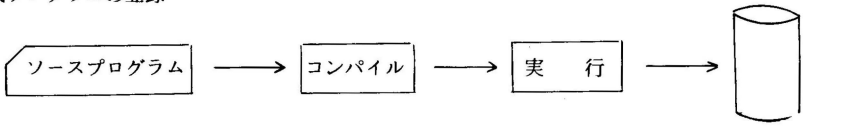
欄	1	8	16	
	$	SNUMB	……	
	$	JOB	……	
	$	SELECT	修飾ファイル名	……修飾ファイル名で示すファイルに登録されているオブジェクト・プログラムを呼出す。

```
欄  1          8
    $          QEXEC
                  ┌──────────────────────┐
                  │  デ　ー　タ          │
                  └──────────────────────┘
    $          ENDJOB
```

注意　$ SELECT文が指定するファイルには，メイン・ルーチンを含む オブジェクト・プログラムが登録されている必要がある。

1.2.3 実行形式プログラムの登録と実行

(1) 実行形式プログラムの登録



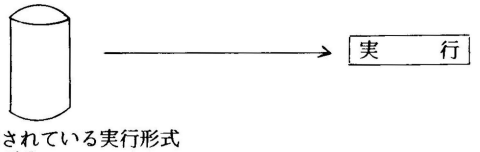

```
欄  1          8          16
    $          SNUMB      ……
    $          JOB        ……
    $          OPTION     FORTRAN, SAVE／プログラム名…プログラムが実行形式プログラム・ファイルに指定したプログラム名で保存される。
    $          FORTRAN
                  ┌──────────────────────┐
                  │ FORTRAN ソース・プログラム │
                  └──────────────────────┘
    $          QEXEC
    $          PRMFL      H*, W, S, 修飾ファイル名   ……登録すべきファイルを実行形式プログラム・ファイルに割当てる。
                  ┌──────────────────────┐
                  │  デ　ー　タ          │
                  └──────────────────────┘
    $          ENDJOB
```

注意　既にプログラムが登録されているファイルに更にプログラムを追加するには，

$ OPTION　FORTRAN, SAVOLD／プログラム名

と指定する（ただし，プログラム名は最初の5文字で一意に識別できる名前である）。

(2) 登録されている実行形式プログラムの実行




```
欄  1          8          16
    $          SNUMB      ……
    $          JOB        ……
    $          PROGRAM    RLHS, NAME／プログラム名   ……プログラム名で指定されたプログラムを登録されている実行形式プログラム・ファイルから呼び出す。
    $          PRMFL      H*, R, R, 修飾ファイル名   ……修飾ファイル名で指定された登録ファイルをH*ファイルに割当てる。
```

```
欄  1          8          16
                ┌──────────────┐
                │ デ ー タ     │
                └──────────────┘
    $           ENDJOB
```

1.2.4 テンポラリ・ファイルを作業ファイルとして利用する

```
欄  1          8          16
    $           SNUMB      ……
    $           JOB        ……
    $           QFORT
                ┌──────────────────┐
                │ FORTRAN ソース・プログラム
                │   ⋮
                │ WRITE (1)  ……
                │   ⋮
                │ WRITE (2)  ……
                │   ⋮
                │ ENDFILE 1
                │ REWIND 1
                │ READ (1, END=999) ……
                │   ⋮
                └──────────────────┘
    $           QEXEC
                ┌──────────────┐
                │ デ ー タ     │
                └──────────────┘
    $           FILE       01
    $           FILE       02
    $           ENDJOB
```

1.2.5 サブルーチン・ライブラリの作成と利用

センター利用者が作成するサブルーチン・ライブラリには，シーケンシャル・ライブラリとランダム・ライブラリとがあるが，ここではこれらのライブラリの作成と利用について述べる。

1.2.5.1 シーケンシャル・ライブラリの作成と利用

バイナリ・デック形式のオブジェクト・プログラムをファイルに登録しておき（→1.2.2）そのファイルをシーケンシャル・ライブラリとして利用することができる。このライブラリを利用する際のデック構成を次に示す。

```
1          8          16
$          SNUMB      ……
$          JOB        ……
$          LIBRARY    S1          ……ローダに対しユーザ・ライブラリの
                                      使用を宣言する（S1はファイルコ
                                      ードを示す）。
$          QFORT
           ┌──────────────────┐
           │ FORTRAN ソース・プログラム
           └──────────────────┘
```

```
欄  1          8          16
    $          QEXEC
    $          PRMFL      S1,R,S,修飾ファイル名        ……$LIBRARY 文で指定したファイル・コードに，シーケンシャル・ライブラリとして利用するオブジェクト・プログラムの登録されているファイルを割当てる。
                          ┌──────────┐
                          │ デ ー タ │
                          └──────────┘
    $          ENDJOB
```

注意 1　シーケンシャル・ライブラリは，シーケンシャルなサーチが行なわれるので，一つのライブラリ上で後にあるサブルーチンから前にあるサブルーチンをコールすることはできない。また複数のライブラリを使用するとき，後のライブラリのサブルーチンから前のライブラリのサブルーチンをコールすることもできない。

2　登録されているオブジェクト・プログラムには，メインルーチンを含んではならない。

1.2.5.2　ランダム・ライブラリの作成と利用

(1)　ランダム・ライブラリの作成

(a)　オブジェクト・プログラムが既にファイルに登録されている場合の作成方法。

```
欄  1          8          16
    $          SNUMB      ……
    $          JOB        ……
    $          PROGRAM    RANLIB                    ……ランダム・ライブラリ・ジェネレータ（RANLIB）を呼出し実行する。
    $          PRMFL      R*,R,S,修飾ファイル名1    ……RANLIB の入力ファイルとして，ファイル・コードR*にオブジェクトプログラムの登録されているファイルを割当てる。
    $          PRMFL      A4,W,R,修飾ファイル名2    ……RANLIB の出力ファイルとして，ファイル・コードA4にランダム・ライブラリを登録するファイルを割当てる。
    $          ENDJOB
```

(b)　オブジェクト・プログラムがファイルに登録されていない場合の作成方法

```
欄  1          8          16
    $          SNUMB      ……
    $          JOB        ……
    $          FORTRAN    DECK      ┐
    $          FILE       C*,A1S,30L┘  ……オブジェクト・プログラムをバイナリ・デック形式でテンポラリ・ファイルに出力する。
                          ┌──────────────────────┐
                          │ FORTRAN ソース・プログラム │
                          └──────────────────────┘
    $          PROGRAM    RANLIB                    ……ランダム・ライブラリ・ジェネレータ（RANLIB）を呼出し実行する。
    $          FILE       R*,A1                     ……前のアクティビティで得られたバイナリ・デック形式のオブジェクト・プログラムのファイルをRANLIBの入力ファイルとする。
    $          PRMFL      A4,W,R,修飾ファイル名      ……RANLIB の出力ファイルとして，ファイル・コードA4にランダム・ライブラリを登録するファイルを割当てる。
    $          ENDJOB
```

注意 1　ランダム・ライブラリ・ファイルを登録する際には，アクセス・モードをRAND（ランダム）に指定しておく必要がある。

2　$ PRRMFL文によるランダム・ライブラリ・ファイルの割当ての際には，第3パラメータ（アクセスモード）の指定欄ではR（ランダム）と指定すること。

(2)　ランダム・ライブラリの利用

```
欄  1          8          16
    $          SNUMB      ……
    $          JOB        ……
    $          LIBRARY    L1          ……ローダに対しユーザ・ライブラリの使用を宣言する（L1はファイル・コード）
    $          QFORT
                    ┌──────────────────────┐
                    │ FORTRAN ソース・プログラム │
                    └──────────────────────┘
    $          QEXEC
    $          PRMFL      L1,R,R, 修飾ファイル名   ……$ LIBRARY 文で指定したファイル・コードに，ランダム・ライブラリの登録されているファイルを割当てる。
                    ┌──────────────────────┐
                    │ デ　ー　タ              │
                    └──────────────────────┘
    $          ENDJOB
```

注意　$ PRMFL 文の第3パラメータ（アクセス・モード）の指定欄には，必ずR（ランダム）と指定すること。

1.3　出力リストの説明

出力リスト・フォーマット

```
××××××××××××××××××××××××
B A T C H                          ……$ SNUMB で指定したジョブ名を印字する。
××××××××××××××××××××××××

××××××××××××××××××××××××
TCYAMADA.                          ……利用者識別名（USER ID）の最初の9文字までを印字
××××××××××××××××××××××××

¥¥   BATCH ENTERED ACOS-6 AT 13.695 FROM CD RDR 0-25-01
0001 ¥      SNUMB  BATCH
0002 ¥*CPROC J/A,, 003, TCYAMADA.050 …
0003# ¥      IDENT  003, TCYAMADA.050
                ⋮      ⋮                    …ジョブ制御文のリスト
0014 ¥      ENDJOB
 TOTAL CARD COUNT THIS JOB= 000045
   ××××××××××××××××××××××××××××××××××××
   ×                                      ×
   ×    CD RDR INPUT  ¥ 4.5        USED  0 %  ×
   ×                                      ×
   ××××××××××××××××××××××××××××××××××××
* ACTY-01  ¥CARD # 0009 #  FORTY  01/24/78  SW= 210210000000
  *NORMAL TERMINATION
```

```
××××××××××××××××××××××××××××××××××××
×  INPUT   ¥ 4.50   BASIC COST ¥ 50.00          ×
×  CPU     ¥ 2.40                               ×
×    ⋮                  ⋮                      ×
××××××××××××××××××(CPU TIME   0.51 SEC)××
```

…ジョブ・アクティビティ1（たとえばソース・プログラムのコンパイル）のアカウント情報

```
* ACTY-02  ¥CARD ♯0012 ♯   GELOAD   01/24/78   SW= 000000000000
  *NORMAL   TERMINATION
××××××××××××××××××××××××××××××××××××
×  CPU     ¥ 8.72   SYSOUT  ¥ 6.10  ……       ×
×  PUNCH   ¥ 0.00     ⋮                        ×
×  DAC     ¥ 0.00     ⋮                        ×
××××××××××××××××××(CPU TIME   2.18 SEC)××
```

…ジョブ・アクティビティ2（たとえばソース・プログラムの実行）のアカウント情報

```
BATCH 01    01-24-78    13.953    ……
ACOS-6 R3.2 FORTRAN (03-31-77)   OPTIONS :   ……
                ソース・プログラム・リスト
                      ⋮
BATCH 02    01-24-78    13.960
  プログラムの計算結果
        ⋮
```

…ジョブ・アクティビティ1の出力結果

…ジョブ・アクティビティ2の出力結果

```
××××××××××××××××××
——BATCH——
××××××××××××××××××
```

……一つのジョブに対する出力リストの最後の頁にSNUMB名を印字する。

```
××××××××××××××××××
CYAMADA.0
××××××××××××××××××
```

……USER IDのはじめの2文字めから9文字を印字する（リモートバッチのときは印字しない）。

```
××××××××××××××××××××××××
×  OUTPUT TO PRINTER PAGE COUNT 8  ×
×  OUTPUT   COST        ¥ 24.00    ×
××××××××××××××××××××××××
```

……出力に対するアカウント情報

2　センター・バッチ・システム用制御文

2.1　ジョブ制御文一覧（→マニ29）

機能	制御文	記述箇所	機能	制御文	記述箇所
基本ジョブ制御文	SNUMB	82	システムコール文	FORTRAN	85
	JOB	82		ALGOL	86
	QFORT	82		COBOL,CBL74	86
	QEXEC	83		PL/I	87
	SELECT	83		GMAP	88
	ETC	83		PROGRAM	89
	JIS	83		FILSYS	89
	ENX	84	ファイル定義文	DATA	89
	ENDJOB	84		ENDCOPY	89
	***EOF	84		FILE	89
ローダ制御文	OPTION	84		REMOTE	90
	DUMP	85		SYSOUT	90
	USE	85		PRMFL	90
	LIBRARY	85			

2.2 ジョブ制御文の一般規則

(1) ジュブ制御文は，カラム1から$で始まる。

(2) $NUMB,ENDJOB,FILE……など，制御文のコントロール・フィールドはカラム8から始まる。

(3) コントロール・フィールドがカラム8からである場合，カラム15はブランクでなければならない。

(4) オペランド・フィールドはカラム16から始まり，注がついている場合を除いてカラム72を超えてはならない。

(5) オペランドは，コンマ "," で区切らなければならない。

(6) フィールド定義の終わりはブランクで表わす。したがって，これ以外にブランクを使うことはできない。

(7) 12文字をこえるオペランドは無視される。

(8) 特に指定されていない場合は，オペランド・フィールドにオプションを指定する順序は任意である。

以上のような規則に加えて，各ジョブ制御言語の説明にはそのジョブ制御文特有の制限がある（以下に述べる各ジョブ制御文のオペランドの欄で下線があるオプションは標準値を示す）。

2.3 各ジョブ制御文の説明

2.3.1 基本ジョブ制御文

システム入力時に処理され，基本ジョブ制御文を構成する。

2.3.1.1 $ SNUMB

欄	1	8	16
	$	SNUMB	ジョブ名

(1) 機　　能　ジョブを識別する。

(2) オペランド　ジョブ名は5桁以内の英数字である。

2.3.1.2 $ JOB （マークFORTRAN使用のときは$MJOBとする）

欄	1	8	16
	$	JOB	登録番号　，アカウント番号，パスワード

〔,LINE=11111〕〔,TIME=mmm〕〔,CORE=kkkK〕〔,{REPT / NRPT}〕

(1) 機　　能　利用資格のチェックおよび利用資源の標準値を変更させる。

(2) オペランド
① 登録番号は利用申請で認可されたものを書く（例　Cddddddddmmm）。
② アカウント番号は支払責任者登録番号を書く。
③ LINEは，出力ライン行数の指定
④ TIMEは，CPU timeの指定（単位 1/100時間）
⑤ COREは，必要なメモリサイズをワード数（単位　KW）で指定する。
（LINE,TIME,COREは一つのジョブに対して指定するものである）
⑥ REPTは，アクティビティごとの統計情報を出力リストに印字する（NRPT）。

備考 1) この文は，$IDENT，$USERID，$LIMITSの制御文をカタログ化したものである。
2) この文は，$SNUMBの次の制御文でなければならない。
3) マークFORTRAN使用のときは$ JOBの代わりに$MJOBとする。
（但し$MJOB文では，LINE,TIME,CORE,REPTなどのオプションは指定できない）
4) LINE,TIME,COREそれぞれ指定がない場合の標準値を次に示す。
LINE = 1000行
TIME = 0.5時間
CORE = 32 KW

2.3.1.3 $ QFORT

欄	1	8	16
	$	QFORT	オプション

(1) 機　　能　FORTRANコンパイラを呼び出す

(2) オペランド　オプションは$FORTRAN文のオプションである。

備考　この文を使うと，欄 1 8 16 は不要である。

$	OPTION	FORTRAN

2.3.1.4 $ QEXEC

欄 1	8	16
$	QEXEC	〔LINE＝llll〕〔,TIME＝mmm〕〔,CORE＝kkkK〕

(1) 機　　能　オブジェクト・プログラムのローディングと実行を要求し，アクティビティの標準値を変更したりする。

(2) オペランド　① LINE ： 出力ライン行数の指定

② TIME ： CPU time の指定（単位 1／100 時間）

③ CORE ： プログラムの実行に必要な主記憶の最大値を示し，10 進数で必要な語数（単位 KW）を指定する。

備考　LINE，TIME，COREのアクティビティごとの標準値は次表のとおりである。

アクティビティ	LINE （行）	TIME（時間）	CORE（キロワード）
FORTRAN	2,000	0.05	28
CBL 74	2,000	0.15	50
ALGOL	2,000	0.08	30
EXECUTE	1,000	0.50	32

2.3.1.5 $ SELECT , $ SELECTD

欄 1	8	16
$	SELECT	修飾ファイル名

欄 1	8	16
$	SELECTD	修飾ファイル名

(1) 機　　能　$ SELECT，$ SELECTD文は，ファイル・システムを呼び出して システム・インプットの入力となるパーマネント・ファイル・カタログを検索する。
（ただし，$ SELECTD文は，$ DATA …,, COPY と $ ENDCOPY 文の間以外で使用すると無視される）

(2) オペランド　修飾ファイル名 ： システム・インプットが読出すための修飾ファイル名

2.3.1.6 $ ETC

欄 1	8	16
$	ETC	前の制御文の続き

(1) 機　　能　前の制御文のオペランド・フィールドを継続して指定するために使用される。

2.3.1.7 $ JIS

欄 1	8	16
$	JIS	

(1) 機　　能　この制御文に続くジョブ・デック（ソース・プログラムおよびデータ）を JISコードとして処理する。

(2) オペランド　使用しない。

備考　1) JISデータは$ JIS文で始まり$ ENX文（または，$ SNUMB文，*** EOF 文）の間に挿入する。

2) $ JIS 文の指定がないときは，BCDレコードとして処理される。

2.3.1.8 $ ENX

欄	1	8	16
	$	ENX	

(1) 機　　能　　$ＪＩＳ文と対に使用され，ＪＩＳデータの終りを示す。

(2) オペランド　　使用しない。

2.3.1.9 $ ENDJOB

欄	1	8	16
	$	ENDIOB	

(1) 機　　能　　ジョブの終りを示す。

(2) オペランド　　使用しない。

2.3.1.10 *** EOF

欄	1	8	16
	***EOF		

(1) 機　　能　　システム入力カードの終りを告げる。

(2) オペランド　　使用しない。

備考　この文は，カード読取装置を使って入力する場合にのみ必要である。
（センター・バッチのみ必要である）

2.3.2 ローダ制御文

ローダ制御文は，ローダへの指示のために使用するものである。

2.3.2.1 $ OPTION

欄	1	8	16	
	$	OPTION	オプション	

(1) 機　　能　　ローディング時に，標準のローダ・オプションを変更する。

(2) オペランド

MAP　：メモリ・マップを作成する（NOMAP）。

CONGO　：ローディング中に致命的でないエラーを検出しても，無視してジョブを実行する。

GO　：ローディング中に何もエラーを検出しなかったときにだけ，ジョブを実行する。

NOGO　：ローディングは終了するが，ロードしたプログラムは実行しない。

SYMREF　：ロードされた各々のサブ・システムに使われているSYMREFのリストをとる（NOSREF…リストをとらない）。

ALGOL　：ALGOLプログラムをローディングする際に必要なすべてのオプションをセットする。

FORTRAN：FORTRAN入出力ルーチンを使用するときに必要であるFCBオプションをセットする。

CBL 74　：COBOL74コンパイラによって作成されたプログラムをロードするときに必要となるオプションをセットする。

PL／I　：PL／I コンパイラによって作成されたプログラムをロードするときに必要となるオプションをセットする。

SAVE／name：6字以内の名前でH*ファイルにプログラムを保存する。セーブするプログラムにFCBが存在すれば，FCBには指定した名前の前にアスタリスク（*）を1つ挿入した名前を付け，H*に保存する。

SAVOLD／name：既にあるH*ファイルに更にプログラムを保存する。

備考　1）各オプションは，コンマで区切る。

2) ALGOL, FORTRAN, CBL 74, PL/I のオブジェクト・デックの ローディングや実行には，それぞれALGOL, FORTRAN, CBL 74, PL/I のオプションを指定しなければならない。

2.3.2.2 $ DUMP

欄	1	8	16
	$	DUMP	プログラム名

(1) 機　　能　この制御文の後続データ（デバッグ文）がDEBUGサブルーチンで処理されることをローダに知らせる。

(2) オペランド　プログラム名には主プログラムのときは……（6つのドット）
副プログラムのときは，副プログラム名
を書く。

備考　$ FORTRAN文にDEBUGオプションを指定すること。

2.3.2.3 $ USE

欄	1	8	16
	$	USE	SYMDEFまたは SYMDEF名／サイズ／

(1) 機　　能　オペランド・フィールドで指定した1つ以上の名前を，SYMREFまたは名前付き共通領域として，ロード・テーブルに登録する。

(2) オペランド　SYMDEF名 ; ロード・テーブルにSYMREFとして登録する。
SYMDEF名／サイズ／ ; 名前付き共通領域として登録する。

(3) 使 用 例

欄	1	8	16
	$	USE	LOC／500／, PROG 1, PROG 2

LOCという500ワードの名前付き共通領域を定義する。

PROG 1, PROG 2をSYMREFとしてロードテーブルに登録する。

2.3.2.4 $ LIBRARY

欄	1	8	16
	$	LIBRARY	ファイル・コード〔,ファイル・コード 2〕…〔,ファイル・コード 10〕

(1) 機　　能　センター・ライブラリあるいは利用者が定義したサブルーチン・ライブラリがあることを示し，システム・サブルーチン・ライブラリをサーチする前に，定義したサブルーチン・ライブラリをファイル・コードの並びの順番に従ってサーチする。

(2) オペランド　ローダがサーチする順番に従って，利用者ライブラリのファイル・コード（英数字2桁からなる）を指定する。

備考　1) 指定したファイル・コードは，ファイル定義文で定義されていなければならない。
2) この制御文は，必らず$ EXECUTE文より前に置く。

2.3.3 システムコール文

種々の言語処理プログラムを呼出したり，これらのプログラムの使用にあたって，使用するオプションを指定するのに使う。

2.3.3.1 $ FORTRAN

欄	1	8	16
	$	FORTRAN	〔{NLSTIN／LSTIN}〕〔{LSTOU／NLSTOU}〕〔{COMDK／NCOMDK}〕〔{DECK／NDECK}〕 〔{DEBUG／NDEBUG}〕〔{XREF／NXREF}〕〔{NFORM／FORM}〕〔{LNO／NLNO}〕 〔{BCD／JIS}〕〔{OPTZ／NOPTZ}〕〔{MAP／NMAP}〕〔,NWARN〕

(1) 機　　能　FORTRANコンパイラを呼出す。

(2) オペランド

LSTIN ：FORTRANコンパイラがソース入力のリストを出力する。（NLSTIN）

LSTOU ：オブジェクト・プログラムのリストを出力する。（NLSTOU）

COMDK ：ソース・プログラムのコンプレスト・デックを出力する。（NCOMDK）

DECK ：オブジェクト・デックを作成する。（NDECK）

DEBUG ：オブジェクト・プログラムにシンボル・テーブル（・SYMT・）を作成する。デバッグ機能を使用するときは，必ずこのオプションを指定しなければならない。（NDEBUG）

XREF ：クロス・リファレンス・リストを出力する。またオブジェクト・コード・リスト上に現われるオフセット値をソース・リストの最左端のフィールドに出力する。（NXREF）

FORM ：ソース・プログラムの形式が標準形式であることを指定する。（NFORM）

LNO ：NFORMの場合，ソース・プログラムに行番号があることを示す。（NLNO）

JIS ：オブジェクト・プログラムの文字セットがＪＩＳであることを指定する。（BCD）

OPTZ ：全体的最適化を行う。（NOPTZ）

MAP ：プログラムの文番号，変数，定数のストーレッジ・マップを出力する。（NOMAP）

NWARN ：コンパイル中に検出される警告メッセージを出力しない。

2.3.3.2 $ ALGOL

欄	1	8	16
	$	ALGOL	〔{LSTIN / NLSTIN}〕〔,{LSTOU / NLSTOU}〕〔,{COMDK / NCOMDK}〕〔,{DECK / NDECK}〕〔,{DEBUG / NDEBUG}〕〔,SYMTB〕

(1) 機　　能　ALGOLコンパイラを呼出す。

(2) オペランド

LSTIN ソース入力のリストを作成する。（NLSTIN）

LSTOU オブジェクト・プログラム出力のリストを作成する。（NLSTOU）

COMDK ソース・プログラムのコンプレスト・デックを作成する。（NCOMDK）

DECK オブジェクト・デックをALGOLによる出力として作成する。（NDECK）

DEBUG ALGOLコンパイラの中間ファイルのリストを作成する。（NDEBUG）

SYMTAB シンボル・リファレンス・テーブルを出力する。

2.3.3.3 $ COBOL , $ CBL 74

欄	1	8	16
	$	COBOL	〔{NLSTIN / LSTIN}〕〔,{LSTOU / NLSTOU}〕〔,{CCOMDK / COMDK / NCOMDK}〕〔,{DECK / NDECK}〕〔,{BCD / JIS}〕〔,{COPY / NCOPY}〕〔,ON6〕〔,SYMTAB〕〔,SEGMNT〕〔,REFON〕〔,LIBCPY〕〔,EALERT〕

欄	1	8	16
	$	CBL 74	〔{NLSTIN / LSTIN}〕〔,{LSTOU / NLSTOU}〕〔,{COMDK / NCOMDK}〕〔,{DECK / NDECK}〕〔,COPY〕〔,XREF〕〔,NWARN〕〔,NRESET〕

(1) 機　　能　$ COBOL ……COBOL コンパイラを呼出す。

$ CBL 74 ……COBOL 74 コンパイラを呼出す。

(2) オペランド　LSTIN　：ソース入力のリストを出力する。（NLSTIN）

LSTOU　：オブジェクト・プログラムのリストを出力する。（NLSTOU）

COMDK　：ソース・プログラムのコンプレスト・デックを作成する。（NCOMDK）

JIS　：COBOL JIS 翻訳ルーチンを呼出す。（BCD）

CCOMDK　：ソース・プログラム内でCOPY命令を使用している場合に登録集原文を複写した後のコムデックを作成する。

ON 6　：実行用プログラムのシンボル・リファレンス・テーブルを作成する。

SYMTAB　：シンボル・リファレンス・テーブルの内容を表示する。

SEGMENT：手続き部の節の見出しに記した区分番号が昇順でない場合に区分番号が昇順となるように節を並び換える。

REFON　：ソース・プログラムのクロスリファレンスを作成表示する。

LIBCPY　：ソース・プログラム内でCOPY命令を使用していることをコンパイラに知らせる。コンパイラは登録集原文を複写する。

COPY　：登録集機能を利用しているとき，必ず指定する。

EALERT　：実行時の "ILLEGAL EIS DATA" および "F7" アボートを解析し，わかり易いメッセージを表示後 "CH" でアボートする。

XREF　：ソース・プログラムのクロス・リファレンスを印刷表示する。

NWARN　：警告メッセージを表示しない。

NRESET　：致命的誤りが検出されても後続の実行アクティビティの起動を許す。

2.3.3.4 $ PL 1

欄	1	8	16
	$	PL 1	[{NLSTIN / LSTIN}][,{LSTOU / NLSTOU}][,{COMDK / NCOMDK}][,{DECK / NDECK}][,{ALTNO / NALTNO}] [,{CHECK / NCHECK}][,{LIST / NLIST}][,{MAP / NMAP}][,{OPTZ / NOPTZ}][,{PARSE / NPARSE}] [,{SEVERITYn / SEVERITY1}][,{SNUMBER / NSNUMBER}][,{STAB / NSTAB}][,{SYMT / NSYMT}] [,{XREF / NXREF}]

(1) 機　　能　PL／I コンパイラを呼出す。

(2) オペランド　LISTIN　：ソース・プログラムのリストを出力する。（NLSTIN）

LSTOU　：オブジェクト・プログラムのリストを出力する。（NLSTOU）

COMDK　：ソース・プログラムのコムデックを作成する。（NCOMDK）

DECK　：オブジェクト・デックを作成する。（NDECK）

ALTNO　：オルタ番号の付いたソース・プログラムのリストをコンパイラの出力リストとして出力することを指定する。（NALTNO）

CHECK　：コンパイラが構文解析と意味解析のみを行い，コード生成処理は行わない。（NCHECK）

LIST　：このオプションを指定すると，LSTOU, MAP, SYMT と XREF オプションが指定されたのと同じ意味になる。（NLIST）

MAP　：コンパイルされて作成されたオブジェクト・プログラムに必要な主記憶領域の大きさに関するテーブルを，コンパイラの出力リストとして出力することを指定する。（NMAP）

OPTZ　：コンパイラは非常に効率の良いオブジェクト・プログラムを生成すべく，作り出される機械語命令を最適化する。（NOPTZ）

PARSE　：構文解析の後になされる意味解析をコンパイラが無条件に止めることを指定する。（NPARSE）

SEVERITY: このオプション，SEVERITYnが指定されると，nよりも小さなエラーメッセージは，コンパイラの出力リストとしては出力されない。ここではnはエラーの度合を示し1,2,3または4である。数が増すほどエラーの度合は大きくなる。

SNUMBER：コンパイラは，実行時に発生するエラーに対してなされるエラー・トレース・バックの情報としてソース・プログラム上の文の番号とラインの番号を追加する。（NSNUMBER）

STAB　：コンパイラは，完全な実行時のシンボルテーブルを生成する。そのソースプログラム上で参照される全ての変数，ラベル，入口名はこのテーブルに実行時のデバック処理にとって便利なように配置されている。（NSTAB）

SYMT　：コンパイラの出力リストとしてソース・プログラム上で使用された名前とそれに与えられた属性のテーブルを出力することを指定する。（NSYMT）

XREF　：ソース・プログラム上で使用されている全ての名前と，それらが宣言されている文の番号およびそれらが参照されている文の番号をコンパイラの出力リストとして出力することを指定する。（NXREF）

備考　1）PL／1コンパイラに対してオプション指定を行なえる制御文としては

$ PL 1制御文

＊オプション文

の2種類がある。

2）$ PL 1文は，PL／1コンパイラを起動するためのものである。

＊オプション文は，PL／1コンパイラに固有の制御文であり，PL／1ソース・プログラムの先頭に置く。

2.3.3.5 $ GMAP

欄	1	8	16
	$	GMAP	〔{NLSTOU / LSTOU}〕〔,{COMDK / NCOMDK}〕〔,{DECK / NDECK}〕〔,{NGMAC / GMAC}〕
			〔,ON 5〕〔,NXEC〕〔,SYMTAB〕

(1) 機　　能　アセンブラを呼出す。

(2) オペランド　LSTOU　：オブジェクト・プログラムのリストを出力する。（NLSTOU）

DECK　：オブジェクト・デックを作る。（NDECK）

GMAC　：アセンブル時にGMAPのシステム・マクロが必要であることを指示する。（NGMAC）

ON 5　：オブジェクト・プログラム・リストには出力されない擬似命令も出力する。ただし，NLSTOUオプションが指定してあると意味がなくなる。

NXEC　：アセンブルの途中で致命的なエラーが検出されると，実行アクティビティを削除する。

SYMTAB　：シンボル・リファレンス・テーブルのリストを，NLSTOUオプションの有無にかかわらず作成する。

COMDK　：ソース・プログラムのコンプレスト・デックを作る。（NCOMDK）

2.3.3.6 $ PROGRAM

欄	1	8	16
	$	PROGRAM	プログラム名〔,オプション〕

(1) 機　　能　システム・プログラム・ファイルに登録されているプログラムを呼出す。

(2) オペランド　プログラム名：呼び出すプログラム名を6字以内の英数字で指定する。

オプション　：$ EXECUTE, $ FORTRAN, $ COBOL, $ GMAP文で使用できるオプションを5個まで指定できる。

2.3.3.7 $ FILSYS

欄	1	8
	$	FILSYS

(1) 機　　能　ファイルの作成や更新を行なうFILSYSアクティビティを呼出す。

2.3.4 ファイル定義文

処理プログラムのファイル情報を定義する。

2.3.4.1 $ DATA

欄	1	8	16
	$	DATA	fc〔,{COPY / NCOPY}〕

(1) 機　　能　アクティビティに対する入力としてテンポラリ・ファイルを作成する。

(2) オペランド　fc　：ファイル・コード（2文字からなる英数字）

COPY　：$ SNUMB, $ JOB, $ DATAを除いたすべてのジョブ制御文を，$ ENDCOPY文があるまで，ファイル・コードで指定したファイルにコピーする。

NCOPY　：次のジョブ制御文までのデータをファイルに書出す。

2.3.4.2 $ ENDCOPY

欄	1	8	16
	$	ENDCOPY	〔fc〕

(1) 機　　能　$ DATA文のコピー機能を停止させる。

(2) オペランド　fc　：$ DATA文で指定されたファイル・コードを指定すると，$ DATAから$ ENDCOPY間のデータを指定されたファイルへ書出す。

2.3.4.3 $ FILE

欄	1	8	16
	$	FILE	fc,〔LUD〕〔,{nn / nnL / nnR}〕

または

1	8	16
$	FILE	fc,NULL

(1) 機　　能　磁気ディスク装置上にテンポラリ・ファイルの割当てを行う。

NULLを指定するとダミーの出力ファイルを割当てる（プログラムで出力が不要のときに用いる）。

(2) オペランド　fc　：ファイル・コード（入出力文の装置番号）

LUD　：ロジカル・ユニット・デジグネータを指定する。（2～3文字の英数字，但し第2文字は数字）

nn
nnL
nnR　：ファイルの大きさとファイルのタイプを指定する。（例えば，10Lは順編成ファイルでその大きさは10リンク，5Rであれば直接編成ファイルでその大きさは5リンクである）英字がなければLとみなす。またこのオプションを省略すると1Lとみなされる。

注) LUD；ロジカル・ユニット・デジグネータは，同一ジョブ内の後続アクティビティにファイルを受渡したり前のアクティビティから定義済のファイルを引継いだりするために用いられるファイル識別子である。

例) 欄 1 8 16

```
$      QFORT                          } アクティビティ 1
       ⋮                              }
$      FILE      02,×1S, 10L          }
$      FILSYS                         } アクティビティ 2
$      FILE      1*, ×1R              }
       ⋮                              }
```

上の例はアクティビティで作成したテンポラリ・ファイルをアクティビティ 2 で利用している例であり，識別子の第 3 文字は，アクティビティの終了時か，ファイルの解放時にファイルに必要な処置（ディスポジション）を示すコードであり，上の例で "S" は後続のアクティビティのためのファイルのセーブ，"R" はファイルのリリースを示している。このコードを省略すると "R" とみなされる。

2.3.4.4 $ REMOTE

欄 1 8 16

```
$      REMOTE    fc〔, id〕
```

(1) 機　　能　出力ファイルを端末に割当てる。1 ジョブで最大 3 個のファイルを端末に割当てることができる（→リ利）。

(2) オペランド　fc　：ファイルを識別するための 2 文字の英数字。

id　：論理端末名（指定がない場合には，そのジョブを入力した端末とみなされる）。

2.3.4.5 $ SYSOUT

欄 1 8 16

```
$      SYSOUT    fc〔,{ORG / 論理端末名}〕
```

(1) 機　　能　① 第 2 オペランドに論理端末名が指定されていなければ，第 1 オペランドに指定されたファイル・コードをもつファイルへの出力は，SYSOUT ファイルに割当てられる。

② 第 2 オペランドに論理端末名が指定されていると，出力は指定された端末へとられる。

(2) オペランド　fc　：SYSOUT ファイルへ割当てるファイルのファイル・コード（2 文字の英数字）。

ORG　：指定されたファイル・コードの SYSOUT ファイルの出力は，ジョブが投入された端末またはセンタ（センター・バッチの場合）へ行われる。

論理端末名　：出力は指定された端末へ行われる。

空白　：指定されたファイル・コードをもつ SYSOUT ファイルの出力はセンターに行われる。

2.3.4.6 $ PRMFL

欄 1 8 16

```
$      PRMFL     fc,{R/W},{S/R},修飾ファイル名
```

(1) 機　　能　作成されているパーマネント・ファイルをアクセスする。

(2) オペランド　fc　：ファイル・コード

{R/W} ：ファイルに対する動作を指定する。

R … READパーミッション

W … READ & WRITEパーミッション

{S/R} ：ファイルが順編成ファイル(S)か直接編成ファイル(R)かを指定する。

修飾ファイル名： fc に割当てるファイル・ストリングを指定する。

リモート・バッチ・システム利用法

1. リモート・バッチ・システム利用例 …… 93
2. リモート・バッチ・システム用制御文 …… 93
   2.1. 制御文の一般形式 …… 93
   2.2. 各制御文の説明 …… 94
3. リモート・バッチ・システム使用中のエラー・メッセージ …… 97
4. リモート・バッチ・システム用端末機器の操作法 …… 99
   4.1 N 6240 データ・ステーション操作法 …… 99

リモート・バッチ制御文（コマンド）一覧

制御文形式	機能	記述箇所
$$$CON,RBATCH	リモート・バッチ・システムとの接続	94
$$$LID,論理端末名	論理端末名の宣言	94
$$$RCD	ジョブの入力指示	94
$$$OUT {snumb / C} [,AUTO]	ジョブの出力要求	94
$$$CAN	自動出力モードの取消	94
$$$STS {snumb / 論理端末名 / !ONL / !LSTR}	ジョブの状態表示と出力先変更	95
$$$ABT snumb	ジョブの強制打切り	95
$$$BYE	リモート・バッチ・システムとの切断	96
$$$CNT	中断動作の再開	96
$$$OAB	ジョブ出力のファイル単位の打切り	96
$$$IAB	入力中ジョブの打切り	97

この章では，遠隔地で発生したジョブを発生した場所から，端末装置およびリモート・コンピュータを介してセンターの計算機に一括投入・実行を行なうリモート・バッチ処理方式の利用法について述べる。データ・ステーション操作法については本号「端末操作法２」を参照されたい。

1. リモート・バッチ利用例

リモート・バッチ方式を利用する際の基本的なカード・デック構成としては，センター・バッチ・ジョブ・デックに，リモート・バッチ処理システムを構成しているところのRBE（リモート・バッチ処理管理プログラム）やACOS 6システムに対する制御文をつけ加えるだけでよい。

使用例①　端末からジョブを入力し，その処理結果の出力を待つ。

```
欄  1        8              16
$$$CON,RBATCH                    ……センターとリモート端末をコネクトする。
$$$LID,R6 注1)                   ……論理端末名を宣言する。
$$$RCD                           ……ジョブ・デックを読み込むことを要求する。
$       SNUMB       yyyyy  ┐
          ⋮                 ├  センターバッチジョブデックと同じ構成
$       ENDJOB              ┘
$$$OUT  yyyyy 注2)               ……SNUMB名（ここではyyyyy）のジョブの出力要求
                  ↑
                  └── 同じ名前を書く
```

注1)　論理端末名は医歯薬地区はＲ６，教養部地区はＲ７である。

注2)　SNUMB名を省略するとその論理端末に対する出力可能な全ジョブの出力を行う。

使用例②　端末からジョブを入力して，その出力を待ち合わせているときに別のジョブを入力する。

```
欄  1        8              16
$$$CON,R BATCH
$$$LID,R6
$$$RCD
$       SNUMB       yyyyy
          ⋮
$       ENDJOB
$$$OUTyyyyy
ブレイク操作｛操作卓の「FUNCT」スイッチ，[7]キーを押し，＄RB009 ENTER ACTION の表示を確認して[1]
             のキー，「FUNCT」スイッチ，[1][1]と打鍵入力し次のカードデックを読み込む。
$$$RCD
$       SNUMB       zzzzz
          ⋮
$       ENDJOB
$$$OUTzzzzz
```

2. リモート・バッチ・システム用制御文

ここでは，リモート・バッチ処理方式を利用する際の制御文（あるいは，コマンド）について説明する。

2.1 制御文の一般形式

RBS（Remote Batch System）に対する制御文の形式は次のどちらかである。

```
欄  1    4       8
$$$xxx,  yyyyy          （例えば　LID）
```

欄　1　　4　　7

$$$xxx yyyyy　　（例えば　OUT,STS）

$$$　：　RBSに対する制御文（あるいはコマンド）であることを示す。

xxx　：　RBE（リモート・バッチ処理管理プログラム）の動作を指示する命令

yyyyy：　パラメータ

zzz　：　パラメータ

2.2　各制御文の説明

2.2.1　$$$CON,RBATCH

(1) 形　　式

欄　1　　4　　8

$$$CON,RBATCH

(2) 機　　能　　リモート・バッチ端局とセンターとをコネクトする。

(3) パラメータ　　な し

2.2.2　$$$LID

(1) 形　　式

欄　1　　4　　8

$$$LID,id

(2) 機　　能　　論理端末名を宣言する。

(3) パラメータ　　id ：論理端末名（医歯薬地区はR6，教養部地区はR7と指定する）

2.2.3　$$$RCD

(1) 形　　式

欄　1　　4

$$$RCD

(2) 機　　能　　リモート・バッチ・ジョブ・デックを読込むことを要求する。

(3) パラメータ　　な し

2.2.4　$$$OUT

(1) 形　　式

欄　1　　4　　7

$$$OUT〔{snumb / C}〕〔,AUTO〕

(2) 機　　能　　リモート・バッチ・ジョブの出力を要求する。

(3) パラメータ　　snumb　：　特定のSNUMB名をもつジョブの出力要求。

C　：　直前に中断された出力を中断された所から継続出力要求。（ただし，若干の出力が重複される）

空白　：　その論理端末に対して出力可能な全ジョブの出力要求。

AUTO　：　出力が終了しても端末はディス・コネクトされず次のジョブ出力が可能になるまで待つ。（このオプションはSNUMB名とともには,指定できない）この自動出力モードは，$$$CAN コマンドで解消されるか，ログオフまで続く。

2.2.5　$$$CAN

(1) 形　　式

欄　1　　4

$$$CAN

(2) 機　　能　　$$$OUTのAUTOによる自動出力モードを取消す。

(3) パラメータ　　な し

2.2.6 $$$STS

(1) 形　　式

```
欄  1    4    7
    $$$STS { snumb
           { 論理端末名
           { /ONL
           { /LSTR
```

(2) 機　　能　ジョブの処理状態の表示，中央システムへ出力変更または出力先変更ジョブの表示を要求する。

(3) パラメータ　空白　：その論理端末に対する全ジョブの状態表示要求。
snumb　：指定したSNUMB名をもつ特定のジョブの状態表示要求。
論理端末名　：指定した論理端末に対する全ジョブの状態表示要求
/ONL　：出力先をセンターに変更する。
（この要求がシステムに受け付けられたかどうかは，直ちに端末へ表示されるが，実際の変更処理は，中央システムの操作員の許可があった後に行われる）
/LSTR　：出力先変更のジョブのリストを要求する。

(4) 使 用 例　端末から複数ジョブを入力し，その後，各ジョブのステータスを調べる。

```
欄  1    4      8              16
    $$$CON,RBATCH
    $$$LID,R6(or R7)
    $$$RCD
    $          SNUMB    xxxxx   ┐
                 ≀                │
    $          ENDJOB           │
    $          SNUMB    yyyyy   │ バッチ・ジョブ・デックの構成と同じ。
                 ≀                │
    $          ENDJOB           ┘
    $$$BYE
    $$$CON,RBATCH
    $$$LID,R6
    $$$STS xxxxx   ┐ ステータスを要求するジョブのSNUMB名を指定する。この論理端末に
    $$$STS yyyyy   ┘ 関する全ジョブのステータスを表示させたいときは，SNUMB名は指定しない。
    $$$BYE
```

2.2.7 $$$ABT

(1) 形　　式

```
欄  1    4    7
    $$$ABT snumb
```

(2) 機　　能　その論理端末から入力したジョブの実行を強制的に打切る（アボート）ことを要求する。

(3) パラメータ　snumb　：打ち切りたいジョブのSNUMB名（必ず指定しなければならない）。

(4) 使 用 例　端末から複数ジョブを入力したが，その後，特定のジョブだけ実行を打切る。

```
欄  1    4      8              16
    $$$CON,RBATCH
    $$$LID,R6(or R7)
```

```
欄  1    4       8              16
    $$$RCD
    $           SNUMB       xxxxx
                  ≀
    $           ENDJOB
    $           SNUMB       yyyyy
                  ≀
    $           ENDJOB
    $$$CON,RBATCH
    $$$LID,R6(or R7)
    $$$ABTxxxxx ←――― 打ち切るべきジョブのSNUMB名を指定する。
    $$$BYE
```

2.2.8 $$$BYE

(1) 形　　式

```
欄  1    4
    $$$BYE
```

(2) 機　　能　リモート・バッチ処理を終了することを，RBEに通知するもので端末はディスコネクトされる。

(3) パラメータ　な し

(4) 使 用 例　端末からジョブを入力後，いったんリモート・バッチ処理を終了し，後で処理結果の出力を要求する。

```
欄  1    4       8              16
    $$$CON,RBATCH
    $$$LID,R6(or R7)
    $$$RCD
    $           SNUMB       xxxxx
                  ≀
    $           ENDJOB
    $$$BYE      ←――― 端末をいったんディスコネクトする。
    $$$CON,RBATCH
    $$$LID,R6(or R7)
    $$$OUT〔xxxxx〕
```

2.2.9 $$$CNT

(1) 形　　式

```
欄  1    4
    $$$CNT
```

(2) 機　　能　ブレークによって中断されたRBEの動作の再開を指示する。

(3) パラメータ　な し

2.2.10 $$$OAB

(1) 形　　式

```
欄  1    4
    $$$OAB
```

(2) 機　　能　ジョブ出力にブレークして入力許可があったとき，ファイル単位に出力を打切る。

(3) パラメータ　な し

2.2.11 $$$IAB

(1) 形　　式

欄　1　　4

$$$IAB

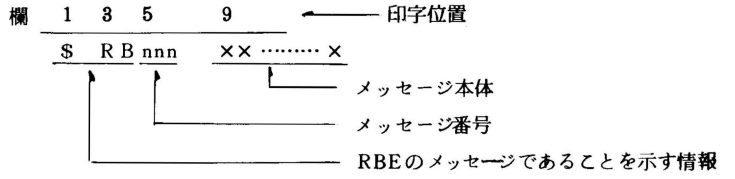
(2) 機　　能　入力中のジョブを打切りたいときに使用する。入力デックの最後に付けるか，入力処理にブレークしてこのコマンドを入力してもよい。

(3) パラメータ　なし

3. リモート・バッチ・システム使用中のエラー・メッセージ

ここでは，リモート・バッチ処理中に障害が発生したり，コマンドなどを間違えた場合のエラー・メッセージについて述べる。

メッセージ形式：



$ RB900 RBS BUSY …… WAIT

〈意　　味〉　端末のコネクト要求は受け付けたが，RBEの資源（端末制御テーブル等）が不足しているため，それ以後のサービスを続けられないことを示す。端末はRBEの資源が空くまで待たされる。

〈処　　置〉　RBEの資源が空くまで待つか，$$$DISコマンドを入力して強制的にディスコネクトする。

〈後続メッセージ〉　資源不足が解消したら，「$ RB002 ENTER ID」が表示され，論理端末名の入力が許される。

$ RB901 INVALID. ENTER ID OR BYE

〈意　　味〉　入力した論理端末名が重複している，すなわち既に他の端末から同じ論理端末名が宣言されていること。または$$$LIID以外のレコードであることを示す。

〈処　　置〉　正しい論理端末名を再入力するか，または$$$BYEコマンドを入力して終了する。

$ RB902 COMMAND CODE ERROR

〈意　　味〉　RBSコマンド以外のデータレコードがあったことを示す。

〈後続メッセージ〉　「$ RB 003 ENTER RBS COMMAND」が表示され，コマンドの再入力が許される。[注1)]

$ RB903 COMMAND PARAMETER ERROR

〈意　　味〉　RBSコマンドのパラメータが誤っていることを示す。

〈後続メッセージ〉　「$ RB 002 ENTER ID」あるいは「$ RB 003 ENTER RBS COMMAND」が表示される。

$ RB904 NO SNUMB, PROC OR IDENT CARD

〈意　　味〉　ジョブの先頭に$ SNUMB, $ PROCまたは$ IDENT文のいずれもなかったことを示す。

〈処　　置〉　なし　（次のジョブまたはRBSコマンドまで読み飛ばされる。ただし，それ以前のジョブは正しく受け入れられている。）

$ RB905 DECK ILLEGAL

〈意　　味〉　入力データ中に伝送制御符号があったことを示す。

〈処　　置〉　なし

$ RB906 INPUT TERMINATED. REASON－×××

〈意　　味〉　テンポラリ・ファイルに関して以下の障害が発生したため，ジョブ入力が中止されたことを示す。

×××は障害の原因を示す。

PTF ： PAT(Peripheral assignment table)エントリがない。

SIZ ： ジョブが長すぎる。

DSK ： 磁気ディスクエラー

LNK ： 磁気ディスクリンクが利用不可能

〈処　　置〉　な し

$ RB907 ILLEGAL BREAK ACTION

〈意　　味〉　ブレーク後に入力した指示が正しくないことを示す。

〈後続メッセージ〉　「$ RB003 ENTER RBS COMMAND」または「$ RB009 ENTER ACTION」が表示される。[注2)]

$ RB908 RJE FILE NOT AVAILABLE

〈意　　味〉　入力ジョブを一時的に格納するテンポラリファイルが確保できなかったことを示す。

〈処　　置〉　な し

$ RB909 SNUMB名 IS NOT IN SYSTEM

〈意　　味〉　RBSコマンドに指示されたSNUMB名をもつジョブがシステムになかったことを示す。

〈後続メッセージ〉　「$ RB003 ENTER RBS COMMAND」が表示される。

$ RB910 SNUMB名 CANNOT BE ABORTED

〈意　　味〉　$$$ABTで打切り指示されたジョブがなんらかの理由でアボートできなかったことを示す。

〈後続メッセージ〉　「$ RB008 ENTER RBS COMMAND」が表示される。

$ RB911 SNUMB名 IS NOT YOUR JOB

〈意　　味〉　$$$ABTコマンドで打切り指示したジョブは，その論理端末から入力されたものではないことを示す。

〈後続メッセージ〉　「$ RB003 ENTER RBS COMMAND」が表示される。

$ RB912 THIS COMMAND CANNOT USE NOW

〈意　　味〉　入力したコマンドの処理に必要なRBEの資源が不足しているため，動作できないことを示す。要求は拒否されている。

〈処　　置〉　RBEのbusy状態解消後再入力する。

$ RB914 INVALID. ENTER USERID

〈意　　味〉　入力したUSER IDは不正である。

〈処　　置〉　再度正しく入力する。

$ RB915 INVALID. ENTER PASS WORD

〈意　　味〉　入力したパスワードは不正である。

〈処　　置〉　再度正しく入力する。

注意1　「$ RB003 ENTER RBS COMMAND」が表示されたときの処置としては，
$$$RCD，$$$OUT，$$$STS，$$$ABTまたは$$$BYEコマンドを入力する。
ただし$$$RCDの後にはジョブ・デックが必要である。

2　「$ RB009 ENTER ACTION」が表示されたときの処置としては，

(1) 出力中にブレークした場合，$$$RCD，$$$ABT，$$$OAB（出力打切り），$$$BYEまたは$$$CNTコマンドを入力する。

(2) 入力中にブレークした場合，$$$OUT（出力要求），$$$STS，$$$ABT，$$$IAB（入力打切り），$$$BYEまたは$$$CNTコマンドを入力する。

(3) アイドル（出力待ち）中にブレークした場合，
$$$RCD，$$$STS，$$$ABT，$$$CAN，$$$BYEまたは$$$CNTコマンドを入力する。

4. リモート・バッチ・システム用端末機器の操作法

4.1 N 6240 データ・ステーション操作法

4.1.1 標準手順

(1) 電源の投入

① 基本制御装置の下部にある電源スイッチをONにする。

② カード読取装置の「POWER」スイッチを押下する。

③ ラインプリンタの操作パネル上の「CLEAR」，「POWER」，「START」のスイッチを押す。

(2) ファームウェアと基本プログラムの格納

④ キャラクタ・ディスプレイ上で「HLT」の表示を確認して（「RUN」の表示のときは「STOP」スイッチを押しHLTさせる。）フロッピィディスク装置0番にコントロール・フロッピィをセットする。

⑤ 基本制御装置操作卓上の「MODE」スイッチを押す。

⑥ 〃 「CLEAR」スイッチを押す。

⑦ 〃 「PRE−SET」スイッチを押す。

⑧ ブザーが鳴り「FD#0 SET IPL」表示（ファームウェア読込み終了）を確認して次にソフトウェア・フロッピィをコントロール・フロッピィと交換する。

⑨ 基本制御装置操作卓上の「START」スイッチを押す。

⑩ 「PROGRAM SELECTION」表示されると[0][0][I]とキーインする。

⑪ 「SHORT COMMAND＝」と表示されると[Y][E][S] [復改] とキーインする。

(3) リモート・バッチ・ジョブの入力

⑫ $$$CON，RBATCHを先頭にカードデックをカード読取装置にセットし，カード読取装置の「RESET」スイッチを押す。

⑬ 操作卓の「FUNCT」スイッチを押す。

⑭ 数字キー部から[1][I]と打鍵入力する。（カードを読取りセンターへ送る。ジョブ入力のたびに⑫ ～ ⑭を繰返す）

(4) 電源の切断

⑮ 「IDLE」状態または「PROGRAM SELECTION」状態であるか確認する。

⑯ フロッピィ・ディスク装置からシステム・フロッピィを取外す。

⑰ 各装置（カードリーダ，ラインプリンタ，モデムなど）の電源を切る。

⑱ 基本制御装置の電源を切断する。

4.1.2 その他の操作法

(1) 操作卓からコマンドまたはデータを入力したいとき．

① 「NK−IN」ランプ点灯を確認して（点灯していなければ「FUNCT」スイッチを押す）[I]キーを押す。

② 「NK−IN」，「KB−IN」ランプ点灯確認後，コマンドまたはデータを入力する。

××……× [復改]

あるいは ××……× [HTAB]

(2) 入力または出力待ち状態になった場合（例えば，カードがうまく読込めないとか，ラインプリンタの用紙がなくなった場合など）の再開始は，「WAIT」表示，「NK−IN」ランプ点灯確認後[II]のキーを押す。

(3) 「FUNCT」スイッチを押した後，このキーインを取消したいときは，[8][0][I]と入力する。

4.1.3 操作卓の機能

ここでは，スイッチ，ランプおよびキーの中で主なものについてその機能を説明する。

(1) スイッチ

① 「FUNCT」スイッチ……このスイッチを押すと，端末プログラムへの割込みが発生する。

端末プログラムがセンターデータの受信中あるいはアイドル中で，けん盤入力

が不可の場合，このスイッチを押下することにより，けん盤入力が可能となる。

② 「CLEAR」スイッチ……システムをイニシャライズする。

③ 「START」スイッチ……停止中のプログラムの実行が再開される。

④ 「STOP」スイッチ……プログラムの実行を停止させる。

(2) ランプ

① NK—IN
○ ……このランプが点灯中は，キーボードの数字キーより入力が可能となる。

② KB—IN
○ ……このランプが点灯中は，キーボードの文字キーより入力が可能となる。

③ 0 ～ 7
○～○……ランプ0～5は，正常動作中は滅灯している。
ランプ6の点灯は送信中
ランプ7の点灯は受信中
を表している。

(3) キーボード

① [復改] ……データ入力の終りを指示する。データはセンターへ送信される。
([HTAB] キーとの相違点は [復改] のときプリンタは復帰改行するが，[HTAB] の場合は復帰改行しない点である)

② [I]…… 入力装置の起動を指示する。
入力装置の選択は，次のように1桁の数字で行う。
指示形式……×I
↑L { 0または省略：操作卓
1 ：カードリーダ
2 ：紙テープリーダ

③ [I]……読取りまたは出力の再開始を指示する再開キーである。

④ [W]…… ブレーク・キー(BK) であり，センター・プログラムの動作に割込む場合にこのキーを押す。

⑤ [リセット]……エラーが発生しているときにエラーのリセットを行う。

⑥ [←]…… カーソルを左へ1文字移動させる。

⑦ [→]…… カーソルを右へ1文字移動させる。行の右端にカーソルがあるときは，次の行の先頭位置に移る。

ファイル・システム利用法

1. ファイル・システムの構成 …… 102
   1.1 ファイル・システムの構造 …… 102
   1.2 カタログの種類と意味 …… 102
   1.3 カタログ名とファイル名の記法 …… 104
   1.4 カタログ名とファイル名の修飾 …… 104
2. ファイルの共用と機密保護 …… 104
   2.1 パスワード …… 104
   2.2 パーミッション …… 105
3. ファイルの編成 …… 106
   3.1 ファイル編成の種類 …… 106
   3.2 ファイル編成の指定 …… 107
4. ファイル領域の割当てと管理 …… 107
   4.1 領域の単位 …… 107
   4.2 領域の割当て …… 107
   4.3 領域の解放 …… 107
5. ファイル制御プログラム …… 108
   5.1 ACCESSとFILSYSの機能 …… 108
   5.2 ACCESSとFILSYSの指令形式 …… 108
   5.3 ファイル用指令 …… 109
   5.4 カタログ用指令 …… 112
   5.5 カタログとファイル用指令 …… 115

注意） ここで述べる内容は，ファイルを有効に利用する上で特に重要となります。それだけに，これらの機能が学生に一度悪用され始めると重大な影響を及ぼしかねません。ファイル・システム利用法を読むに当たってこの点を十分銘記しておいて頂きたい。

ACOS－6においては，ファイルはパーマネント・ファイルとテンポラリ・ファイルの2種類に分類される。パーマネント・ファイルは，ジョブやセッション終了後も必要となるデータを格納する保存ファイルであり，テンポラリ・ファイルは，ジョブやセッション内でのみ一時的に発生するデータを格納する一時ファイルである。ファイル・システムで管理の対象となるのはパーマネント・ファイルのみである。

パーマネント・ファイルは，ファイル・システムにより，カタログを用いて集中的，統一的な管理がなされ，種々の処理形態の下で異なった利用者からも共通に利用することができる。パーマネント・ファイルは，大きさの指定により必要な領域が自動的にディスク上に確保され，その名前すなわちファイル名を指定するだけで簡単にディスク上のファイルをアクセスすることができる。またファイルを複数の利用者で共用することもでき，その際，ファイルの機密保護や共用の度合いの設定などができる。

1. ファイル・システムの構成

1.1 ファイル・システムの構造

パーマネント・ファイルは，カタログにより階層化された木構造として集中的かつ統一的に管理される。ファイル・システムの構造を下図に示す。木構造の幹にはシステム・マスタ・カタログ（SMC）があり，その直下の第1ノード（節）は利用者のマスタ・カタログ（UMC）であり，これより下のレベルのノードは利用者が作成したサブカタログである。そしてファイル自身が木構造の末端を構成している。ファイル・システムのレベル（UMCから数えたカタログ構造の深さ）は8レベルまでに限られている。

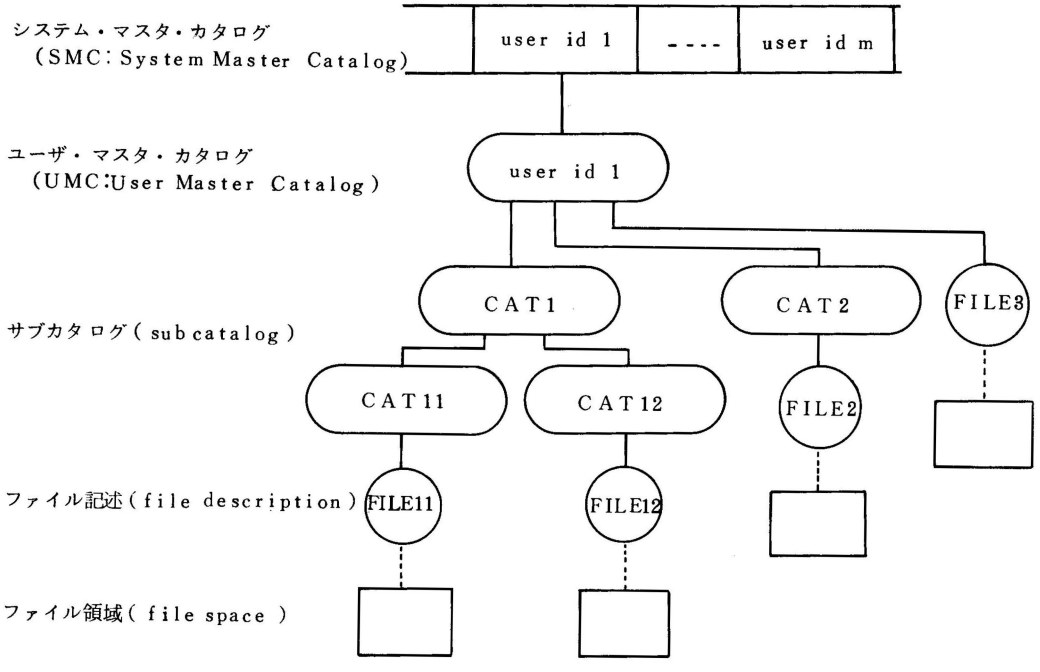


図1.1 ファイル・システムの構造

1.2 カタログの種類と意味

カタログには次の4つのものがある。

(1) システム・マスタ・カタログ（SMC）

SMCは，各利用者について次の情報が記述されている。

- SMC名すなわち利用者識別名（user id）
- 利用者パスワード
- 利用可能ファイル領域の大きさ
- 予算額
- 利用実績記録

SMCは利用申請書に基づいてセンターが作成する。利用者は，バッチでは$ JOB制御文で，タイムシェアリングではログオン時に，SMC名であるuser idとパスワードを指定しなければならない。

タイムシェアリング・システムでは

USER ID－ user id

PASSWORD－－パスワード

バッチ・システムでは

$ JOB user id, account number, パスワード

(2) ユーザ・マスタ・カタログ（UMC）

UMCはこれより下位のサブカタログやファイルを作成する際に自動的に作られ，UMC名としてはSMC同様user idが用いられる。利用者は，タイムシェアリング・システムで他人のファイルを利用する場合，バッチ・システムでファイルを利用する場合には，修飾ファイル名の先頭でUMC名としてuser idを指定する必要がある。

タイムシェアリング・システムでは

＊ ファイル利用コマンド

または

ACCESSシステムのFUNCTIONで 〔user id／カタログ名／・・・／〕ファイル名

⇓

修飾ファイル名

バッチ・システムでは（→バ利）

$ PRMFL ファイル・コード，割当てタイプ，アクセス・モード，

user id／〔カタログ名／・・・／〕ファイル名

⇓

修飾ファイル名

(3) サブカタログ

サブカタログはUMCと(4)で述べるファイル記述との中間に位置するもので，これに従属する下位のサブカタログは，第5章で述べるように，タイムシェアリング・システムではACCESSサブシステムを，バッチ・システムではFILSYSアクティビティを実行することにより作成される。サブカタログは単にカタログとも呼ばれ，利用者は修飾ファイル名の中で木構造に沿ってカタログ名を指定する。

(4) ファイル記述

ファイル記述は木構造の末端に位置し，次の情報が記録されている。

- ファイル名
- パスワード
- ファイルの現在のサイズと最大サイズ
- ファイル領域のディスク上の位置
- パーミッション
- 作成，変更，最終アクセスなどの年月日

ファイル・システムでは，ファイルという場合このファイル記述を意味し，利用者は，タイムシェリアリング・システムのACCESSサブシステムやバッチ・システムのFILSYSアクティビティによってファイル記述を作成する。また，ファイル名は(3)で述べた修飾ファイル名の最後で指定する。なおサブカタログを経由しないで，UMCの直下に作られるファイルを特にクイック・アクセス・ファイルと呼ぶ。

1.3 カタログ名とファイル名の記法

個々のカタログおよびファイルはその名前によって識別される。カタログ名とファイル名は共に1～12文字から成り，名前に使用できる文字は次のものである。

大文字アルファベット

数　字

ピリオド（ . ）

マイナス（ － ）

ただし，すべて数字0（ゼロ）で構成される名前は使用できない。またタイムシェアリング・システムでは，ファイル名は最大8文字である。

1.4 カタログ名とファイル名の修飾

ファイルを利用する際あるいはカタログを取扱う際には，カタログまたはファイル名を木構造に沿って上位のカタログ名で修飾してゆく。これをカタログ名またはファイル名の修飾といい，修飾を受けたカタログ名またはファイル名を修飾カタログ名または修飾ファイル名と呼ぶ。なお，修飾ファイル名はファイル・ストリングあるいは filedescr（ファイル・デスクリプション）ということもある。

修飾カタログ名または修飾ファイル名の先頭はUMC名としてのuser id（ただしタイムシェアリング・システムでは省略できる場合がある）であり，以下木構造に沿って上位のカタログから順次スラッシュ（／）で区切って書き，最後にカタログ名またはファイル名を書く。

修飾カタログ名と修飾ファイル名の指定を次に示す（→図1.1）。

カタログ	CAT1	〔user id〕注)／CAT1
カタログ	CAT11	〔user id〕／CAT1／CAT11
カタログ	CAT12	〔user id〕／CAT1／CAT12
カタログ	CAT2	〔user id〕／CAT2
ファイル	FILE11	〔user id〕／CAT1／CAT11／FILE11
ファイル	FILE12	〔user id〕／CAT1／CAT12／FILE12
ファイル	FILE2	〔user id〕／CAT2／FILE2
ファイル	FILE3	〔user id〕 FILE 3

注）タイムシェアリング・システムでは，セッションを開いた利用者のファイルであれば省略できる。

2. ファイルの共用と機密保護

2.1 パスワード

複数の利用者間でファイルを共用する場合に，アクセス権をもたない利用者からのアクセスを防止する目的でカタログおよびファイルにパスワードを付与することができる。カタログまたはファイルに一旦パスワードが付けられると，これをアクセスする場合には常にパスワードを指定しなければならない。ただし，タイムシェアリング・システムでの利用において，AFT（アベイラブル・ファイル・テーブル）にファイル名が登録された場合にはこの限りではない。パスワードは1～12文字からなり，使用できる文字は次のものである。

大文字アルファベッド

数　字

ピリオド（ . ）

マイナス記号（－）

利用者はカタログ名またはファイル名を指定するときには，この後に$パスワードとしてパスワードを与える。図1.1でカタログCAT2にパスワードPCAT 2，ファイルFILE 2にパスワードPFILE 2 が付与されている場合には，次のように修飾ファイル名を指定する。

例）　〔user id〕／CAT2 $ PCAT 2／FILE 2 $ PFILE 2

2.2　パーミッション

(1) パーミッションの機能と分類

ファイルは通常そのファイルの作成者自身しかアクセスすることができないが，パーミッションを取付けることにより，他の利用者との間でそのファイルを共同利用することが可能になる。パーミッションとは，自分が作成したファイルを他人に使わせる際の許可操作のことである。

パーミッションには，すべての利用者を対象とする一般パーミッション，特定の利用者を対象とする特定パーミッション，および一般パーミッションから特定利用者を除外するEXCLUDEパーミッションがある。

(2) パーミッションの種類

パーミッションの種類には次のものがあり，これらは特定パーミッションとしてもあるいは一般パーミッションとしても指定できる。

① READまたはR

ファイルからの読込みと実行を許す。

② WRITEまたはW

ファイルからの読込みおよびファイルへの書出しを許す。

③ APPENDまたはA

現時点で終端とみなされるファイルの部分に対してのみ情報の転送を許す。

④ EXECUTEまたはE

ファイルからの実行のみを許す。

⑤ PURGEまたはP

ファイルの抹消を許す。同時にファイルに対する読込み・書出しも許される。

⑥ MODIFYまたはM

ファイル記述の変更を許す。上記の全パーミッションを包含する。

その他，カタログに対してのみ有効となる CREATEがある。

上記パーミッションの包含関係を図示すると図2.1のようになる。

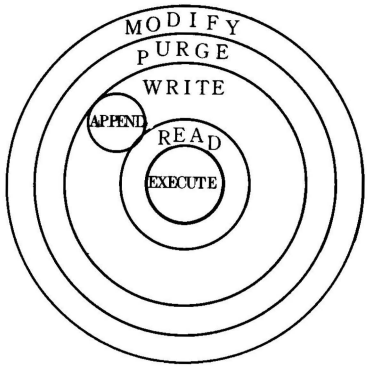


図2.1　パーミッションの包含関係

(3) 一般，特定，EXCLUDEパーミッションの関係

一つのファイルまたはカタログには一般パーミッションと特定パーミッションとの両方のパーミッションを付けることができる。しかしながらこの両方を付けたときには，利用者に対して，特定パーミッションを指定された利用者のみ特定パーミッションが適用され，特定パーミッションの指定がない利用者には一般パーミッションが適用される。

EXCLUDEパーミッションを指定すると，特定パーミッションを与えることなしに特定の利用者を一般パーミッションから除外することができる。

パーシッションを省略した場合には，一般パーミッション，特定パーミッションおよびEXCLUDEパーミッションは無指定とみなされる。

(4) パーミッションの指定法

① 一般パーミッション

許可操作は任意の順序でその全体または短縮形を書く

許可操作，……，許可操作

② 特定パーミッション(EXCLUDEを含む)

{EXCLUDE / 許可操作} / user id，……/，{EXCLUDE / 許可操作} / user id，……/，……

3. ファイルの編成

ACOS－6では，3つの形式のファイル編成すなわち順編成，直接編成，索引順編成を許している。

3.1 ファイル編成の種類

3.1.1 順編成ファイル

順編成ファイルは，ファイル内のレコードの順序と処理されるレコードの順序が一致するファイルで，テープなどの上に作成される。物理的に識別できるレコードを物理レコードといって，この中は1個以上の論理レコード(ソフトウェアで処理する単位)から成っている。

3.1.2 直接編成ファイル

直接編成ファイルはディスク上にだけ作成されるファイルで，64語のブロックに分割されている。各ブロックにはファイルの先頭から相対アドレスが付けられているので，このアドレスを指定することにより直接アクセスが行える。

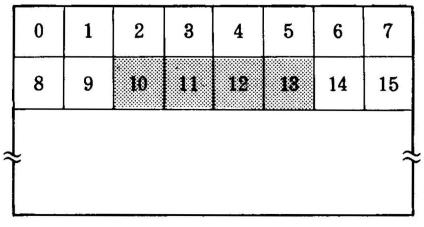


図3.1 直接編成ファイルの構造

図3.1 で斜線部分を処理したいときは，相対ブロック番号10と入力データ長256語(64語×4)を指定してREAD／WRITE命令を出せばよい。

3.1.3 索引順編成ファイル

索引順編成ファイルは，ディスク上にだけ作成されるファイルで索引ファイルとデータ・ファイルの2種類のファイルで構成されている。データ・ファイルは1個から最大8個までのファイルから成っている。索引ファイル，データ・ファイルともページと呼ぶ一定長の単位に分割され，主記憶とディスク間の情報転送はページ単位で行われる。アクセスは，キー値指定による直接アクセスと順次アクセスの二つの方法がある(→図3.2)

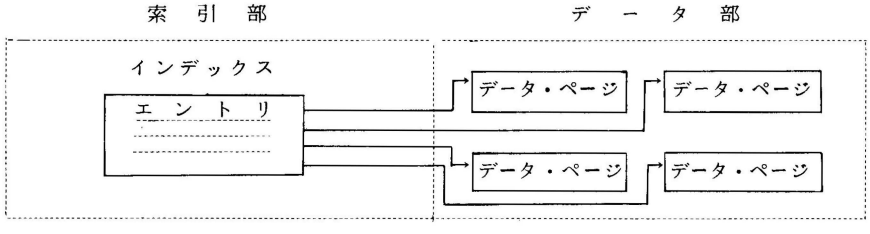


図3.2 索引順編成ファイルの構造

3.2 ファイル編成の指定

(1) ファイル・システムでは { 順編成ファイル SEQ / 直接編成ファイル RAND }

(2) ジョブ制御文では { 順編成ファイル S / 直接編成ファイル R }

と指定する。

4. ファイル領域の割当てと管理

4.1 領域の単位

利用者がファイルを開設する際，必要なファイル領域はその大きさを指定するだけで，ファイル・システムによって自動的に確保され割当てられる。割当ての単位はLリンク（LLINK）またはリンク（LINK）であり，1Lリンクは320語の容量をもち，1リンクは12リンクすなわち3840語の容量をもつ。これらの単位と容量の関係を表4.1に示す。

表4.1 ファイル領域の単位と容量

容量 / 単位	Lリンク	語数（語）	BCD（6ビット／字）文字数（字）	JIS（9ビット／字）文字数（字）
Lリンク	1	320	1,920	1,280
リンク	12	3,840	23,040	15,360

タイムシェアリング・システムでソースプログラム／データをファイルに格納する場合にはJISコードの可変長テキスト・イメージ形式が用いられるので，1Lリンクに格納できるテキスト（行）数は一定でないが，80字／行とした場合には1Lリンクには16枚格納できる。バッチ・システムでのソース・プログラム／データのコードは通常BCDのカード・イメージ形式なので1Lリンクには21枚のカードを格納できる。

4.2 領域の割当て

(1) 利用者に対する領域の割当て

利用者が使用できるファイル領域の最大値として，利用申請書のファイル使用量の欄に記入された値がSMCに記録される。この値を上限値としてファイルの作成，拡張が許可される。ただし，システムがカタログやファイル記述を記録するために1リンク必要とするので，利用者が実際のファイルのために使用できる領域は申請時の値よりも1リンク少なくなる。

(2) ファイルに対する領域の割当て

タイムシェアリング・システムのACCESSサブシステムやバッチ・システムのFILSYSアクティビティを実行してファイルを作成するときに，ファイルに割当てる領域の初期値と最大値を指定できる。初期値は最初にファイルで使用する領域で指定がなければ1リンクがとられる。最大値はファイル領域の制限値で，利用者の使用できる領域の大きさの範囲内であれば任意の値（≧初期値）に設定できる。最大値の指定がないと最大値は初期値と同じになり拡張が許されなくなる。

タイムシェアリング・システムでSAVEコマンド（→タ利3.5.3）によりファイルが自動的に作成されるときには，初期値はプログラム／データを格納するのに必要な領域がとられ，最大値はその20倍の大きさの値が設定される。

4.3 領域の解放

ファイルに割当てられている領域は，ファイルやファイルが属するカタログを抹消したときにのみ解放される。したがって，現在使用中のファイル領域が，新たなデータの書込みなどによって小さくなったとしても元の領域の大きさのままである。

5. ファイル制御プログラム（ACCESS, FILSYS）

5.1 ACCESSとFILSYSの機能

ファイル・システムにおいてカタログやファイルの作成・変更などは，タイムシェアリング・システムではACCESSサブシステムを，バッチ・システムではFILSYSアクティビティを実行することによって行われる。これらのプログラムは次の機能をもつ。

(1) カタログおよびファイル記述と領域の作成

(2) カタログおよびファイル記述の変更

(3) カタログ，ファイル記述およびファイルの抹消

カタログを抹消すると，そのカタログの下位に属するすべてのカタログ，ファイル記述およびファイル自身も消去される。

(4) カタログおよびファイル記述のリスティング

5.2 ACCESSとFILSYSの指令形式

5.2.1 入力形式

ACCESSサブシステムにおいては，次の形式で入力する。

SYSTEM?　　ACCESS (ST) 注1)

FUNCTION?　[ACCESS　指令形式 1]　(ST)

FUNCTION?　[ACCESS　指令形式 2]　(ST)

⋮　注2)

FUNCTON?　[ACCESS　指令形式 n]　(ST)

FUNCTION?　{ (ST) / DONE (ST) }　注2)　システム選択レベルに戻す

SYSTEM?　　他のサブシステム名またはBYE

注1)　(ST) 1行の入力終了キーを表す。

注2)　システムからのメッセージは次のFUNCTION?までの間に出力される（リスト出力の機能使用中）

指令形式が1行におさまらない場合には，(ST)の前にカタログのレベルを表す／（スラッシュ），パラメータを分離する，（コンマ）あるいはパスワードの前に付加する$（ドル）などのデリミタ（分離記号）を置けば，次行にMORE?

と出力されるので，この後に残りの部分を入力してゆく。

FILSYSアクティビティにおいては，次のようにカード・デックを構成する（ただしJOB文などは省略している）。

欄　1　　8　　72

$　　FILSYS

[FILSYS　指令形式 1]

[FILSYS　指令形式 2]

⋮

[FILSYS　指令形式 n]

$　　(next activity)

FILSYS指令カードは複数枚に渡ってもよいが，一枚のカードは1～72欄が有効であり，次のカードに継続する場合は，／（スラッシュ）または，（コンマ）で終らせる。指令の最後は␣（スペース）で終らせる。

FILSYS指令カードは指定された順序で一枚ずつ処理され，エラーが発見されるとその報告がされ，残りの指令カードについては構文チエックのみ行われる。

ただしリスティング用の指令は，前のカードにエラーがあっても処理される。

5.2.2 指令の一般形

ACCESSおよびFILSYS指令形式の一般形は次のようになる。

	オペレーション部	オペランド部
ACCESS	機能名 ,	修飾名，オプション
FILSYS	機能名 ␣	

以下，指令形式の説明では次の記法を用いる。

- 大カッコ（{ }）は選択項目で，そのなかの一つを必ず指定する。
- カギカッコ（〔 〕）は指定してもしなくてもよい項目を表す。
- 日本語または英小文字は必要なもので置き換える。
- 英大文字はその通りに書く。
- 反復記号（…）は繰返しを意味する。

5.3 ファイル用指令

5.3.1 ファイルの作成

(1) 形　　式

ACCESS	{ CREATE FILE / CF } ,	修飾ファイル名，ファイルのサイズ〔，パスワード〕〔，パーミッション〕〔，アクセス・モード〕
FILSYS	{ FCREAT / FC / CF } ␣	

(2) 機　能　　指定された属性のファイル記述を作成し，ファイル領域を開設する。

(3) パラメータの説明

① パスワード

PASSWORD／名前／

② パーミッション

2.2の(4)で述べた記法に従って指定する。

③ ファイルのサイズ

{ LLINKS / LINKS } ／初期値 [, { UNLIMITED / 最大値 }] ／

UNLIMITEDを指定すれば，SMC（System Master Catalog）の制限値までファイルのサイズを拡張できる。大きさの単位はLLINKS＝320語（1語＝36ビット），LINKS＝3840語（12LLINKS）である。

④ アクセス・モード

MODE ／ { SEQ / RAND } ／

省略するとSEQが指定されたものと見なされる。SEQは順アクセスおよび直接アクセスが可能であるが，RANDは直接アクセスのみしか許されない。

(4) 注　　意

① ファイルを直接アクセスとして使用するときには，ファイルのサイズは初期値＝最大値としなければならない。

② ファイルは通常リンク単位に拡張されるので，次のように初期値を1Lリンクとする指定は避けた方がよい。

LLINKS ／ 1， UNLIMITED ／　または　LLINKS ／ 1， 10／

(5) 使 用 例

(a) ACCESS（タイムシェアリング）

SYSTEM? ACCESS (SD)

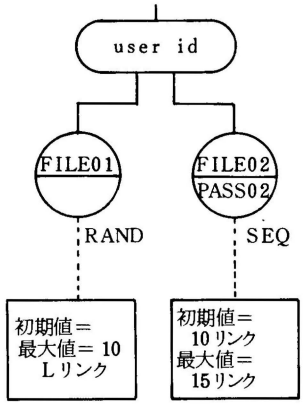
FUNCTION? ␣ CF, user id／FILE01, MODE／RAND／, LLINKS／10, 10／(SD)

FUNCTION? ␣ CF, user id／FILE02, PASSWORD／PASS02, (SD)

MORE? LINKS／10, 15／(SD)

FUNCTION? ␣ (SD)

SYSTEM? （next subsystem）またはBYE



(b) FILSYS（バッチ）

欄	1	8	72
	$	FILSYS	
	CF	user id／FILE01, MODE／RAND／, LLINKS／10, 10／	
	CF	user id／FILE02, PASSWORD／PASS02,	
		R, LINKS／10, 15／	
	$	(next activity)	

(c) 説　　明

user id で指定されたUMCの直下に，クイック・アクセス・ファイルFILE01，FILE02を作成する。

FILE01は，パスワードはなく，ファイルのサイズは10リンクで拡張はできない。パーミッションの指定がないので，他の利用者はアクセスできない。アクセス・モードは指定によりRANDで直接アクセスのみ可能である。

FILE02は，パスワードはPASS02で，一般パーミッションREADが指定されているので，他の利用者はこのファイルの参照のみ許される。ファイルのサイズは最初10リンクで最大15リンクまで拡張できる。アクセス・モードはSEQで順アクセスおよび直接アクセスが許される。

5.3.2 ファイル記述の変更

(1) 形　　式

ACCESS	{MODIFY␣FILE / MF} ,	修飾ファイル名〔，パスワード〕〔，追加パーミッション〕〔，変更最大値〕〔，変更ファイル名〕〔，パーミッションの除去〕〔，変更アクセス・モード〕
FILSYS	{FMOD / FM / MF} ␣	

(2) 機　能　　指定された名前のファイルの属性を変更する。

(3) パラメータの説明

① パスワード

{PASSWORD／名前／ / PASSWORD}　　PASSWORD／名前／　：パスワードの変更

PASSWORD　：パスワードの除去

② 追加パーミッション

追加すべきパーミッションを指定する（→2.2）

③ 変更最大値

{LLINKS / LINKS} ／ {UNLIMITED / 最大値} ／　　最大値は既に指定しているサイズ以上でないといけない。

④ 変更ファイル名

NEWNAM／パスワードのないファイル名／　　注1）

注1）　サブカタログなどのないファイル名

⑤ パーミッションの除去

DELETE／{〔GEN'L,〕user id，…，user id}／

GEN'L注2）は一般パーミッションを除去する。　注2）'は省略記号（アポストロフィー）である。

user id，…，user id はそれぞれの利用者に与えられている特定パーミッションを除去する。

⑥ 変更アクセス・モード

MODE／{SEQ / RAND} ／

(4) 注　　意

① パスワード付ファイルの場合には，修飾ファイル名にはパスワードが必要である。

② 修飾ファイル名のほかに必ず1つ以上のオプション・パラメータを指定しなければならない。

(5) 使 用 例

(a) ACCESS（タイムシェアリング）

```
SYSTEM? ACCESS (SD)
① FUNCTION? MF, user id／FILE01, LLINKS／20／(SD)
② FUNCTION? MF, user id／FILE02$ PASS02, NEWNAM／FILE002／(SD)
  FUNCTION? (SD)
  SYSTEM?  (next subsystem) またはBYE
```

(b) FILSYS（バッチ）

```
欄   1          8
     $          FILSYS
①    MF         user id／FILE01, LLINKS／20／
②    MF         user id／FILE02$ PASS02, NEWNAM／FILE002／
     $          (next activity)
```

(c) 説　　明

① ファイルFILE01の最大サイズを20 Lリンクに変更する。

② ファイルFILE02のファイル名をFILE002に変更する。従って以後修飾ファイル名はuser id／FILE002$PASS02となる。

5.3.3 ファイルの抹消

(1) 形　　式

ACCESS	{RELEASE␣FILE / RF} ,	修飾ファイル名
FILSYS	{FRELES / FR / RF} ␣	

または

ACCESS	{PURGE␣FILE / PF} ,	修飾ファイル名
FILSYS	{FPURGE / FP / PF} ␣	

(2) 機　能　　指定されたファイルの領域を解放する（RELEASE）またはファイルの領域をクリア（帰零）後解放する（PURGE）。

(3) 使 用 例

(a) ACCESS（タイムシェアリング）

SYSTEM? ACCESS (SD)
FUNCTION? RF, user id／FILF01 (SD)
FUNCTION? PF, user id／EILE02$PASS02 (SD)
FUNCTION? (SD)
SYSTEM? （next subsystem）またはBYE

(b) FILSYS（バッチ）

欄　1　　8

```
$       FILSYS
RF      user id／FILE01
PF      user／FILE02$PASS02
$       (next activity)
```

5.4 カタログ用指令

5.4.1 カタログの作成

(1) 形　　式

ACCESS	{CREATE␣CATALOG / CC} ,	修飾カタログ名〔, パスワード〕〔,パーミッション〕
FILSYS	{CCREAT / CC} ␣	

(2) 機　能　　指定された名前のカタログを作成する。

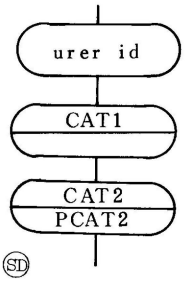
(3) パラメータの説明

① パスワード

PASSWORD／名前／

② パーミッション

2.2 の(4)で述べた記法に従って指定する。



(4) 使 用 例

(a) ACCESS（タイムシェアリング）

SYSTEM? ACCESS (SD)

FUNCTION? CC, user id／CAT1 (SD)

FUNCTION? CC, user id／CAT1／CAT2, PASSWORD／PCAT2／(SD)

FUNCTION? (SD)

SYSTEM? （next subsystem） またはBYE

(b) FILSYS（バッチ）

欄	1	8	72
	$	FILSYS	
	CC	user id／CAT1	
	CC	user id／CAT1／CAT2, PASSWORD／PCAT2／	
	$	(next activity)	

(c) 説 明

user id で指定されたUMCの下にカタログCAT2を作成する。

5.4.2 カタログの変更

(1) 形 式

ACCESS	{MODIFY CATALOG / MC} ,	修飾カタログ名〔, 変更カタログ名〕〔, パスワード〕〔, パーミッション〕〔, パーミッションの除去〕
FILSYS	{CMOD / CM / MC} ␣	

(2) 機 能 カタログ名を指定されたように変更する。

(3) パラメータの説明

① 修飾カタログ名 変更の対象となるカタログ名を修飾された形で指定する。
パスワードがあればこれも含めて指定する。

② 変更カタログ名

NEWNAM／パスワードのない非修飾カタログ名／

③ パスワード

{PASSWORD／名前／ / PASSWORD} PASSWORD／名前／ :（パスワードの変更）
PASSWORD :（パスワードの除去）

④ パーミッション

追加すべきパーミッションを指定する（→2.2）。

⑤ パーミッションの除去 注)

DELETE／{〔GEN'L,〕user id, …, user id}／

GEN'Lは一般パーミッションを除去する。

user id, …, user id はそれぞれの利用者に与えられている特定パーミッションを除去する。

注) 'は省略記号（アポストロフィー）である。

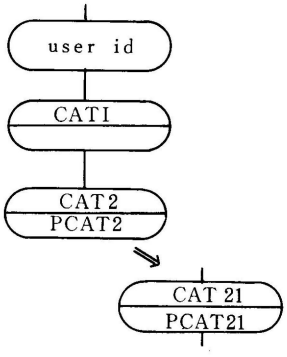
(4) 使 用 例

(a) ACCESS（タイムシェアリング）

```
SYSTEM?  ACCESS (SD)
FUNCTION?  MC, user id／CAT1／CAT 2 $ PCA 2 , NEWNAM／CAT21／(SD)
FUNCTION?  MC , user id／CAT1／CAT21 $ PCAT 2 , PASSWORD／PCAT21／(SD)
FUNCTION?  (SD)
SYSTEM?  （next subsystem）またはBYE
```

(b) FILSYS（バッチ）

```
欄   1          8
$          FILSYS
MC         user id／CAT 1／CAT 2 $ PCAT 2 , NEWNAM／CAT21／
MC         user id／CAT1／CAT21 $ PCA 2 , PASSWORD／PCAT21／
$          (next activity)
```



5.4.3 カタログの抹消

(1) 形　　式

ACCESS	{ RELEASE␣CATALOG / RC } ,	修飾カタログ名
FILSYS	{ CRELES / CR / RC } ␣	

または

ACCESS	{ PURGE␣CATALOG / PC } ,	修飾カタログ名
FILSYS	{ CPURGE / CP / PC } ␣	

(2) 機　能　指定されたカタログおよびこのカタログの下に属するすべてのカタログとファイルとを抹消し，ファイルのスペースを解放する。ファイルの領域は，PURGEの場合にはクリア（帰零）後解放され，RELEASEの場合にはそのままの状態で解放される。

(3) 使 用 例

(a) ACCESS（タイムシェアリング）

SYSTEM? ACCESS (SD)

FUNCTION? RC, user id／CAT1／CAT21 (SD)

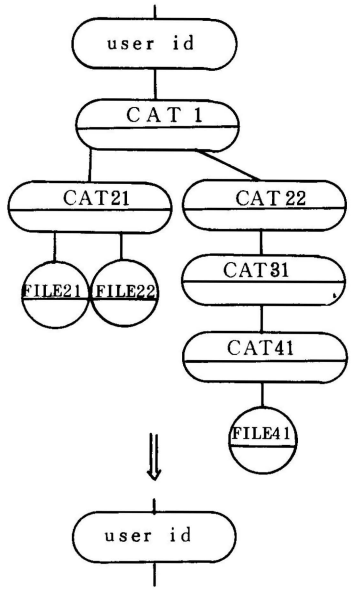
FUNCTION? PC, user id／CAT1 (SD)

FUNCTION? (SD)

SYSTEM? （next subsystem）またはBYE

(b) FILSYS（バッチ）

欄 1	8
$	FILSYS
RC	user id／CAT1／CAT21
PC	user id／CAT1
$	(next activity)



5.5 カタログとファイル用指令

5.5.1 カタログやファイルのリスト（タイムシェアリング）

(1) 形 式

ACCESS	{ LIST CATALOG / LC },	修飾カタログ名〔, x(*mm-dd-yy*, n, R)〕〔, FIRST／name／〕

(2) 機 能 指定されたカタログに属するカタログ名およびファイル名を端末上に表示する。

(3) パラメータの説明

(a) x (*mm-dd-yy*, n, R)

リストするカタログあるいはファイルを限定したい場合に指定する。

x： C……作成日付による限定

A……最終アクセス日付による限定

L……最終修正日付による限定

mm-dd-yy xに対応する日付を*mm*月*dd*日19*yy*年で指定する。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

R……出力の順番を逆（新しい順）にする。

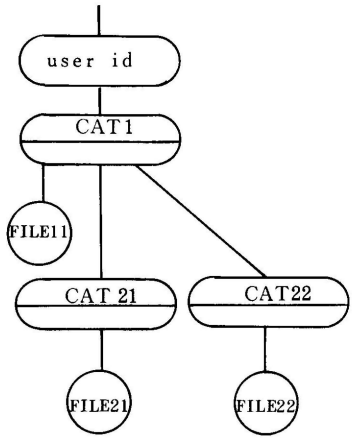
(b) FIRST／name／

nameで指定したところからカタログのリストを取り始めることができる。

(4) 使 用 例

```
SYSTEM?  ACCESS (SD)
FUNCTION? LC, user id (SD)
  LIST OF CATALOG - user id
  CATALOGS
    FILES
  CAT1
    FILE11
  CAT21
    FILE21
```

```
    CAT22
      FILE22
  FUNCTION? LC, user id／CAT1 (SD)
    LIST OF CATALOG - CAT 1
    CATALOGS
      FILES
      FILE11
    CAT21
      FILE21
    CAT22
      FILE22
  FUNCTION? (SD)
  SYSTEM?  (next subsystem)またはBYE
```



5.5.2 カタログやファイル記述のリスト（タイムシェアリング）

(1) 形　　式

ACCESS	{ LIST SPECIFIC / LS } ,	修 飾 名

(2) 機　能　　指定されたカタログまたはファイル記述を端末上に表示する。

(3) パラメータの説明

修飾名……修飾カタログ名または修飾ファイル名を指定する。

(4) 使 用 例

```
SYSTEM? ACCESS (SD)
FUNCTION? LS, user id／CAT1 (SD)
  CATALOG NAME - CAT1
  ORIGINATOR - user id
  DATE CREATED - 12 10 77
  DEVICE - ST1
  GENERAL PERMISSIONS - NONE
  SPECIFIC PERMISSIONS - NONE
FUNCTION? LS, user id／CAT 1／FILE11 (SD)
  FILENAME - FILE11
  ORIGINATOR - user id
  DATE CREATED - 12 10 77
  DATE CHANGED - 12 12 77 ( 15.750 )
  LAST DATE ACCESSED - 12 15 77
  NUMBER OF ACCESSES - 9
  MAX FILE SIZE - 20 LLINKS
  CURRENT FILE SIZE - 2 LLINKS
  FILE TYPE - LINKED
  DEVICE - ST 2
  GERERAL PERMISSIONS - NONE
  SPECIFIC PERMISSIONS - NONE
FUNCTION? (SD)
SYSTEM? (next subsystem ) またはBYE
```

5.5.3 カタログやファイル記述のリスト（バッチ）

(1) 形　　式

FILSYS	{CLIST / LIST} ␣	修飾名〔，リスト指定〕〔，エクステント指定〕

(2) 機　能　　指定されたカタログまたはファイル記述を印刷する。

(3) パラメータの説明

① 修飾名……修飾カタログ名または修飾ファイル名を指定する。

② リスト指定

LISTOPT／{ALL / ONLY / BUSY} ／ 指定がない場合の既定値はALLである。

ALL　…… 指定されたカタログに属するすべてのカタログやファイルが印刷される。

ONLY …… 指定されたカタログ直下のカタログやファイルのみが印刷される。

BUSY …… ビジーファイルだけがリストされる。

③ エクステント指定

EXTENTS／{NORM / ALL / NONE} ／ 指定がない場合の既定値はNORMである。

NORM …… ファイルのエクステントは印刷される。

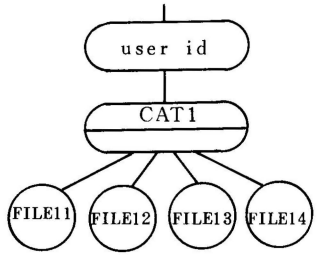
ALL　…… すべてのファイルのエクステントが印刷される。

NONE …… ファイルのエクステントが印刷されない。

(4) 使 用 例

欄　1　　8

```
$          FILSYS
LIST       user id
```



(5) 出力例と説明（→図5.1）

① ディレクティブ：カード番号とこのリストを作ったディレクティブ

② 名前：ファイルまたはカタログ名

③ 一般パーミッション：パーミッションの省略形

④ 装置タイプ，名前：カタログについては，それを格納している装置またはパック。ファイルについては，それが割当てられている最初の装置（ファイル記述は，常にそのファイルが属しているカタログと同じ装置またはパック上にある）

⑤ 作成日：ファイルまたはカタログが作成された日付（mmddyy）

⑥ 最新変更日付：ファイル記述あるいはカタログが最後に変更された日付（mmddyy）

⑦ 最後の割当て日付：ファイルが最後に割当てられた日付（mmddyy）

⑧ 最新更新日付：ファイルの内容が最後に更新された日付（mmddyy）

⑨ 最新更新 時刻：ファイルに対して書込みを行なった最後のジョブがファイルを解放した時刻。
形式はhh．hhh

⑩ 最大Lリンク数：ファイルの拡張可能限界値。Lリンク数。

⑪ 現在のLリンク数：現在ファイルに割当てられてるLリンク数。

⑫ 最新退避ボリューム番号：ファイルの最新の退避の第1ボリュームのテープシリアル番号。

⑬ 退避以降更新インディケータ：N＝最後の退避以降ファイル内容は更新されていない。

Y＝最後の退避以降，更新が行われた。W ＝書き込み中に退避が行われた。

⑭ 割当てカウント：ファイルの割当て回数（最大 262144 ）。ペリフェラル・アロケータは，全資源の割当てが成功するまで，何回か一つのファイルを割当てたり解放したりすることに注意されたい。

⑮ ファイル状態：NULL— そのファイルが作成されて以来一度も書込まれていない。

⑯ ファイル・タイプ：指定されたファイル・タイプを示す略号

SEQ —順アクセスまたは直接アクセスが許されているファイル

RAND —直接アクセスのみが許されているファイル

⑰ 特定パーミッション

```
FLSYS 01  03-10-78   19.287                R 3.3      F I L S Y S    A C T I V I T Y                PAGE      1
                     1①  LIST    user - id                                                  PRIOR STATUS ON PG
 ②                              ③      ④        ⑤      ⑥       ⑦        ⑧     ⑨     ⑩    ⑪     ⑫      ⑬    ⑭   ⑮    ⑯
   CATALOG NAME                GEN'L DEVICE   CREATE      DATE OF LAST     TIME   MAX CURRENT LAST SAVE ALLOC  FILE
      FILE NAME                PERMS TYP:NAME  DATE   MODIFY ALLOC  CHANGE      LLINKS LLINKS VOL-CHG? CNT  STAT TYPE
user - id
   CAT1                              450:ST1  120677 120677
      FILE11                         450:ST1  120677 012478
            ⑰                   R    450:ST2  120677 012478 012478 120677 12.700  240    240   99999N      9        RAND
            PROTECTION:  ACCESS/CONC/
      FILE12                         450:ST2  012378 012378 012578 012378 21.552   20     1    99999N      4        SEQ
      FILE13                         450:ST2  012378 012378 012378 012378 21.552   20    20    99999N      0  NULL  SEQ
      FILE14                         450:ST2  012478 012478 013078 012478 19.889   20     1    99999N      6        SEQ
```

図5.1　ファイル・システム出力例