

47-S003

遠隔情報処理システムの応用実験

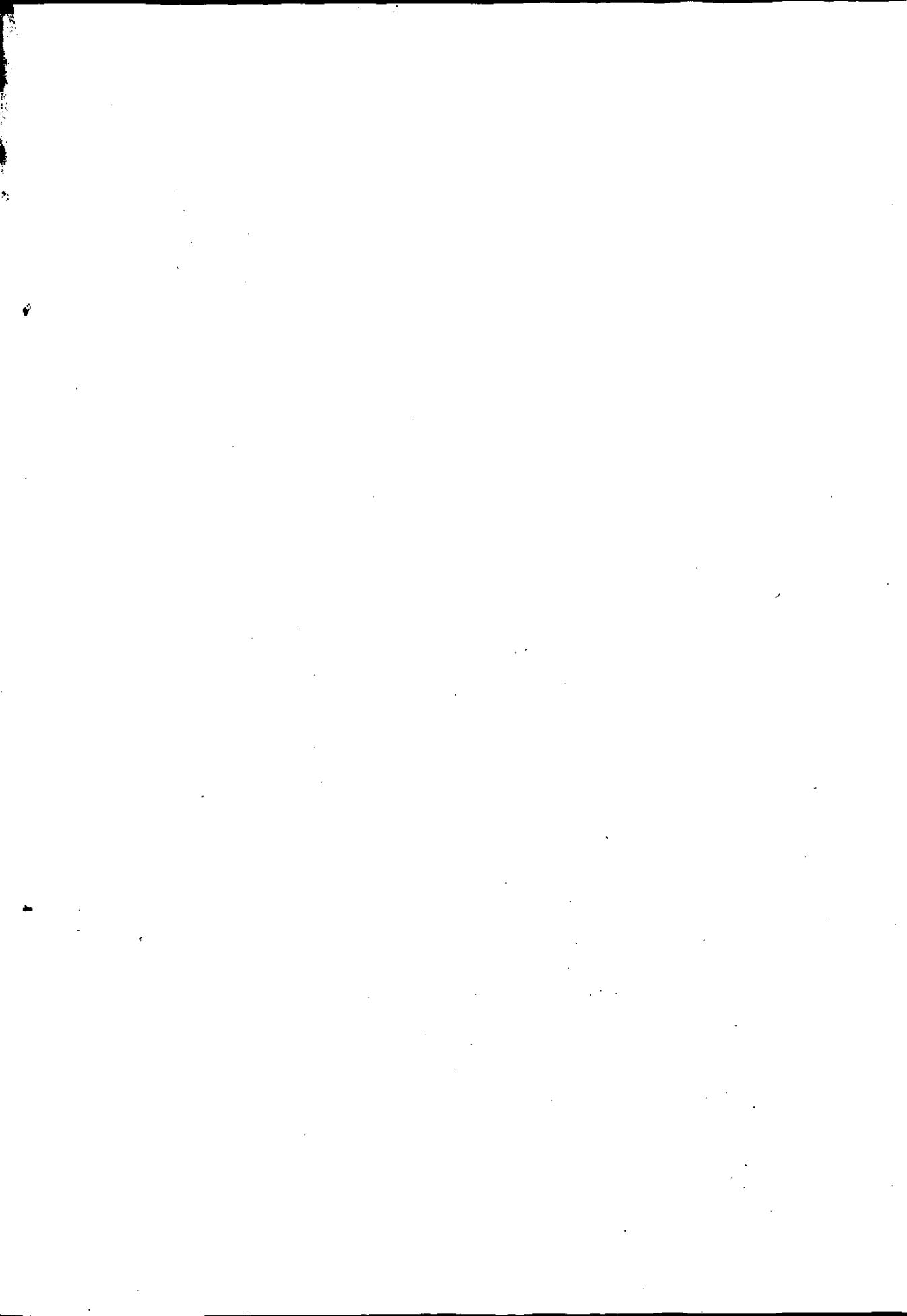
—遠隔情報処理システムの研究開発—

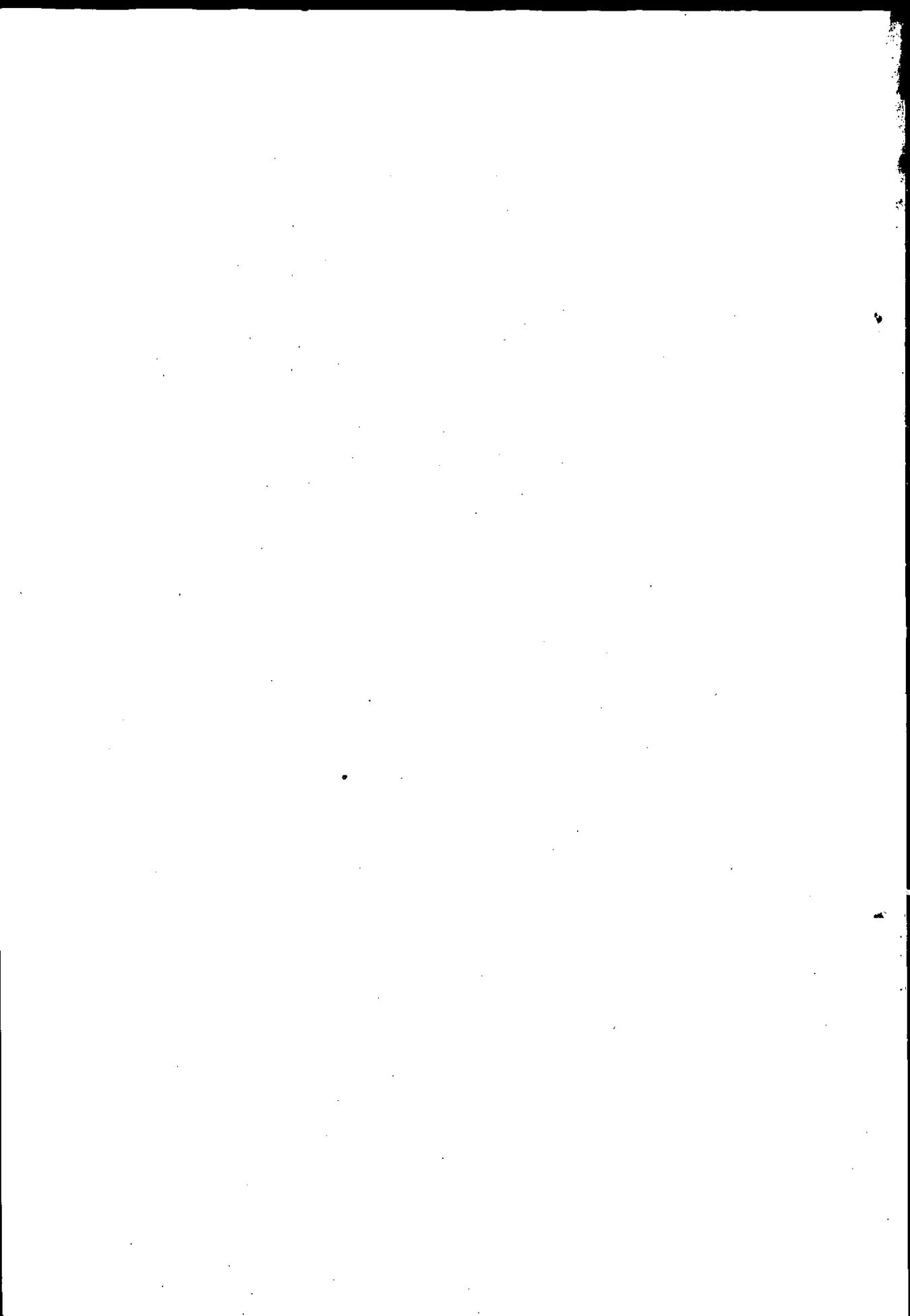
昭和 48 年 3 月



財団法人 日本情報処理開発センター

この報告書は、日本自転車振興会から競輪収益の一部である機械
工業振興資金の補助を受けて昭和47年度に実施した「遠隔情報処
理システムの研究開発」の一環としてとりまとめたものであります。





序

通信回線と電子計算機を結んで行なう遠隔情報処理システムは、将来非常に大きく発展する分野と考えられておりますが、現在のところ未解決の多くの技術的問題を含んでおります。

当財団では昨年に引き続き、この部門において最も重要と考えられるタイム・シェアリング・システムについて基本的な応用実験を行ない、その結果をここに報告書としてとりまとめた次第であります。

この報告書に記載された研究の実施については、下記の機関の協力によって行なわれたものであり、ここに深く感謝の意を表するとともに、本研究が社会に寄与することを念願しております。

昭和48年3月

財団法人 日本情報処理開発センター

会長 難波捷吾

協力団体： 通商産業省

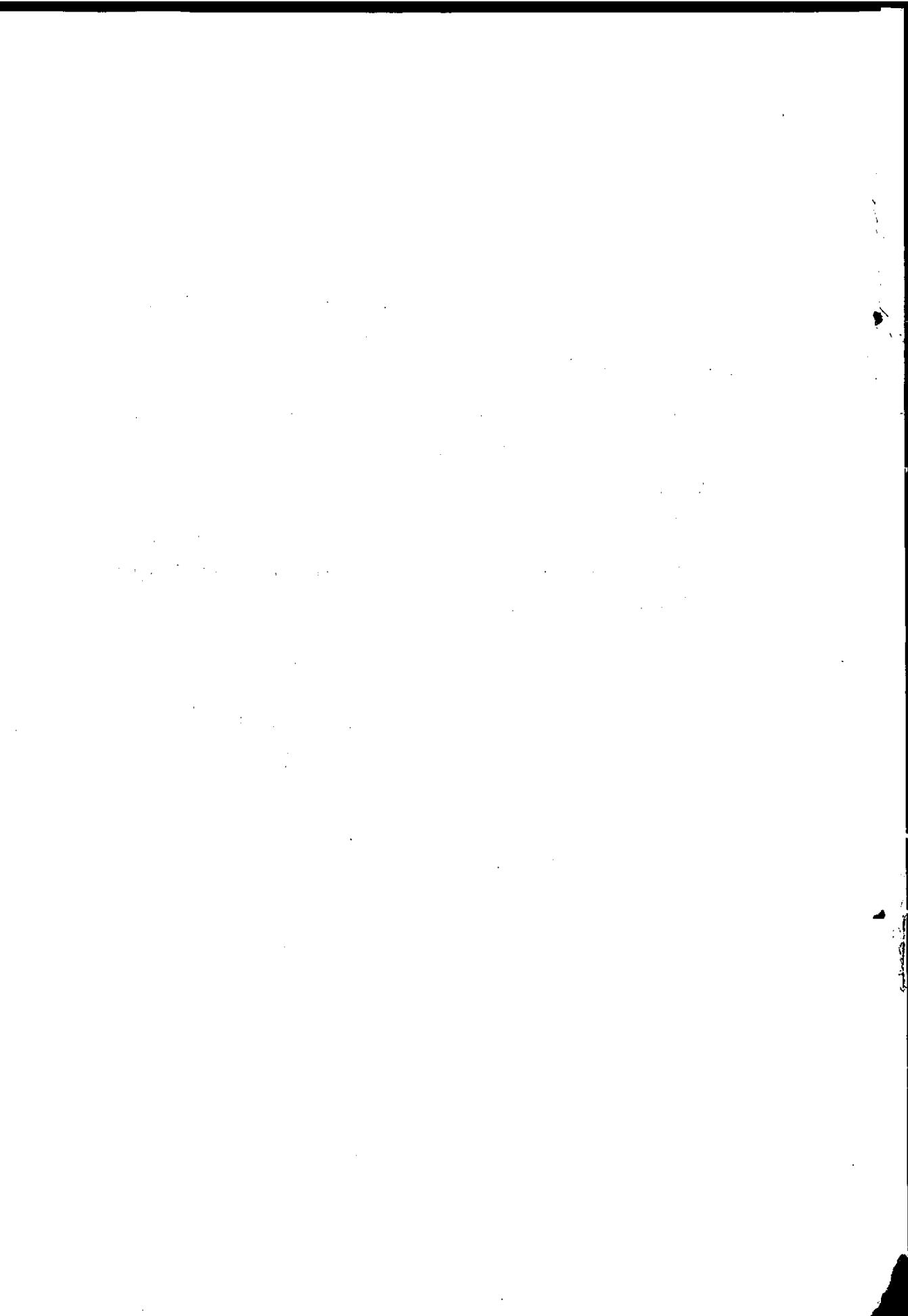
(順不同) 郵政省

日本電信電話公社

(財) 機械振興協会

通商産業省・工業技術院

財東京繊維情報センター



は　じ　め　に

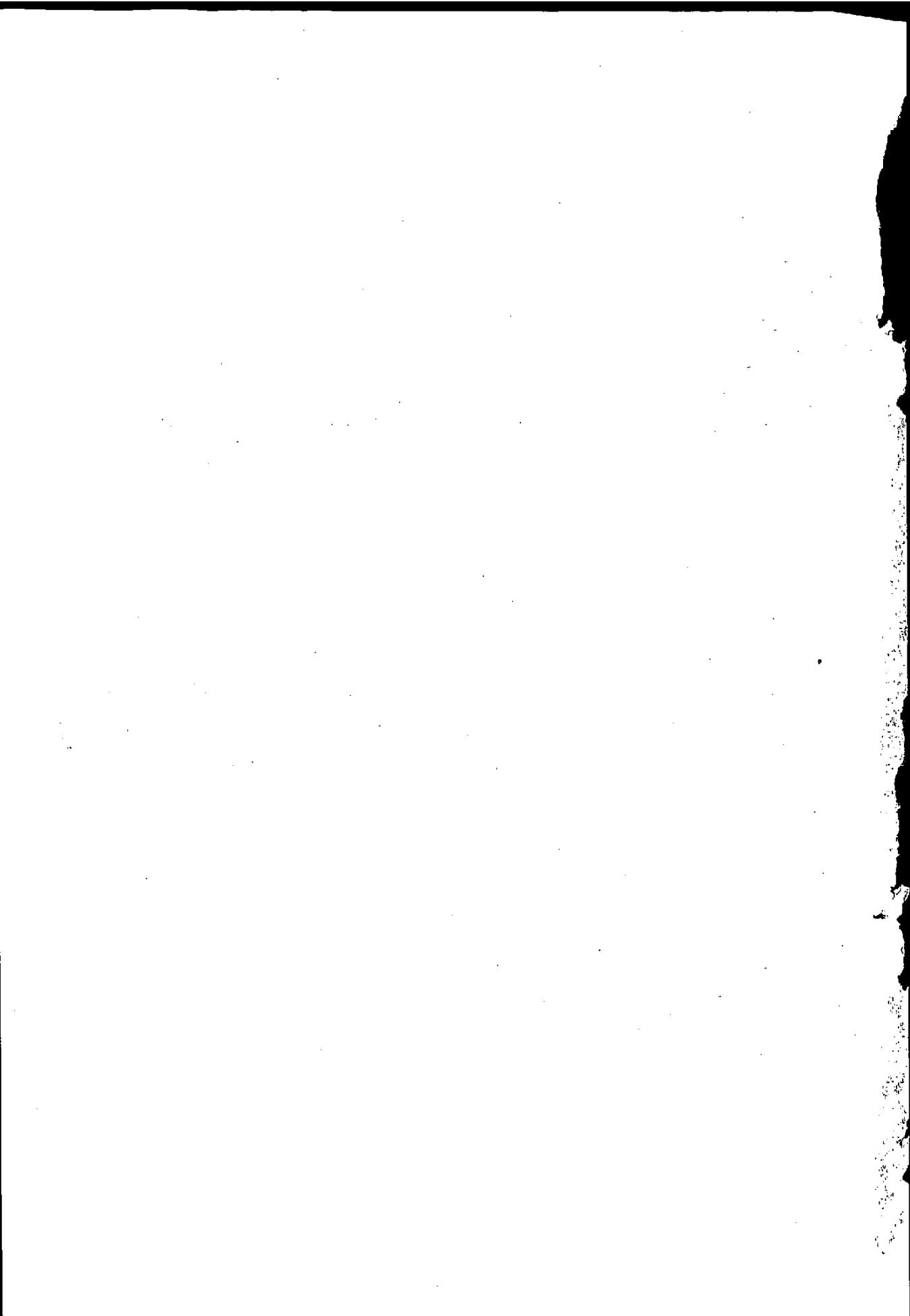
本報告書は、遠隔情報処理システムの研究開発の一環として、タイム・シェアリングを対象にし、その研究成果を骨子に述べたものである。

このタイム・シェアリング・システムの応用試験は、昭和44年10月から約2年半NEAC 2200-500/200のタイム・シェアリング・システム(MACシステム)で行ない、かなりの実績をあげたが、より規模の大きいシステムの要求がでたため、昭和47年2月下旬からFACOM 230-60 MONITOR-Vによるタイム・シェアリング・システムに変更されたものである。

MONITOR-Vシステムは、現在実用化されているオペレーティング・システムでは最大規模に属するもので、バッチ処理、リモート・バッチ処理、リアルタイム処理、デマンド処理が同時に可能な4次元のオペレーティング・システムであるが、本応用実験はリアルタイム処理を除いた3元処理で行なった。

リモート・バッチ処理方式については、昭和46年から実験が開始されているが、デマンド処理も含めた3元処理による本格的な応用実験は今年度が初めてである。

本報告書は、この応用実験から得られた運用方式、ソフトウェア、各端末の利用状況とその評価、また、センターの効率的運用を図るための端末ユーザの調査などが述べられている。これらの内容が今後のタイム・シェアリング・システムに対していくらかでも参考になれば幸いである。



目 次

1. 応用実験の概要	1
2. システムの概要	3
2.1 ハードウェアの構成	3
2.2 ソフトウェアの構成	9
2.3 端末装置	13
2.3.1 F1591A/F1520M端末装置	13
2.3.2 F1592A端末装置	14
2.3.3 FACOM-R/FACOM-RE端末装置	15
3. タイム・シェアリング・システムの運用	17
3.1 運用方式	17
3.1.1 クローズ制	17
3.1.2 障害対策	17
3.1.3 ファイル割付け	33
3.1.4 サービス体制	36
3.1.5 センター標準値	42
3.1.6 ジョブ制御マクロ	46
3.1.7 ファイル・リカバリー	46
3.1.8 ファイル保護	49
3.2 計算センター・ルーチン	51
3.2.1 計算センター・ルーチンの概要	51
3.2.2 計算センタールー・ルーチンとして 制御プログラムに追加した機能と目的	56

3.2.3	処理プログラム	64
3.3	安定性	66
4. デマンド処理		69
4.1	利用状況	69
4.2	処理プログラムについて	74
4.2.1	言語仕様	74
4.2.2	コマンド	86
4.2.3	エラー検出機能とメッセージ	96
4.3	各端末の利用状況	98
4.3.1	通商産業省における利用状況	98
4.3.2	郵政省における利用状況	103
4.3.3	日本電信電話公社における利用状況	112
4.3.4	(財)機械振興協会における利用状況	128
4.3.5	当財団における利用状況	139
4.4	端末ユーザの要望に対する検討	146
5. リモート・バッチ処理		149
5.1	概要	149
5.2	利用状況	150
5.3	機械技術研究所におけるリモート・バッチ処理方式の利用	152
5.3.1	利用の目的と経緯	152
5.3.2	リモート・バッチ方式採用の理由	159
5.3.3	機械研のリモート・ターミナルの構成	163
5.3.4	旋削用自動プログラミング・システムMELTS	164
5.3.5	インプリメンテーション	176
5.3.6	リモート・ターミナルの利用結果	181
5.3.7	リモート・バッチ処理方式を利用して	187

5.3.8 むすび	189
6. MONITOR-Vについて	191
7. 調査	201
7.1 調査Ⅰ	202
7.1.1 調査目的	202
7.1.2 端末装置導入目的とその効果	203
7.1.3 安定性	204
7.1.4 ソフトウェア	205
7.1.5 端末装置の操作性	207
7.1.6 センター側に対する要望	208
7.1.7 その他	208
7.2 調査Ⅱ	210
7.2.1 調査目的	210
7.2.2 調査対象センターの性格	211
7.2.3 主な業務の概要	211
7.2.4 コンピュータの利用方法の検討	215
8. プログラム開発	221
8.1 C P L用科学計算ライブラリの開発	222
8.1.1 ライブラリの特徴	222
8.1.2 各ライブラリの説明	226
8.2 TSS-FORTRAN YREPLACEサブコマンドの新設	301
8.2.1 概説	301
8.2.2 詳説	302
8.3 JOLDOR-IIの開発	307
8.3.1 概要	307

8.3.2 オンライン文献検索の持つ意味	308
8.3.3 検索に対する JOLDOR-II の姿勢	309
8.3.4 情報蓄積経路	310
8.3.5 情報検索経路	311
8.3.6 システムの概要	312
8.3.7 今後の問題	331
8.3.8 JOLDOR-II システムの検討および反省	333
8.3.9 おわりに	339
付録-I FACOM利用の手引書	341
付録-II ジョブ制御マクロ登録集	359
付録-III 会計情報処理ルーチンの出力例	397

1. 応用実験の概要



1. 応用実験の概要

遠隔情報処理システムは電子計算機と電気通信設備の通信回線による結合を意味するものである。即ち、電子計算機に接続された回線制御装置によって通信系と直結している体系をさし、遠隔地の利用者が端末を介して中央の電子計算機を操作できるシステムである。

遠隔情報処理システムが要請される背景には種々考えられるが、その最大の要因は経済社会の発展に伴って増大する情報を迅速、正確かつ高度に処理することを要求する情報化社会への推移がある。そこには情報の供給・加工・蓄積に対する技術的な裏付けが必要とされ、ハードウェア・ソフトウェア・通信の三位一体の技術開発が前提とされる。

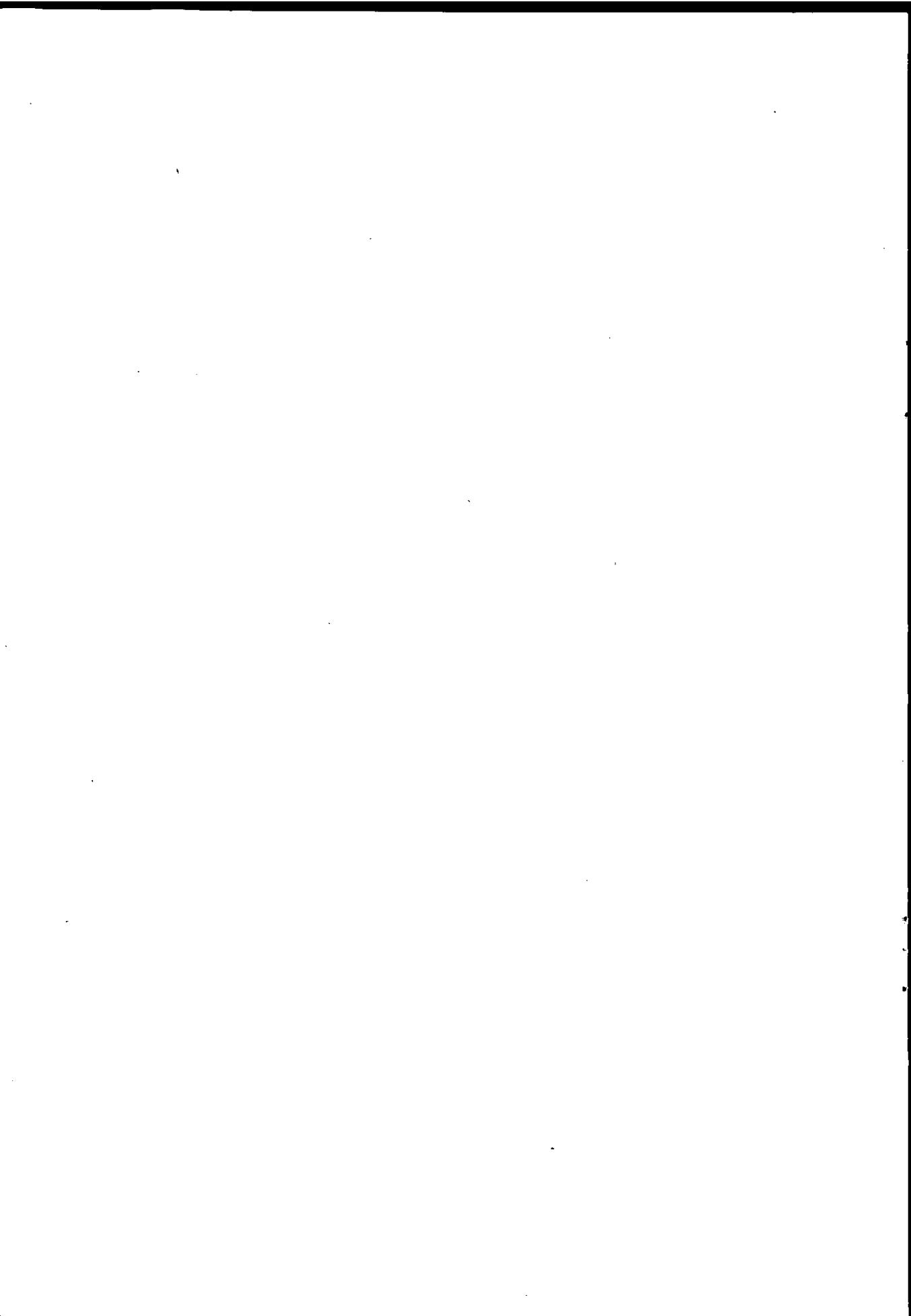
本応用実験は、遠隔情報処理方式の適用分野の拡大ならびにその技術的向上に資するため、当財団設置のコンピュータ FACOM 230-60 およびその端末機群を使用して、オンラインによるアプリケーション・プログラムの開発、およびオンライン・システムの応用実験を行ない、TSS の利用とともにハードウェア、オペレーティング・システムあるいは、TSS 運用上の問題点等について端末利用者よりデータを収集し、問題点を把握し、その解決をはかるとともに共同利用の効率経済性等について追求するものである。

なお、このシステムに接続されている端末は、計12台で設置場所は次のとおりである。

通商産業省重工業局情報処理振興課	F1591A	1台
郵政省電気通信監理官室	"	1台
日本電信電話公社技術局調査部門	"	1台
(財)機械振興協会経済研究所	"	1台
通商産業省・工業技術院機械技術研究所	FACOM-R E	1台
㈱東京繊維情報センター	FACOM-R	1台

当財團	F 1591A	1 台
	F 1592A	1 台
	F 1520M	1 台
	FACOM-R	3 台

2. システムの概要



2. システムの概要

2.1 ハードウェアの構成

応用実験の対象となった FACOM 230-60 の構成については図 2-1 の通りであるが、当該コンピュータは、1語42ビット（フラグ部4ビット、データ部36ビットおよびそれぞれの奇偶検査用の2ビット）で構成されるワードマシンであり、1命令当たりの平均実行時間は、Gibson Mixの場合約 $1.6\mu s$ となっている。

主な装置の性能については次のとおりである。

・中央処理装置

命令の種類		演算時間(μs)	精度
固定小数点	加減算	1.26	36ビット (10進10.5桁相当)
	乗算	4.06	
	除算	10.18	
1倍精度 浮動小数点	加減算	2.27	仮数部 27ビット 指数部 9ビット (10進換算における範囲は精度8桁で $10^{-77} \sim 10^{76}$)
	乗算	3.68	
	除算	5.12	
2倍精度 浮動小数点	加減算	2.27	仮数部 62ビット 指数部 9ビット (10進換算における範囲は精度18桁で $10^{-77} \sim 10^{76}$)
	乗算	6.20	
	除算	10.02	

・記憶装置

高速磁心記憶装置 (HCM)

容量 192 kW (256 kWまで拡張可能)

サイクルタイム $0.92\mu s/2W$

大容量磁心記憶装置 (LCM)

容量 256 kW (768 kWまで拡張可能)

サイクルタイム $6.0\mu s/2W$

• 補助記憶装置

高速磁気ドラム装置	1台
記憶容量	3MB/台
アクセスタイム	10ms
情報転送速度	0.312MB/S
磁気ドラム装置	3台
記憶容量	2.56MB/台
アクセスタイム	17ms
情報転送速度	0.156MB/S

集合磁気ディスクパック装置 2セット(1セット9台, 内1台は予備)

記憶容量	29.17MB/台
アクセスタイム	75ms
情報転送速度	260KB/S
磁気テープ装置	7台(9トラック6台, 7トラック1台)
情報転送速度	60KC/S

• データチャネル装置

チャネル制御装置	
最高転送速度	500,000 2W/S(高速磁心記憶装置との転送)
セレクタ・チャネル装置	
最高転送速度	800,000C/S(入出力装置との転送)
マルチプレクサ・チャネル装置	
最高転送速度	
バーストモード	400,000C/S(入出力装置との転送)
マルチプレクサモード	100,000C/S(入出力装置との転送)

• 通信制御装置

収容回線容量	50 bit/s - 60回線	速度混在を行なわない場合 の各速度最大収容回線数を 示す。
	200 bit/s - 30回線	
	1,200 bit/s - 5回線	

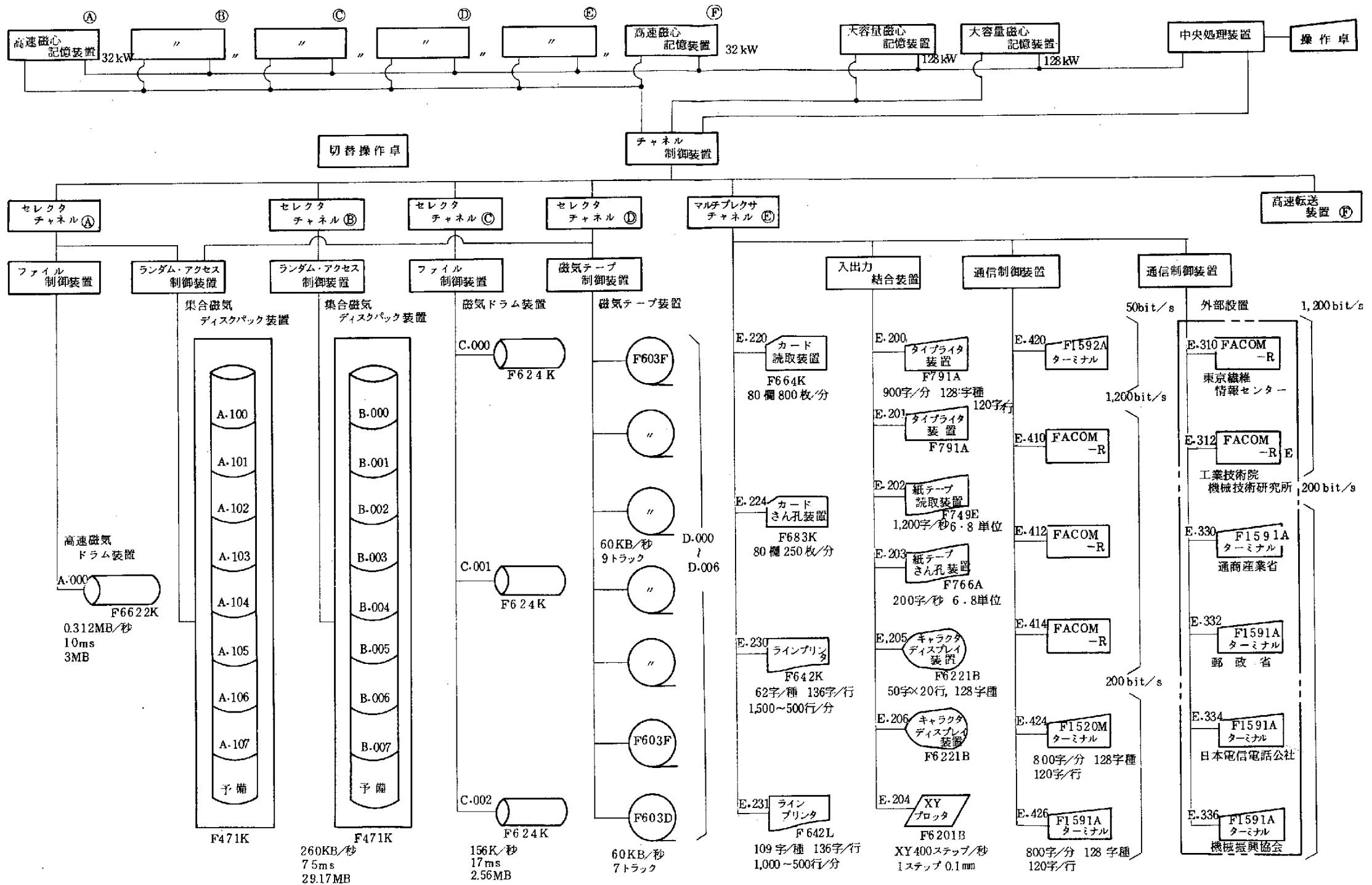
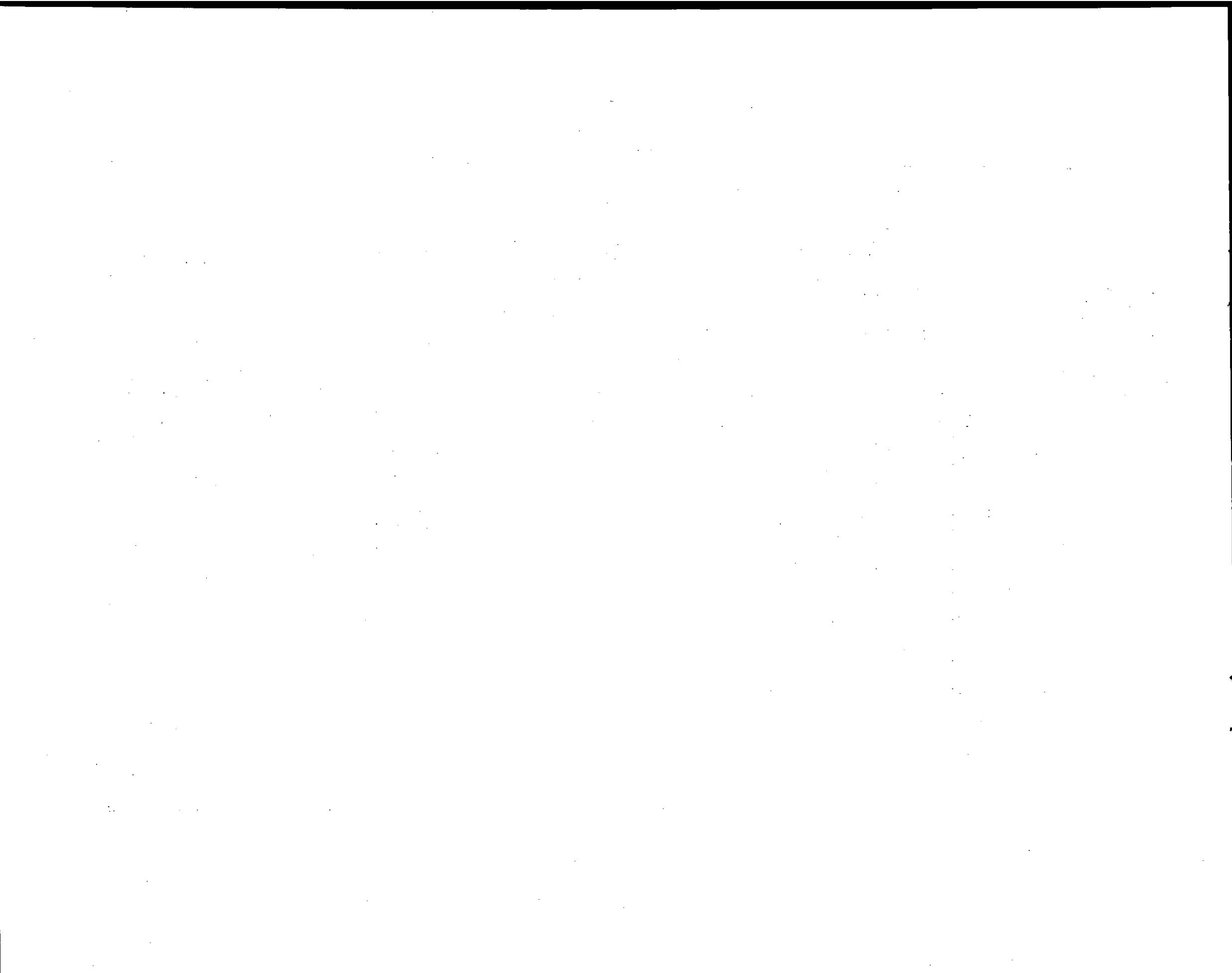


図 2-1 FACOM 230-60 機器構成図



通信速度	50 200 1,200 bit/s
通信方式	全二重通信方式, 半二重通信方式
単位数	6 ~ 8 単位(データ信号ビットを示し, スタート, ストップビットは含まない)
符号系	任意の伝送符号, ISO 標準符号
同期方式	調歩同期方式
誤り検出機能	垂直水平パリティチェック, 調歩監視, 文字シーケンス監視

次に, システム構成上の特徴を述べると, 次のような点があげられよう。

1. 単一中央処理装置のシンプレックス・システムである。
2. コンソール・タイプライタ F791A が故障すると, システムダウンとなってしまうので, コンソール・タイプライタのマルチ化が図られている。
3. セレクタチャネル#Aには, 後述のとおり, TSS 用ファイルなどの重要なファイルが接続されているので, セレクタチャネル#Aが故障した時の対策として, セレクタチャネル#Dとチャネル間のクロスコールを行なっている。
4. 記憶装置を高速(高速磁心記憶装置)と低速(大容量磁心記憶装置)との2種類に分け, 大容量磁心記憶装置を Virtual Memory として使用している。
5. モニタを磁気ドラム装置 C.000 に格納しているが, この装置のダウン対策として, 磁気ドラム装置の C.002 を代替用装置として常に同じレベルのモニタを格納している。

なお、リモートバッチ用端末として接続されているミニ・コンピュータの
FACOM-RおよびFACOM-REの機器構成は図2-2のとおりである。

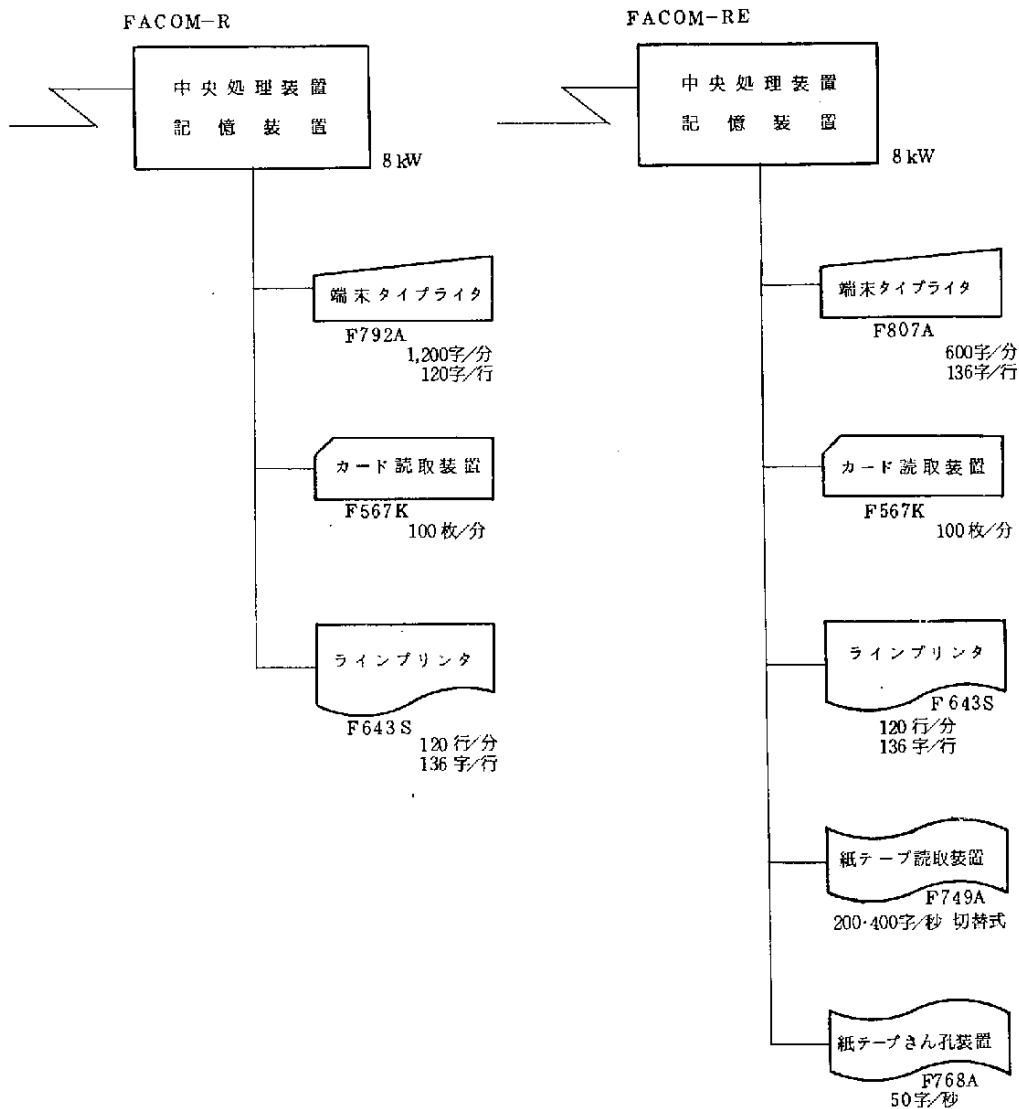
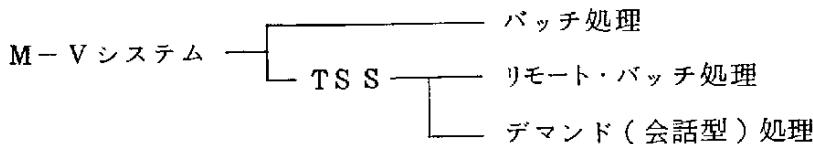


図2-2 FACOM-R・FACOM-RE機器構成図

2.2 ソフトウェアの構成

FACOM 230-60 のオペレーティング・システムは、 MONITOR-V システムと呼ばれ、今までのオペレーティング・システムのようにバッチ処理専用、リアルタイム処理専用といったそれぞれの応用分野に固有なものでなく、また、バッチ処理のみのオペレーティング・システムであったものを後で改良して、比較的限られた範囲の TSS 機能を追加して作りあげたものでもない。

すなわち、制御プログラムがもつべき基本機能を、すべての分野に共通して応用したいわゆる汎用のオペレーティング・システムであり、バッチ処理、リモート・バッチ処理、リアルタイム処理、およびデマンド処理のいずれにも適用できるよう最初から設計されている。当実験では、次の図のように 3 種の処理機能を同時使用した。



この 4 次元処理のオペレーティング・システム MONITOR-V システムは、現在実用化されている国産コンピュータの中では、最大規模のオペレーティングシステムの 1 つであり、制御プログラムと処理プログラムに分けられ、図 2-3 のような構成になっている。

制御プログラムが受け持つエリアコントロールは、デマンド処理、バッチ処理、リモートバッチ処理の区別なくベースレジスタによるダイナミックリレーション機能を持っている。スケジューリングはジョブ優先権、実行優先権、主記憶駐在優先権による複合コントロール方式で、0~7 または 0~15 のレベルを持ち、システム・ジェネレーション時に各優先権を各処理形態との関連を考慮して指定する。また、この値は、実行時に各ジョブのコントロール・パラメータによって変更が可能となっている。ファイル形式は順編成、分割型順編成、順インデックス編成、直接編成をデータ管理で取扱う。また、ファイルの扱いに関してはドラム、ディスク、磁気テープ、ラインプリンタ、カードリ

ーダなどの間でデバイス・インディペンデントに処理することができる。

処理プログラムとしては、デマンド処理用言語として JOSS 型の BACCUS, エディティング専用の LINED がメーカから提供されている。

この BACCUS は、最初から科学計算向きにデマンド処理用として開発された会話型言語で、プロセッサもインタプリティブな処理を行なっている。

この他、当財団で開発したデマンド処理用言語として、TSS-FORTRAN および会話型 PL/I の CPL がある。

TSS-FORTRAN については、従来 FORTRAN を使用してきたユーザの FORTRAN に対する要求が根強いため、バッチ用 FORTRAN に端末入出力、フリーフォーマットによる入力、ステートメント単位のシンタックスチェック、および簡単なエディティング機能を追加し、デマンド型 FORTRAN として使用できるようにした。

CPL については、FACOM の TSS における分担が技術計算もさることながら、データ処理あるいは、システム記述的要素をもつアプリケーションのために役立たせようというねらいから、会話型 PL/I が開発された。しかし、バッチの PL/I 同様、予想以上にプロセッサが大きくなつたため、高速磁心記憶装置にプロセッサを駐在させるのは、現在の容量ではやや無理が生じるので、大容量磁心記憶装置に駐在させている。このシステムは、リエントラントに作られているが、大容量磁心記憶装置ではベースレジスタの数の制限から、残念ながらリエントラント機能が使用制限となっている。

その他、特殊アプリケーションとしてオンライン文献検索システム JOLDO R, オンライン・シミュレーション・システム SIMBOL, および CPL の自動学習システム CLASS を当財団で開発した。しかし、これらのシステムはディスプレイ端末を使用するように設計されているため、現在のところセンター内のみで使用している。

デマンド処理用言語について要約すると、実際にユーザにサービスしているものは次のとおりである。

- デマンド処理プログラム
- LINED(Lined Data Set Editor)
ファイルの作成, 追加, 変更を端末から会話形式で行なうプログラム。
 - BACCUSインタプリンタ (Basic Calculus)
科学計算向き会話型プログラム言語であり, 会話形式でプログラムの作成や実行ができる。
 - CPL (Conversational PL/I)
会話形式のPL/I。
 - TSS-FORTRAN
会話形式のFORTRAN

リモート・バッチ処理としては, 端末からシステム入力情報を一括してシステムに送り, いったん大記憶に蓄え, それ以降はバッチ処理として処理される。従って, 処理言語としては, バッチ処理で扱う処理言語と同じと考えてよく, 特別な処理言語はない。

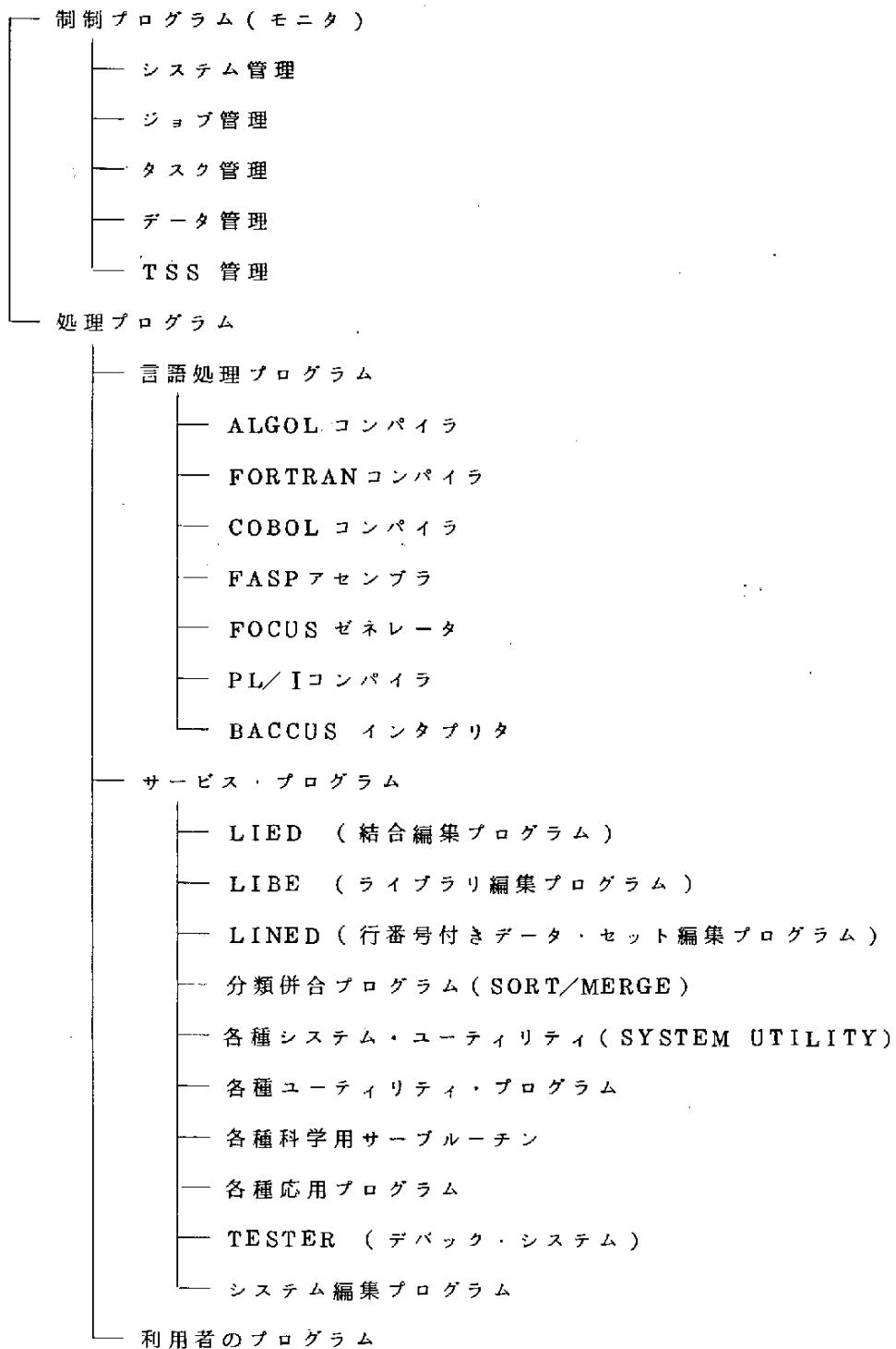


図2-3 FACOM 230-60 MONITOR-V システム

2.3 端末装置

現在、端末装置としてF1591A, F1592A, F1520M, 8K語(17bit/w)のメモリとカードリーダ, ラインプリンタ, タイプライタ, 紙テープ・リーダ, 紙テープ・パンチからなるミニコンピュータFACOM-RおよびFACOM-REの5種類が接続されている。

この5種類の端末はデマンド処理, リモート・バッチ処理の両方が処理できるように工夫されているが, F1591A, F1592AおよびF1520Mの3種類は, 主にデマンド処理用として使用し, FACOM-RおよびFACOM-REは, 主にリモート・バッチ処理用として使用している。

各端末装置の概要は次のとおりである。

2.3.1 F1591A/F1520M端末装置

F1591AおよびF1520Mは200BPS用の端末装置である。この端末装置は, タイプライタ部, 紙テープ読取部, 紙テープさん孔部から構成されている。

この2つの端末装置には操作上いくぶん相違があるが, 装置そのものにはあまり相違はみられない。

(1) 規 格

適用回線	普通第2規格回線(専用回線)
通信方式	半二重通信方式
同期方式	調歩同期方式
通信速度	200bit/s
使用コード	ISO 7ビットコード+パリティビット
通信コード	会話通信
電 源	AC 100V 50または60Hz
誤り制御	垂直・水平パリティ・チェック 手動再送訂正→F1591Aの場合 自動再送訂正→F1520Mの場合
伝送方 式	ブロック伝送(プリンタの1行印字ごとにブロック構成)

→ F1591Aの場合

バッファ・メモリによる定ブロック伝送

→ F1520Mの場合

(2) けん盤部

- ① 鉄配列 JIS 事務機械配列
- ② 打鍵入力字数 英字, 数字, カナ, 記号, 計128字
復・改, スペース, 後退, エラー, 抹消, TAB, STX,
STB, ETX, EOT。
- ③ 打鍵速度 800字/分
- ④ シフト シフト鉄打鍵は必ならずシフト・コードを発生, プリンタのシフト状態と連動する。

(3) プリンタ部

- ① 印字数 SI側 64字, SO側 64字, 計128字。
- ② 1行印字数 最大120字/行
- ③ 印字速度 20字/秒

(4) 紙テープリーダ・パンチ部

- ① 入出力テープ 8単位紙テープ
- ② 入出力速度 1,200字/分
- ③ テープ・アウト あり
- ④ テープ・エンド なし
- ⑤ フィード・チェック なし

2.3.2 F1592A端末装置

F1592Aは50BPS用の端末装置である。

この端末装置は、タイプライタ部, 紙テープ読取部, 紙テープさん孔部から構成されている。

(1) 規格

通信方式 半二重通信方式

同期方式 調歩同期方式
通信速度 50 bit/s
電源 AC 100V 50または60Hz

(2) けん盤部

① 打鍵入力字数 英字, 数字, カナ, 記号 計 126 字
上段, 中段, 下段, 改行, 復帰, 後退, 訂正, 消去,
スペース, 始, 終, 空白
② 打鍵速度 最大 700 字/分

(3) プリンタ部

① 印字数 126 字
② 1行印字数 最大 80 字/行
③ 印字速度 最大 375 字/分

(4) 紙テープリーダ・パンチ部

① 入出力テープ 6 単位紙テープ
② さん孔速度 最大 700 字/分(鍵盤さん孔時)

2.3.3 FACOM-R/FACOM-RE 端末装置

ミニコンピュータ FACOM-R および FACOM-RE は、1,200 BPS の通信回線で FACOM 230-60 と接続されており、先に述べたようにリモート・バッチ処理、デマンド処理のいずれの処理形態でも利用できる。

これは、ASC制御プログラム（通信制御プログラム）に「一括送信」と「会話送信」の2つの送信モードがあり、これらのモードの設定は操作指令により行なわれるが、一般にリモート・バッチ処理には一括送信モード、デマンド処理には会話送信モードが使用されることになる。

この端末装置は、データ送受信の入出力動作を効率よく働かせるように、リモートバッチ処理として使用するのが適当である。

この2つの端末装置は、機能的にはほとんど相違がない。

(1) システム構成

図 2-2 FACOM-R および FACOM-RE 機器構成図を参照されたい。

(2) 性能および規格

• 記憶装置

容 量	8 kW (最大 32 kWまで拡張可能)
語 長	16 ピット + 1 バリティ ピット
サイクルタイム	1.5 μs

• 回線制御アダプタ (ASA)

同期 方式	調歩同期
通信 方式	半二重通信方式
接続 方式	コンテンツ 方式
通信 速度	1,200 bit/s
伝送 コード	I S O
誤 制 御	垂直・水平バリティ 自動再送方式

• 直流電源装置

入力 電圧	100 V ± 10 V
周 波 数	50 または 60 Hz

• 入出力装置

タイプライタ装置	F 792 A	印字 1,200 字/分
カード読取装置	F 567 K	100 枚/分
ラインプリンタ装置	F 643 S	120 字/分・136 字/行
紙テープ読取装置	F 749 A	200・400 字/秒, 切替式
紙テープ穿孔装置	F 768 A	50 字/秒

3. タイム・シェアリング・システムの運用



3. タイム・シェアリング・システムの運用

3.1 運用方式

3.1.1 クローズ制

TSSサービス開始までの計算機利用方式は、オペレーティング・システムであるMONITOR-V システムがバッチ処理、リモート・バッチ処理、デマンド処理の同時利用、すなわち3元処理を行なうには十分な安定性を持っていなかったことから、オープン制を採用し、各プロジェクトが必要なだけ計算機時間を専有して、その範囲内で、センター内TSSおよびバッチ処理を組み合わせて利用したり、センター内TSSの時間を別に設けたりしてきた。

しかし、オープン制では、不特定多数の利用者がオペレーションを行なうため、オペレーションミスを起しやすく、それが原因でシステム・ダウンになることもあります。また、円滑な運用ができないことや計算機の効率的利用ができないなどの欠点もある。

そこでTSSサービスの開始に当っては、これらの欠点を改善するため必然的にクローズ制へと移行した。

オープン制からクローズ制への移行は、TSSサービス時間帯に合わせて段階的に行なっていった。

クローズ制実施に先立ち、クローズ制推進グループを設け、必要諸事項の立案および検討を行ない、「FACOM利用の手引書」を作成した。

また、利用者にはクローズ制の実施に当って、「FACOM利用の手引書」を基に説明会を開いて、利用方法についての徹底を図った。

「FACOM利用の手引書」の内容については巻末の付録-Iを参照されたい。

3.1.2 障害対策

オペレーションミスやハードウェア、ソフトウェア、回線などの障害が原因

となって、誤処理、中断、あるいはシステムダウンのような異常状態が発生する場合がある。

3元処理形態のシステムでは、このような異常状態の発生に対し、その原因を調査して適切な処置を行なう事により誤処理をなくし、システムの運用サービスの停止を最少限にとどめて、速やかに運転を再開するとともに、その再発を防止することが非常に重要な問題となる。

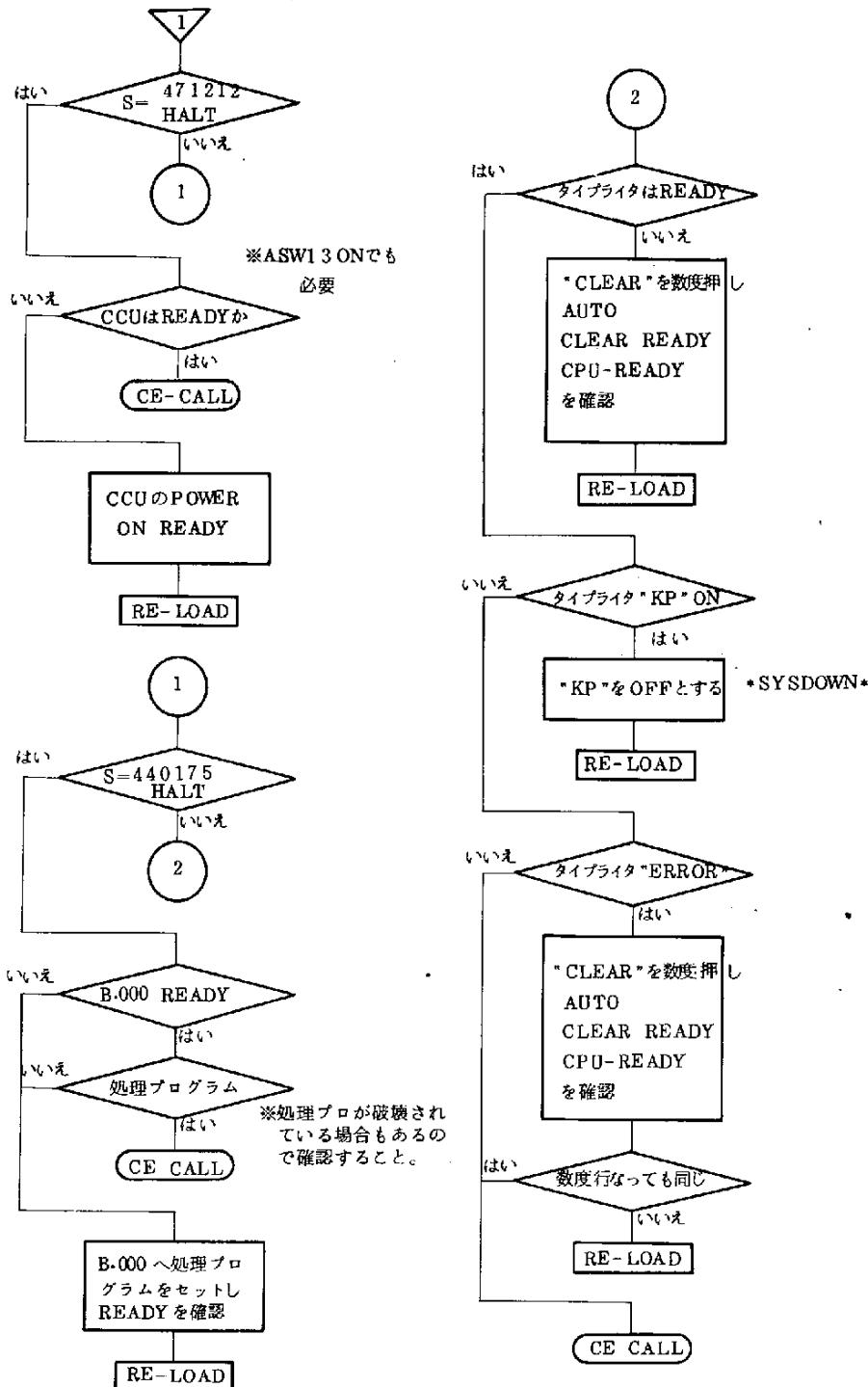
オペレーションミスを防ぐ方法としては、コンソール・タイプライタから必要以外の割り込みをさけることと、モニタからの返答要求メッセージの見落しといった初步的なミスをなくすよう操作員に注意を喚起すると同時に、物理的には注意を要するスイッチ、ボタンなどにプラスチックカバーを装着し、オペレーションミスによる障害を極力さけるようにした。

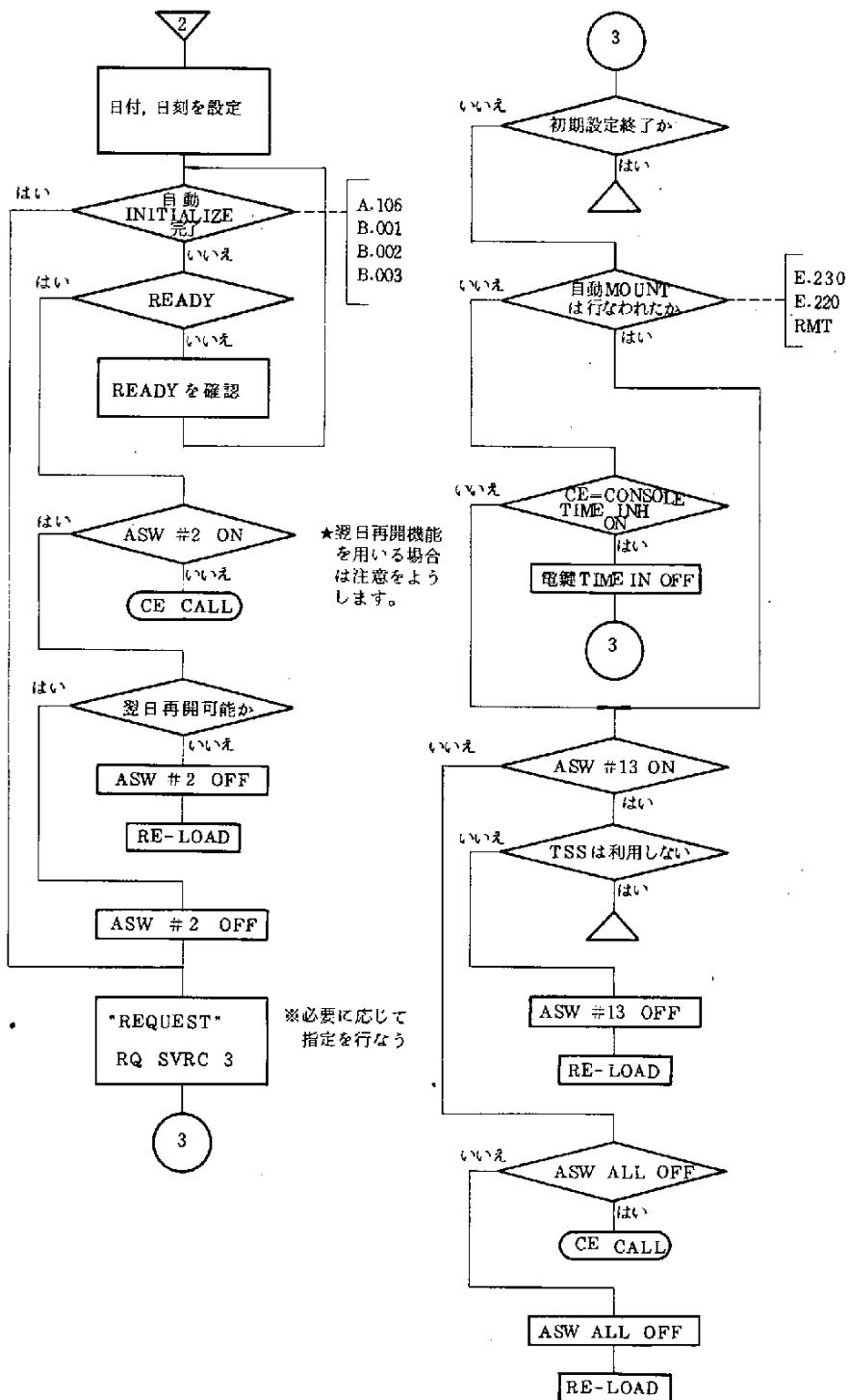
また、あらかじめ予測される異常状態を想定して、その原因調査手順をフローチャートにまとめ、いつでもオペレータが異常状態に対処できるようにした。

この操作員による調査で原因がつかめない場合は、S EまたはC Eに本格的調査を依頼することにした。

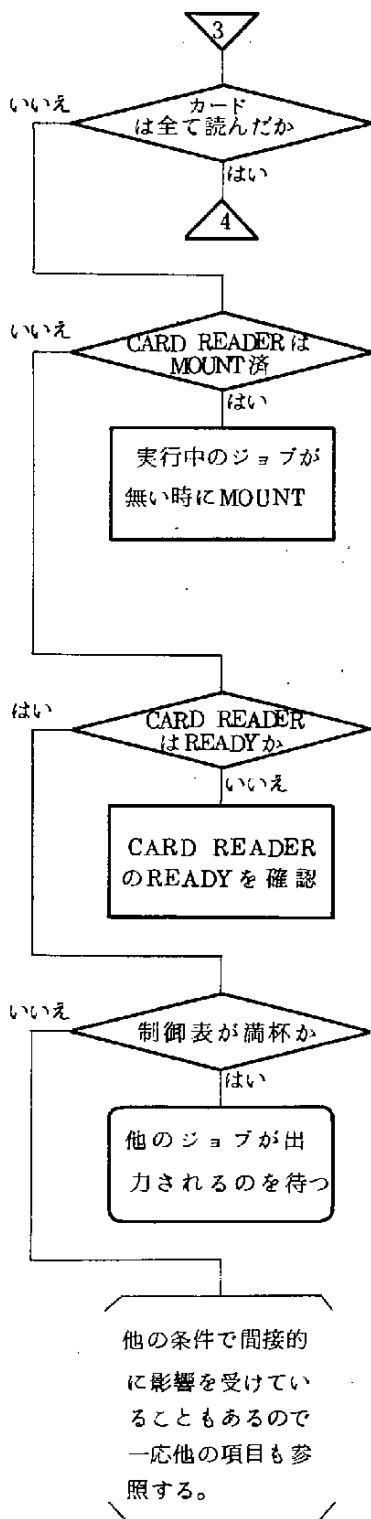
操作員の主な調査手順のフローチャートは次のとおりである。

(1) モニタのイニシャルロードができない場合





(2) カードを読まない場合



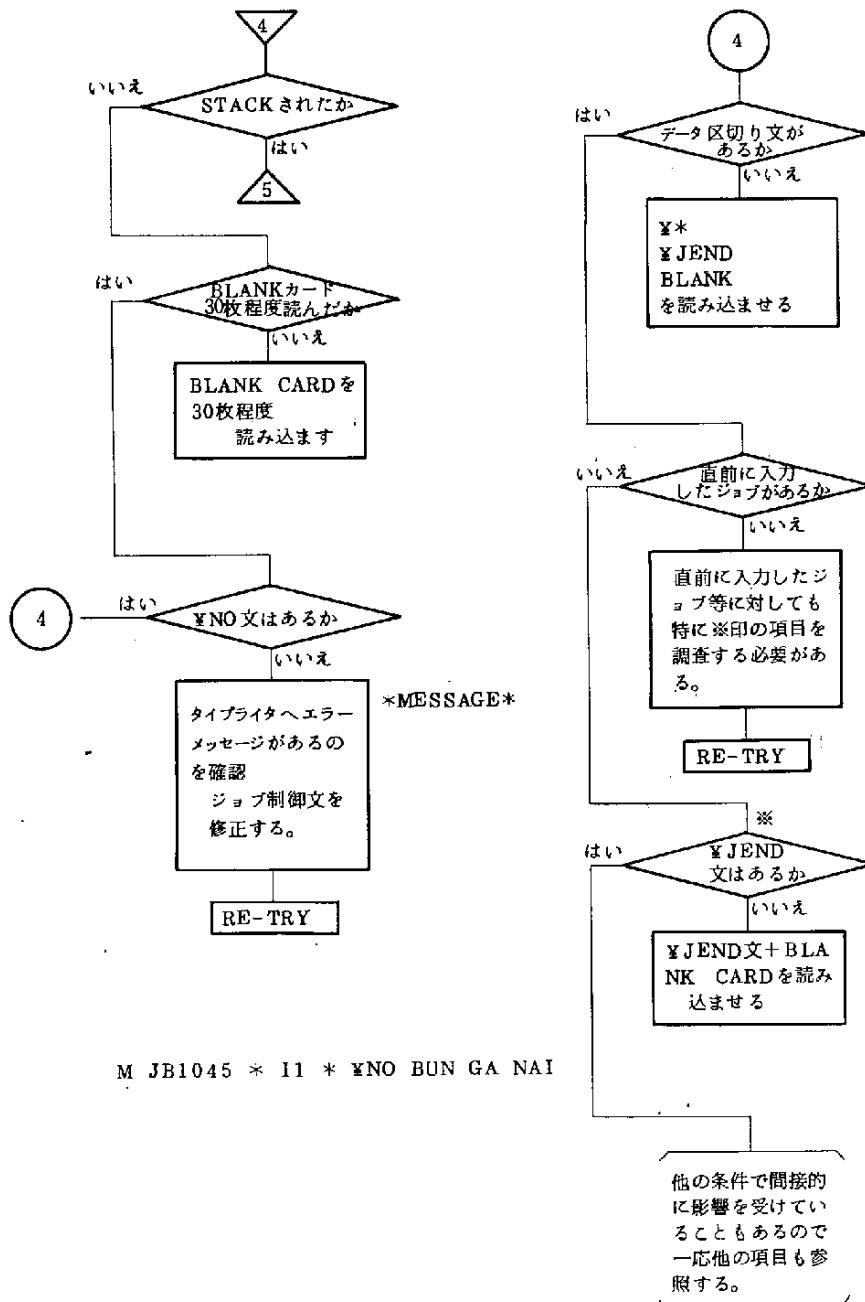
* OPERATION *

RQ M E.220,,E00
M JB0059 OWARI.....
RQ I I1,E.220,E00220

※ジョブの出力が終了し新たなジョブが入力され
ない状態で確認することが可能ではあるが、端
末からの利用が行なわれている場合は、必ずしも
ジョブ終了で入力が行なわれない。

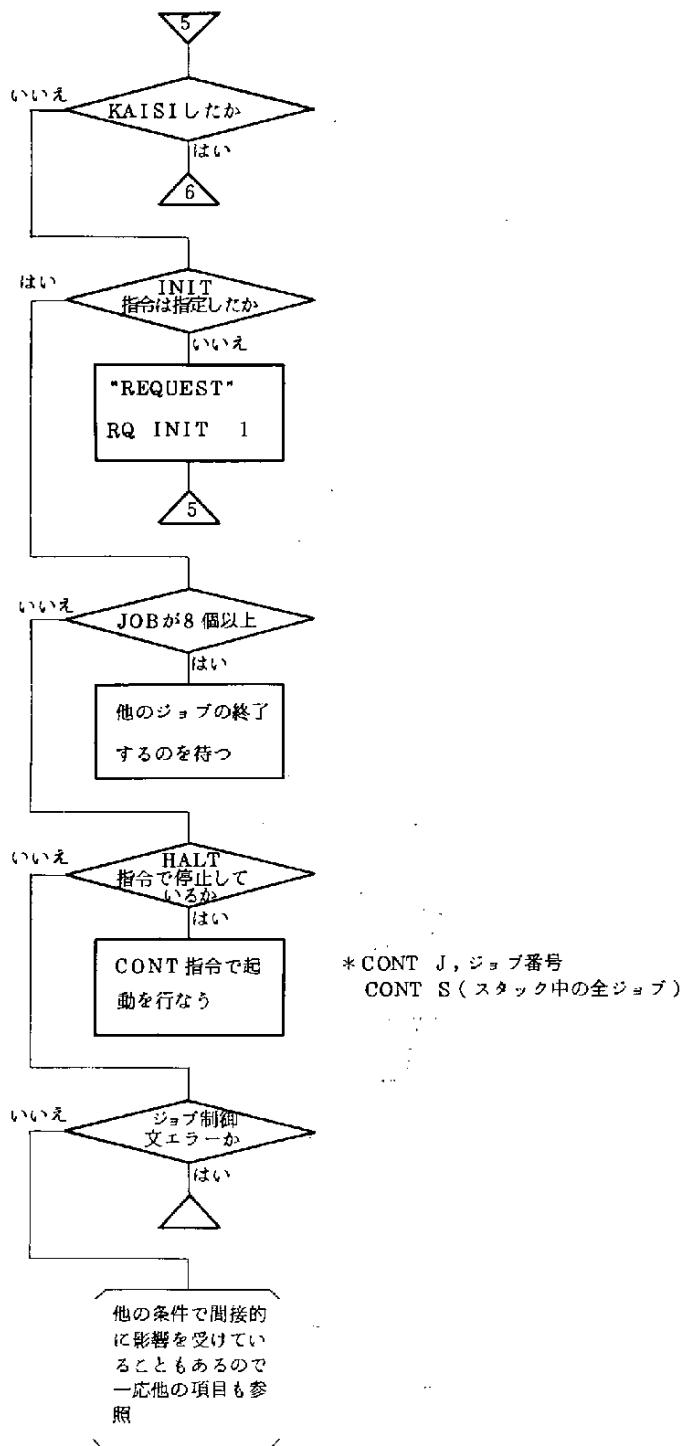
間接的とは他に処理中のジョブがREADY待ちの
ため他の処理が進まない場合等がある。

(3) ジョブがスタックされない場合

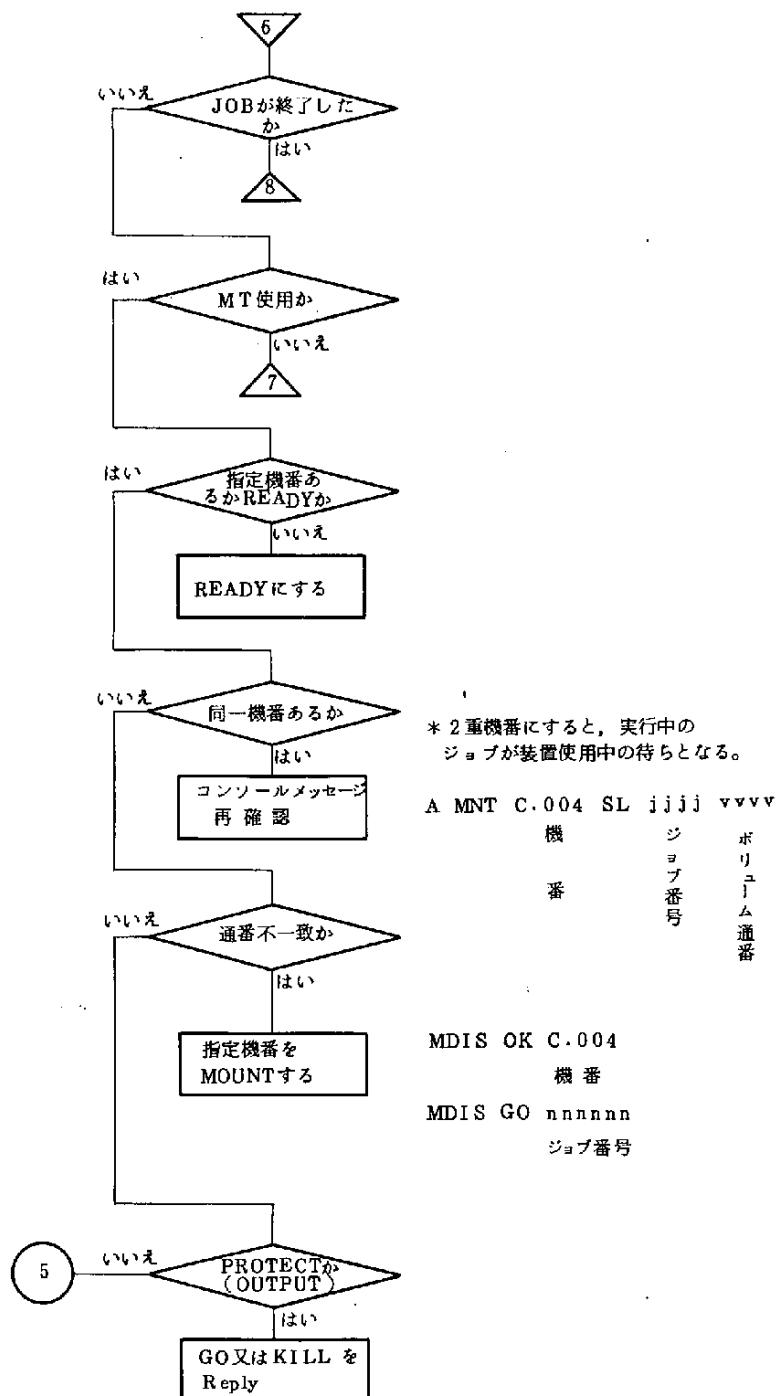


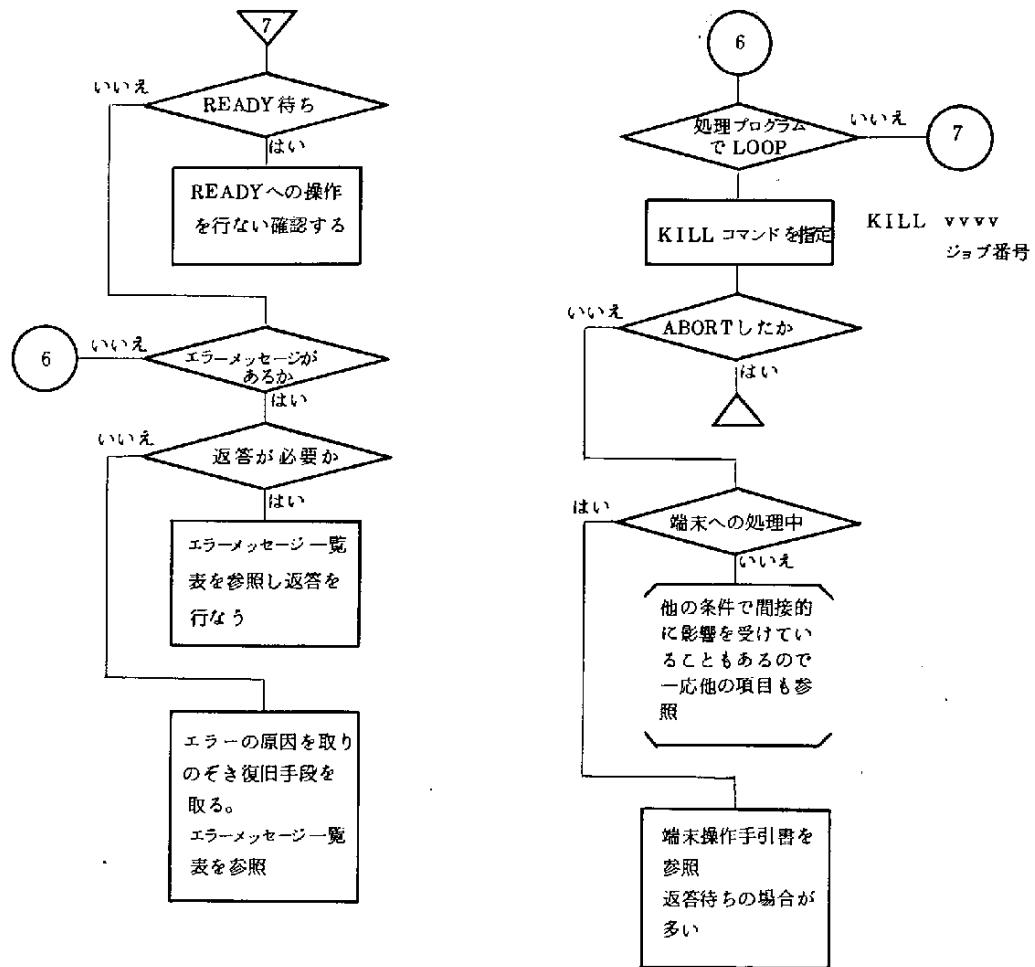
データ区切り文は、¥FD SYSIN, DATAの場合に必要であるが、
DATAの機能は¥NO, ¥JEND, ¥*のところまでである。

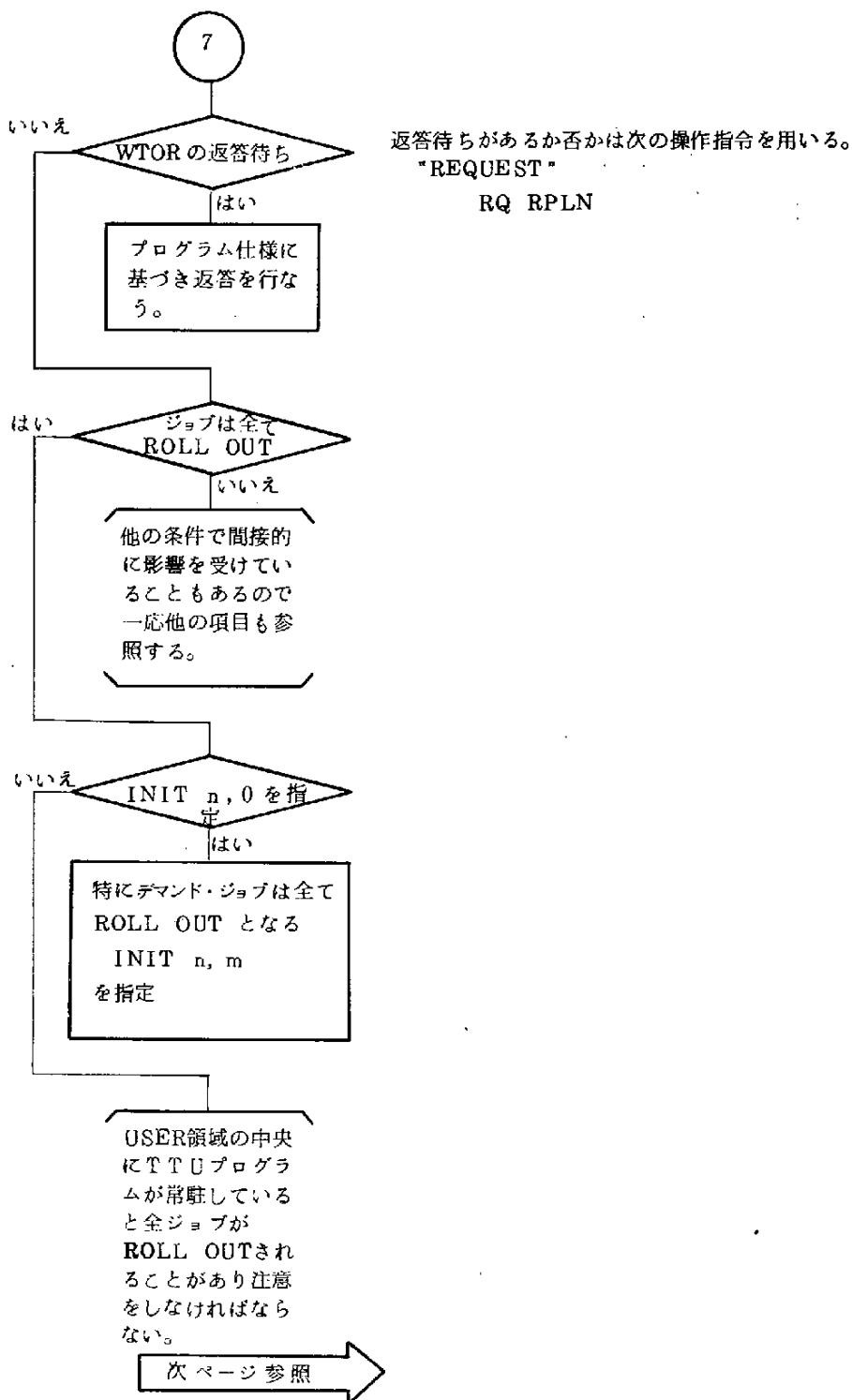
(4) ジョブが開始されない場合

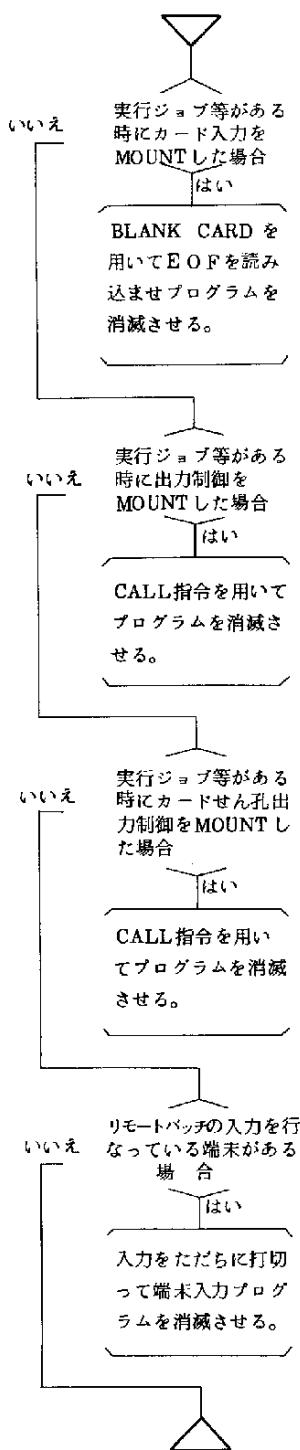


(5) ジョブが終了しない場合

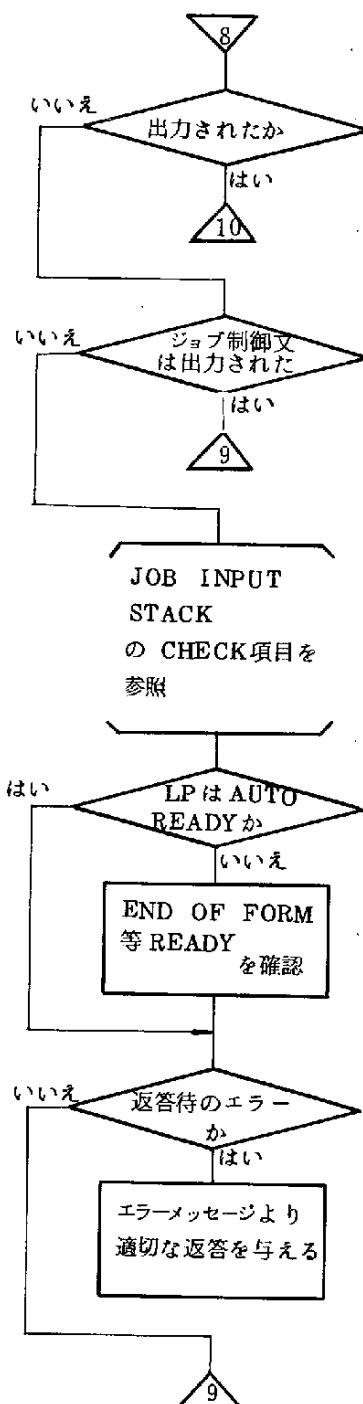




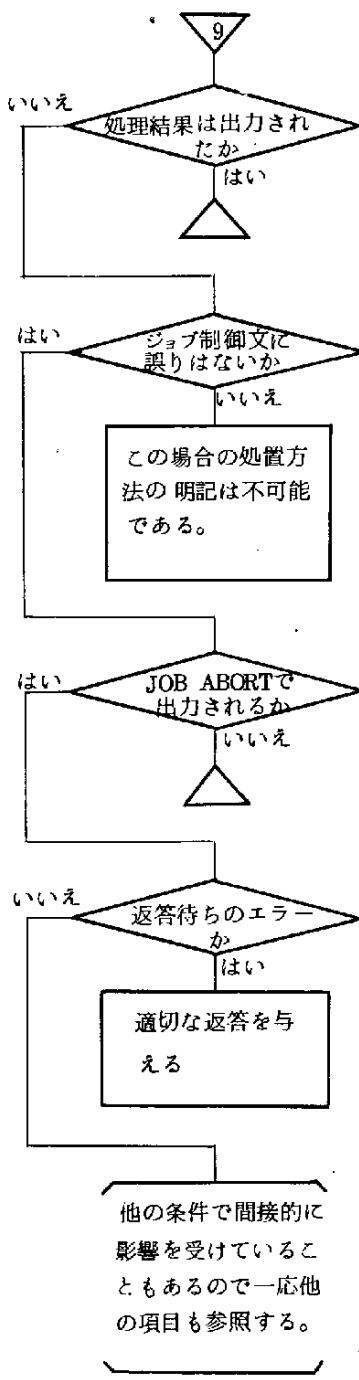




(6) 結果が出力されない場合



★リモート・バッチで SYSOUT=PRINT と指定した場合でも端末へ SMF (ジョブ制御文) の出力が行なわれる所以、必ず RECEIVE 状態としなければセンターへの出力も行なわれない。



SYSOUT=PRINT1 等を指定した場合等、他方 LPをPRINT1でMOUNTし、使用後直ちにDIS MOUNT(プログラムの消滅)を行なわなければいけない。

また、ソフトウェアあるいはハードウェアの何らかの原因でシステムダウンとなった場合には、まず、ソフトウェアに起因するものか、あるいはハードウェアに起因するものかをまず突き止め、ソフトウェアによるものであれば速やかにコアダンプや各レジスタの値、モニタ・メッセージなどの必要資料を収集した後にモニタを再ロードし、システムの運用サービスを再開する。集められた各資料は保守部門に回され、原因が追求されることになる。原因が判明した場合には、モニタの修正を行なって同じ障害の再開を防ぐ。

ハードウェアによるものであれば、どの装置に障害が発生したのかを突き止め、その装置が判明したならば、どの位の時間で復旧できるかを判断する。もし、30分以内で復旧できる障害であれば速やかに修理を行ない、システムの運用サービスを再開するが、30分以上修理に時間を要する障害であればFall Back System を適用し、機能を低下させてシステムの運用サービスを再開させる。その障害が、システムの運用サービスを続行させながら並行修理できるものであれば並行修理を行ない、並行修理ができないものであれば、システムの運用サービスが修了した後に行なうこととした。

Fall Back System は次のとおりである。

(1) C P U が障害の場合

1 C P U システムであるため、C P U が障害の場合には復旧するまでシステムの運用サービスを中断する。

(2) セレクタチャネル # A が障害の場合

セレクタチャネル # A とセレクタチャネル # D とは、チャネル間のクロスコール機能を取り入れているので、セレクタチャネル # D に切り替える。

(3) セレクタチャネル # B が障害の場合

セレクタチャネル # B に接続されている集合磁気ディスクパック装置には処理プログラムファイル、ロールイン／ロールアウト・ファイル、Temporary ファイルなどの重要ファイルがある。これらのファイルをデマンド用共用ファイルなどが収められているセレクタチャネル # A に接続の集合磁気ディスクパック装置に移したとしても、最適なファイルの組み合わせがまづか

しく、デマンド処理機能を止めるか、リモートバッチ処理機能を止めるか、あるいはバッチ処理機能を止めるかのいずれかの処置を行なわなければならなくなる。また、いずれかの機能を犠牲にしたとしても、すべてのジョブコントロールカードを変更しなければならなくなり、そのために発生するであろう混乱、あるいは、利用者に対する変更連絡を徹底させることのむつかしさなどを考慮すると、適当な手段がないので復旧するまでシステムの運用サービスを中断する。

(4) セレクタチャネル#Cが障害の場合

セレクタチャネル#Cに接続されている磁気ドラム装置にはモニタファイル、エラーロギングファイル、エラーメッセージファイル、センタルーチンファイル、ジョブ制御表ファイル、システムカタログファイルなどの重要ファイルがおさめられている。これらのファイルをFall Back用として、別に集合磁気ディスクパック2パックに作成しておき、このパックモニタをセレクタチャネル#Aに接続されている集合磁気ディスクパック装置のWORD(Temporary)ファイルがあるA. 106、および持込用ファイルであるA. 107の2デッキを犠牲にしてこれに当て、システムの運用サービスを続行させる。

(5) セレクタチャネル#Dが障害の場合

セレクタチャネル#Dには磁気テープ装置が接続されているが、セレクタチャネル#Dが障害の場合には、復旧するまで磁気テープ使用ジョブを犠牲にしてシステムの運用サービスを続行させる。

(6) マルチプレクサチャネル#Eが障害の場合

マルチプレクサチャネル#Eに接続されているI/Oにはカードリーダ、カードパンチ、ラインプリンタ、コンソールタイプライタ、紙テープリーダ、紙テープパンチ、端末などの装置が接続されている。マルチプレクサチャネルは1チャネルしかなく、他チャネルへの切り替えが不可能となるため、これらの装置が使用できなくなる。従って、復旧するまでシステムの運用サービスを中断する。

(7) 高速磁心記憶装置が障害の場合

高速磁心記憶装置は32 kW単位で1台のスタッカーに収納されており、現在6台から構成され、192 kWの容量となっている。

この内モニタが約92 kWの領域を占有するため、ユーザエリアとしては約100 kWとなる。

1台が障害で使用不能になった場合は、ユーザエリアが約68 kWとなり、容量が小さくなつて能率が低下するが、バッチ処理の多度を調整しながら1台を切り放してシステムの運用サービスを再開する。

しかし、2台以上が同時に障害で使用不能になった場合は、ユーザエリアが約36 kW以下となり、ユーザプログラムの大きさを約40 kW位に規制していることと、3元処理が実質的に不可能になってくることから1台が復旧するまでか、あるいは2台とも復旧するまでシステムの運用サービスを中断する。

また、3台が同時に使用不可となった場合には、2台が復旧するまでか、あるいは3台が復旧するまでシステムの運用サービスを中断する。

(8) 大容量磁心記憶装置が障害の場合

大容量磁心記憶装置は、1台256 kWからなり、1バンク128 kWの2バンクからなっている。

大容量磁心記憶装置には、入出力制御表が作成される。また、この記憶装置はバーチュアルメモリとして使用され、デマンドジョブ、あるいはリモート・バッチジョブやバッチジョブでも使用するように運営上設計されている。

従って、1バンクが障害で使用不能となった場合には、他のバンクに入出力制御表を作成して、この記憶装置を使用するジョブの制限を行なえば、システムの運用サービスを続行させることができる。しかし、2バンクが使用不能となった場合には、1バンクが復旧するまでか、あるいは2バンクが復旧するまでシステムの運用サービスを中断することになる。

(9) モニタ格納用磁気ドラム装置C.000が障害の場合

モニタが格納されている磁気ドラム装置C.000が障害の場合には、代替用モニタとしてあらかじめ作成して格納してある磁気ドラム装置C.002に切り

替えて、システムの運用を続行する。

(10) コンソールタイプライタが障害の場合

コンソールタイプライタが障害で装置ダウンになると、モニタとのコミュニケーションがなくなるため、単なる装置ダウンにとどまらず、システムダウンとなってしまう。この場合には、代替用コンソールタイプライタ E. 201 に切り替えてシステムの運用サービスを続行する。

3.1.3 ファイルの割り付け

ジョブを実行するために必要な制御表ファイル、ジョブ入力データファイル、ジョブエラーメッセージファイル、ジョブ制御マクロファイルなどの各種ファイルの割り付けについては、TSSサービス開始時点では次のように定めて運用した。

機器装置名	機番	ファイル名
磁気ドラム装置 F624K	A.001	計算センタールーチン・マスターファイル 制御表ファイル 二次ロールアウトファイル システムカタログファイル
	A.002	モニタプログラムファイル ジョブエラーメッセージファイル エラーロギングファイル 二次ジョブ制御マクロファイル
集合磁気ディスク パック装置 F471K	B.000	処理プログラムファイル 一次ジョブ制御マクロファイル
	B.001	一次ロールアウトファイル
	B.002	ジョブ入力データファイル 寸借ファイル 20 トラック × 20 ファイル 50 " × 20 " ワークファイル 端末入力ジョブバッファ
	B.003	ジョブ入力データファイル 寸借ファイル 100 トラック × 10 ファイル 200 " × 5 " ワークファイル

機器装置名	機番	ファイル名
	B.004	デマンド用共用ファイル
	B.005	デマンド用共用ファイル
	B.006	リモートバッチ用共用ファイル
	B.007	バッチ用共用ファイル

しかし、その後、オンラインアプリケーションである文献検索システム JOLDORの実用化に当っては（現在、キャラクタディスプレイ装置が端末として利用できないため、センター内のみで使用している）、ファイルの絶対量が不足すること、ワークファイル、リモート・バッチ用共用ファイルなども多少不足すること、また、制御プログラムが格納されている磁気ドラムのダウン対策として、代替制御プログラムファイルを他の磁気ドラム装置にファイルしておきたいことなどの理由により、昭和47年9月に集合磁気ディスクパック装置、磁気ドラム装置を増設した。

これらのファイル装置の増設により、ファイルの割り付けを次のように変更して運用した。

機器装置名	機番	ファイル名
磁気ドラム装置 F624K	C.000	モニタプログラムファイル ジョブエラーメッセージファイル エラーロギングファイル 二次ジョブ制御マクロファイル
	C.001	計算センタルーチンマスターファイル 制御表ファイル 二次ロールアウトファイル システムカタログファイル
	C.002	モニタプログラムファイル ジョブエラーメッセージファイル エラーロギングファイル 二次ジョブ制御マクロファイル (C.000 の代替用装置)

機 器 装 置 名	機 番	フ ァ イ ル 名
集合磁気ディスク パック装置 F471K	B.000	処理プログラムファイル 一次ジョブ制御マクロファイル
	B.001	一次ロールアウトファイル
	B.002	ジョブ入力データファイル 寸寸借ファイル 20 トラック × 20 ファイル 40 " × 20 " ワークファイル 端末入力ジョブバッファ
	B.003	ジョブ入力データファイル 寸寸借ファイル 50 トラック × 20 ファイル 100 " × 10 " ワークファイル
	B.004	JOLDOR 用データファイル
	B.005	JOLDOR 用データファイル
	B.006	JOLDOR 用データファイル
	B.007	JOLDOR 用データファイル
集合磁気ディスク パック装置 F471K	A.100	デマンド用共用ファイル
	A.101	デマンド用共用ファイル
	A.102	デマンド用共用ファイル
	A.103	バッチ用共用ファイル
	A.104	バッチ用共用ファイル
	A.105	リモートバッチ用共用ファイル
	A.106	寸寸借ファイル 150 トラック × 10 ファイル 200 " × 5 " ワークファイル
	A.107	持込用ファイル

さらに今後のファイル割り付け改善案としては、ロールイン／ロールアウトの処理能力の向上を図るため、現在集合磁気ディスクパック装置に作成されているロールアウトファイルを、より転送速度の速い高速磁気ドラム装置に移す予定である。今のところ高速磁気ドラム装置は、ハード的に接続されているが、

モニタプログラムとインターフェイスがとれていないため、ソフト的には接続されていない。従って、O Sのバージョンアップを行なって、昭和48年4月頃から実施する予定である。

3.1.4 サービス体制

(1) TSSサービスタイム

TSSのサービスタイムは、TSSサービス開始当初から長時間のサービスを避け、段階的延長を図った。その理由としては3元処理形態の運用面に不慣れであったこと、バッチ処理のオープン制からクローズ制へ切り替える時期であったため、まだ多くの時間をTSSサービスにさくことができなかつたこと、また、この時期のオペレーティングシステムが3元処理を行なうには、その安定度に不安があったことなどが上げられる。従って、これらの点を徐々に改善しながら、TSSサービスの時間帯を段階的に延長して、最終的なものに持っていくことにした。

その結果、次の表に示すような経過をたどって現在に至っている。

項目 年月	サービス日	サービスタイム	サービス時間帯
47年2月	月、水、金	3H	10:00 ~ 13:00
3月	月、水、金	3〃	10:00 ~ 13:00
4月	月、水、金	4〃	10:00 ~ 14:00
5月	毎日	4〃	10:00 ~ 14:00
6月以降	毎日	5〃	10:00 ~ 15:00

(日・祭日および土曜日のサービスは行なわなかった)

また、実際には各端局からサービスタイムの延長依頼が比較的多くあったため、延長依頼があった日に対しては、運用上許せる範囲で延長するようつとめ、TSSサービスタイムの弾力的運用を行なった。

(2) TSSサービス中のセントラルマシンダウンの処置

TSSサービス時間中に、セントラルマシンがソフトウェアあるいはハードウェアの障害のため、システムダウンとなった場合の端末利用者に対する

処置を次のように行なった。

まず、センター係員は端局責任者あるいは直接利用者に対して、必要に応じて今どのような理由でストップしているのかを状況説明し、予想復旧時間を連絡する。それと同時にセンター側では、復旧の処置を速やかに行なう。

復旧次第、関係端局責任者あるいは直接利用者にその旨連絡し、ジョブを開始してもらうわけであるが、もし、ダウンが原因でユーザファイルが破壊されていることが確認されていたならば、その状況を説明するとともに、必要処置を合わせて連絡し、それに対処してもらった。

これらの連絡手段は、すべて電話で行なったが、連絡時に問題となつたことは、端末室に電話が設置されていない端局があったために、システムダウンやその他のトラブル時の連絡がスムーズに行なえず、不便を感じたことがしばしばあったことである。やはり端末室には必ず電話を設置しておくべきであろう。

この問題で考えさせられたことは、現在のモデムに装着されている受話機は、センタ側と端末側（端末装置によってモデムが内蔵されているものもあり、受話機が装着されていない機種もある）の両方が通話モードでなければ話し合いができない機構になっているが、通常はセンター側も端末側もデータモードでジョブを処理しているため、システムダウンなどの時に端末側と通信するためにセンタ側のモデムを通話モードにしても、端末側のモデムを通話モードにする手段がない。今のところ端末側に電話して通話モードに切り替えてもらう方法しかないが、このような手段を取るなら、初めから電話で用件をすませてしまえばよいわけである。そこでこのモデムがセントラルマシンがシステムダウンした時に、両方とも自動的に通話モードに切り替わるとか、通話したい時には相手側を呼び出して、通話モードにさせるための発信音を出すなどの機能があれば便利であり、端末室に電話がなくてもスムーズに用件が伝えられるのではないかということである。

モデムの機構上、このような機能を付加することは不可能であるのかも知れないが、このようなことが実現すれば、非常に便利であると思われる。

(3) 端末装置のハード障害処置

各端局で TSS 利用中に、端末装置のハード障害が発生した場合の処置を次のように行なった。

端末側からその異常状態をセンタ係員に連絡してもらい、それに対してセンタ係員は、保守部門に速やかに連絡を取る。保守員はただちに障害発生端局へ直行して、その障害状況を調査判定し、復旧に必要な措置を取り、速やかに復旧を図るようにした。

(4) 回線障害の処置

バッチ処理システムのハード障害であれば、機器の障害だけですむが、TSS 機能が付加されたシステムの場合には、これとは別に回線の障害が加わる。単なる機器障害であれば、保守部門に修理を依頼することになるが、明らかに回線障害の場合には、電信電話公社の保全部門に修理を依頼しなければならない。

しかし、端局側からの連絡、あるいは、センタ側で何らかの障害があることがわかったとき、その障害が機器にあるか、回線にあるかを判別することが困難なことがしばしばある。

このような場合、センタ係員はできるだけすみやかに、どの部分の障害があるのかを発見し、適切な処置をとる責任があるわけであるが、次のような手順で判別し処置をした。

まず、回線と端末、回線とセントラルマシンを切り放し、端末およびセントラルマシンに異常がないかを個別にテストをする。もし、端末およびセントラルマシンに異常が認められなければ、回線の調査を行なう。

調査方法としては、テスト符号を送って回線の状態を調べ、テスト符号の異常により回線が悪いと判断されたときに、電信電話公社の保全部門にアクションを取って復旧させた。

(5) 利用相談

計算機システムを利用するに当って、利用に関する全般的な相談、プログラムに関する質問や問い合わせが発生する。

これらの質問や問い合わせの受け付け、および回答を次のように行なった。

- 電話による方法
- 「問合せ票」による方法
- 来所による方法
- センタから出張する方法

電話による方法は、後の管理資料を作る場合には比較的記録として残しづらいこと、電話では要領を得ない場合があることなどの理由から、極力「問合せ票」による方法を用いるようにしていたが、やはり緊急を要する質問や問合せが多いため、電話あるいは出張による方法を多く用い、処理のスピード化を図った。

「問合せ票」の形式は次のとおりである。

※16 _____

問合せ票

年 月 日

社名 _____

所属 _____

氏名 _____

受付番号※	受付者※	種別	方法
		1.質問 2.要望 3.その他	1.来所 2.書面 3.電話

(質問内容) :

				書類返送
				要否
電話番号	回答年月日※	回答者名※	回答区分※	
	年 月 日		1.解決 2.検討 3.打切	

(解答) :

※欄は記入不要です。

(6) 講習会

TSSサービスを利用するに当って必要な教育を行なうため、当財団では必要が生じた場合、あるいはユーザの要望に応じて講習会を開催した。その開催実績は次のとおりである。

第1回講習会

a 日 時

- 昭和47年1月31日(月)～2月8日(水)

b 内 容

- FACOM-TSS の概要説明
- 端末機F-1591Aについての説明
- BACCUSについての説明
- LINEDについての説明
- TSS-FORTRANについての説明
- CPLについての説明
- BACCUSおよびLINEDの実習

c スケジュール

表3-1のとおり。

表3-1 第1回講習会スケジュール

月 日	1/31	2/1	2	3	4		7	8
曜 日	月	火	水	木	金		月	火
10:00 午 前 12:00	FACOM-TSS の概要	BACCUS	BACCUS	LINED	FORTRAN	10:00 BACCUS→ 11:00 LINED 12:00	実 習(A)	実 習(C)
13:00 午 後 16:00	端末機 について	BACCUS	BACCUS	LINED	CPL	13:00 BACCUS→ 14:30 LINED→ 16:00	実 習(B)	実 習(D)

第2回講習会

a 日 時

- 昭和47年2月28日(月)～3月7日(火)

b 内 容

- FACOM-TSSの概要説明
- 端末機F-1591Aについての説明
- BACCUSについての説明
- LINEDについての説明
- TSS-FORTRANについての説明
- CPLについての説明
- BACCUSおよびLINEDの実習

c スケジュール

前回と同様表3-1に基いて行なった。

第3回講習会

a 日 時

- 昭和47年11月13日(月)～11月16日(木)

b 内 容

- FACOM-TSSの概要
- CPLの説明
- システムコマンドの説明
- TSS-FORTRANの概要
- CPLおよびTSS-FORTRANの実習

c スケジュール

月 日	11/13	15	16	
曜 日	月	水	木	
10:00 午 前	F A C O M T S S	の 概 要 C P L	シ ド の 説 明 C P L コ マ ン	実 習
12:00				
13:00 午 後 16:00		C P L	T S S I F O R T R A N	実 習

3.1.5 センタ標準値

オペレーティングシステムMONITOR-Vを創成するには、システム編集（System Generation）を行なう。

その時に定められるジョブ打ち切り処理装置使用時間、端末出力打ち切り行数、ジョブ優先権、または、効率的なシステム運用を図るために定められるジョブの多重度、各種制限値などのセンタ標準値を次のように決めた。

(1) 各種優先権

モニタで持っている優先権には、ジョブ優先権、実行優先権、主記憶駐在優先権があり、複合コントロール方式を取っている。

ジョブ優先権とは、大記憶の中にシステム入力装置から入れられて貯えられた、ジョブの実行開始の順序を決定するために利用されるものであり、ジョブの優先権の高いものから選び出されて開始される。レベルは0～7まであり、数字の大きいほど優先権が高くなっている。

実行優先権とは、タスクの実行順位を定めるもので、実行待タスクは実行優先権にしたがって実行に移される。すなわち、中央処理装置の使用優先順位である。レベルは0～15まであり、数字の大きいほど優先権が高くなっている。

主記憶駐在優先権とは、主記憶を占有する際の優先順位を定める。レベル

は、0～15まであり、数字の大きいほど優先権が高くなっている。

これらの優先権をバッチジョブ、リモート・バッチジョブ、デマンドジョブとの関連を持たせ、次のように定めた。

優先権 ジョブ処理形態	ジョブ優先権	実行優先権	主記憶駐在 優先権
バッチジョブ	5	5	5
リモート・バッチジョブ	6	6	6
デマンドジョブ	7	15	3

ジョブ処理形態の性質上、デマンドジョブのジョブ優先権、実行優先権を最高レベルに設定した。主記憶駐在優先権については、端末操作中などの実行待ち時間の無駄な主記憶駐在を防ぐため、バッチジョブ、リモート・バッチジョブのそれよりも低くした。

リモート・バッチジョブについては、ジョブ優先権と実行優先権をバッチジョブよりも高く、デマンドジョブよりも低いレベルに設定し、ターンアラウンドタイムを速めるため主記憶駐在優先権をバッチジョブやデマンドジョブよりも高いレベルにした。

バッチジョブは、全体から見て他の処理形態よりも各優先権のレベルが低く、ただ主記憶駐在優先権だけはデマンドジョブの実行待ち時間を利用するため、デマンドジョブよりもレベルを高くしている。しかし、リモート・バッチジョブよりも低く定め、リモート・バッチジョブの優先を図った。

(2) ジョブ打ち切り処理装置使用時間

1ジョブが処理装置を使用することができる最大時間を決めるもので、それぞれのジョブの内容を検討し、次のように設定した。

バッチジョブ 5分

リモート・バッチジョブ 5分

デマンドジョブ 20分

(3) ジョブ打ち切り磁心記憶占有時間

1ジョブが磁心記憶装置を占有することができる最大時間を決めるもので、

それぞれのジョブの内容を検討し、各ジョブ処理形態とも60分と設定した。
ただし、デマンドジョブについては180分とした。

(4) ジョブ打ち切りカードせん孔枚数

1ジョブがカードせん孔装置に出力できる最大枚数を決めるもので、それ
ぞのジョブの内容を検討し、各処理形態とも10,000枚と設定した。

(5) ジョブ打ち切り印刷ページ数

1ジョブがラインプリンタ装置に出力できる最大印刷ページ数を決めるも
ので、それぞれのジョブの内容を検討し、また処理形態とも（デマンドジョ
ブ、リモート・バッチジョブについてはセンタ出力の場合）200ページと設定
した。

(6) ジョブ打ち切り印刷行数

1ジョブがラインプリンタ装置に出力できる最大印刷行数を決めるもので、
(5)で述べたジョブ打ち切り印刷ページ数と関連させ、1ページ50行印刷する
ものとして、各処理形態とも（デマンドジョブ、リモート・バッチジョブにつ
いてはセンタ出力の場合）10,000行と設定した。

(7) 端末出力打ち切り行数

端末装置に出力される1ジョブの打ち切り行数を決めるもので、ジョブの
内容や通信回線の通信速度との関係であまり多量の出力をさけるために、
1,000行と設定し、これを越えるものはセンタ出力させるようにした。

(8) ジョブ打ち切りHCM占有語数

1ジョブが高速磁心記憶装置を占有できる最大量を決めるもので、プログラ
ムを作成する場合には、先に述べたように運用面のことを考慮して、40
kWを標準とし、あまり大きくならないように申し合せているが、モニタで
持つジョブ打ち切りHCM占有語数は70kWに設定した。

(9) ファイル定義文パラメータの標準値

ファイル定義文は、プログラムのなかで定義されたファイル制御ブロック
と、実際のデータファイルとの対応関係を確立するものであるが、ファイル
定義文のSPACEパラメータが省略された時に取られる標準値は、ジョブの

内容を検討し、次のように設定した。

スペース割り当て一次量 40 トラック

" 二次量 40 "

(10) ジョブ入力データファイルのスペース割り当て量

ジョブ入力データファイルは、1個のジョブ入力データにつき1個ずつ開設するもので、その割り当て量はジョブの内容を検討し、次のように設定した。

ジョブ入力データファイルのスペース割り当て一次量 40 トラック

" 二次量 40 "

(11) ジョブの最大多重度

ジョブの最大多重度を決めるもので、15多重に設定した。しかし、実際のシステム運用サービス中は、その時の全体のジョブの状態で、オペレータのシステム操作指令で1~4多重の範囲で運用した。

(12) 端末入力制御ジョブバッファの割り当て量

端末入力制御が作成するジョブバッファの割り当て量を決定するもので、端末入力ジョブの内容を検討し、次のように設定した。

スペース割り当て一次量 10 トラック

" 二次量 10 "

(13) 端末入力の制御の最大値

端末入力制御が同時に処理しうる端末装置の最大数を決めるもので、端末利用頻度を考慮に入れて7台と設定した。

(14) 端末出力制御の最大値

端末出力制御が同時に処理しうる端末装置の最大数を決めるもので、(13)で述べた端末入力制御の最大値に合わせて7台と設定した。

(15) デマンドジョブの処理許容値

デマンドジョブの同時利用可能な最大数を決めるもので、端末数に合せて12台と設定した。

(16) 端末装置台数

TSS制御の管理下にあるすべての端末装置の台数を指定するもので、現在12台の端末が設置されているが、増設のことも考慮して15台と設定した。

3.1.6 ジョブ制御マクロ

ジョブの処理の方法は、各言語ごとに原プログラムの翻訳—結合—実行といった手順を踏むのが一般的である。このようにジョブの制御には繰り返し使用される一定の型がある。このようなジョブ制御文の並びを一まとめにして名前をつけて、ジョブ制御マクロとして登録しておき、必要なたびに登録した名前で呼び出して使えば便利なわけである。このような方法で利用者のジョブ制御カード作成の手間を省くとともに、ジョブ制御カードの誤りも省くことができる。また、ジョブ制御カードの作成を各利用者にゆだねた場合には各種優先権、磁心記憶占有量、ファイルのSPACE指定などの値を不当に指定して、システムの運用サービスに重大な影響を与えることもある。従って、3元処理でのシステムの運用サービスを円滑に進めるためにも、運用部門でセンタの標準値を指定したジョブ制御カードをジョブ制御マクロとして登録して、一般利用者に開放した。また、例外的なジョブ制御マクロの登録は、利用者から運用部門へ登録申請され、運用部門でチェックの上登録して開放した。

ジョブ制御マクロの開放に当って、バッチジョブ用（リモート・バッチジョブも含む）には「ジョブ制御マクロ登録集」を、デマンドジョブ用には「デマンド用マクロコマンド説明書」を作成し、利用者に配布した。

「ジョブ制御マクロ登録集」については巻末の付録-IIを参照されたい。

なお、「デマンド用マクロコマンド説明書」については、4.2処理プログラムについての項と重複する部分があるので省略する。

3.1.7 ファイル・リカバリー

ファイル（大記憶ファイル）のリカバリーについては、その方法、タイミング等いろいろ考えられ、「ファイル単位で、破壊される直前の状態」に戻すことが望ましいが、3元処理においてはデマンド処理、リモート・バッチ処理の

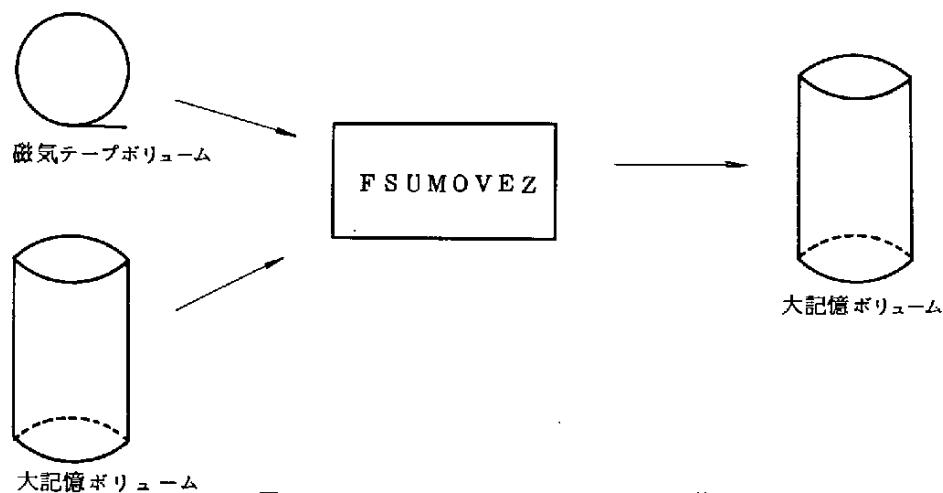
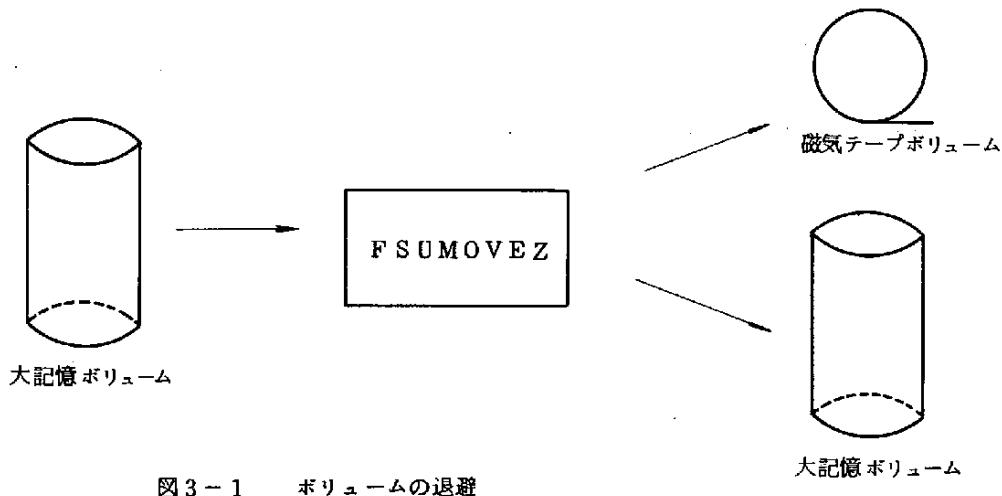
ように各端末毎あるいはジョブ毎に入力データの形式や出力ファイルの形式が異なるので、各ファイルを破壊される直前の状態に戻すことは必ずしも容易ではない。また、仮りにそうできるようにするとすれば、そのために必要とするファイル容量、CPUタイム等は無視できない程のものとなる。それゆえ、ファイル・リカバリーは業務の性質、すなわち、ファイルの重要性によって、その方法が異なるであろう。

当センターのTSSサービスにおけるファイル・リカバリーの方法は、TSSサービス終了後または開始前にシステム・ユティリティ FSUMOVEZ を用いてファイルを磁気テープに退避しておき(退避するボリュームはデマンド用2個、リモート・パッチ用1個、センター用1個の計4ボリュームである)，ファイルが破壊された場合は、この磁気テープから破壊されたファイルのみを復元するという方法をとることとした。この方法では、仮りにファイルが破壊された場合、そのファイルに当日UPDATEをしていたとしても1日前の状態に戻ってしまう(破壊されたファイルのみ)という欠点があるが、ファイルを破壊するといったハード、ソフト的な原因、確率および完全なファイル・リカバリーをするためのファイル容量、CPUタイム等を考慮すればやむを得ないと考えられた。

システム・ユティリティ「FSUMOVEZ」は、大記憶ボリューム間ないしは大記憶ボリュームと磁気テープ間でファイルの転送や複写を行なうものであり、次のような機能をもっている。

- 大記憶ボリューム上の全ファイルを同時に別の大記憶ボリュームまたは、磁気テープへ転送ないしは複写する。
- (FSUMOVEZで作成された)磁気テープから大記憶ボリュームへ
 - (1) 全ファイルを同時に複写する。
 - (2) 指定されたファイルのみを複写する。
- 大記憶ボリューム上の指定されたファイルのみを別の大記憶ボリュームまたは、磁気テープへ転送または複写する。
- ファイル編成形式は次のものを許し、これらが混在していてもかまわない。

- (1) 順編成ファイル
 - (2) 直接編成ファイル
 - (3) 分割型順編成ファイル
 - (4) 順インデックスファイル
- ・ファイルの数やレコード形式に制限はない。



試行期間中に「FSUMOVEZ」自身の障害が2,3発見され、例えばトラックオーバーフローを使用したファイルについては退避した磁気テープから元のファイルが復元できず、ファイルを最初から作り直したケースがあったが、オンラインのサービスをする場合には、ファイル(ボリューム)の退避、復元とい

ったものは極く基本的なものであり、TSSモニタを提供する以前にこのようなシステムユティリティといったもののテストはメーカーとしても十分しておくべきであろうと思われた。

実際の運用に際しては各外部端末ユーザのファイルが完全に破壊されたというケースはハード的にもソフト的にも1件もなかったといってもいいと思われる。

ただ、次のようなケースが1件生じた。それは、デマンド処理において、すでにセーブしてあるプログラムを修正(UPDATE)しようとして該当するプログラムをロードし、修正後セーブしてあるプログラムを消去(PURGE)して、再び修正済のプログラムをセーブしようとしたとき、消去までの動作は完全であったが、修正済のプログラムをセーブしようとしたとき、ハードエラーであるといった、メッセージが端末に出力されて、何度繰り返えしてもセーブされなかった。これはオープン動作が正常でなかったためと、それに対する処理プログラムの対策が不備であったためと思われる(そのファイル自身は正常で同一時点他の端末からは何等支障なくアクセスできた)。

このため、やむを得ずそのジョブを打切り、翌日、消去してしまったプログラム(メンバ)のみを退避してある磁気デープから、前日の状態に戻し、再び修正作業から再実行してもらった。

これらの経験から、システムのデータ管理が安定していれば、ファイルが壊されるといったケースはほとんどないはずであり、人為的に壊されないような手段を講ずることが重要であろう。

3.1.8 ファイル保護

ここでいうファイルの保護とは、MONITOR-Vのデータ管理レベルのものではなく、他人のファイルを消去してしまうといった利用者のちょっとした間違いによって生ずる事故を防止するものである。

リモート・バッチ処理やデマンド処理のように各端末利用者が自由にファイルの作成あるいは消去ができる、しかもファイル名が全く任意に指定できるもの

であれば、間違って他人（他の端末）のファイルを消去してしまうといったケースが生ずるかも知れない。また、異なる端末間で作成したファイルをパスワードも使用せずに自由に参照できるということは機密保護という面からも望ましくないであろう。かといって、例えば A という端末から作成するファイルのファイル名は必ずファイル名の頭文字を A とし、B という端末から作成するファイル名の頭文字は必ず B とするといった、端末利用者にファイル名に対する制限を加えることは利用者にとって面倒であるばかりでなく、システム側にとっても絶えず制限に違反しないかどうか、監視しなければならないといった面があり、端末利用者に使用するファイルの名前に対して制限を課すことは望ましくないと考えられる。

それゆえ、端末利用者が指定するファイル名は任意であっても、それを指定した端末によってファイル名を区別できるような方法をとることとした。具体的には、利用者が指定したファイル名の前に計算センター・ルーチン（後述）が、端末名を付加するという方法である。

この方法によれば、異なる端末間では全く同じファイル名が指定されることはなく（すなわち、他の端末から消去される可能性は全くない）、しかも端末利用者にとっては端末名が付加されているかどうかといったことは全く問題にならず、自分が指定したファイル名でアクセスすることができる。もちろん、特殊なパスワードを付加することによって異なる端末間でも同一ファイルをアクセスできるようになっている。これは、共用ファイルの存在やセンタのように端末が複数台設置してある所では、同一人が同一端末のみを使用するとは限らないためであり、また、バッチ処理で作成したファイルをデマンド処理やりモード・バッチ処理でアクセスするといったケースが生ずるためである。

3.2 計算センター・ルーチン

3.2.1 計算センター・ルーチンの概要*

計算センター・ルーチン（以下単にセンター・ルーチンという）とは、OSの基本モジュールに含めることができないようなものに対して、各センターがそのセンターの条件に合うようにOSの一部として組み込むルーチンおよび運営管理に供する各種統計表作成プログラム等をいう。

たとえば、ジョブ番号の指定方法、ファイルの指定方法、利用者の資格審査、会計情報の収集などは、センターの運用目的、条件等がそれぞれ異なるため当然各センターごとに異なるのが普通である。もちろん、これらの条件を考慮したうえで最も一般的、あるいは標準的な処理方法をOSは提供するものであるが、センター独自の要求、特にTSSを運用するようなセンターではそのセンターの運用目的、ハードウェアの構成等に合った処理方法がとられなければならない場合がある。

それゆえ、OSの設計思想としてはこのようなセンター・ルーチンが容易に組み込めるような配慮がなされていなければならない。

(1) TSSの運用にあたって考慮されるべきセンター・ルーチンの機能

TSSの場合はジョブの依頼がセンター受付けを通らず端末から直接入力されるといったことからローカル・バッチ処理とは異なった問題が発生するので、センター・ルーチンとしては次のような点を考慮しなければならない。

イ. 利用者の資格審査

TSSにおける利用者の資格審査は、ローカル・バッチ処理のようにジョブの依頼がセンター受付けを通らないため特に重要である場合があるが、パスワード等の使用を含めた審査の方法は各センターごとに異なり、機密保護等も考慮したうえで作成されなければならない。

*計算センター・ルーチンの概要については、その機能等の説明の都合上46年度の報告書の「センター・ルーチンについて」と一部重複する。

ロ. ジョブ番号(名)の発行

O.S.によって指定方法は異なるが、それが“ジョブを識別するもの”であるとすれば同一時点で同じジョブ番号をもつジョブが入力されてはならないし、同一時点でなくても会計情報等の各種統計をとる場合には不都合な点を生ずる。

ローカル・バッチ処理においては、ジョブ番号(名)は、ジョブ依頼者およびセンター受付者によって、そのジョブが入力される前にチェックされるが、TSS処理においてはセンター受付けを通らないためにO.S.(センター・ルーチン)がジョブ番号を決定し、かつ、端末にそのジョブ番号を知らせなければならない。これは端末利用者に指定させると異なる端末から同一のジョブ番号が指定される可能性があるためである。

ハ. ファイル名

端末利用者にファイル名に対して特に制限を加えることは望ましくない。しかし、端末利用者に任意のファイル名の指定を許すとすればジョブ番号と同様同じファイル名が異なる端末から指定される可能性がある。それゆえ、端末利用者が指定するファイル名は任意であってもそれを指定した端末によってファイル名を区別できるようにするような方法がとられるべきであろう。これはファイル保護という面からも必要である。また、逆に公用ファイルのような異なる端末から同一ファイルをアクセスする場合もあり、特定の端末群のみ特定のファイルをアクセスするといった場合もあるのでこれらを満足するような方法がとられなければならない。

ニ. 優先権の指定

ジョブ優先権、実行優先権、主記憶優先権等の指定は、たとえそれがO.S.の許容範囲内であっても、ローカル・バッチ処理、リモート・バッチ処理、デマンド処理といった処理が混在する場合には他のジョブに与える影響が非常に大きい。それゆえ、これらの指定に関してはそれぞれの処理形態によってある程度の制約をしなければならない。

ホ. ファイルの容量チェック

外部記憶装置の記憶容量には当然センターごとに限界があり、特に保存される(KEEPされる)ようなファイルに対しては使用量の制限を加えなければならないであろう。規定の容量を越えるような場合はその旨端末に通知しなければならない。

ヘ. 会計情報ファイルの作成

日毎あるいは月毎の個人別、プロジェクト別等の使用記録(処理ジョブ数およびジョブ・ステップ数、CPU使用時間、主記憶使用量、外部記憶使用量等)をとるための会計情報ファイルを作成しなければならない。

(2) 主なセンター・ルーチンの目的と機能

センター・ルーチンに必要な機能については、(1)で述べたが、具体的なプログラムとしては次のようなものがある。

① 初期設定ルーチン

他のセンター・ルーチンを動作させるための初期設定を行なうルーチンであり、次のようなことが可能である。

イ. 計算センター・ルーチンが使用するファイル(会計情報ファイル等)のOPEN処理。

ロ. その他の初期値の設定。

② ジョブ・チェック・ルーチン

ジョブ・チェック・ルーチンはジョブに関する各種パラメータ(ジョブ番号、登録番号、各種優先権、CPU使用時間、完了条件等)のチェックを行なうのが目的であり、次のようなことが可能である。

イ. パラメータの一部変更。

ロ. パラメータで与えられなかったものに関しては、センターで規定した標準値を与える。また、センターとして禁止しているようなパラメータが指定されている場合は強制的に変更するかエラーとして通知する。

ハ. 会計用情報の収集。

ニ. その他。

③ ジョブ・ステップ・チェック・ルーチン

ジョブ・ステップ・チェック・ルーチンは、ジョブ・ステップに関する各種パラメータ（実行優先権、主記憶優先権、主記憶使用量等）のチェックを行なうのが目的であり、次のようなことが可能である。

イ. パラメータの一部変更。

ロ. パラメータで与えられなかったものに関しては、センターで規定した標準値を与える。また、センターとして禁止しているようなパラメータが指定されている場合は、強制的に変更するかエラーとして通知する。

ハ. 会計用情報の収集。

ニ. その他。

④ ファイル・チェック・ルーチン

ファイル・チェック・ルーチンはファイルに関する各種指定（ファイル名、UNIT指定、装置指定、容量の指定等）のチェックを行なう。

イ. パラメータの一部変更。

ロ. パラメータで与えられなかったものに関しては、センターで規定した標準値を与える。また、センターとして禁止しているようなパラメータが指定されている場合は、強制的に変更するかエラーとして通知する。

ハ. 会計用情報の収集。

ニ. その他。

⑤ ジョブ制御マクロ・チェック・ルーチン

ジョブ制御マクロ・チェック・ルーチンはジョブ制御マクロ呼び出し文のチェックを行ない、パラメータの変更等が可能である。

⑥ デマンド・ジョブ・チェック・ルーチン

デマンド・ジョブ・チェック・ルーチンは、デマンド・ジョブを開始させるための各種情報の作成、チェックを行なう。

イ. 端末利用者の利用資格審査。

ロ. ジョブ番号の作成、通知。

ハ. ジョブ制御文の作成。

⑦ ジョブ番号作成ルーチン

ジョブ番号作成ルーチンは、リモート・バッチ処理のジョブ番号が作成するルーチンである。これは異なるターミナルから同一ジョブ番号が指定されるのを避けるためである。

⑧ 会計情報収集ルーチン

会計情報収集ルーチンは、それぞれ次のような時点で呼び出され、他のルーチン、たとえばファイル・チェック・ルーチン等で収集された情報を含めた各種会計用情報を収集する。

イ. ジョブが開始されるとき。

ロ. ジョブ・ステップが開始されるとき。

ハ. ジョブ・ステップが終了したとき。

ニ. ジョブの実行が終了したとき。

ホ. ジョブのシステム出力が終了したとき。

ヘ. リモート・バッチ・ジョブのシステム出力が終了したとき。

ト. デマンド・ジョブが終了したとき。

⑨ 仕分け情報作成ルーチン

仕分け情報は、システム出力装置上にシステム出力を行なう場合、ジョブごとの区別ができるようシステム出力の最初または最後に出力されるが、仕分け情報作成ルーチンでは、この仕分け情報を作成する。

主なセンター・ルーチンは以上のようなものであるが、センター・ルーチンは狭い意味では制御プログラムの一部として組み込まれるものであり、広い意味では制御プログラムのセンター・ルーチンによって作成された会計情報ファイルから各利用者毎の料金あるいは統計等をとるユーザ・プログラムもセンター・ルーチンに含まれる。これらのルーチンの構成は次図のとおりである。

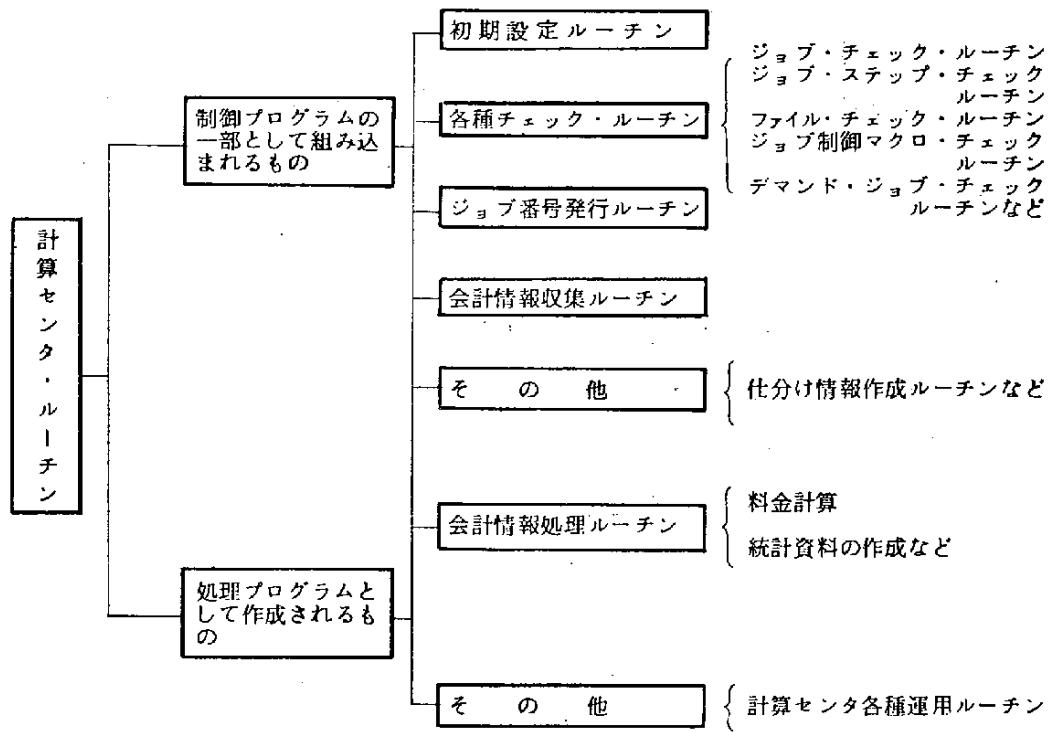


図3-3 計算センター・ルーチンの構成

3.2.2 計算センター・ルーチンとして制御プログラムに追加した機能と目的

TSSの運用にあたって、センター・ルーチンとして制御プログラムに追加したものはセンター・ルーチンの概要(1)で述べたようなものについてである。

(1) 利用者の資格審査

デマンド処理における利用者の資格審査に関するMONITOR-Vの標準的な機能は、下に示すようにDEMANDコマンドを入力後、利用者名とパスワードを要求してくることである。そして、それらに関するチェックは、12文字以内の英数字であるかどうかだけである。

これは、前述のように各センターの運用目的、条件等がそれぞれ異なるため当然パスワード等のチェック方法も異なるため、それ以上のチェックができないためであろう。

DEMAND デマンドジョブを開始するための
ANATA NO NAMAE=利用者名 コマンド
ANATA NO PASSWORD=パスワード 利用者名、パスワードはそれぞれ
12文字以内の英数字である。

当センターの T S S の運用にあたっては、これを次のように変更した。

DEMAND

?USER-ID.....利用者名、登録番号〔マスタ・パスワード〕

登録番号は、プロジェクト毎に定められた 4 桁の数字であるが、デマンド、リモート・バッチにおける外部端末ユーザには特定のコードが割り当てられている。

登録番号をプロジェクト毎にしたのは各種統計表を作成するのに便利だからであり、外部端末に対しては端末毎にそれぞれ特定のコードを定め、そのコード以外のコードが入力されるとデマンド・ジョブを利用できないようにしてある。省略された場合は、再入力を要求する。

例 1. 登録番号を正しく入力した場合

```
?USER-ID.....xxxxx,xxxx  
/YOUR JOB-NO....DJIA001  
MACRO BUN NO NYUURYOKU.  
**  
:  
:  
:
```

例 2. 登録番号を入力しない場合

```
?USER-ID.....xxxxxx  
/YOUR-PASSWORD....xxxx  
/YOUR JOB-NO.....DJIA001 -----正しいコード  
MACRO BUN NO NYUURYOKU.  
**  
:  
:  
:
```

例3. 間違った登録番号を入力した場合

```
?USER-ID.....XXXXX,XXXX  
/*PASSWORD ERROR*
```

⋮
⋮
⋮

OFF-LINE状態への移行

また、センターの運用部門に与えられているコードは、ASW8をONにすることによって、外部端末からも入力できるようにしてある。これは障害が発生した場合、ハード、ソフトのチェックを外部端末からしてみる場合、入力を特定コードだけに限定するとこのようなセンタ業務として使用する場合であっても、外部端末が使用したという形で会計ファイルに記録されてしまうという不都合を生ずるためである。

マスター・パスワードは、(3)のファイル名の項でも述べるように、ある端末から作成したファイルは、ソースプログラムのセーブ・ファイルも含めて、原則としてそのファイルを作成した端末からでないとアクセスできないようにしてあるが、センターのように端末が複数台設置してある場合は、同一人が何時も同一端末を使用する、あるいは使用できるとは限らないため、他の端末で作成したファイルもアクセスできなければ不都合となる。このような場合に、このマスター・パスワードを入力することによって他の端末で作成したファイルでもアクセスできるようにしたものである。そのため、このマスター・パスワードは外部端末の利用者には全く連絡されていない。

(2) ジョブ番号の発行

デマンド処理における端末利用者にとっては、ジョブ番号といったものは特に必要としないものであるが、MONITOR-Vでは入力されるジョブはすべてジョブ番号(名)で区別されているので、デマンド・ジョブにおいてもジョブ番号といったものが必要となる。

デマンド・ジョブのジョブ番号の発行に関するMONITOR-Vの標準は、

DJ 99999 (99999 は数字部で 0 からの連番号) である。すなわち、 デマンド・ジョブが開設されるたびに 1 ずつ増加するカウンターがあり、 その内容の頭に DJ (Demand Job) という文字を付加したものがジョブ番号となる。

ANATA NO JOB BANGO = DJ 00001

これだと、 ジョブ番号が同一になる心配はないが、 センタ側のコンソール、 シート上では、 どこの端末から起動されたジョブであるか直ちに判断することができないため、 操作上不便を感じる場合がある。

例えば、 操作指令 D JA (Display Job All) において現在実行中のジョブを見たい場合、

ZIKKOTYU DJ 00001, Z 015, DJ 00002

というように表示された場合、 DJ 00001 と DJ 00002 がどの端末から起動されたジョブであるかは、 これを見ただけではわからない。そのため、 DJ 99999 の数字部分を 2 行へらし J 99 のかわりに端末名を付加することとした。

各端末には、 System Generation 時にそれぞれ名前を指定することになっており、 DJ のかわりにこの端末名が付加されていれば今実行中のデマンド・ジョブはどの端末から起動されたものであるか、 また、 どの端末が現在使用中であるかを直ちに知ることができる。

例 J IA という端末名をもつ端末からデマンド・ジョブを起動すると、 次のようになる。

(修正前)

DEMAND

ANATA NO NAMAE = ××× · · · ×

ANATA NO PASSWORD = ××× · · · ×

ANATA NO JOB BANGOO = DJ 00001

MACRO BUN NO NYUURYOKU.

**

(修正後)

DEMAND

?USER-ID...XXXXXX, XXXX

/YOUR JOB-NO....DJIA001

MACRO BUN NYUURYOKU.

**

(3) ファイル名

概要でも述べたとおり、デマンド処理やリモート・バッチ処理においてもファイル名に対する制限はなるべくさけるべきであろう。

しかし、そのためにはOS側で何等かの手段を講じなければならないが、ファイル名に関してはセンター・ルーチンの一部として次のような機能をもたせた。

- ① デマンド・ジョブ、リモート・バッチ・ジョブで指定されたファイル名には、システムが使用する特定なファイル名以外には、そのジョブを入力した端末名と・（ピリオド）を付加する。

例

¥FD F01, FILE=(NEW, F1),.....

この制御カードがデマンドあるいはリモート・ジョブとしてJIA端末から入力されると、ファイル名の頭に端末名が付加されて

¥FD F01, FILE=(NEW, JIA·F1),.....

という制御カードに変更される。

- ② 特殊なパスワード（マスター・パスワード）が指定された場合は、端末名を付加しない。

①の機能によって端末利用者は任意のファイル名を指定することができる。例えば、端末名JIAからAというファイルを指定すればそれはJIA·Aというファイル名となり、端末名がJIBからAというファイル名を指定すれば、それはJIB·Aというファイル名となる。それゆえ、端末利用者はファイル名の頭に端末名が付加されているということは、一切知

る必要がなく、ファイル名としてAまたはBと指定することによって自由にアクセスすることができる。

②の機能は異なる端末間でファイルを共用する場合や、バッチ処理によって作成したファイルをデマンドやリモートバッチでアクセスする場合に必要である。これはバッチで指定したファイル名は当然指定したままのファイル名となっており、これをそのままのままデマンドやリモートで参照すると、そのファイル名の頭に端末名が付加されるために異なるファイル名となってしまうためである。

(4) 優先権の指定

MONITOR-Vでは、3.1.5 センタ標準値(1)各種優先権の項で述べたように、各ジョブ(バッチ、リモート・バッチ・ジョブ)にユーザが優先権を指定できるようになっている。当然通常は、これらを指定せず System Generation 時に定められた標準的な優先権が使用されるが、利用者が意識的に高い優先権を指定すると他のジョブに与える影響は非常に大きく、例えば、バッチ・ジョブやリモート・バッチ・ジョブの実行優先権をデマンドと同じ、あるいはそれ以上にするとデマンド・ジョブがほとんど動かなくなってしまう。そのためバッチ、リモート・バッチ、デマンドといったもの同時にサービスする場合には、ユーザの優先権の指定を禁止するといった方法をとらなければならない。バッチ処理においては、受付時に人間による優先権の指定の有無をチェックすることも可能であるが、リモート・バッチ処理では不可能である。このため、バッチ、リモート・バッチともセンタ・ルーチンによって特殊なキーが付されていない場合は、たとえ優先権の指定があってもそれを無視するようにした。

(5) 端末の“OFF-LINE”移行時点の変更

デマンド処理において、処理プログラムで RETURNマクロや ABORTマクロ(FORTRANのSTOPステートメントや\$STOPコマンドに相当)を実行したり、利用者が KILL コマンドを入力すると、MONITOR-Vは端末の状態を“OFF-LINE”に移行させるのが標準である。

OFF-LINEに移行してしまうと、同じ利用者が続けて次の作業をしよ

うとする場合でも、再び DEMAND コマンドや、利用者名、登録番号といったものの入力から始めなければならない。デマンド処理においては、同じプログラムでデータを変えて何度も実行させたり、セーブ (Save) してあるプログラムを呼び出して次々に実行したりすることが多いが、そのたびに DEMAND コマンドから再入力しなければなるとかなり面倒である。そのため、例えば、KILL というコマンドを実行しても OFF-LINE に移行させず、次のマクロ・コマンドの入力を持つようにし、その状態で "BY E" というコマンドが入力されたときに限り、OFF-LINE へ移行するようにした。

例1. 修正前

DEMAND

?USER-ID....XXXXXX,XXXX

·
·
·

**DFORT

·
·
·
·
·

¥STOP

DEMAND JOB OWARI

ZIKOKU XX:XX NI OFF-LINE NI NATTA SIYOOZIKAN XX:XX

DEMAND ← 次のジョブの開始 } 再入力
?USER-ID....

**DFORT

·
·
·
·

例2 修正後

DEMAND

?USER-ID.....XXXXXX,XXXX

**DFORT

¥STOP

//TSUGI WA? COMMAND OR "BYE"? ZIKOKU XX:XX XX:XX:XX

JOB OWARI

/YOUR JOB-NO....XXXXXXX

MACRO BUN NYUURYOKU

**DFORT ← 前のジョブが終了すると、マクロ文
の入力待状態になる

¥STOP

**BYE ← OFF-LINEへ移行させるコマンド

//CPU-TIME XX:XX:XX.XXX

ANATA WA DEMAND NO RIYOO O YAMETA.

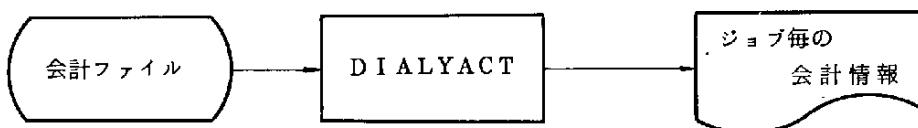
ZIKOKU.....

3.2.3 処理プログラム

処理プログラムとして作成したものは主として会計情報ファイルの処理プログラムであり、以下のようなものがある。

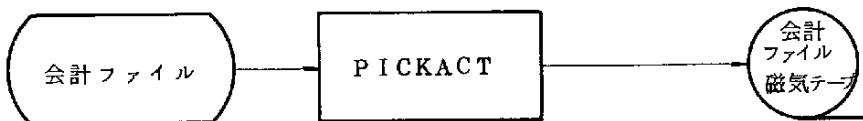
- (1) ジョブ毎の会計情報をプリントするプログラム (DAILYACT)

制御プログラムが作成した会計ファイルを読み、各ジョブ毎の完了状態、C P U 使用時間、CORE 占有時間、各種優先権、データ入力量、出力量等を印刷する（出力結果は付録一Ⅲ参照）。



- (2) 会計ファイルを磁気テープへ書き出すプログラム (PICKACT)

制御プログラムが作成した会計ファイルを読み出し、ブロッキングして磁気テープファイルを作成する。



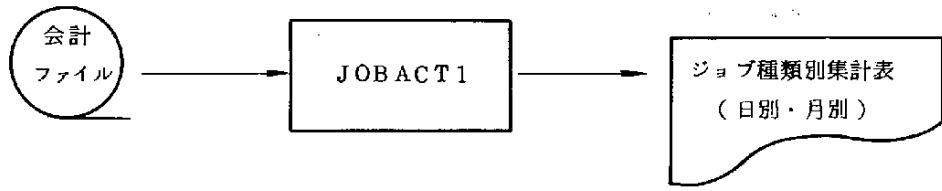
- (3) 会計ファイルを累積するプログラム (ACT·MERGE)

(2)で作成される磁気テープを累積するプログラム



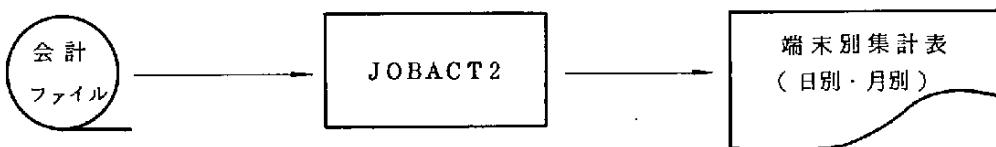
- (4) ジョブ種類別会計情報集計プログラム (JOBACT1)

ローカル・バッチ、リモート・バッチ、デマンドのジョブ種類別に日別、月別の使用件数、C P U 使用時間、CORE 占有時間、入力データ量、出力データ量等を集計する。



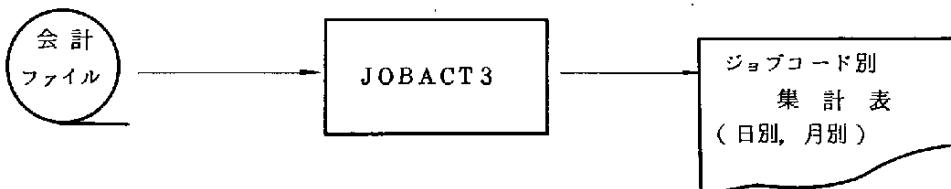
(5) 端末別会計情報集計プログラム (JOB ACT 2)

端末毎のCPU使用時間, CORE占有時間, 入力データ量, 出力データ量等を日別, 月別に集計する。



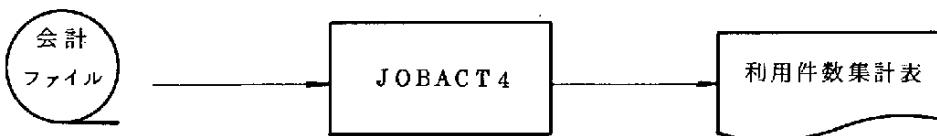
(6) ジョブコード別会計情報集計プログラム (JOB ACT 3)

ジョブコード毎のCPU使用時間, CORE占有時間, 入力データ量, 出力データ量等を日別, 月別に集計する。



(7) システム・プログラムの利用件数集計プログラム (JOB ACT 4)

アッセンブラー (FASP), コンパイラ (FORTRAN, COBOL, PL/I等) リンケージエディタ (LIED) 等がどの位利用されているかを集計する。



3.3 安定性

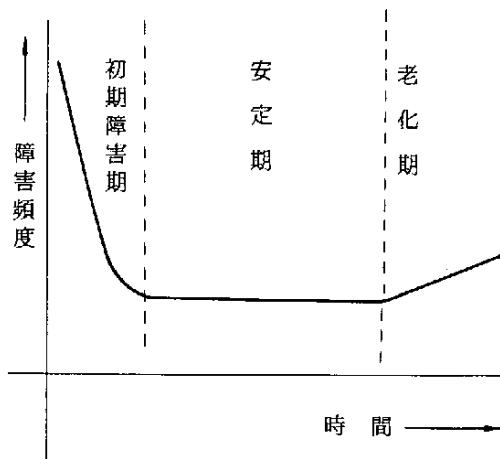
計算機システムの基本的条件は、ハードウェア、ソフトウェア全般にわたる安定性にあるといつても過言ではない。バッチ処理専用システム、あるいはTSS専用システムであっても勿論であるが、バッチ処理形態、リモート・バッチ処理形態、デマンド処理形態の3元処理でのシステム運用サービスにおいては、特にこの安定性ということが生命となってくる。

ある処理形態での障害が、その処理形態の範囲内でカバーできず、他の処理形態のジョブに影響を与えたたり、または、ただちにシステム全体の停止、すなわちシステム・ダウンにつながることもある。タイム・シェアリング・システムでは、バッチ処理などと違ってデータの発生源が外部にあり、かつ、データの発生がまったく不規則であるため、システム・ダウン時点におけるデータおよび関係ファイルなどの復元作業が、非常にむつかしい問題となってくる。これが原因でユーザとのトラブルが発生することもある。

このように専用システムであれば、その範囲内ですんでしまう問題でも、バッチ処理を含んだ多重処理を行なっているTSSシステムでは、その影響がきわめて大となる。

一般的に、ハードウェアの障害頻度は、右のグラフに表わした傾向をたどるのが通念となっている。つまり、導入または、増設当初の調整不十分とか、不安定であった部分が不良部品となってしまうというようなことから障害が頻発する初期障害期を経て、安定期に入り、ついに部品劣化などが原因で障害が発生する老化期に入り、再び障害が多発するようになる。

当実験のハードウェア・システムは、昭和47年8月に高速磁心記憶装置、昭和47年9



月に磁気ドラム装置、集合磁気ディスクパック装置、セレクタ・チャネル、ファイル・コントロール装置を増設し、昭和47年10月にチャネル装置、高速磁心記憶装置、中央処理装置の改造作業を行なった。増設あるいは改造作業後の総合テストをメーカー側では十分行なっているはずであるが、やはり総合テストで発見できなかった障害が多発し、初期障害期に見られる現象が発生した。これらの障害は単なる装置障害にとどまらず、システムダウンになってしまふものが大多数を占めていた。

主に、メモリ関係、チャネル関係の重要装置にダウンの原因があった。現在では比較的安定してきているが、障害が多発した初期障害期から定期移るまでに、約2ヶ月から3ヶ月の時間が必要であった。このようなことから、できることならばTSSサービスの開始当初から、最終的な機器構成でスタートし、途中で機器を増設することは、極力避けたほうが賢明であるようである。

何らかの事情で、やむを得ず増設する場合には、細心の注意を払う必要がある。

次に、ソフトウェア関係について述べると、バッチ処理機能だけのオペレーティング・システムMONITOR-Vでは、さほど問題はないと思われるが、TSS機能が追加されたMONITOR-Vでは、それだけ負担が大きくなり、各機能相互のからみ合いなどで障害の発生率も多くなってくる。

システムが導入されてから当実験が開始された昭和47年2月までの過去3年間に、数多くのバージョン・アップとシステム・テストの繰り返しを行ない、一応の安定をみせてきたが、まだまだ満足する状態ではなかった。その後、約1年を経過して現在に至る間、さらに数度のバージョン・アップを行なって安定化を図ったため、実験当初より安定度は徐々に良くなっている。

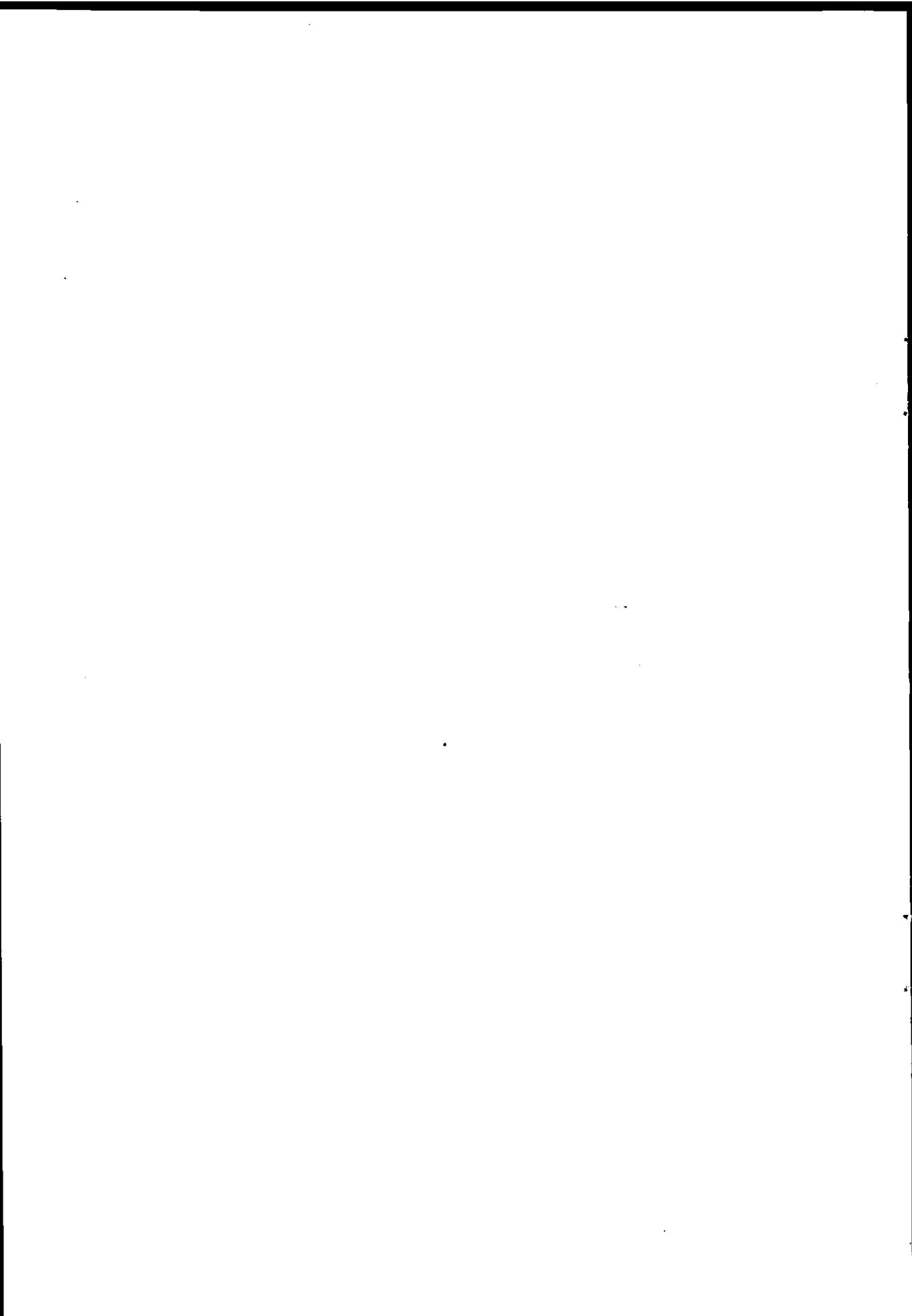
しかし、障害が発生するときは集中的に発生する傾向にあるように感じられる。また、バージョン・アップされた時に感じられることは、修正ミスによって時にはレベルダウンとなって、バージョンアップ直後の障害が多くなる傾向があり、残念に思われる。バージョン・アップ時の問題点として、十分注意を払わなければならないところであろう。

ソフトウェア障害の内訳は、やはりハードウェア障害と同じく、システムダウンとなる障害がシステムダウンに結び付かない比較的軽い障害よりも多く発生している。

3元処理のシステム運用サービスで、ハードウェアおよびソフトウェアが原因となって発生するシステム・ダウンを、どれ位の割り合いで許せるかは人によってそれぞれ意見が異なり、ある基準を設けるのは困難な問題であると思うが、利用者の立場からしてみれば、一度も発生しなければそれにこしたことではなく、理想的なことであろう。

しかし、これはあくまで理想であり、相手が人間の作ったハードウェアやソフトウェアであれば現実的でなくなってくる。現実的に考えてみると、ある計算センタでは1ヶ月に1回の割り合いならば一応満足する数字ではないかとしているが、当センタのシステムでは、MTBF (Mean Time Between Failure. 一度システム・ダウンしてから次のシステム・ダウンまでの平均時間) が50時間位になることを当面の希望としている。つまり、現在のTSSサービスは、月曜日から金曜日の10時から15時までとなっているので、10日間に1度位の割り合いになる。今後、種々のハード的チェック機能、ソフト的チェック機能の充実、各部品の品質向上、ソフト技術、ハード技術の向上などによって安定化を図ることを希望する。メーカー側も、エラーチェック回路の付加または、PADIA (Patrol Diagnosis Program)と呼ばれる巡回診断プログラムの開発などによって、障害を早期に積極的に検出する手段を講じており、これらの手段によってMTBFあるいはMTTR (Mean Time To Repair. システムダウンから回復するまでの平均時間) が改善されることを期待したい。

4. デマンド処理



4. デマンド処理

FACOM 230-60 MONITOR-Vによる外部端末(通産省、郵政省、日本電信電話公社、(財)機械振興協会経済研究所)も含めたデマンド処理の応用実験は、昭和47年2月から開始した。

当初は隔日サービスから始め、徐々に時間帯を延長し、5月より土曜日を除く毎日(10時~15時)がサービスの対象となった。また、デマンド処理の対象となる処理プログラムも、最初はメーカ提供のBACCUS、LINEDから開始し、5月からは当センターで開発したTSS-FORTRAN、10月にはCPLのサービスを開始した。

過去3年間NEAC2200-500 MACシステムによるタイム・シェアリングの応用実験を行なった経験があるが、本年度はFACOM230-60 MONITOR-Vの3元処理による応用実験であり、MACシステムとはまた異った面が実験結果として提起されたことと思われる。

この章では、各端末の利用状況、サービスの対象となっている各処理プログラムの簡単な言語仕様、およびそれに対する検討、最後に各端末の利用者自身による利用状況とその所感について述べられている。

4.1 利用状況

昭和47年5月~昭和48年2月までの利用状況は次のとおりである。

- (1) 表4-1 ジョブ種類別月別利用件数
- (2) 図4-1 "
- (3) 表4-2 端末別月別利用件数
(デマンド処理およびリモート・バッチ処理)
- (4) 表4-3 プログラム別端末別利用件数(デマンド)
- (5) 表4-4 プログラム別利用件数

表 4 - 1 ジョブ種類別月別利用件数

項目 月	使 用 件 数			C P U 使用 時 間 (HH:MM' SS)			C O R E 占 有 時 間 (HH:MM' SS)		
	デ マ ン ド	リ モ ト ・ パ ッ チ	ロ カ ル ・ パ ッ チ	デ マ ン ド	リ モ ト ・ パ ッ チ	ロ カ ル ・ パ ッ チ	デ マ ン ド	リ モ ト ・ パ ッ チ	ロ カ ル ・ パ ッ チ
5	621	113	1,520	2:21' 27	0:01' 01	13:43' 30	83:55' 15	0:34' 23	44:30' 10
6	939	91	1,630	2:32' 07	0:08' 35	20:29' 45	94:11' 48	0:50' 46	75:19' 46
7	938	18	1,627	4:33' 40	0:00' 29	16:54' 12	203:25' 45	0:25' 29	57:03' 47
8	568	105	2,143	2:40' 37	0:11' 59	25:26' 31	109:54' 20	1:23' 53	84:01' 20
9	467	124	2,087	2:36' 19	0:24' 43	20:28' 58	94:59' 59	2:13' 34	80:30' 13
10	516	291	2,240	3:05' 51	1:02' 14	19:31' 56	128:16' 56	5:24' 48	95:57' 12
11	733	538	3,050	3:34' 18	5:59' 03	42:00' 59	178:42' 24	16:04' 56	180:02' 26
12	417	361	2,901	1:31' 33	3:05' 50	53:39' 20	95:16' 23	10:49' 29	169:06' 07
1	795	293	2,266	5:47' 24	5:43' 45	35:10' 55	187:55' 38	14:33' 01	145:32' 05
2	1,036	204	2,050	5:13' 28	1:08' 54	33:47' 57	274:26' 20	6:28' 38	112:47' 41
計	7,030	2,138	21,514	33:56' 44	17:40' 33	281:10' 03	1452:14' 48	58:48' 57	1044:50' 47
平均	703	214	2,152	3:24' 07	1:47' 03	28:07' 00	145:13' 00	5:53' 00	104:29' 00

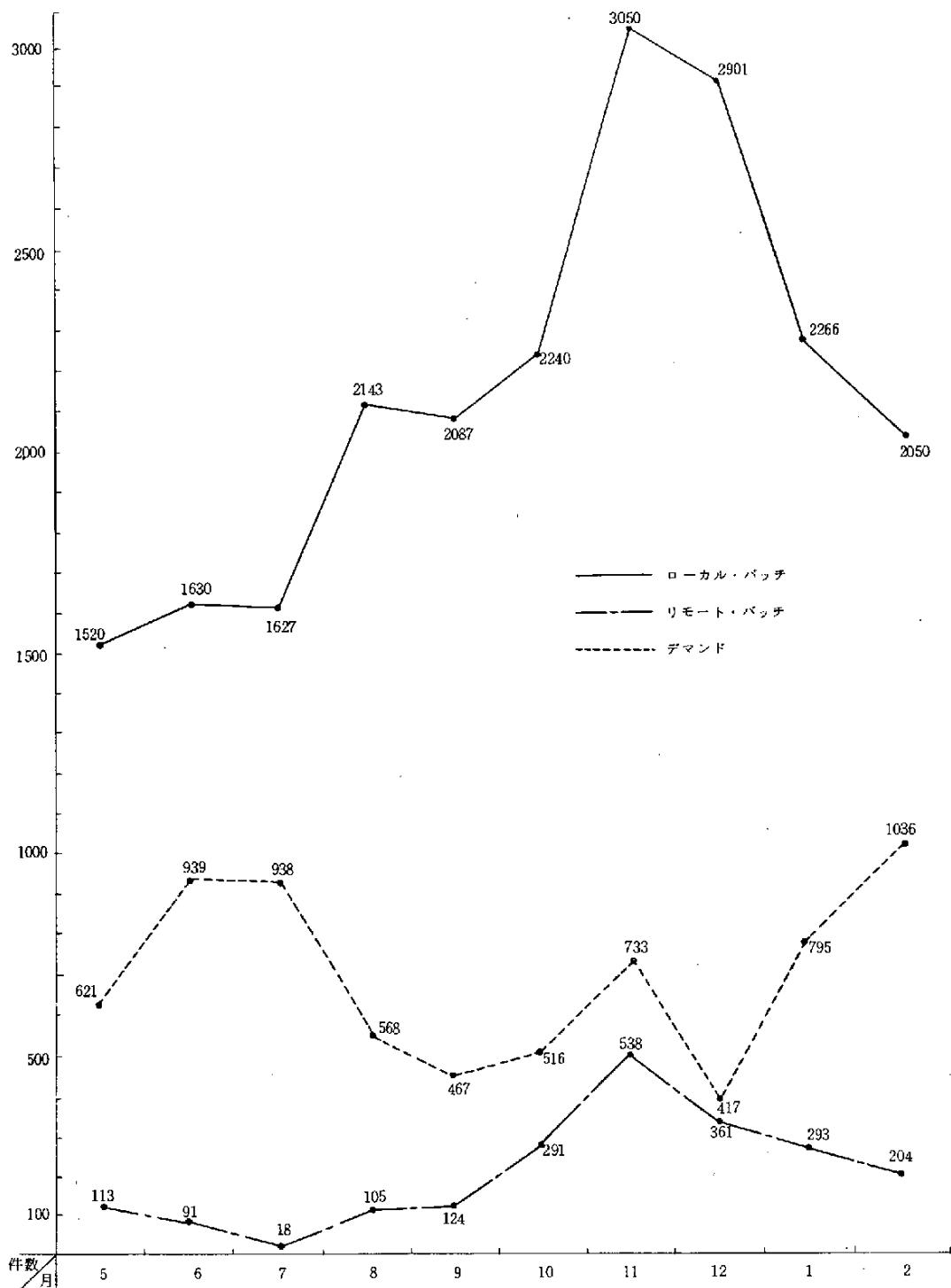


図 4-1 ジョブ種類別月別利用件数

表 4 - 2 端末別・月別利用件数(デマンドおよびリモート・バッチ)

端末名 月	JIA (センタ)	JIB (センタ)	JIC (センタ)	JID (センタ)	JIE (センタ)	JIF (センタ)	MIT (通産省)	MPS (郵政省)	NTT (電電公社)	JSP (機械協)	MEL (工技院)	TTI (総務)	計	平均
5	126	68	14	198	253	59	5	0	1	2	0	8	734	61.2
6	162	114	35	107	482	33	0	7	44	24	0	22	1,030	85.8
7	173	165	127	136	183	34	0	0	25	107	0	6	956	79.7
8	105	119	22	142	141	28	13	43	0	29	31	0	673	56.1
9	54	98	46	68	206	23	3	18	10	17	47	1	591	49.3
10	58	47	97	146	303	17	0	66	2	17	54	0	807	67.3
11	125	58	146	272	398	40	1	41	35	5	149	1	1,271	106.0
12	58	98	73	191	259	10	4	53	26	0	6	0	778	64.9
1	178	236	102	94	82	47	5	16	16	25	0	0	795	66.3
2	293	280	58	142	316	56	1	6	45	17	26	0	1,240	103.4
計	1,332	1,283	720	1,496	2,623	347	32	244	204	243	313	38	8,875	739.6
平均	1,332	1,283	720	1,496	2,623	347	3.2	244	20.4	24.3	31.3	3.8	887.5	

CPU 使用時間計	9:45' 48	4:06' 43	.5:48' 59	9:23' 29	8:17' 49	0:39' 46	0:03' 51	2:33' 02	0:17' 15	0:26' 39	4:10' 43	0:06' 40
CORE 占有時間計	251:26' 07	156:00' 28	119:19' 53	137:54' 11	161:54' 11	35:21' 55	9:21' 18	168:39' 00	76:34' 05	88:22' 44	11:45' 44	2:07' 21

表 4 - 3 プログラム別・端末別利用件数(デマンド 47.5 ~ 48.1)

端末名 使用された言語	JIA (センタ)	JIB (センタ)	JIC (センタ)	JID (センタ)	JIE (センタ)	JIF (センタ)	MIT (通産省)	MPS (郵政省)	NTT (電電公社)	JSP (機械協)	計	平均
TSS-FORTRAN	185	89	253	263	456	17	28	224	107	220	1,842	184.2
CPL	556	169	103	235	266	105	0	0	8	0	1,442	144.2
BACCUS	29	60	50	93	140	33	2	0	12	3	422	42.2
LINED	6	1	0	4	12	0	0	0	6	0	29	2.9
JOLDOR	40	206	0	0	12	66	0	0	0	0	324	32.4
GETF(Get file)	4	1	1	13	8	0	1	0	0	0	28	2.8
DELF(Delete file)	8	2	23	22	23	0	0	0	6	0	84	8.4
BELL	10	0	1	23	0	0	0	0	0	0	34	3.4
CLASFF	189	192	14	31	213	10	0	0	0	0	649	64.9
TEST	0	146	0	55	282	38	0	0	0	0	521	52.1
SIMBOL	0	0	0	0	71	0	0	0	0	0	71	7.1
その他	33	103	12	31	123	19	0	0	0	0	321	32.1

表 4 - 4 プログラム別利用件数(47.5~48.1)

処理プログラム名	利 用 回 数
LIED	10,096 回
LIBE	6,217 "
FASP	5,946 "
FORTRAN	5,727 "
COBOL	2,984 "
FSUPROGM	1,013 "
FSUMMOVEZ	736 "
SORT	599 "
FSUDADMP	566 "
FSULIST	528 "
PL/I	426 "
BACCUS	425 "
FSUMTDMF	229 "
GPSS	39 "
SYSGEN	30 "
LINED	30 "
DYNAMO	26 "
ALGOL	20 "
その他	60 "
TSS-FORTRAN	1,842 "
CPL	1,448 "
JOLDOR	813 "
CLASS	649 "
DIAL	589 "
PLD	437 "
TEST	997 "
JMACCT	263 "
JYONINOS	389 "
他のユーザプログラム	10,717 "
計	53,144 "

メーカー提供のシステム・プログラム

ユーザ・プログラム

4.2 処理プログラムについて

現在 TSS サービスで利用者に提供している言語処理プログラムおよびユーティリティには次のものがある。

TSS-FORTRAN (会話型 FORTRAN)

CPL (会話型 PL/I)

BACCUS (Basic Calculas)

LINED (Lined data set editor)

各種編集用ユーティリティ

上記中、BACCUS までは言語処理プログラムである。次に言語処理プログラムについて簡単に説明する。

4.2.1 言語仕様

(1) TSS-FORTRAN

FORTRAN と CPL (会話型 PL/I) は従来バッチに使用されてきた言語であるが、一般に、表現能力が大きく、利用価値の高いもので、TSS を行なうについて一応用意しなければならない言語ではないだろうか。特に、FORTRAN は科学技術計算用のために 1954 年から約 2 年半の月日をかけて開発され、その後修正を重ね、1962 年には FORTRAN IV が発表されており、文法も比較的簡単であり、経験のうすい人でも理解しやすいため現在最も広く使用されているプログラミング言語であり、TSS サービスを行なう上で不可欠な言語である。

この TSS-FORTRAN プロセッサは、バッチ用 FORTRAN コンパイラーに手を加え、会話的使用を可能にしたいわゆる一括コンパイラー型システムである。一括コンパイラー型の特長としては、バッチ用コンパイラーを使うので、事実良い目的プログラムが得られるし、バッチとの互換性もある。しかし、ステートメントの追加、削除、変更等の修正があるつど翻訳、結合の処理をする必要があり、この点が少々煩わしい。このプロセッサーのもつ機能は、

1. 1ステートメント入力毎にシンタックス・チェックを行なう。
2. ソース・プログラム入力はフリー・フォーマットを許す。
3. READ文, WRITE 文ではシステムで定めた標準フォーマットにより入出力できる。
4. 端末を入出力の対象にできる。
5. システムは、入力プログラムに行番号をつけ、利用者はこの行番号を使って各種編集〔置換、挿入、削除〕を行なうことができる。

また、入出力文の装置番号と実際の装置との対応は、次の表のようになっていいる。

装 置 番 号	バ ッ チ	会 話 型
1 ~ 4	磁気テープ 磁気ディスク	磁気ディスク
5	カード読み取り装置	ソース・データ・ファイルの入力
6	ラインプリンタ	端末タイプライタ
7	カードせん孔装置	端末ラインプリンタ〔FACOM-Rのみ〕
8	磁気テープ	端末入力〔FACOM-Rはカード・リーダも含む〕
9 ~ 16	磁気ディスク	未 使用

次に会話型FORTRANとFACOM 230-60バッチ用との相違点を列記してみると、

1. BLOCK DATA文の禁止。
2. PAUSE文の禁止。
3. REAL FUNCTION 等のように型宣言を伴ったFUNCTION文の禁止。
4. ソース・プログラムの第1行目がコメント行であってはならない。

以上はJIS FORTRAN水準7000にあってTSS-FORTRANで禁止しているもの。

5. ELEMENT文の禁止。
6. OPTION文の禁止。
7. NAMELIST文の禁止。

8. LCM文の禁止。

以上は FACOM 230-60 FORTRAN固有の機能で,TSS-FORTRANで禁止しているもの。

このTSS-FORTRANのもつているコマンドには、次のようなものがある。

INPUT	RUN
IEND	SAVE
UPDATE	PURGE
UEND	LOAD
LIST	NAMELIST
DELETE	CHMODE
￥行番号	STOP
RESEQ	SYNTAX

これらのコマンドのパラメータおよび機能は表4-5コマンド一覧表のTSS-FORTRANコマンドを参照されたい。

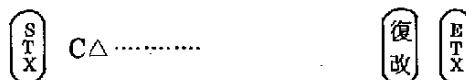
ステートメントの機能、一般形等は原則としてFORTRANの文法に従う。

TSS-FORTRANの入力形式は次の形をしている。

1. ソース・プログラムの入力

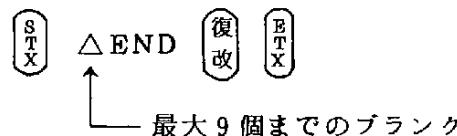


2. コメント行

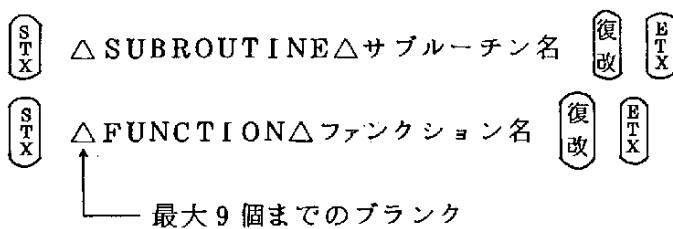


コメント行に書ける文字数は72文字までである。

3. END行



4. SUBROUTINE文およびFUNCTION文



5. 上記以外のFORTRAN文



6. 繙続行

継続行は **復改** の直前に & を入力する。

&は一つの文に対して最大8個まで使用できる。

特に指示されていない△印は1個のブランクを表わす。

端末装置としてFACOM-Rを使う場合は、 **S T X** **E T X** は不用である。

データの入力はREADステートメントを実行すると、

DATA △

と出力して入力状態になるので適当なデータをタイブインする。

TSS-FORTRANの簡単な使用例は、83ページにあるので参照されたい。

(2) CPL

CPLは会話型のPL/Iで従来、プログラミング言語は科学技術計算用とか事務計算用とかに分かれていた。例えば前者はFORTRAN, ALGOL等で後者はCOBOLに代表されるが、言語を主要目的に限定してしまうとコンピュータのもつている性能や機能をフルに活用できなくなる。そこで考えられたのがPL/Iであり、幅広いプログラミングが可能となった。CPLをサービスすることにより端末利用者は柔軟性をもったプログラミングができる。

CPLはインタプリタ型とコンパイラ型の中間をとり、できるだけ機械

語のオブジェクトを出すようにしてある。CPLの本体は、次の各処理ルーチンより成っている。

- コンパイラ

入力されたソースプログラムを翻訳し、オブジェクトを作る。また、ステートメントの挿入、置換、削除もこのルーチンで行なっている。

- エグゼキュータ

コンパイラによって作られたオブジェクトの実行をコントロールする。

ダイレクト・ステートメントが入力されたときは、それを翻訳したあとコントロールをエグゼキュータに移す。

- インタプリタ

入出力関係あるいは代入ステートメントの一部等、完全な機械語のオブジェクトを作りにくいステートメントを中間言語にしておき、それを実行時に翻訳する。

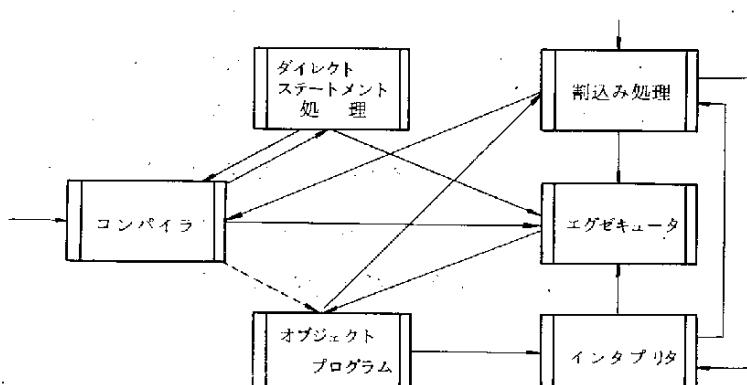
- 割込み処理

オブジェクト・プログラムおよび、インタプリタ実行中に割込み条件が発生した場合その処理をする。

- ダイレクト・ステートメント処理

¥RUN（実行開始の指示）や¥SAVE（プログラムの保存指示）等のステートメントの先頭に¥又は%を付したものを見つけると、それを翻訳、実行する。

これらの関係は次の図のようになる。



CPLで使用禁止となっている主なものを列記してみると、

1. 多重タスク。
2. INITIAL 指定の代りにダイレクト・ステートメントを許し、コンパイル時に実行させる。
3. 構造体の配列。
4. 添字つきの修飾された名前。
5. 配列の断面。
6. BEGIN ブロック。
7. 再帰的手続き。
8. ポインタなしの CONTROLLED 変数。

CPLにおけるステートメントは、コレクト・ステートメントとダイレクト・ステートメントに分かれる。コレクト・ステートメントは通常のプログラム・ステートメントである。ダイレクト・ステートメントは先頭に¥または%を付したもので、入力されるとただちに実行される。また、ある種のコレクト・ステートメント（代入、GET、PUT、STOP等）の先頭に¥または%をつけることによりダイレクト・ステートメントとして使うことができる。

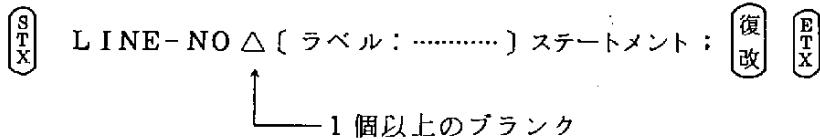
CPLのもっているコマンド（ダイレクト・ステートメント）には、次のものがある。

RUN	NAMELIST
LIST	NOMSS
SAVE	PURGE
LOAD	MSS
ERASE	GET
COM	PUT
NORECOM	STOP

これらのコマンドのパラメータおよび機能は表4-5 コマンド一覧表の CPL コマンドを参照されたい。

ステートメントの機能、一般形等は原則としてPL/Iの文法に従う。

ステートメントの入力形式は一般に次に示す通りである。



LINE-NO は整数部 4 桁、小数部 2 桁までの10進数を入力することができるが、この LINE-NO によってステートメントの挿入、交換が任意に行なえるようになっている。

端末装置として FACOM-R を使う場合、復改 ETX は不用である。

コメントは LINE-NO の後であれば /* と */ で囲んで任意に入れることができる。ステートメントの継続は、CPLでは復改 を任意の時点で入力することができるので、ステートメントを 2 行以上にすることも可能である。

CPLの簡単な使用例は、84ページにあるので参照されたい。

(3) BACCUS

BACCUSはメーカが開発した会話型言語である。この言語の特長は、

1. 端末からプログラムの作成、実行、管理を行なえる。
2. 必要ならばFORTRANサブルーチン副プログラムを結合して使用できる。
3. 会話を一時中断しても、後でこれを復元して会話を続行させることができる。
4. 簡単な計算から、高度な問題解法のプログラム作成に至るまで取り扱うことができる。

文法自体はFORTRANに似ているとも、ALGOLに似ているという人もおり、いづれにしてもこれ等の言語の経験者には覚えやすい言語ではないかと思われる。しかし、他方例に示すとおり条件節、くり返し節等がうしろにあるため、他のプログラミング言語を知っている人は、異和感をもち、使いにくいと言う人もあるようである。

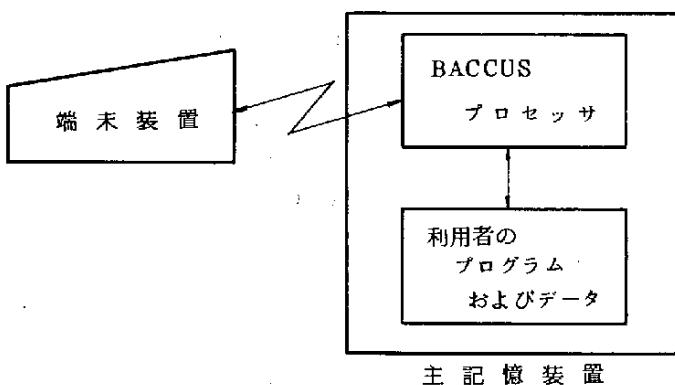
例

GO TO OWARI IF N=0 :

条件節

機能の面においては、現在のものでは文字の代入、比較やファイルを扱うことができないし、精度の点でも FORTRAN と比較してみると、倍精度が使えないため少々不満は残る。

BACCUS プロセッサは端末装置から入力されたプログラムを翻訳して、これを実行させる。



文は直接文と間接文に分かれ、直接文には文番号は付いておらず、入力された時点ですぐ実行される。間接文は文番号が付された文で、これは入力された時点ではなくては実行されず、直接文（例えば TO 文）で実行を指示してやらなければならない。そして、直接文が入力できる状態（MANUAL 状態という）と間接文が入力できる状態（AUTO 状態という）の切り換えは AUTO, MANUAL というコマンドを入力することにより行なえる。また、間接文入力中に一時的に直接文を入力したいときは、直接文の頭に／を付けて入力すればよい。なお、BACCUS プロセッサが開設された時点では、MANUAL 状態にある。

BACCUS プロセッサのもつ命令（コマンド等）は次のようなものがある。
直接文としてのみ使用することができる文

AUTO 文 DELETE 文 LIST 文 PAUSE 文 ARRAY 文

Q文 RUN文 SAVE文

間接文としてのみ使用することができる文

EXTERNAL文 FORMAT文 PAUSE文 ARRAY文

INTEGER文 REAL文 空文(ラベル等)

共通文として使用することができる文

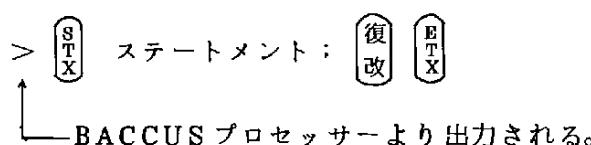
ACT文 CALL文 COMMENT文 SET文 STOP文

TO文 CRLF文 DEMAND文 TYPE文

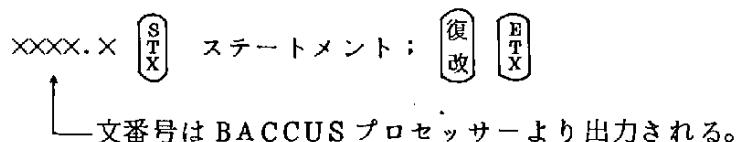
上記、共通文とは直接文としても、間接文としても使用できる文のことである。

ステートメントの入力形式は次のとおりである。

直接文入力の場合



間接文入力の場合



データの入力



端末装置として FACOM-R を使う場合は   は不用である。

BACCUS の簡単な使用例は、85ページを参照されたい。

各言語の使用例

TSS-FORTRAN

```
***D
? USER-ID...HOSHIYA.0512
/YOUR JOB-NR...DJIC040
MACRO BUN NYUURYOKU.
**DFORT
JOB KAISI.
$LOAD SQRTF
  SQRTF A LOAD カレタ
*LIST SQRTF
  00010 10  READ(8,END=9999)I
  00020      J=N+9
  00030  DD 20 I=N,J
  00040      X=1.0
  00050  30  DELT=(-X+I/X)/2.0
  00060      X=(X+I/X)/2.0
  00070  IF(ABS(DELT).LT.0.0001) GO TO 8888
  00080  GO TO 30
  00090  8888  WRITE(6,100)I,X
  00100  100  FORMAT(' *   SQRT (',I4,') = ',F11.8)
  00110  20  CONTINUE
  00120  GO TO 10
  00130  9999  STOP
  00140  END
*$UPDATE SQRTF
>$DELETE 20
>30 DD 20 I=N,N+9
>$END
$RESET Q SQRTF
*LIST SQRTF
  00010 10  READ(8,END=9999)I
  00020      DO 20 I=N,N+9
  00030      X=1.0
  00040  30  DELT=(-X+I/X)/2.0
  00050      X=(X+I/X)/2.0
  00060  IF(ABS(DELT).LT.0.0001) GO TO 8888
  00070  GO TO 30
  00080  8888  WRITE(6,100)I,X
  00090  100  FORMAT(' *   SQRT (',I4,') = ',F11.8)
  00100  20  CONTINUE
  00110  GO TO 10
  00120  9999  STOP
  00130  END
$RUN SQRTF
-----
```

```
DATA 1
*   SQRT (  1) =  1.00000000
*   SQRT (  2) =  1.41421357
*   SQRT (  3) =  1.73205081
*   SQRT (  4) =  2.00000000
*   SQRT (  5) =  2.23606795
*   SQRT (  6) =  2.44948977
*   SQRT (  7) =  2.64575130
*   SQRT (  8) =  2.82842714
*   SQRT (  9) =  3.00000000
*   SQRT ( 10) =  3.16227764
DATA 1E0F
-----
```

```
* END OF FORTRAN *
$PURGE SQRTF
```

CPL

```

***0
? USE R-ID...HOSHIYA,0512
/YOUR JDB-N0....DJIE055
MACRG BUN NYUURYOKU.
**C PL
JDB KAISI.
? ¥LOAD CSORT;
LOAD SWARI
LAST LINE(END ST) 0150.00
? ¥LIST;
0010.00 CSORT:PROC;
0020.00 START: PUT SKIP (3);
0030.00 GET LIST (N);
0040.00 IF N = 0 THEN GO TO SWARI;
0050.00 DO I=N TO N+9;
0060.00 X=1.0;
0070.00 KEI: DELT=(-X+I/X)/2.0;
0080.00 X=(X+I/X)/2.0;
0090.00 IF ABS(DELT) < 0.0001 THEN GO TO KAKU;
0100.00 GO TO KEI;
0110.00 KAKU: PUT EDIT(*      SORT (*,I,") = ",X)(A,F(4),A,F(11,8));
0120.00 END;
0130.00 GO TO START;
0140.00 SWARI: STOP;
0150.00 END;
? 40 IF N = 0 TIEN GO TO SWARI;
LAST LINE(END ST) 0150.00
? ¥LIST;
0010.00 CSORT:PROC;
0020.00 START: PUT SKIP (3);
0030.00 GET LIST (N);
0040.00 IF N = 0 THEN GO TO SWARI;
0050.00 DO I=N TO N+9;
0060.00 X=1.0;
0070.00 KEI: DELT=(-X+I/X)/2.0;
0080.00 X=(X+I/X)/2.0;
0090.00 IF ABS(DELT) < 0.0001 THEN GO TO KAKU;
0100.00 GO TO KEI;
0110.00 KAKU: PUT EDIT(*      SORT (*,I,") = ",X)(A,F(4),A,F(1
0120.00 END;
0130.00 GO TO START;
0140.00 SWARI: STOP;
0150.00 END;
? ¥RUN;

```

DATA 0030.00	N? 1
*	SQRT (1) = 1.00000000
*	SQRT (2) = 1.41421350
*	SQRT (3) = 1.73205080
*	SQRT (4) = 2.00000000
*	SQRT (5) = 2.23606800
*	SQRT (6) = 2.44948900
*	SQRT (7) = 2.64575130
*	SQRT (8) = 2.82842710
*	SQRT (9) = 3.00000000
*	SQRT (10) = 3.16227760

DATA 0030.00 N? 0
 //TSUGI WA ? COMMAND GR "BYE" ? ZIKOKU=02.20 15:50:42
 JCB SWARI.
 /YRIN 100 300 81000000

BACCUS

```
***  
? USER-ID...HOSHIYA.0512  
/YOUR JOB-NR...DJIC044  
MACRO BUN NYUURYOKU.  
**BACCUS  
JOB KAISI.  
    BC0001 BACCUS-1 (02-04) HAZIMARI.  
>LOAD SOURCE FROM BSQRT;  
    BC605M LOAD OWARI: SOURCE  
>LIST ALL:  
    1.0  START;  
    2.0  DEMAND N;  
    3.0  GO TO OWARI IF N=0;  
    4.0  ACT KEI1. OWARI FOR I=N UNTIL N+9;  
    5.0  GO TO START;  
    6.0  KEI1;  
    7.0  X=1.0;  
    8.0  K1;  
    9.0  DELT=(-X+I/X)/2.0;  
   10.0  X=(X+I/X)/2.0;  
   11.0  GO TO KEI2 IF ABS(DELT)<0.0001;  
   12.0  GO TO K1;  
   13.0  KEI2;  
   14.0  TYPE FORMAT 10: I,X;  
   15.0  FORMAT 10:*      SQRT ( %%% ) =%%.%%%%%%%%%;  
   16.0  OWARI;  
   17.0  PAUSE;  
>#1.1 CRLF 2;  
>1.2 TYPE "DATA INPUT (BSQRT )";  
>LIST 1.0,5.0;  
    1.0  START;  
    1.1  CRLF 2;  
    1.2  TYPE "DATA INPUT (BSQRT )";  
    2.0  DEMAND N;  
    3.0  GO TO OWARI IF N=0;  
    4.0  ACT KEI1. OWARI FOR I=N UNTIL N+9;  
    5.0  GO TO START;  
>GU TO 1.0;
```

```
DATA INPUT (BSQRT )  
N=1  
*     SQRT ( 1 ) = 1.00000000  
*     SQRT ( 2 ) = 1.41421357  
*     SQRT ( 3 ) = 1.73205081  
*     SQRT ( 4 ) = 2.00000000  
*     SQRT ( 5 ) = 2.23606801  
*     SQRT ( 6 ) = 2.44948977  
*     SQRT ( 7 ) = 2.54575130  
*     SQRT ( 8 ) = 2.82842714  
*     SQRT ( 9 ) = 3.00000000  
*     SQRT ( 10 ) = 3.16227770
```

```
DATA INPUT (BSQRT )  
N=0  
    BC6020  17.0  PAUSE.  
>STOP;  
    BC604M BACCUS-1 OWARI.  
//TSUGI WA ? COMMAND UR 'BYE' ?      ZIKOKU=02.16 19:20:38  
JOB OWARI.  
//YOUR JOB-NR...DJIC044  
MACRO BUN NYUURYOKU.  
**BYE  
//CPU-TIME 00:00:00.435
```

4.2.2 コマンド

コマンドとは、端末装置を使って利用者がOS（オペレーティング・システム）に対していろいろなサービスを受けるために用いる指令のことをいう。

コマンドはシステムと会話をするという意味から容易に使えるような構成が望ましく、通常次のような形式になっている。

コマンド名△[パラメータ1, [パラメータ2,]]
↑
1個以上のブランク

実際にコマンドを使うとき、コマンドの種類によっては、パラメータ情報が必要なときがあるが、このようなときは省略時の標準値を決めておき、変更の必要があるときパラメータを書くようにしておけば、利用の面で非常に便利である。

TSS関係のコマンドを大別するとシステム制御コマンド、マクロ・コマンド、それに各言語独自のコマンドに分けることができる。

(1) システム制御コマンド

利用者の要求によりシステムに対して問い合わせ、システムの状態変更等をおこなう。

TSSで使うシステム制御コマンドは別表のとおりだが、一応これだけそろっていれば不自由ないと思われるが、DISPLAY機能に全ジョブ（デマンド、バッチ）の稼動状態を表示できれば会話中応答時間が多少遅くなても納得いくし、また、その時間の使用を見合せることもできる。事実、端末操作中応答時間が遅くなると利用者は操作ミスなのか、センターのシステム・ダウンなのか少々不安になってくるものである。

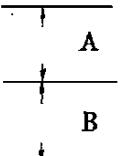
(2) マクロ・コマンド

処理プログラムを呼び出すために用いるコマンドで、これはジョブ制御言語カードをマクロ・ファイルに登録したものである。このコマンドは端末装置をデマンド状態にしたあと（システム制御コマンド“D”を入力する）最初に入れるコマンドであり、これによってどのような言語あるいはユーティリティを使うかがきまる。一般にこの種のコマンドはセンターによってコマ

ンドはセンターによってコマンド名、パラメータ情報が異なるため、利用するセンターのマクロ・コマンドは十分知っておくことが必要である。

このコマンドを使ってみて感じたことはDFORTコマンドを入力後、実際にTSS会話ができるまでにかなり時間がかかるということである。コマンド入力後会話可能状態になるまでには図に示すとおりA、Bの2つの過程を

**DFORT
JOB KAISI.
¥



経るが、Aにおいてはかなりの枚数のジョブ制御言語カードを解読するので時間がかかるのである。Bではデマンド・ジョブ開設後、会話に入る前にファイルの消去、編集をおこなうので時間がかかる。A、Bとも時間的には約30秒かかるが、このことは端末利用者にとって気になる1つである。ともすれば、初心者などは故障か、自分の誤操作と勘違いして他のコマンドを入れてみたり、端末の電源を切ってしまうことにもなりかねないのではないか。この辺の改善を望みたい。

(3) 言語独自のコマンド

これは各言語ごとにある直接文がそれに相当するものと思われる。勿論、そうでないものもある。このコマンドの使用に際しては各言語の文法等をよく理解しておくことが必要である。言語別のそれは別表を参照されたい。

これ等のコマンドを参考してみて改良して欲しい点、追加して欲しい機能等、若干気が付いたことを言語別に述べる。

• TSS-FORTRAN

1. NAMELIST機能により保存ファイルにあるプログラム名を端末に印刷するとき、プログラム名のうしろに作成日付、あるいは更新日付と一緒に印刷できないものだろうか。また、保存ファイルをBACCUSと共に用いているため、FORTRANプログラムとBACCUSプログラムとの区別がつきにくく何か印のようなものを付けて欲しい。

例

PROG1	73.01.23	(F)
プログラム名	作成日付あるいは 更新日付	FORTRANプログラムの意味

2. サブコマンド`¥DELETE`はプログラム入力中または`UPDATE`中のみ使用できるが、コマンドとしていつでも使えるようにしたほうが便利ではないか。そのため、若干パラメータ部の修正を要するが、次のようにしたらどうか。

`¥DELETE [プログラム名,] 行番号-1 [, 行番号-2]`

プログラム名はプログラム入力中または`UPDATE`中には省略できるが、その他のときは必ず付ける。行番号に関しては今まで通りとする。
例

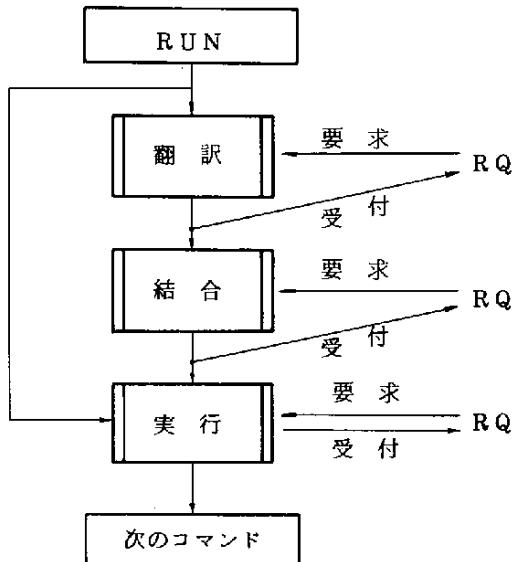
プログラムの入力中または`UPDATE`中に削除の必要が生じたとき。

`> ¥DELETE 70 70 のみ削除`

その他のとき

`¥DELETE PROG1, 40, 60 40~60 を削除`

3. ソース・プログラムの翻訳、結合時においてはシステム制御コマンド`RQ`は受けつけられないが、端末利用者としては何かの事情で処理を打ち切りたいのであるから、下図のように`RQ`コマンドを待ち状態にして翻訳終了時または、結合終了時に受けつけるようにすれば無駄な時間を使わずにすむのではないか。



4 PURGE コマンドに対して処理完了メッセージが出力されることを望む。

• CPL

1. CPLは行番号をマニュアルで入力するようになっているが、TSS FORTRAN のように自動的に出力されれば利用者にとって非常に助かるのではないか。このコマンドの追加を望む。
2. NAMELIST コマンドで共用ファイルのプログラム名を印刷させると、パラメータとして(C)を指定するが、単にCだけでよいのではないか。特にカッコはシフト・キーを押してから打つため煩わしい。
3. リシーケンス機能を望む。

• BACCUS

1. BACCUSでは一度セーブしたプログラムを更新して再セーブするとき、保存ファイル中のプログラムを削除する機能がないため、他の名前でセーブしなければならず不便である。PURGE コマンドのような機能を望む。
2. リシーケンス機能を望む。
3. NAMELIST 機能を望む。できれば TSS-FORTRAN の項で述べたようになればよい。

各言語共通して欲しい機能としてステートメントの部分修正機能がある。端末利用者はよく1文字修正するのに1ステートメントそっくり入力することを経験するが、目的の部分だけを入力するだけで済めば大へん助かる。

表 4 - 5 コマンド一覧表

システム制御コマンド

〔 〕内は省略形

コマンド名	パラメータ	機能	事柄
DEMAND [D]		デマンド・ジョブを開始する。	
KILL		デマンド・ジョブを強制消去する。	
OFF		端末装置をOFF状態に移行させる。	
CONT		データ入力状態で操作を誤り、入力不可になったとき、再び入力状態にする。	
REQUEST [RQ]		デマンド・ジョブに割込みを指示する。	
DISPLAY [DSP]	ジョブ番号	デマンド・ジョブの状態を表示する。	その端末のジョブのみ。
BYE		デマンド・ジョブを終了する。	

マクロ・コマンド(言語プロセッサー)

コマンド名	パラメータ	機能	事柄
DFORT	[Tn=OLD, Nn=ファイル名, Kn=KEEP]	TSS-FORTRANプロセッサーを起動する。	nは入出力文で使用している1～4までの整数。
BACCUS	[FN=ファイル名, PARAM=L32, VOL=0]	BACCUS プロセッサーを起動する。	ファイル名は保存用ファイル。
CPL	[FNn=ファイル名, TPn={ NEW, OLD }, DPn=DELETE, TNn=トラック数]	CPL プロセッサーを起動する。	nは1～5までの整数で何個目のファイルに対するパラメータかを区別する。 各コマンドの応答時間(つまりコマンド入力後、ステートメント入力記号が出るまでの

コマンド名	パラメータ	機能	事柄
			時間)は DFORT が一番長く、 平均約 1 分位かかる。 (マルチ・ジョブ実行中)

マクロ・コマンド(編集)

コマンド名	パラメータ	機能	事柄
FLMNT		デマンド用ファイルの再編成をする。	DFORT, BACCUSに使用できる。
FOR TDEL		SYSTEM ワークの消去をする。	
FLDEL	ファイル名	FORTRAN で作ったデータ・ファイルを消去する。	
LINED	[OLDファイルのエレメント名], [INファイル1のエレメント名], [INファイル2のエレメント名]	LINED プロセッサーを起動する。	

TSS-FORTRANコマンド

コマンド	パラメータ	機能	事例
INPUT	[プログラム名]	FORTRAN ソース・プログラムを入力する。	
IEND		INPUT コマンドを終了させる。	
UPDATE	[プログラム名]	すでに入力したソース・プログラムを修正する。	
DELETE	行番号-1 [, 行番号-2]	ソース・プログラムの一部あるいは全部を消す。	INPUT コマンドまたは,

コマンド名	パラメータ	機能	事柄
1行修正	FORTRAN文	去する。 1行分の置き換え、挿入を行なう。	UPDATEコマンド実行中のみ使用可。 "￥行番号△本文"の型で入力する。
UEND		UPDATEコマンドを終了させる。	
RESEQ	[プログラム名] [, 整数]	ソース・プログラムの行番号をつけかえる。	
LIST	[プログラム名] [, 開始行番号, 終了行番号]	ソース・プログラミリストを端末タイプライタに出力する。	
SATV	プログラム名-1 [, プログラム名-2, プログラム名-3,プログラム名-n]	プログラムをファイルに保存する。	
LOAD	プログラム名-1 [, プログラム名-2, プログラム名-3,プログラム名-n]	保存ファイル上にあるソース・プログラムを呼び出す。	
PURGE	プログラム名-1 [, プログラム名-2, プログラム名-3,プログラム名-n]	ソース・プログラミリストを端末タイプライタに出力する。	
RUN	[主プログラム]	入力したプログラムの実行を指示する。	
CHMODE	旧プログラム名, 新プログラム名 [, M]	プログラム名の変更, 主プログラム ←→副プログラムのモード変換。	
NAMELIST		保存ファイルにあるプログラム名を端末に出力する。	
STOP		TSS-FORTRANを終了する。	
RESTART		プログラムの実行を続ける。	システム制御コマンドの "REQUEST"により割込み入力する。
END		プログラムの実行をやめる。	
SYNTAX	ON OFF	シンタックス・チェックをする。 シンタックス・チェックをしない。	特に指定が無い限りONの状態とみなされる。

CPLコマンド

-93-

コマンド名	パラメータ	機能	事柄
RUN	[文番号[,文番号]];	プログラムの実行を指示する。	
LIST	[文番号[,文番号]];	ソース・プログラムのリストを端末に出力する。	
SAVE	;	プログラムをファイルに保存する。	
LOAD	プログラム名;	保存ファイル上にあるソース・プログラムを主記憶に読み込む。	
ERASE	[文番号[,文番号]];	ステートメントの削除を行なう。	
COM	;	任意の時点で入力されたプログラムをリコンパイルする。	
NORECOM	;	プログラムの追加、置き換え等があった場合コンパイラは通常リコンパイルをするが、それをさせないことをコンパイルに指示する。	
NAMELIST	[(C)];	保存ファイルにあるプログラム名を端末に出力する。	
NOMSS	;	エラーメッセージを印刷する際メッセージを全部印刷せず、エラー番号のみを印刷する。	
PURGE	プログラム名;	ソース・プログラムを保存ファイル上から消去する。	
MSS	;	コマンドNOMSSの効力を無効とする。	
GET	[STRING(文字ストリング名)]; データ指定	データ・リストに指定された変数に、端末装置または、文字ストリング変数から取り出した値を代入するものである。	

コマンド名	パラメータ	機能	事柄
PUT	[STRING (文字ストリング名)] ; [データ指定] [PAGE] [SKIP [(式)]]	指定された変数の値を端末装置に出力する。	
STOP	;	CPL プロセッサーの実行を停止させる。	

BACCUSコマンド

コマンド名	パラメータ	機能	事柄
AUTO	[文番号] [, 文番号]	間接文としての文番号をBACCUS プロセッサーが制御する。	
MANUAL		BACCUS プロセッサーで制御されていた文番号を端末より制御する。	
SAVE	SOURCE ON プログラム名	入力されたソース・プログラムをファイルに保存する。	
LOAD	SOURCE FROM プログラム名	保存ファイルよりソース・プログラムを主記憶に読み込む。	
LIST	[ALL] [, 文番号] [, 文番号]	プログラムの全部または一部を端末に出力する。	
DELETE	[ALL] [, 文番号] [, 文番号]	プログラムの全部または一部を削除する。	
Q		PAUSE 状態にある文の実行を続開させ、1つの文の実行が終了した場合、再びPAUSE 状態となる。	
RUN		PAUSE 状態にある文の実行を続開させる。	

コマンド名	パラメータ	機能	事柄
DEMAND	変数	指定された変数の値を端末から入力する。	
TYPE	'文字列' 実パラメータの並び FORMAT 番号:パラメータの並び	記号列、変数の値、式の値等を端末に書き出す。 'で囲まれた文字列を書き出す。 実パラメータの値を標準書式で書き出す。 実パラメータの値を、書式の番号で与えられた書式に従って書き出す。	
STOP		BACCUSによる処理を終了する。	
TO	文番号	指定された文番号を持つ文から間接文の実行を開始させる。	
ACT	[文番号] [, 文番号]	1つまたは、一連の文を実行させる。	
SET	算術式	算術式の値を左辺の変数に与える。	SETは省略可。

4.2.3 エラー検出機能とメッセージ

エラー検出機能は、プログラムを作成する過程および実行において非常に重要な機能である。また、エラーの表現のしかたで利用者にとってそのプロセッサーの使用感が良くも悪くもあるのではないか。一般に冗長的なメッセージは端末利用者にとってかえって煩わしいものであるため、その文章は端的に述べているものが良い。

各言語共、大体正しいメッセージは出たが、気が付いた点をいくつかあげてみる。

1. 構文チェック時のエラーメッセージ出力の2段構えについて

これはBACCUSで採用していることであるが、もし、間違ったステートメントを入力した場合、そのステートメントが間違いであることを表わす簡単な印を出力するだけでも、利用者は大部分の誤りを発見することができるのではないか。しかし、どうしても誤りの個所がわからないような場合にはこのままでは困るので、問合せの記号を入力するとそのエラーに対する詳細なメッセージを出力する。

例えば、BACCUSを例にすれば、

- ① > I = I + J FOR J=1 UNTL 100 :
- ② EH?
- ③ >?
- ④ BC044Q SAYOSO GA TARINAI.

(下線の部分は利用者が入力する。)

①でステートメントを入力したが、UNTILがUNITLになっている。これは少し馴れば容易にわかることがある。

② おかしなステートメントなので、システムよりEH? (エッ?)と聞き返えしてきた。

③ 誤りがわかったならば正しいステートメントを入力すれば良いが、ど

うしても誤りがわからなかったら問合せ記号?を入力する。

- ④ そのエラーに対する具体的なメッセージが出力される。

一見めんどくさそうであるが、実際使ってみたところでは便利である。

TSS-FORTRAN, CPL にもこのような方法が採用されることを望む。

2. TSS-FORTRANの実行時のメッセージ出力について

ここで実行時とは翻訳、結合、実行のことであるがTSS-FORTRANではバッチ用のメッセージをそのまま使っている。従ってメッセージもTSSではあまり使い道のないエラー・アドレス等も出力され、長いメッセージになっている。

例えば、

FT999W DIVIDE CHECK ERROR AT(XXXXXX)
余 分

こういったものを編集して出力できないものだろうか。

3. CPLにおいてメッセージを出力する際アスタリスク(*)の長い列が前の方に出るが、あまり長いのも煩わしい。

4. CPLに省略型（エラー番号のみ）が選択できる機能があるのは良いと思う。

4.3 各端末の利用状況

4.3.1 通商産業省における利用状況

1. 概 要

昭和44年10月に(財)日本情報処理開発センターの遠隔情報処理システムの研究開発の一環としてNEAC 2200-500/200MACシステムの端末装置が通商産業省重工業局内に設置され、昭和46年12月まで使用された。昭和47年1月からはFACOM230-60システムの端末装置が設置されている。

通商産業省内ではおもに次のような目的で利用されてきた。

- (1) 省内業務、特に予測、調査、統計などに関連のある業務での実際運用（プロダクション）。
- (2) 省内職員に対して情報処理技術に関する知識および技術の習得を目的とした教育訓練用。
- (3) 情報処理システム（TSSを主とした）に関する知識の啓蒙、普及に資するための省内外の関係者に対して行なうデモンストレーション用。

上記の中で最も多く使用しているのは、昭和44年度からの実績をみれば、実務と直結している(1)の実際運用である。

現在、端末装置は、通商産業省本館7階726号室（TSS・ROOM）に設置されており、管理は重工業局情報処理振興課が行なっている。

また、端末機利用については、省内職員全員に開放しており、利用希望者が所定の手続きで情報処理振興課の担当職員に申し込めば端末機使用中以外は自由に使用できるようにしている。

2. 実際業務の運用における利用状況

昨年と同様一般行政部門におけるTSSの利用で多いものは需要予測などであり、プログラムは、ほとんどの場合、一般行政部門の担当者が独自に開発している。

以下、主要な2事例についてその概要を述べることとする。

(事例1)

• TSS を利用した需要予測

昭和47年11月から12月にかけて、財日本情報処理センターのFACOM 230-60システムのTSS端末機を利用して、特殊鋼（特殊鋼熱間圧延鋼材）の内、工具鋼、構造用炭素鋼、バネ鋼、軸受鋼、ステンレス鋼、高張力鋼、その他の鋼の8種の鋼についての需要予測を行なった。

従来から、特殊鋼熱間圧延鋼材ベースの需要予測は行なわれており、上記各種鋼種別の需要予測も行なわれていた。特殊鋼熱間圧延ベースでの需要予測については、IIP（鉱工業生産指数）と特殊鋼熱間圧延材の需要間に有意な相関関係があることが認められていることから、線型回帰分析等統計的手法によって求めてきた。最少二乗法による回帰分析は前年度TSSを使って体験すみである。工具鋼、構造用炭素鋼、バネ鋼、軸受鋼、ステンレス鋼、高張力鋼、その他鋼の需要予測についても、これらの鋼の需要先の景気動向、需要の季節的変動等を基にして需要量の予測をたて、それらの積み上げた値が、別途IIPを基に求めた特殊鋼熱間圧延鋼材の需要量予測と整合性のとれるようにfeed backを繰り返し調整してきた。

しかし、feed backに伴う計算は単純計算ではあるが、鋼種別の繰り返し計算となり、手計算では面倒であるし、四半期ごとに必ず必要な作業でもあるので、今回計算機を利用して能率化を企ることにした。一部、ライブラリーに登録されている最小二乗法のプログラムを使うほかは極簡単な計算内容である。

(a) 予測の方法

予測の手順を、動作分析的に細い作業ごとの表にしてみると以下のとおりである。

手順 1

各特殊鋼別過去四半期ごとの生産実績を調べる。

基準年度を定め、指標については、基準年度各期の生産実績を100とする。

季節的な需要量変動の影響を少なくするため、基準年度各期の生産量を基準として各年度各期ごとの生産実績を指標で表す。

以上で求めた各期の生産実績指標を時系列のグラフに表わす。このグラフにより季節的な変動を含めた需要増減のパターンを各鋼種別に把握することができる。

グラフのパターン、各鋼種別需要先の景気動向等を考慮し、グラフを延長して次期の需要見込み指標を求める。

見込み生産指標を需要量に換算する。

以上で求めた各鋼種別見込み需要量を合計し、特殊鋼熱間圧延鋼材ベースの需要見込み量を得る。

以上で求めた特殊鋼熱間圧延鋼材の需要見込み量を別途(手順2)で求めた需要見込み量と比較し、大きな差があるときは、先に戻って各鋼種ごとの次期需要見込み指標の見直しをする。

YES

差が大きい

NO

各鋼種別次期需要見込み量を得る。

手順 2

IIPの実績、特殊鋼熱間圧延鋼材生産実績を調べる。

基準年度を定めて、IIP、特殊鋼熱間圧延鋼材生産実績の対前年同期比を求める。

それぞれの指標間の相関関係を最小二乗法をもじいて回帰分析する。

回帰式にIIPの次期見込み指標を代入し、次期特殊鋼熱間圧延鋼材ベースでの需要見込みの対前年同期比を得る。

以上求めた指標を需要見込み量に換算する。

(b) 計算機の利用

以上の手順からも明らかなように、この作業は大型コンピュータの性能をフルに利用する性格のものではない。しかし、日常の作業の内では手計算でするには面倒な部類にはいるものである。コンピュータを日常の役所の仕事にどのようになじませていくかは、能率化、合理化の上からも重要な問題だが、今回のような **routine work** は比較的コンピュータとなじみやすいであろう。問題はむしろ、随時に要求される資料をいかにコンピュータを機動的に使って作成できるようになるかということであろう。

(事例2)

景気の変動は企業の収益に大きな影響を与えることは衆知のことである。しかし、各産業、例えば商社、機械工業、輸送工業等の間ではその影響の度合は異なる。各産業を商社、銀行、運輸業、機械工業、化学工業、鉄鋼業等の16産業に分類し、それぞれ各別産業間の影響度合をTSS 端末機を利用し、分析してみた。

分析の方法は2元回帰分析手法を用いて、相関係数を出すことにした。すなわち、42年から46年までの間における全産業の経常利益（上期・下期の年2回）を出して、これを基準にし、各個別産業の経常利益（42年～46年、上期、下期の2期）の全産業の経常利益に対する相関度を出したわけである。ここで、各個別産業の相関度が全産業に近づく程景気の変動が少ないと解釈し、逆に相関度が低い程（変動率が低い程）景気の変動を受けやすい産業であるという解釈をとったわけである。

(a) 計算式

与えられたデータについて回帰直線 $y = a + bx$ を求める。

$$a = \frac{1}{n} \left(\sum_{i=1}^n y_i - b \sum_{i=1}^n x_i \right) \quad \dots \dots \dots \quad (1)$$

$$b = \frac{s_{xy}}{s_{xx}} \quad \dots \dots \dots \quad (2)$$

$$s_{xx} = \sum_{i=1}^n x_i^2 - \frac{1}{n} (\sum_{i=1}^n x_i)^2 \quad \dots \quad (3)$$

$$s_{xy} = \sum_{i=1}^n x_i y_i - \frac{1}{n} (\sum_{i=1}^n x_i) (\sum_{i=1}^n y_i) \quad \dots \quad (4)$$

r : 相関係数

$$r = \frac{\sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{\sqrt{\left\{ \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2 \right\} \left\{ \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2 \right\}}} \quad \dots \quad (5)$$

(b) 結 果

結果については、相関係数が商社：0.8155161、不動産電鉄業：0.9287744、機械工業：0.8620189、建設業：0.9174873、自動車工業：0.7663727等の結果が得られた。なお、入力データ数について各産業毎に若干の違いがあり、必ずしも精度が高いと言えないが、おおまかな傾向がつかめれば十分であった。上記相関係数から、私共は、乱暴ではあるが、景気変動の大きい産業として、自動車工業、商社、機械工業、建設業、不動産業等というようなランク付けを行なったのである。

3. TSS を使用しての所感

(1) システムについて

システムの安定性を試す程のプログラムでなかったのでシステムの特性をうんぬん出来ないが、計算中に幾度かシステム・ダウンがあったことは確かだ。もちろんこちらの側の操作ミスによるシステム・ダウンもあると思うが、はっきり判るようにしてもらいたい。我々の仕事には超高能率、超高性能のコンピュータより、どんなことをしても絶対ダウンせずに動いてくれるコンピュータの方が頼もしい。

データのインプットはタイプライター、テープ両方使ったが、時間がかかるのが難点である。また、コンピュータがインプットを要求してくる際そのたびに「DATA」と打つのは、目ざわりもあるし、時間も無駄なの

で何か他の記号で代用してはどうか。

プログラムのインプットはテープを利用するには少し面倒なので、タイプライターからインプットしたが、時間もかかるし、端末を専有する時間も長くなるので、テープで簡単にインプットできるようにしてもらいたい。

データの保存については、ファイルを使うのは、なんとなく不安だし、データは手元に置きたいという気持もあってテープに頼った。しかし、紙テープは保存性がよくないので、紙テープあるいは磁気テープのデータ用カセットのようなものがあれば便利だ。

データのアウトプットについては、作表、作図が面倒だ。表のフォームを幾つか記憶させておき、計算中必要なときにすぐ呼び出すことは出来ないか。

(2) 応答時間と処理時間

応答時間については、タイプアウトに時間がかかるので、別回線でアウトプット専用の端末機があれば便利であろう。

(3) 端末装置

端末装置はかなり大きく、また機械の性質上専用の室が必要である。ポータブルな端末機というのは無理かも知れないが、テープ穿孔機の小さなものがいれば場所を選ばないので、便利であるし、端末機を専有している時間も少なくてすむ。

(4) ソフトウェア

細いことだが、FORMATで指定して印字させる場合、頭文字が出ない場合があった。

(5) 運営管理体制

サービス時間帯が不便である。10時～15時という時間は会議が多く、資料作成等コンピュータを使いたい作業は夕方から夜にかけてが多い。

4.3.2 郵政省における利用状況

1. 概要

郵政省の端末装置は、リモート・バッチ・システムおよびタイム・シェア

リング・システム（デマンド処理）による各種アプリケーションならびにソフトウェアの研究開発等、遠隔情報処理システム開発の一テストケースとして、財日本情報処理開発センター（以下センターと略す）によって通産省、電々公社、財機械振興協会と同時に設置されたものであり、昭和47年5月の本格的稼動以来10ヶ月近く経過した。

このシステムは、昭和44年末に開始されたNEACタイム・シェアリング・システム（MACシステム）と軌を一にするものであるが、今回は使用機能もFACOM 230-60と大型化され、使用可能言語も多種にわたり、ユーザーサイドからみると、かなり利用しやすいものとなっている。

以下は簡単に、現在までの当省における利用状況と、このシステムに対する所感、要望等を記述したものである。

2 利用状況

(1) 概 况

当省においては、複雑な科学計算、あるいは大量情報処理を日常業務とするセクションでは、それぞれ各業務に合致した機種と言語を用いており、このシステムの利用はほとんど経営分析事務に限られてきた。

記録によると、昭和48年1月末までの利用日数は65日、使用時間は150時間程度である。

センター側の記録によれば、利用日数は58日、CORE占有時間は168時間程度である。当省の記録簿の日数がより多くなっているのは、昭和47年4月以前のテスト使用の日数を加算したためであり、使用時間がより少なくなっているのは、計算方法の違いと記録洩れによるものである。

センター側の記録によると、CORE占有時間（DFORTからSTOPまで）168時間のうちCPU使用時間は2時間半と通常の使用状態から考えてやや多目に出ているが、これは後述するように、無意味なループをくり返して、〔ZIKAN GIRE〕のメッセージを数回受け取った事例によるところが大きかったと思われる。

また、アクセス件数（DFORTからSTOPまでの件数）は231件であ

り、1日平均4回の件数となっている。これは実際の使用にあたり、プログラムの修正に時間を要するため一時STOPをかけるとか、データによって補助記憶装置の番号が異なっているため、DFORTからやり直す必要があったこと等によるものである。

(2) 適用業務

今日のFACOM 230-60システムの各種の利用例のうち、その主な項目とその内容を挙げれば次のとおりである。

ア. 需要関数の作成

(説明) 所要のデータから回帰方程式

$$Y = a + b X_1 + c X_2 + \dots + z X_n$$

または

$$Y = a X_1^b X_2^c \dots X_n^z$$

を求めるものである。

- a. メインプログラムと20個のサブルーチンで処理。サブルーチンはデータの読み込み、計算、出力に分かれ、その指定によって読み込み形態、出力数値を決定する。
- b. 説明変数の数：最大5変数まで
- c. 入力：当初は紙テープによったが、説明変数を入れ替えて何回も反復計算する必要があるので、現在ではデータを一括ディスク・ファイルに記録させ、必要なデータをプログラムで指定して呼び出す方式を用いている。
- d. 出力：一次線型か対数線型かの選択推定式のパラメーター、推定値と観察値の残差、推定式の不偏分散（回帰の推定値の分散）、パラメーターの標準誤差、決定係数、自由度修正済相関係数、予測値のE（ Y^o ）および Y^o に対する理論的な誤差、残差変動に系列相関があるか否かの判定のためのダービン・ワトソン比等。

また、作成された回帰式の予測力評価のためのシミュレーションおよびそれに基づく予測等がプログラムの後半である。

イ. 標本数の早見表の作成

(説明) 信頼度係数, 誤差率(相対誤差), 母集団数を与えて, 変動係数の指定されたきざみ毎に標本数を計算して作表する。

ウ. 複利計算表の作成

(説明) 初期値 P_0 と n 年後の数値 P_n から n 年間の平均増加率を知る早見表の作成。

以上のように, FACOM230-60 システムの利用は, 回帰分析, シミュレーション, 作表等であるが, 利用の主たるものは, その回数および時間数からいって需要予測である。

これらの処理のためのプログラムは, 利用者がその都度作成したり, あるいは既存のプログラムを修正したりして用いている。

3. 所感

FACOM230-60 システムの端末装置利用状況は大体以上のとおりであるが, 使用開始当時(昭和47年3月)は TSS フォートランが使用出来ず, LINE ED によるコマンドによらざるを得なかったため, 前回の NEAC タイム・シェアリング・システムに比して使いにくい感じがした。しかし, 昭和47年5月以降本格的サービスの開始と共に, TSS フォートランの使用も可能になり, また, サービス時間も延長されたので, 実務にかなり利用出来る状態になった。

ただ利用の当初においては, 端末機自体の操作方法が前回の機種と相違していたこと, また, 今回のシステムが初経験である初心者が多かったこと等のために操作ミスが生じ, あわてて不必要的キーを打鍵したりしてキーがロックされ, 作動しなくなってしまう状態もしばしば生じた。

以下端末装置のユーザーサイドから, 今回のシステムに対する感想を記し, あわせて要望も述べてみたい。

(1) システムについて

ア. システムの安定性について

a. センター

昭和47年10月頃まではかなりの頻度でシステム・ダウンがあり, 当時

の使用記録法をみても日に最低1回、多い時には3、4回もダウンし、作業が全然渉らないことが多かった。そのダウンの都度キーがロックされて動かなくなってしまい、電話でセンター側の回復状況を問い合わせることも再三、再四あり、回復待ちのために予想外の時間を費したりした。特に9月下旬の機器増設以来ダウンの回数が増したように感じられた。

このようなダウンが生じた場合、センター側の好意により使用時間の延長をある程度認めてもらったが、操作途中でダウンするために最初からやり直したり、あるいはプログラムがページされてしまったりして、スケジュールの回復にはかなり手間だった。

しかし、11月頃からは、かなり徹底したチェックが行なわれた様子で、ダウンの回復もかなり減り、現在は安心して使用出来る状態になっている。

b. 端末

端末装置自体はかなり安定しており、端末自体が原因と思われる故障は1回しか経験しなかった。この時はセンターの係員の方に来省を要請し、いろいろ調査してもらったが、原因が分からず、何回か開始ボタンを押しているうちに正常に動き始めた。

c. ファイル

ディスク・ファイルにデータを書き込みその呼び出し形式をとったプログラムは需要関数の作成だけであるが、10月頃ダウンが頻発していた時期に、ディスク・ファイルに書き込んだデータ90個が消滅してしまった事例があった。この事例の場合は、データを再度書き込む手間も余りかからないので時間的ロスも少なかったが、今後かなりのデータを書き込む場合も予想されるので、データの保存については十分の配意を願いたいと思う。

プログラムについては、セーブされたものが消滅した事例は現在までない。

イ. 応答時間と処理時間

タイム・シェアリング・システムとしては当然のこととして、システム加入者が同時にレスポンスを求める場合に、その応答に時間がかかりすぎるよう感じられる。

特に12月以降、レスポンスに6~7秒、短い時でも2~3秒要する場合が時々見受けられるようになってきている。

1回のレスポンスに上記のような時分を要すると、実際に端末を操作している者にとってはその数字以上に長く感じられ、ひいては操作のリズムを失なってささいな打鍵ミスや誘発することがある。

タイム・シェアリング・システムである以上、単独で本体を専有しているようにはいかないのは当然と思うが、全加入者が同時使用している場合でも、もう少し何とか応答時間を短縮出来ないものか。

ウ. エラーメッセージ

FORTRAN エラーメッセージは、前回の MAC システムに比較すると大分判読し易くなっているが、プログラムの修正も容易になっている。

ただ、LINKAGE EDITOR エラーメッセージは解説書にもコメントされておらず、出力された場合はセンター側その都度電話で問い合わせている状況である。

以下のようなメッセージが输出された事例では、STOP 指令を出して DFORTからやり直したら実行が可能になったが、どのような状態の場合に下の様なメッセージが输出されるのかマニュアルで解説してもらえばと思う。

〔例〕

```
**..... LIED ERROR MESSAGE LIST ** .....
```

```
***** LD017Z FILE NO AREA GA TARI NAI
```

```
***** LD021Z HARD ERROR
```

エ. 原因不明なメッセージ

システムの安定性に関する事例であるが、以下の様な事例が生じたことを参考に書き添えておく。

- a. プログラム OUTPUTをアップデートし、ファイル保存のプログラムOUTPUTをページし、次いでアップデートされたOUTPUTをセーブする段階で次ぎのようなメッセージが出力され、とうとうセーブすることが出来なかった。

```
⋮ ⋮  
¥ SAVE OUTPUT  
SAVE FILE ノ ハードウェアエラーデス  
⋮ ⋮  
¥ SAVE OUTPUT  
SAVE FILE ノ ハードウェア エラーデス  
⋮ ⋮
```

この場合はセンターの磁気デーブにOUTPUTの記録があったので当方の負担は少なかった。

- b. リストの最中に出力が止まり以下のような操作をしたところジョブアボートされてしまった。

```
¥LIST A  
00010 .....  
⋮ ⋮  
..... n ..... ※出力が停止  
R Q ..... ※割り込みキーを押して R Q を打鍵  
R Q ..... ※全然応答がないので再度 R Q を打鍵  
>END ..... ※>を出力してきたので END を打鍵  
JOB ABORT  
ABORT SITANOWA .....
```

*JB4101 X EXEC FILE GA NAI (DELETE)

*SP723Z(WARIKOMI) MEIREI YOMIDASI

SINGAI

c. ディスク・ファイルを使用してデータを書き込むプログラムを実行したところ、当方のマクロコマンドのパラメータにエラーがあったため、データの読み込みを行なわず一定時間後〔ZIKAN GIRE〕が 出力された。

このためにCPUの使用時間が非常に大きくなつた。（センター側の記録によれば約30分）

このような場合に備えて、この無意味なループをチェックする機能を追加してもらえたうと思う。

才. 端末装置について

a. 操作性

$$A = B + C$$

のステートメントを打鍵する場合、=、+を打鍵するためにはいちいち中段シフトを押さえなくてはいけないので、急いでいる場合にはつい押し忘れてミスしてしまう。前回のMACシステムの端末はこの手間がないだけ操作が楽だったようだ。手間をかけるだけ何らかのチェック機能があると思われるが、上のような記号はよく使うだけに上段キーで打てるようにして欲しい。

b. 速度に関しては入力速度はともかく（データが大量で反復計算の必要がある場合はディスク・ファイルを使用）出力速度はかなり時間がかかり、特に入力データは簡単であるが、出力が膨大な場合等には出力速度の一層の向上が必要であると思われる。

c. 端末装置に付加してもらいたい機能はシステム・ダウンの際の何らかの表示機能である。最近はほとんどダウンがないようであるが、ダウンの続発していた時期には正常に稼動しているが、ロールアウトされた状態とダウンの区別がつかないで、センターに問い合わせたこと

が何回もある。

ロール・アウトの時間は長くても 1 分程度と聞いているが、それ以上の場合もあるようであり、このような点から続行中あるいは故障の場合に何らかの表示機能を追加して欲しい。

また、上記のような状態に関連したことであるが、端末自体に電話もセットされていたら非常に便利であると思う。

さらに欲を言えば、データ・シートを固定する装置を装備して欲しい。実際に作業を行なう段になると机を脇に持ってきているが、やはり不便である。この装置は、是非検討していただきたい。

カ. ソフトウェア

言語の種類と機能に関しては、現在当方で使用している言語はすべて FORTRAN であり、PL/I (CPL), BACCUS 等の使用可能言語は使用していない。

FORTRAN の機能は J I S 規格 7000 以上のものであり、工夫すればかなり汎用性があるので、現在のところ特に意見はない。

(2) サービス体制について

ア. センターのサービス体制について概括的に言えば、ユーザーの立場をかなり考慮されているようである。当方の手違いによるミスについてはその都度親切な指導をいただいているが、電話連絡では処理困難な場合にも係員の方に来省していただく等かなり助かっている。

また、ダウン統発時にも、センター側としては可能な限りの事後処理をされていたと思う。

イ. サービス時間帯については、緊急な案件の場合にはその都度延長を依頼しているが、今後は制度として少なくとも 1 時間、できれば 2 時間程度延長して欲しい。

ウ. ユーザープログラムの管理について

当方使用のプログラムの一つがたまたま大きなエリアを占めたために、処理速度がスローダウンし、他のユーザーの方々にも御迷惑をおかけし

た事例があったが、このような場合にはセンター側でプログラムの分割なり、その他の処理方法のアドバイスを事前にしてもらえないものか。

特にこの事例では、そのプログラムは何回か実行済みであったので、そのCORE占有状態も大体見当がついているのであるから、何らかの連絡が欲しかった。

私達のような初心者にとって、自分のプログラムがどの程度のCOREを占有しているのかなかなか分かりにくいものであるから、今後この様な場合には出来るだけ早いアドバイスを是非お願いしたい。

4. む す び

FACOM 230-60 の端末装置の利用状況とその感想等は大体以上である。当該システムは若干の不安定性の問題があったものの、大型コンピューターの迅速な処理と大きな容量を手軽に利用できるシステムとして、今後一層の利用を図っていきたいと思う。

ただ、このシステムの利用のためには、利用者自身がプログラムを作成し、あるいはまた開発されたパッケージプログラムを利用するため若干の機能を追加する等の場合には、どうしても言語の問題がある。

さいわい習得し易い簡易な会話型言語BACCUSおよびCPLが開発されているのであるから、その講習会を開催するとか、またはアプリケーションプログラムをより一層整備するとか等により、今後より多数の人が簡単に利用できるシステムとなるよう検討、開発を期待したい。

4.3.3 日本電信電話公社における利用状況

日本電信電話公社に助日本情報処理開発センターのTSS端末が入ってから3年になるが、その間当初のNEAC-TSSから昭和47年3月にFACOM 230-60を中心としたシステムに変更されている。この論文はFACOM-TSSになってから最初の使用状況および意見等の報告である。このTSS端末の利用は、本社部内で特に限定はしていないが、端末の管理を技術局で行なっている関係から、またその日常の業務の性格からも技術局勤務者が中心となって利用している。

る。前2回NEAC-TSSのサービスについてのアンケート調査を技術局第二調査係から報告しているが、今回もほぼ同様の内容の調査を行なったので、その結果を報告する。

1. アンケートの概要

アンケートは前後2回行なった。すなわち、昭和47年6月および昭和48年2月である。調査は下記の項目について行なった。

- (i) 利用者層
- (ii) 使用頻度・目的
- (iii) システムの安定性・処理速度に対する意見
- (iv) 端末でのオペレーションおよびプログラム関連

また公社技術局では、FACOM-TSSと同室に公社の科学技術計算システム(DEMOS)の端末も併置しているので、上記同一項目のアンケートをDEMOS利用者に対して行なった。両システムの端末はともに200bit/sのキーボード・プリンタ形式のものであり、両システムの利用者習性に似かよったもの多かった。FACOM-TSSおよびDEMOS利用者共通の意見・傾向等はこの種のサービスに対する一般的な評価であると考えるので、本論文ではそれについても合わせて考察してみる。

2. 利用形態および状況

両システムの端末は技術局端末室に設置している。その利用については、特に時間割当あるいは、予約制等の管理は行なわず、利用者各自の自由に任せている。これはタイム・シェアリング・システム本来の目的あるいは利用者の要望が、いつでも自由に大型電算機の処理能力を利用して業務処理を行なえるという事にあると考えるからである。従って利用者はいつでも、端末室に行けば使えるという体制をとっている。この1年間のFACOM-TSSの利用者は10数人で昭和45年度調査時の60人、昭和46年調査時の15人に比べて減少の傾向にある。特に昭和47年6月～10月にかけての利用は少なく、昭和47年末から昭和48年初にかけて利用は増加の傾向にある。これは昨年末CPLの訓練を技術局への新配属者に対して、センターにお願いし

て行なったためと考えられる。

DEMO Sの利用状況については、FACOM-TSSに比してはるかに使用頻度は高い。固定的に利用している人の数は20名程度であるが、テンポラリーの利用者を含めると30名前後の人人が利用している。DEMO Sについては、本社内には別に施設局および建築局にも設置してサービスを行なっているので、その利用はかなり広い層におよんでいると考えられる。

また、両システムをともに利用している人は少なく数名程度である。結局両システムの2端末で技術局内のタイムシェアリング・システムのサービスに対する需要を分担しているという状態である。

3. アンケート結果

アンケートに対して22件の回答がよせられた。このうちFACOM-TSSの利用者からの回答は9件である。この9件に対して統計的分析をするのは余り意味はないが、この9件に現われた傾向あるいは意見等は、DEMO S利用者を含めた22件の中でみると特に特異な集合ではないと考えられる。以下にその結果を述べる。

3.1 利用者層調査

利用者の年令構成および在職期間の分布を図4-2および図4-3に示す。また、公社のエンジニアの専門は、大きく分けると交換、線路、搬送、無線およびデータ通信（含む電信）に分類できるが、各専門毎の利用者の分布は図4-4に示す。また、学生時代の専門は図4-5に示すが81%の者が電気・電子を専門としていた人々である。

これらの図が示すように、FACOM-TSSおよびDEMO Sの利用者層はほぼ一致している。利用者の66%は入社3年以内のものであり、86%のものが20代である。すなわち技術開発等に実際に携わっている実務者レベルで活発に利用していると言えよう。

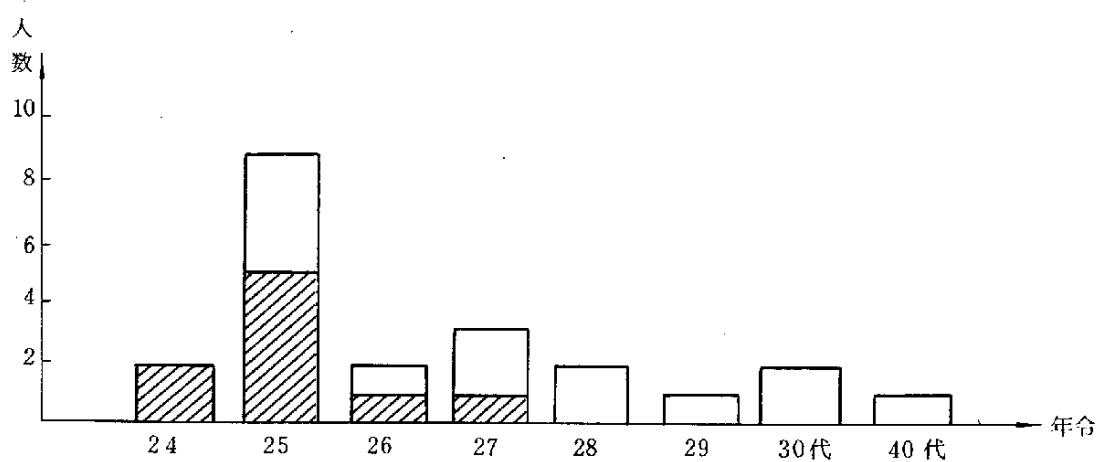


図 4 - 2 利用者年令構成

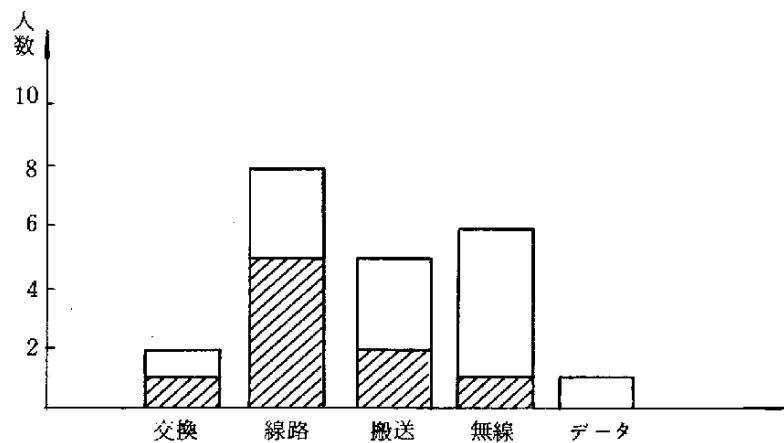


図 4 - 3 利用者の専門別分布

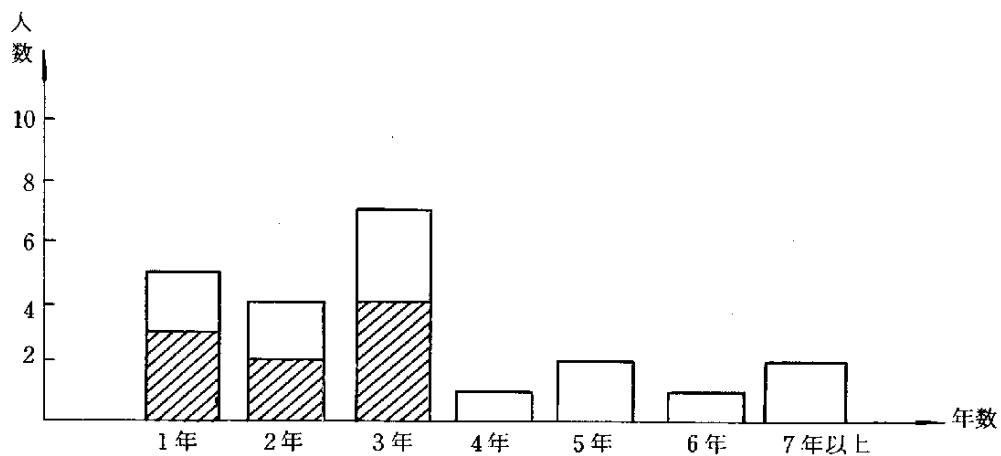


図 4 - 4 利用者の公社在職期間

注：各図中で斜線部がFOCOM-TSS利用者で、残りがDEMONS利用者

電気・電子	土木	その他	回答者全員
81%	14%	5%	

図 4 - 5 学生時代の専攻

3.2 電算機の使用経験について

FACOM-TSSあるいはDEMO-Sの利用以前の電算機使用経験についても調査を行なった。

アンケート回答者は全員両システムの端末利用以前から電算機の使用経験がある。また、そのうちの57%は入社以前からの使用経験がある。この特徴をあげると、

- 入社以前の経験

- (i) 経験者のほとんどがバッチ処理の経験のみで、TSS経験者は数名にすぎない。
- (ii) その使用言語はフォートランが大部分で他の言語としては、アルゴル、コボル、アセンブラー等である。

- 入社以後の訓練等による経験

- (i) ソフトウェア(言語)訓練の経験者が主である。
- (ii) その言語はフォートランとコボルがほぼ半々を占め、他にはPL/Iがある。

例えば、バッチ処理主体であってもこのような経験者が多くなっているので、この種のサービスに何の抵抗も感じることなく利用しているようである。

3.3 使用方法の修得について

使用方法の修得については、マニュアルによる自習、先輩について教わる、あるいは訓練等によっており、その割合を図4-6に示す。修得するまでの時間も、数時間から半日という回答が多く、特に修得に際して困難を感じるものはいないようである。(図4-7)

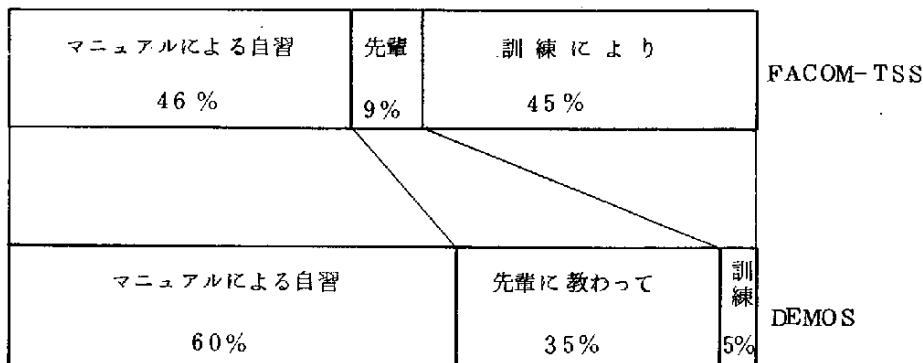


図 4-6 使用方法の修得について

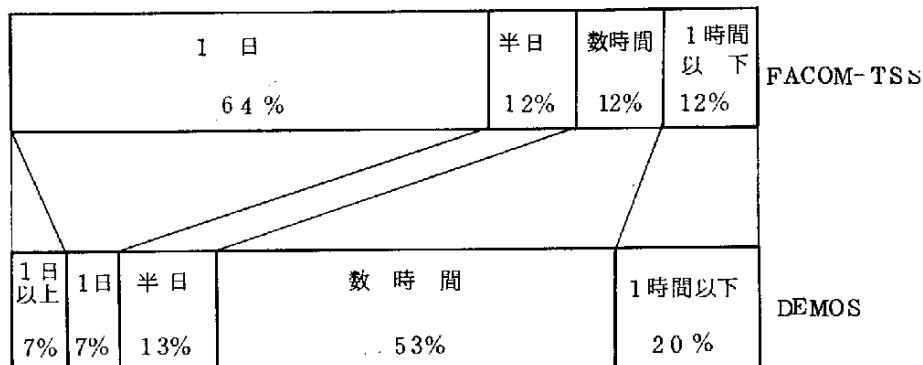


図 4-7 一通り使えるまで何時間かかったか

3.4 端末使用頻度

1人あたりの月平均の使用回数、1回当たりの端末占有時間および1つの問題を解決するまでの端末使用回数の調査結果を図4-8、図4-9および図4-10に示す。平均的な利用者は1つの問題を解決するまでに、4～6回端末を使用し、しかも使用1回当たりの端末占有時間は1時間前後である。

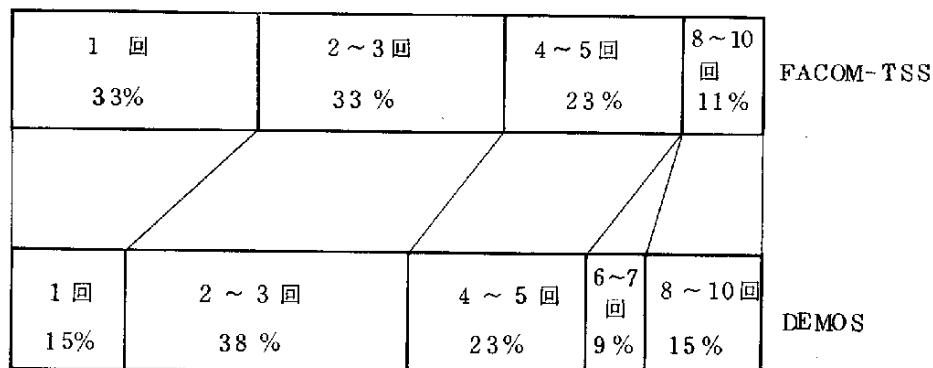


図 4-8 1人当たり月平均の端末使用頻度

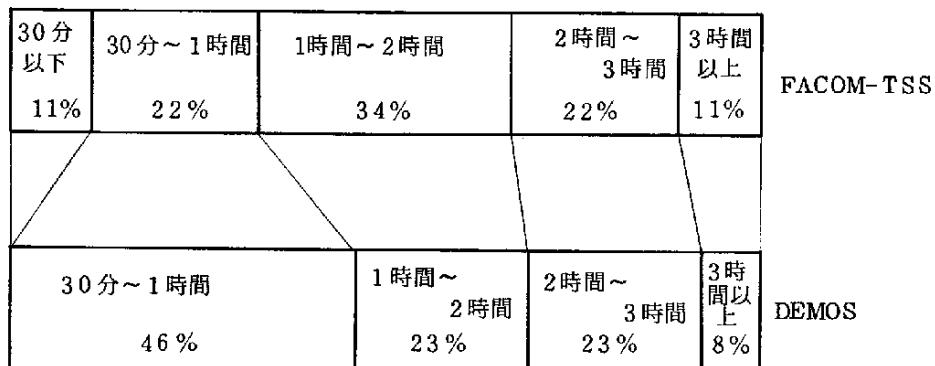


図 4-9 1回当たりの端末占有時間

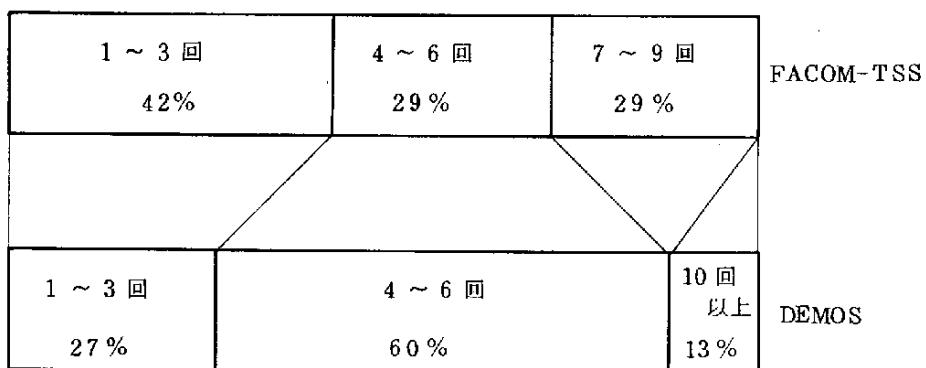


図 4-10 問題1件当たりの端末使用回数

3.5 使用目的(処理内容)

利用者の処理内容が、下記1～5の分類のいずれに相当するかの調査を行なった。

1. 科学技術的理論の考察
2. 物理的諸現象の解析・シミュレーション
3. 電気通信システムのモデリングあるいはシミュレーション
4. 各種実験・試験のデータ分析整理
5. 経済比較あるいは予測

その結果を図4-11に示す。

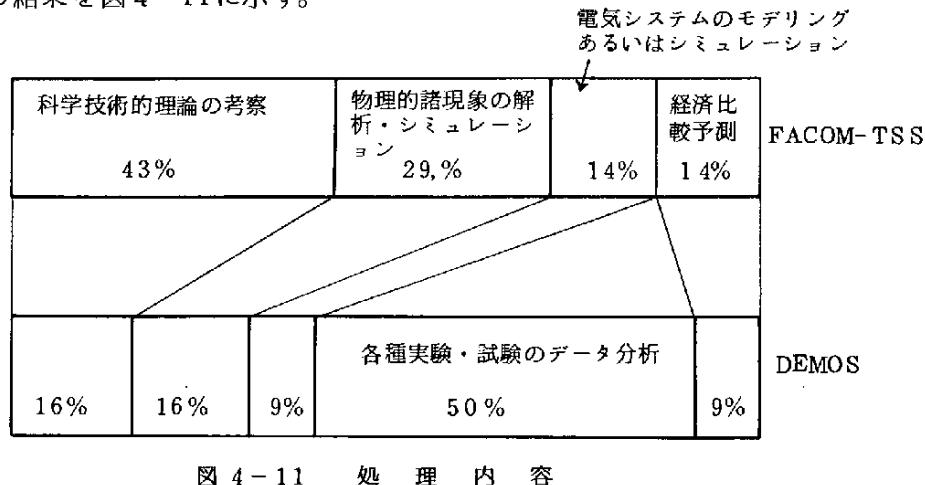


図4-11 処理内容

3.6 障害経験について

FACOM-TSSの使用経験者の約半数が障害を経験しており、その障害の種類はセンターダウンおよび端末装置の障害である。特にセンターダウンについては、実行中のプログラムの消滅、リカバリー時間の長さ等に対する苦情が述べられている。また端末の安定度についても多少の苦情が利用者から述べられているが、先に述べたように、昭和47年6月～10月期間の使用実績が極めて少ないので、現実の障害件数は少なかった。直接使用者達と話してみての感想であるが、端末障害よりもセンターダウンに対する苦情が多い事は意外であった。

3.7 端末のオペレーションについて

端末でのオペレーションに関しては、デマンド処理のレスポンス・タイム、リモート・バッチ処理の場合のターンアラウンドタイム、操作性、装置の速度、騒音、プログラムおよびデータの入力方法等について調査した。その結果は、図4-12～図4-17に示すが、リモート・バッチに関しては、無回答であった。

主な特徴は、

1. デマンドのレスポンス・タイムについては約70%の人が遅いと感じている。特に実行中急に応答が遅くなることが時々あるとの苦情がみられた。
2. 操作性について、使いにくいという人の苦情の例をあげると、ペーパーテープからの入力が不便である等の他、STXやETXの操作が煩雑である等がみられた。
3. 装置の動作速度については、特に意識していない人が多かった。
4. 騒音について、静かであるという意見はさすがに見られなかつたが、まづがまんできるというのが平均的意見であった。
5. プログラム、データの入力方法については、ペーパーテープとキーボードを併用している人が約半数であるが、キーボードだけで済している人も相当ある。

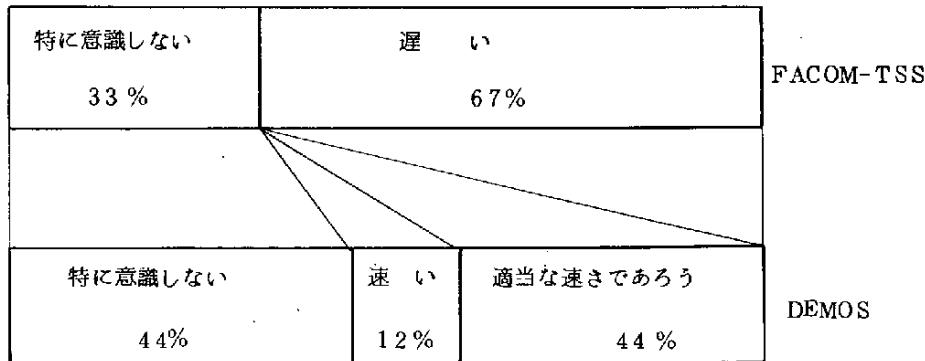


図4-12 デマンド処理の場合のレスポンス・タイム

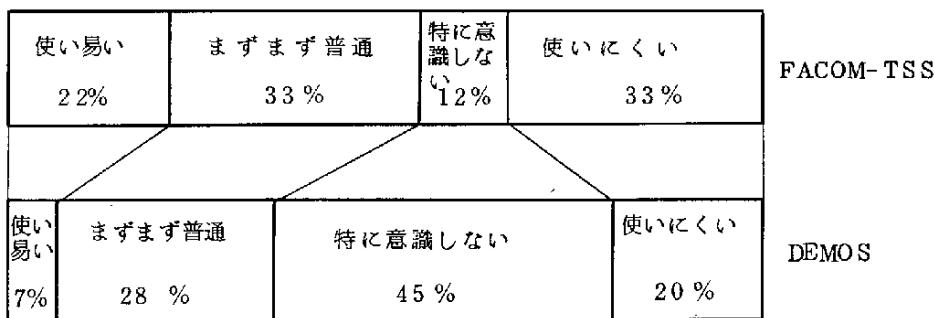


図 4-13 端末装置の操作性

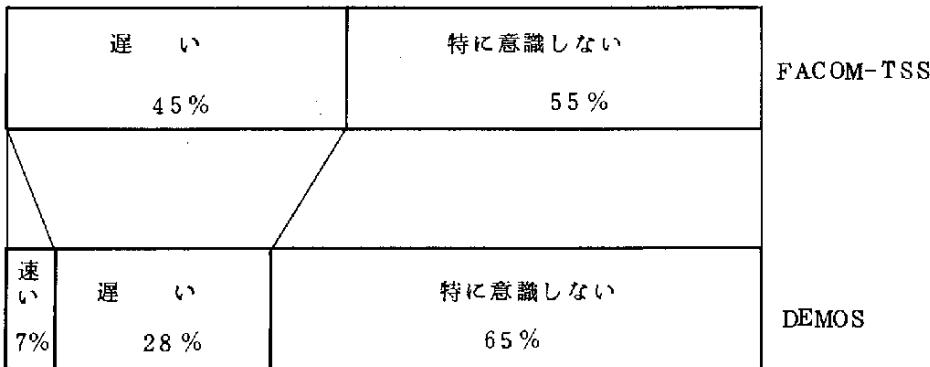


図 4-14 端末装置の速度

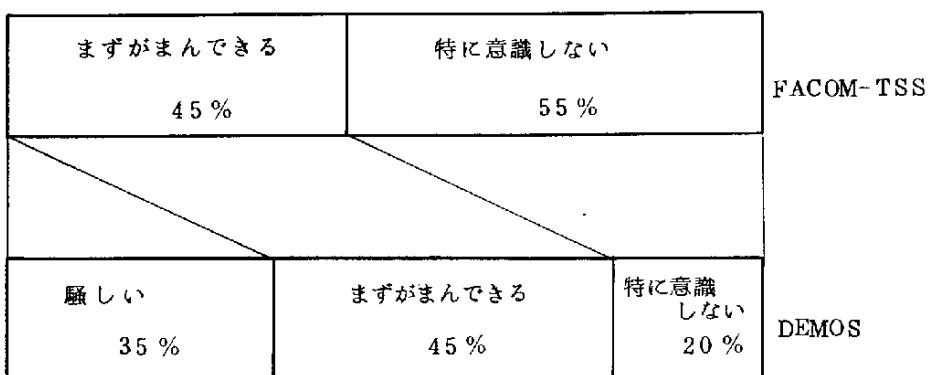


図 4-15 端末装置の騒音

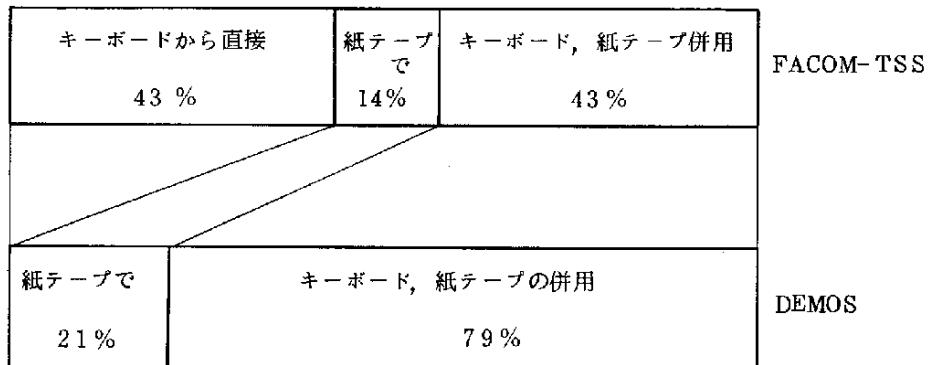


図 4-16 プログラムの入力方法

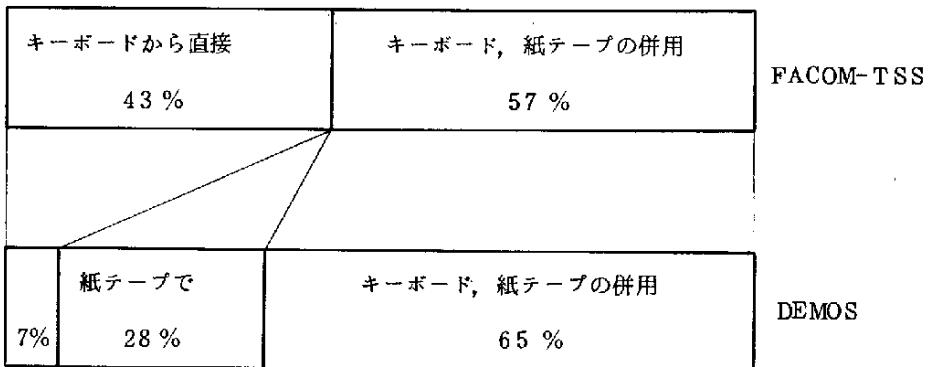


図 4-17 データの入力方法

3.8 ソフトウェアについて

ソフトウェアに関しては、使用言語の満足度、コマンドの使用感想、コマンド・ネーム、コマンド・パラメータの構成、ファイル容量、エラーメッセージおよびライブラリー等について利用者の意見を求めた。その結果を図4-18~図4-24に示す。

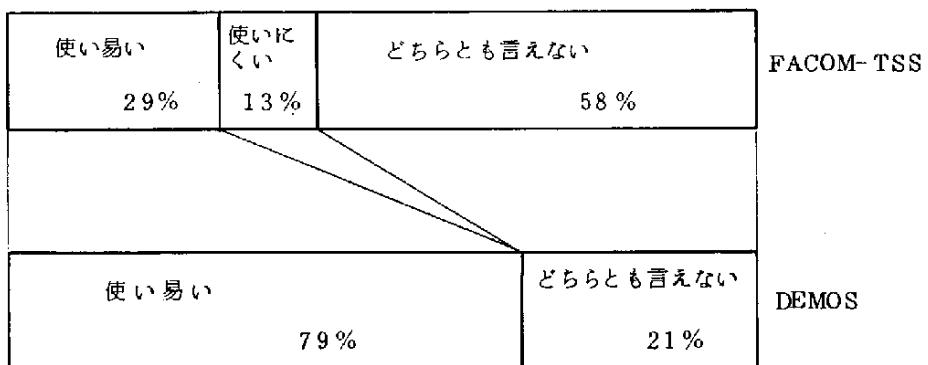


図 4-18 コマンドの使用感想

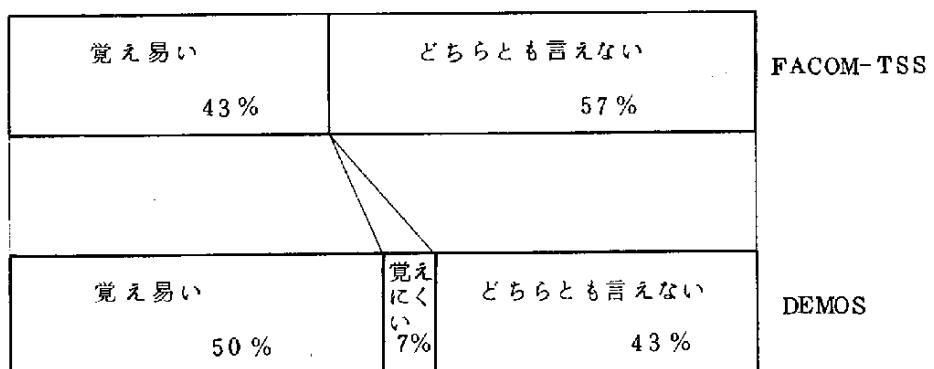


図 4-19 コマンド・ネーム

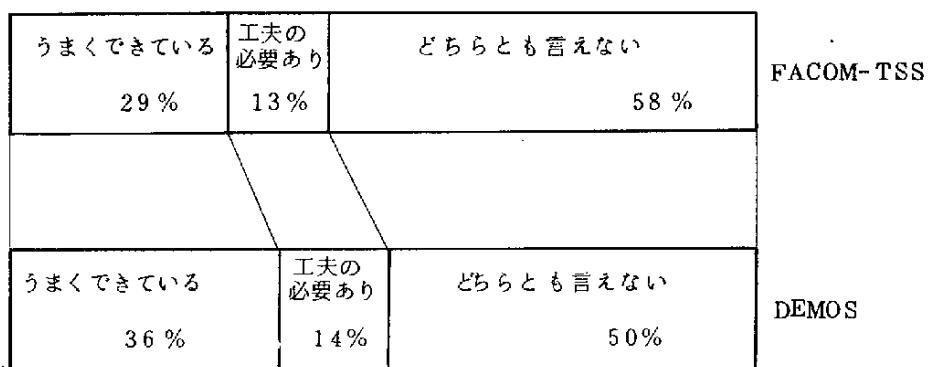


図 4-20 コマンド・パラメータ

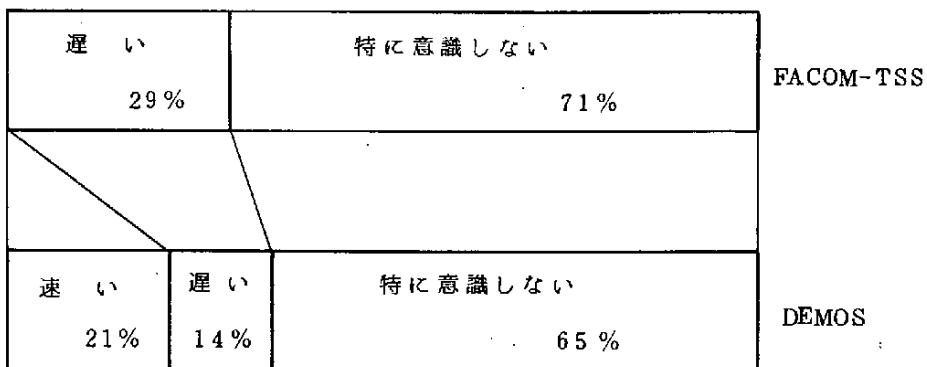


図 4-21 コマンドの応答速度

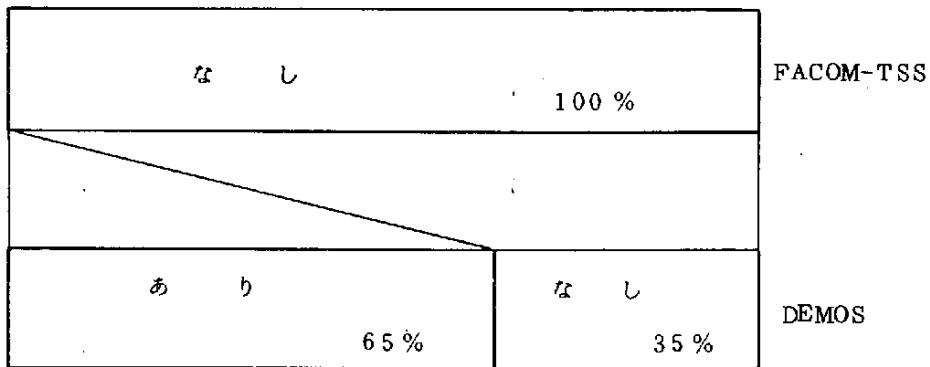


図 4-22 一割当ファイル制限で計算できなかつことは？

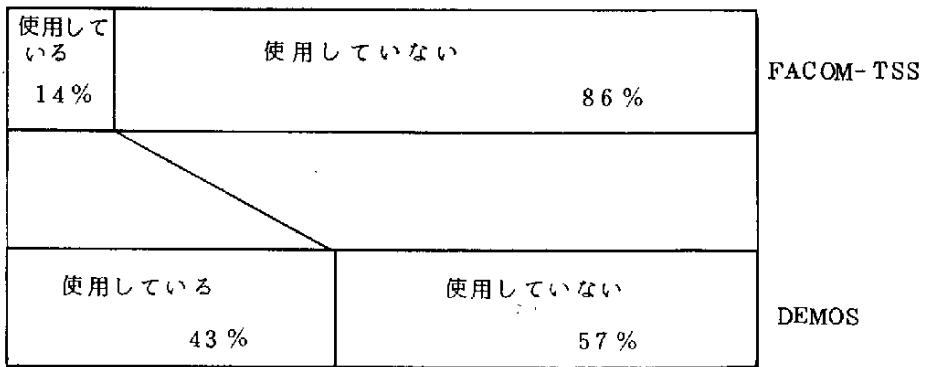


図 4-23 ライブライ一はよく使用しているか？

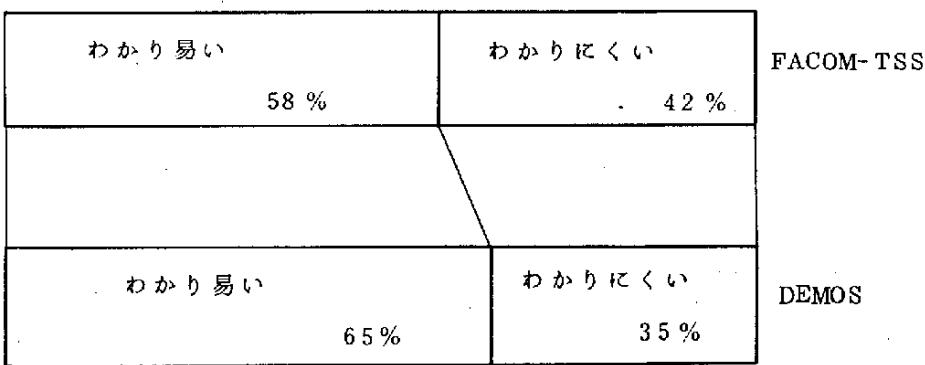


図 4-24 エラーメッセージはわかり易いか？

3.9 T S S 形式のサービスについて

T S S 形式の処理において、1つの問題を解く過程が、バッチ処理のときとどう変化しているかを調査するため、下記の設問をアンケートの中に加えた。すなわち、問題を解く過程を

- (i) 問題の検討
- (ii) プログラミング(含むフローチャーティング、コーディング)
- (iii) データ作成
- (iv) 入出力
- (v) 実行(含む デバッグ)
- (vi) 結果の検討

の6ステップに分解して、最も手まどり時間がかかる順にならべえる。この結果を、

- (a) 非常に手まどる
- (b) 手まどる
- (c) 簡単

の3つのグループに分解して上記6項目がこの(a), (b), (c)に現われる割合を図示すると、図 4-25 になる。図において左側に机上の作業を右側に端末室内での作業を配した。図上では必ずしも明確とは言えないかも知れないが、机上での仕事の時間の割合が、バッチ処理のそれよりもかなり大きくなり、

電算機にとられる時間が減少している事は望ましい傾向であると考える。特に図中の“実行”がデバッグ作業を含んでいる事を考えれば、デバッグのたび毎に待行列を経験したバッチ処理に対する便利さは非常に大とすべきであろう。また、この図からTSS形式の端末からの処理の問題等も推測されよう。

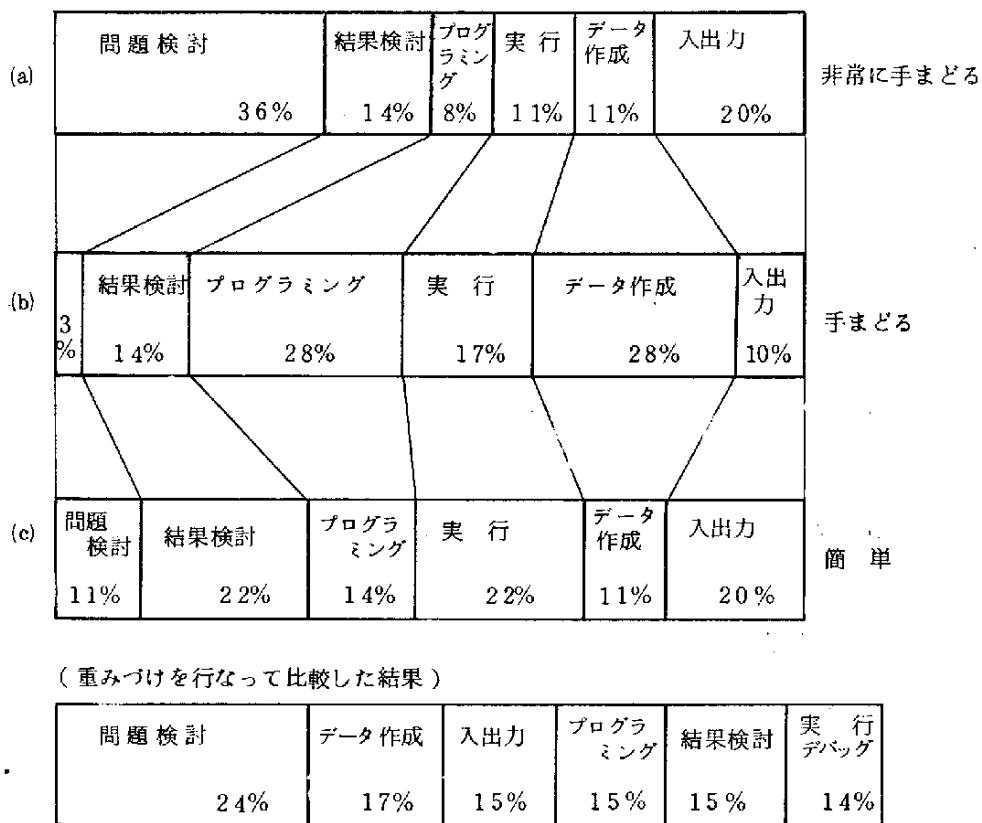


図 4-25 最も煩雑で時間のかかるステップ

ま と め

以上、FACOM-TSSを中心として、公社、技術局端末の利用状況を、アンケート調査をもとに述べて来た。現在、端末室の利用はかなり混雑しており（特にDEMONSについて）、本来、バッチ処理のプログラム実行以前の手続きあるいは待時間等種々の不便さを克服する方式として登場したタイムシェアリング・システムも、少なくとも技術局においては端末装置の前に待行列ができている状態である。

こういった状態において、FACOM-TSSの端末がそのサービスに対する要求を分担して稼動している形ではあるが、利用の伸びはいま一歩である。その理由はいくつか考えられる。

1. サービス時間

現在のサービス時間は、10時～15時であるが、利用者の間では時間延長の要望が多く聞かれる。タイム・シェアリング・システムの何時でも利用できるという特徴がかなり限定されている。DEMONSが、日曜日でも使用可能であることと比べると、アベイラビリティにおいて問題があると言えよう。

2. サービスの広域性

これはFACOM-TSSが実験的システムであるので、ある程度やむを得ない事ではある。

人事異動による転勤のため、以前からの利用者がそのまま残っている事が少なく、この3年間に利用者はかなり減少した。これに対してDEMONSについては、東京、大阪、名古屋すでに200端末を越え、サービスもかなり一般化して、DEMONSの経験者は多いが、FACOM-TSSの経験者は少ない。

3. その他

利用者をサポートするユーティリティ的なライブラリ等の利用状況は、DEMONSと比べると少ない。これはFACOM 230-60のライブラリに利用者のなじみが薄いせいもあるが、マニュアルが不足している等の二次的要素も多少影響しているかもしれない。これについては、端末の管理体制の不備で反省している。ライブラリーが完備して、常に利用者にその利便を提供できる事も、1つの必要な条件ではないかと考える。

以上、細部にわたって分析すればあるいは他の理由が浮んでくる可能性はあるかも知れないが、上記3点で、2、3については、一見二義的理由のように見えるが、タイムシェアリングのサービスを公衆データ通信という観点からみると案外に重みを増していく点かもしれない。

一方、両システムの利用者からオフラインのさん孔タイプライタ設置の要望が多く寄せられた。これは利用者の数が増加して端末の前に待行列ができるためである。先に3・9の図4-25に示される如く、端末装置を前にして時間のかかる作業はデータ作成と入出力である。端末装置がパーソナルコンピュータとして使えるなどまだ先の話と考へれば、何らかの対策を講じる必要があろう。

また、現在の業務処理程度であれば、リモート・バッチに注目する必要がある。端末での待行列の解消には、デマンド処理をプログラムのデバッグに主として使い、実行はリモート・バッチで行なってそれで間に合えば利用者にとって申し分ない訳である。この場合センター処理装置の処理能力と端末の入出力の速度さえ十分であれば可能である。情報処理開発センターでは、ミニコンを用いたリモート・バッチ専用の端末も、このFACOM-TSSで実験しているという事なので、我々としてもその結果に、大きな関心をもっている。特に最近インテリジェント・ターミナルの概念に注目が集まりつつあるのでなおさらである。

以上で、アンケート調査にもとづくFACOM-TSSの使用状況とDEMOSをも含めた利用者の習性等の考察と報告を終る。情報処理開発センターの方々のシステム開発への御努力に感謝すると同時に、この報告が多少とも貢献することができれば幸いである。

4.3.4 (財)機械振興協会における利用状況

1. 概況

当研究所に遠隔情報処理システムの研究開発のため端末機が設置されてから4年になる。

昭和47年2月には、従来のNEACシリーズ2200モデル500/200によるタイム・シェアリング・システム(時間分割システム)を一步進め、汎用性を持たせたFACOM 230-60、汎用端末機FACOM 1591Aに変換されて現在一年が経過した。

現在のFACOM 230-60によるTSSはNEAC 2200と比べてコンパクトになり、また、タイプライター、データ・ステーション温度調節用ファンから発する騒音に悩まされずにすむようになった。このことはオペレーターにとって作業が行ない易くなり、作業効率からみて非常に大きなメリットがある。

以下、昨年2月のFACOM、TSS変換時から本年1月までの約1年間について利用状況、処理業務内容と利用上の問題点を述べてみる。

2. 利用状況と適用業務

利用状況についてみれば、NEAC 2200 TSSよりFACOM 230-60、TSSに変換された結果、初期段階での利用効率は当然落ちている。

その原因としてNEACで利用していたデータ、プログラムが利用できない。

また各人が完全にFACOM 230-60 TSSを使い切れなかった点があった。

従来、手計算、卓上電子計算機などを使って計算していたものを端末機に移換してきた。この方がデータミス、入力ミスさえなければ正確さと時間的なロスがなく有益である。

また、各個人のプログラムもすぐ利用できるようにしている。以上のように日常的な計算も端末機で処理する方向をとっている。

適用業務についてみれば、当研究所は「機械工業」を中心とする経済問題を解明し、明日の「機械工業」の発展をうながし、より良いものとするため、予測および分析を研究業務としている。

次に、分析・予測に関する研究業務の電子計算機（端末機）による処理業務は、

1. ◦需要予測
 - KSKモデル
2. ◦貿易結合度分析

○輸出結合度分折

3. ○その他日常の計算業務

○成長率

○加重平均

○移動平均

○最小自乗法

○増加率

○その他個別作業

以上のように、端末機は予測・分折に利用している。今まで処理されたジョブはすべて F O R T R A N 言語によるプログラムで実行されてきている。

1 の需要予測は、従来予測といえば計量経済的手法を用いてきたが、この計量モデルに産業連関モデルを連動させ、各産業間に齊合性のある予測としてメリーランドモデルがある。このメリーランドモデルを基盤に「機械工業」用に拡充して、機械振興協会のモデルとして立案されたのが KSK モデル（機械振興協会モデル）である。

K S K モデルによる予測の方法を簡単に説明すると、K S K モデルは計量モデルと産業連関モデルからなる連動モデルである。

そして、産業連関モデルに与える項目別最終需用を計量モデルで推定しておく。また最終需用の品目分割をもコンバーターマトリックスを利用して行なう。

産業連関モデルは、産業連関表および機械工業分折用産業連関表から投入係数予測を行なう。

次に計量モデルで得られた最終需要を産業連関モデルで得た投入係数によりチェックし、さらにフィードバックして逆行列係数マトリックスの計算を行ない、そして産業別、職種別投入係数によってもチェックし、最終目的である各部門の生産額を求めるものである。

K S K モデルについての端末機への利用については前回くわしく述べているので本稿では主として適用業務 2 の輸出結合度計算に当っての端末機

の利用例を述べる。

元来日本経済の発展は、産業構造の農業中心から繊維工業化へ、さらに重化学工業化に由来してきた。産業構造の変動が即輸出とは限定できないが、日本は概して輸出依存度の高い国である。これを類別に見ると食料品・原料品から中間製品、そして完成製品という順序で雁行的輸出発展が繰返されている。

したがってこのような輸出構造の変動を解明するに当って貿易結合度分析により、その一要因をさぐることが出来る。つまり、世界の輸出入のパターン分析を通して、日本の海外市場の把握、および予測をおこなえる。

さらにわかりやすく云えば日本の輸出が相手国にとって、どれほどの重要性があるか、また競争輸出国との関係を分析するものである。これを、普通の長期的概観によって輸出構成を見た場合では、日本の輸出市場の地域拡充と商品の多様化しかみれない。これは日本側から見ただけの輸出市の変動にすぎない、云わば一方的分析である。

そこで輸出結合度分析が必要であり、以下の3つの数式がある。

1. 国別輸出結合度

$$A_j = \frac{X_j}{X} / \frac{W_j}{W} = x_j / w_j = \frac{X_j}{W_j} / \frac{X}{W} = S_j / S$$

2. 商品別輸出結合度

$$B_i = \frac{X_i}{X} / \frac{W_i}{W} = x_i / w_i = \frac{X_i}{W_i} / \frac{X}{W} = S_i / S$$

3. 商品別国別輸出結合度

$$C_{ij} = \frac{X_{ij}}{X} / \frac{W_{ij}}{W} = x_{ij} / w_{ij} = \frac{X_{ij}}{W_{ij}} / \frac{X}{W} = S_{ij} / S$$

X …… 日本の輸出総額

X_j …… 日本の j 国向け輸出額 ($j = 1, 2, \dots, n$ 国)

X_i …… 日本の i 商品輸出額 ($i = 1, 2, \dots, n$ 商品)

X_{ij} …… 日本の i 商品 j 国向け輸出額

w …… 世界の輸入額 (除日本)

w_j …… j 国の世界からの輸入額

w_i	世界の i 商品輸入額
w_{ij}	世界からの j 国の i 商品輸入額
$x_j = X_j / X$	国別日本輸出構造
$x_i = X_i / X$	商品別日本輸出構造
$x_{ij} = X_{ij} / X$	商品別国別日本輸出構造
$w_j = W_j / W$	国別世界輸入需要構造
$w_i = W_i / W$	商品別世界輸入需要構造
$w_{ij} = W_{ij} / W$	商品別国別世界輸入需要構造
$S = X / W$	(世界貿易に占める日本輸出の) 総シェアー
$S_j = X_j / W_j$	" 国別シェアー
$S_i = X_i / W_i$	" 商品別シェアー
$S_{ij} = X_{ij} / W_{ij}$	" 商品別国別シェアー

輸出結合度の概念であるが、日本輸出のある取引関係（国別、商品別、商品別国別輸出）の日本側と相手側との双方から見た重要性は、 S_i 、 S_j 。

S_{ij} という日本輸出の世界輸入に占めるシェアーによって大体評価できる。

こういう国別、商品別、商品別国別の日本輸出の独占度ないし市場参加度が大きい程、その取引関係は緊密であるといえる。それが結合度に近い概念である。だがもう一步進めたい。

1971年の日本の機械総輸出額の世界機械総輸入額に占める総シェアー (S) は 11.68 % であった。国別シェアー S_j 、商品別シェアー S_i 、商品別国別シェアー S_{ij} を総シェアー S で割ったものを結合度指数としたい。

すなわち $A_j = S_j / S$ 、 $B_i = S_i / S$ 、 $C_{ij} = S_{ij} / S$ である。ただし総シェアー S の 11.68 % は日本の各取引関係についての平均的輸出能力を意味し、どの取引も一様の緊密さであれば、 S_i 、 S_j 、 S_{ij} はそれぞれ 11.68 % に達しうるはずである。又、それ以下である場合とくに疎遠な関係であると云える。

つまり輸出結合度は、日本の平均的輸出能力にくらべて、日本側と相手側との双方から見てのある取引関係の重要さの大小をあらわしているものである。

Sを日本の平均的輸出能力と述べたが、それを世界市場に対する日本の平均的な輸出競争力といい直してもよい。けだしほかの輸出国の競争に立向かいそれを排除してSだけの世界貿易シェアを獲得しているからである。

そうすると結合度1(指数100)より大きい場合は、その取引について日本の輸出競争力がとくに強いことであり、逆に結合度が1より小さい場合は、その取引きについて日本の輸出競争力が弱いことをあらわしているといえる。

つまり各種の輸出結合度は国別、商品別、商品別国別の日本輸出の国際競争の大小をあらわしているといえる。そしてそれは世界市場に対する第三国との輸出競争という視点からする考え方である。

以上が輸出結合度の数式とその概念であるが、それ以前の問題としてデータに問題点がある。

- 1 結合度算出の最大の隘路は、世界の輸出総額、輸入総額のデータがとれないことである。特に品目が細分化されるに従って、各国間の品目分類の齊合性がなくなり、ますます世界輸出・輸入の総額は得られなくなる。
- 2 世界の輸入総額は理論的には輸出総額と合致するはずであるが、輸出総額はFOB(本船積み込み渡し値段)で評価され、輸入総額はCIF(保険料運賃込み値段)で評価されるためと、FOBとCIFの差の外に統計期間の締切り日によるタイム・ラグ、あるいは最終仕向地をとるか中間荷上国にするか、仕出国を原産地国か最終仕出地にするかで理論上正しくとも実際上は差が出て來るのである。
- 3 機械についていえば、機械輸入国の仕入先別輸入額も、実際には相当の開きがある。また、貿易総額において開きが少なくても品目別になるとその差は大きく、細分化された品目については極めて危険である。
- 4 品目の分類上の問題点についても相違点がある。ある貿易商品を如何なる機械と認定するかは、各国税関窓口で認められるもので、分類表が如何に整理されていようと、格付認定に誤りがあれば、統計上の誤差も当然あらわれて来る。

上記のように輸出入の乖離率は分類が細分化されるほど大きくなること

がわかる。そのような問題点に関する認識をふまえた上で、結合度指数算出のためのデータとして、当経済研究所は OECD 貿易統計を採用した。

とくに機械に関しては全世界貿易の大部分が把握される。OECD 貿易統計は 22 の加盟国から貿易統計磁気テープを提供してもらい、S.I.T.C (国際標準貿易商品分類) 4 衔分類による統一フォーマットの磁気テープを作成している。

OECD 貿易統計は、

1. 分類、数量単位、金額単位が SITC により統一されているので国際比較が出来る。
 2. 磁気テープ、フォーマットが統一されているので電子計算機による再編成、および結合度指数のような解折数学の算出に便利である。
- のような利点がある。

以上、結合度分析について内容、DATA について述べてきたが、分析にあたって端末機をどのように適用するかについて述べよう。

貿易結合度算出のフロー・チャートは図 4-26 のとおりである。

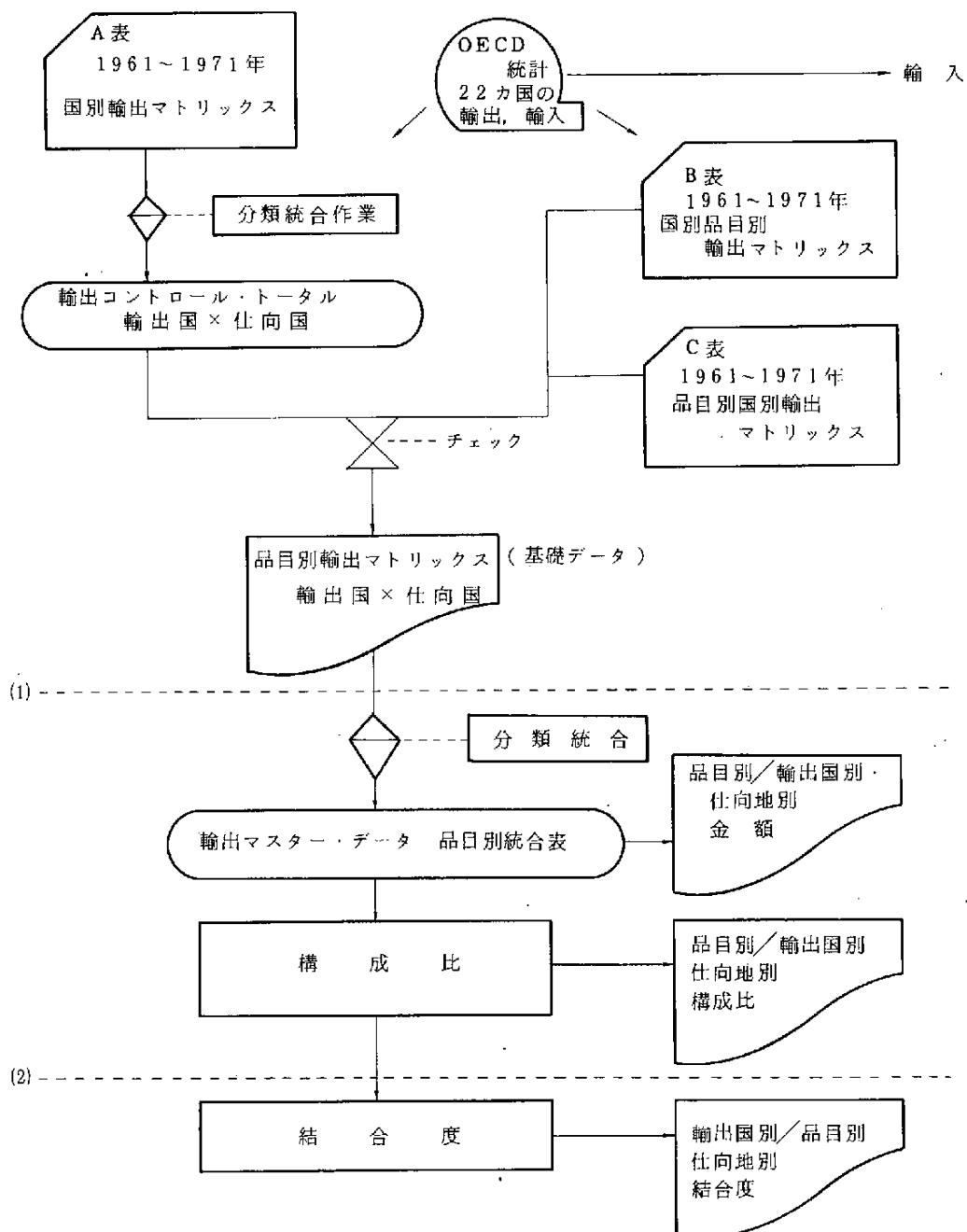


図4-26 輸出結合度フローチャート

品目番号 71 機械類(除電気機器)

輸出国 <i>i</i> 仕向国 <i>j</i>	日 本	西 ドイツ	ア メリ カ 22カ国	TOTAL
日 本	-				
西 ドイツ		-			
ア メリ カ			-		
J 国	$XJ_{j=18}^{i=1}$	$WJ_{j=18}^{i=2}$			$WJ_{j=18}^{i=23}$
TOTAL	$X_{j=23}^{i=1}$	$X_{j=23}^{i=2}$			$W_{j=23}^{i=23}$

図 4-27 各品目の輸出国別仕向地別金額統合表

O E C D 貿易統計は、A表(国別輸出入マトリックス)、B表(国別品目別輸出入マトリックス)、C表(品目別国別輸出入マトリックス)の3表からなっている。A表から各年次の22カ国の世界輸出入総計、相手地域別輸出入合計、および各国別の相手地域別輸出入合計を算出する(コントロール・トータル)。B表、C表から相手国、品目の選択を行なう。B表の表域分類に含まれる国名コードは4桁よりなっている。このコード番号により地域および国名の配列を行ない、A表の金額をあてはめる。そしてC表から品目別の輸出入を対象国別に配列し輸出入コントロール・トータルでチェックする(輸出入マトリックス基礎データ)。品目の分類はS I T C Section 7(機械類)の3桁分類まで統合している。

この作業はデータが膨大な量になるため現在の端末機の能力では出来ない。この部分については外部に委託している。

結合度算出にわたっては、品目別統合表が最も重要である。基礎データを元にして各品目ごとに相手国を統合するのであるが、この作業は最も人為的作業である。

すなわち相手国の選択をするわけであるが、それは品目別に相手国の輸出入金額を輸出入コントロール・トータルとチェックしながら行なうわけである(図4-27)。この統合表が、構成比、結合度を算出する場合のマスターデータ(図4-26)である。

さらに、構成比についても、コントロール・トータルを基に品目別国別構成比を演算している。このように、この作業で端末機を利用できるのは、図4-26の点線(1)または(2)で示した位置より後の作業であり、全体のシステムから見ればわずかであると言わねばならない。

次の表は、マスターデータをもとにした品目733(道路走行車輛)のアメリカからO E C D諸国向けの輸出結合度である。

アメリカからO E C D諸国向け商品別輸出結合表

C.Y	XJ	XJA	JAJ/J	JA	HI	AJ	JP-MJ=7	AM-J	XJ/M	XJA/M	クロスド
62	3009675.	20399.	0.3	197115.	20392393.	1813851.	20435532.	0.7	30.24	10.17	35.6
63	2173925.	20299.	0.2	215735.	30733135.	1957920.	20755256.	0.8	20.43	9.37	32.9
64	9355300.	20395.	0.3	203156.	34215039.	2462898.	32332141.	0.9	23.92	10.53	35.3
65	10015729.	35513.	0.3	290850.	39055533.	3223930.	35351343.	0.8	27.93	11.34	41.7
66	11150559.	36893.	0.3	315745.	44529421.	4374307.	39355034.	0.8	23.14	12.35	43.9
67	12572999.	41350.	0.3	349505.	48725551.	5931473.	42795083.	0.8	29.38	11.82	40.2
68	14451974.	40122.	0.3	392237.	55552899.	3129374.	43542225.	0.8	20.73	12.32	42.0
69	15233395.	53257.	0.4	459079.	57057053.	9263973.	57203915.	0.9	23.53	11.91	41.5
70	17331111.	53912.	0.3	501569.	77305913.	11159453.	51751557.	0.8	20.80	9.77	36.5
71	18415301.	70366.	0.4	578560.	93571200.	13972666.	73592100.	0.8	26.55	10.55	46.5

3. 所感

前にも若干述べたようだが、NEAC2200からFACOM230-60に変換されたが、当研究所では、業務の性格から当然計算業務がそのほとんどである。この事は、NEAC2200端末システムが、TSS用として開発された事もあって、ファイル操作、処理能力が現在のように汎用化されてない面もあって、非常に使いやすいように感じられる。ただ、我々の能力不足やNEAC2200に使い慣れていた事によるかもしれない。

ここでは、以前のNEAC2200と現在のFACOM230-60端末機の相違を比較しながら、使用上感じられた点とそれに対する我々の希望を述べてみたい。

- TSS作業の準備を完了するまでに待ち時間が非常に長く感じられる。NE-

A C の場合は待ち時間がほとんどなかったのだが、 F A C O M の場合、 " D " を最初に入力してから J O B K A I S I および ¥ が出力されるまでの待ち時間が非常に長く感じられる。この待ち時間を短かく出来れば、短かくしていただきたい。

- ・ソース・プログラム・ファイルおよび修正用プログラム・ファイルの入力が非常に便利になった。すなわち、コーディングシートの形式を無視出来るようになり、カラム数を数える手間がはぶけ作業効率の面で大きなプラスになっている。
- ・すべてのコマンドおよびデータ入力に際して、 STX コマンドおよびデータ ETX のフォームを取らねばならず、操作上わずらわしさが加わった。
- ・ファイル操作のうち、特にデータファイルが各端末割当のディスクエリア上の操作でなくなった事から、 NAME LIST コマンドによってデータファイルを調べる事、あるいはこのデータファイルの一部修正が不可能になってしまった。また、このことは、 INPUT コマンドによってデータファイルを直接外部から編集することが出来なくなった事なのだが、あらかじめ登録しておいたデータの内容を見たり、このデータを使用して何らかの作業を遂行する場合に、そのデータの内容を知らねばならない時、直接 OUTPUT 出来ず、プログラムを通してねばならず、作業が N E A C の時のようにスムーズに行なえなくなっている。我々の作業は比較的にデータ量も多く、特に、需要予測などの作業を行なう場合、産業連関分析が柱になっているため、コアの制約から、一度に全ての作業を終えられず、何段階にも分けて実行している。このため、最初に入力したデータからその計算結果を次のステップのデータとし、何度も繰返して最終結果を得るのである。この繰返し計算過程でのデータ修正等を行なうに当っては、現在のシステムで行なうのは非常に面倒な点が多い。ここで強く要請したいのは、データの INPUT およびアウトプットを N E A C システムの状態に近づけていただきたい。

最後に、現在の利用状況という点ではかなり落ちてきているが、これは、N E A C 2200 端末機用として作成してきた、プログラムおよびデータのライブラリー（紙テープによるもの）を F A C O M に変換しているためであるが、端末機の効率的運用を目指す方向として、磁気テープの利用という面とカードリーダーの設

置が残されている。カードリーダーに関しては、現在の端末機にFACOM-Rを設置すれば解決するとの事であるが、レンタル料の大きさにかかっている。しかし、磁気テープ利用という面では、前回の報告でも述べているが、最近の各省庁の統計資料がコンピュータ需要の増大につれて磁気テープ化されてきている。

当研究所でも大蔵省「貿易統計」や「産業連関表（昭和40年）」等を入手している。これら磁気テープによる資料の分析はほとんど外部の計算会社を利用しているが、これを端末機への組み込みを考えている。センター側にも都合があるかもしれないが、磁気テープの利用を含めた意味で端末機の有効的な利用面での体制を早期に確立していただければ、TSSシステムの利用価値は一層高まるだろう。

4.3.5 当財団における利用状況

当財団におけるデマンド処理の利用状況は表4-2、表4-3(P72)、表4-6に示すとおりであり、利用件数、利用時間からいっても相当な数字を記録しており、デマンド処理がコンピュータ利用の面で有用なものであることを示していると言える。

センターにおけるデマンド処理の利用状態は、次の5つに大別され、外部端末がほとんど(1)の目的に使用されるのに対し、(2)～(5)の利用形態が割合大きな比重をしめることが、利用方法として若干異なるところである。

- (1) 問題解決のための利用。
- (2) サブルーチン等共用プログラムの作成。
- (3) 開発中のデマンド用処理プログラムのデバックのための利用。
- (4) MONITOR-V自身のテストのための利用。
- (5) デマンド・ジョブを通して、MONITOR-Vおよび処理プログラムの性能評価のための利用。

(2)のサブルーチン等共用プログラムの作成は、主としてCPLの科学計算用ライブラリー(8.1参照)の作成等であり、(3)は、現在開発中のデマンド処理

表 4-6 センタ内におけるデマンド処理の利用状況

端末 月名	JIA		JIB		JIC		JID		JIE		JIF		計		平均	
	利用件数	CPU使用時間	利用件数	CPU使用時間	利用件数	CPU使用時間	利用件数	CPU使用時間	利用件数	CPU使用時間	利用件数	CPU使用時間	利用件数	CPU使用時間	利用件数	CPU使用時間
5	126	1:29' 52	68	30' 56	7	0' 10	140	4' 20	210	13' 13	59	02' 14	610	2:20' 45	101.7	23' 49
6	162	1:33' 46	111	19' 46	34	11' 28	58	4' 45	460	4' 35	33	13' 40	858	2:28' 00	143.0	24' 40
7	173	54' 46	165	1:18' 10	126	1:14' 00	131	27' 52	174	27' 58	34	0' 11	803	4:22' 57	133.8	43' 49
8	105	28' 59	118	21' 33	7	0' 03	116	20' 34	105	26' 52	28	06' 33	479	1:44' 34	79.8	17' 25
9	54	40' 04	90	3' 27	36	31' 03	46	21' 59	170	51' 11	23	0' 34	419	2:28' 18	69.8	24' 43
10	53	22' 31	46	2' 49	69	57' 03	82	7' 41	166	13' 10	15	11' 56	431	1:55' 10	71.8	19' 11
11	124	1:10' 20	58	3' 32	102	35' 56	109	18' 38	218	1:08' 02	40	1' 13	651	3:17' 41	108.5	32' 56
12	58	23' 27	98	24' 10	9	2' 09	43	6' 26	116	12' 40	10	0' 08	334	1:09' 00	55.7	10' 01
1	178	2:36' 55	236	1:00' 19	102	45' 57	94	57' 13	82	13' 11	47	3' 20	739	5:36' 55	123.2	56' 09
計	1033	9:40' 40	990	4:04' 42	492	4:17' 49	819	2:49' 28	1,701	3:50' 52	289	39' 49	5,324	25:23' 20	887.3	4:13' 53
平均	114.8	1:04' 31	110.0	27' 11	54.7	28' 38	91.0	18' 49	189.0	25' 39	32.1	4' 25	591.6	2:49' 15		

用プログラムには、オンライン文献検索システム JOLDOR-II (JOLDOR-I の改良型)、オンライン・シミュレーション・システム SIMBOL、インタラクティブ自動学習システム CLASS 等がある。

これらの利用を通じて MONITOR-V におけるデマンド処理について感じたことを述べてみる。

① 端末装置

現在センター内に設置されている端末装置は、50 BPS 用 F1592A-1 台、200 BPS 用 F1591A-1 台、F1520M-1 台、1,200 BPS 用 FACOM-R リモート・ターミナルー 3 台の計 6 台である。

デマンド処理用としては、外部端末として接続されている 200 BPS 用 F1591A が標準のようである。

以前、タイムシェアリングの使用実験を行なった MAC システムでは、速度、騒音等が非常に問題とされたようであるが、その点についてはこの F1591A ではさほど問題とはならないようである。問題は、F1591A では電文の前後に必ず (STX) , (ETX) というキーを押さなければならぬことである。これは他の端末では必要なく、利用者から必ず苦情のでるところである。この (STX) , (ETX) を入力するメリットがどこにあるかは、専門家でないと何とも言えないが、端末装置を利用する人間が、いわゆるタイピストやパンチャーのようなタイプすることが専門の人ではなく、むしろそれを苦手とする部類の人であることを考え、行番号といったものを処理プログラム側で自動的に打ち出してやったり、省略形といったものを導入していくらかでもタイプする文字数を減らすように努力するのであり、(STX) , (ETX) といったものは入力せずにすむのであれば、それにこしたことはない。

また、この (STX) , (ETX) の問題はオフラインで紙テープを作成するときにも起る。これは (STX) , (ETX) は打鍵しても何の文字も印刷されないため、孔の途中で果して、(STX) , (ETX) を押しただろうか、押さなかっただろうか迷うような場合があるし、出来上ったテープをオフラインで印刷してみてもそれを検出することができないので不便である。それゆえ、オフライ

ンでテープを作成する場合には、切換えかなにかの操作によって [STX] , [ETX] に対応する特殊文字みたいなものが印字されると便利ではなかろうか。また、デマンド用だけの目的から言えば、数字キーの配列を電卓のような配列にするとか、よく使われる記号 [=, +, -, *, /, (,), ,] などは、上段シフトで使用できるようになっていると非常に便利である。

端末装置自身の故障については過去 2 ~ 3 回程度で比較的少ないようである。

② 応答時間

応答時間については、特に MONITOR-V のような 4 元処理システムでは同時に動いているジョブ数、CORE 容量、ロールイン／ロールアウト・ファイルのスピード等ソフト・ハード、運用形態等が作用し、いちがいには言えないが、MONITOR-Vにおいては、マクロコマンドの解釈時間、ロールイン／ロールアウトの方法（領域が分れていてもプログラム全体がロールイン／ロールアウトの対象となる）、リエントラント属性の機能がない、各種優先権が動的に変化せず全く機械的であるといった点に問題があるようであり、一般的にいって応答時間が速いとは言えないようである。

③ 言語

現在サービスしている言語には「2.2 ソフトウェアの構成」の項で述べたようなものがある。

各端末毎の言語別利用状況は、表4-3(P72)に示すとおりであり、外部端末ではほとんどが FORTRAN を利用している。これは、一般の外部端末利用者のように、プログラムの作成が専門ではなく、いわゆる実際業務のためにコンピュータを利用するといった立場の人では、これから言語を覚えようとする人でない限り、既に FORTRAN を知っており、それが使用出来るのであれば例え新しい言語が多少便利であってもあらためてそれを覚えようとする気は起らないであろう。また、その時間もないかも知れない。それゆえ、逆に言えば利用する業務は FORTRAN で記述可能なものに限定されるであろう。

こういう意味では、デマンド処理の分野においても、FORTRANは今後とも根強い人気をもつものと思われる。

TSS-FORTRANについて言えば、以前使用実験を行なったNEACによるMACシステムの教訓を生かし、ソース・ステートメントのフリーフォーマットの入力の他にシンタックス・チェッカーを付け、ソース・ステートメントの入力から実行までを最低3つのコマンド(INPUT, IEND, RUN)で済むようにした。しかし、このFORTRANもバッチ型のコンパイラを利用しているため、ファイルの介在等でTSS用としてはコンパイルおよびリンクに多少時間がかかることに問題がある。できれば、ソース・ステートメントをLCM上におき、コア内でコンパイルを行なうといった、TSS用のより高速なコンパイラが欲しいところである。リンクエディタ(LIED)についても同様である。

いわゆる会話型では、どうしてもインタプリタ的な要素あるいは、余分な操作が必要なため、オブジェクトの効率が一括型コンパイラより悪く、実行時間がかかるが、コンパイルやリンクに要する時間が非常に短縮され、簡単なシンタックス・チェックさえ行なわれれば、特に1ステートメント毎の実行や一部分の実行をさせるといった、いわゆる会話型である必要性もうすぐなる。ただ、これはそれを使用する人が、その言語をどの位知っているかにもより、初心者が試行錯誤的に使用するのであれば、会話型のコンパイラもぜひ必要となる。それゆえ、非常にぜいたくな望みではあるが、プログラムの作成、デバックは会話型で行ない、実際の実行は一括型のコンパイラによって行なえるようになっていなければそれにこしたことはないであろう。

なお、V-2.(Version-2)のTSS-FORTRANにはステートメントの部分変更の機能がなく、ステートメントのある一部分を変更したいときでもそのステートメントを全部打ち直さなければならず不便であるので、V-3ではこの機能を追加した。(「8. プログラムの開発」参照)

CPLについては、会話型としては多少機能が大きすぎ、その結果としてコンパイラが大きくなり、他のジョブとの関係を考え、LCM上に置かざる

を得なくなつた(MONITOR-Vにリエントラント機能が付加されれば、HCM上でも可能である)。LCM上にデマンド処理用のプロセッサを置くということは、CPUの有効利用、処理速度といった面で望ましいことではないが、PL/Iのような汎用言語を会話型にする場合は、どのへんまで機能をもたせるかが問題であろう。

BACCUSは、会話型専用の言語のため従来のBATCH処理用言語と多少ニュアンスが異なる。そのため、この言語に慣れてしまえば使い易い言語であるが、BATCH処理用言語に慣れている人にとってはとりつきにくいといった面もある。また、機能面から言えば技術計算用であるからかも知れないが、文字の処理(代入、読み込み、比較等)ができないことと、ファイルが使用できないといった点が問題ではなかろうか。

LINEDは、ソース・プログラムやデータのファイルを作成するためのエディタであり、これによって実際のプログラムを入力して実行させるには、デマンド処理用としてはかなり抵抗を感じるようである。

全般的には、MONITOR-Vという汎用の強力なOSがあるのであるから、メーカ側としても処理プログラムの方にももう少し力を注いで欲しいものである。

④ システムの安定性

システムの安定性という場合、もちろんハードウェア、ソフトウェアの両方を合せたものであるが、「3.3 安定性」の項でも述べたような状態であり、運用者側、利用者側からみた場合、過去の使用実験期間中のデータからでは機器の増設、改造、それにともなうSystem Generation等考慮しなければならない点があったにせよ、必ずしも満足できるものではなかった。

ソフトウェアについては、バージョンアップとシステムテストのくり返しを行ない実際の運用に入ったが、実際の運用となると、やはり当初はかなり多くのシステムダウンを起した。これは、メーカ側のテスト方法だけでは必ずしも実際の使用方法通りにはいかず、使ってみなければわからないといった点があるので、ある程度はやむを得ない場合もあるが、問題となるのは、

バージョンアップすなわち SGによるシステムバックである。MONITORのレベルや機器の変更等があった場合に SGを行なうわけであるが、オンライン・サービスをしているような場合十分なテスト期間がとれずには開放せねばならないため、 SG後2~3週間は何かと故障が多発するようである。各ユーザ毎に標準値、機器構成、センター・ルーチン等が異り、確かに面倒な作業ではあるが、もう少し簡単に、しかも正確な SGができるものであろう。

全般的に言えば、ハード的な障害は集中的な傾向があり、機器の増設直後、 SG直後のトラブルを除けば最近2~3ヶ月はハード、ソフト共大分安定してきたようであり、今後もこのような状態であればさほど問題とならないであろう。

4.4 端末ユーザの要望に対する検討

端末ユーザからセンターに対する要望事項および、ソフトウェアの不備な点としていくつかあげられたので、それらの主なものについてセンター側の説明不足、資料不足等も考えられるので検討してみたい。

要望事項およびソフトウェアの不備な点としてあげられた主なものは、次のようなものである。

- ① 紙テープの入力について。
- ② **FORMAT**指定で印字させた場合、頭文字が出ない場合があった。
- ③ サービス時間を延長してほしい。
- ④ データファイルを直接外部から編集できるようにしてほしい。
- ⑤ 磁気デープの使用を考慮してほしい。
- ⑥ マクロコマンドのパラメータにエラーがあったとき、エラーメッセージが出力されず、ループ状態になり、C P U使用時間切れとなった。
- ⑦ エラーメッセージ。

①の紙テープ入力については、現在でもすべての入力データ（コマンド、ソースプログラム、データ等）について可能である。この場合、紙テープは
〔**STX**〕 データ 〔**ETX**〕 の形式でなければならない。また、連続読み取り、一行読み取り（会話モード）も可能である。

②については、このような現象は今まで報告がないので今後調査、検討するが、回線等に障害があればその旨メッセージが出力されるはずである。ただ **FORMAT** 指定における第1文字目はコントロール・キャラクターとみなされるので注意が必要である。

③については、NEACによるMACシステムのときにも指摘された問題であり、応用実験という性格から定時間内における延長は要請があった場合はできるだけ弾力的に運用した。

しかし、プログラムの開発ということからTSSサービス中には処理のむずかしい特殊ジョブ（COREダンプを探るといったもの）、長時間ジョブ、ハード、ソフトの各種データの収集およびTSS用ファイルのセーブ等があり、これらの業務のために定時間内にどうしても2時間位必要なためTSSサービス時間帯は10~15時とした。

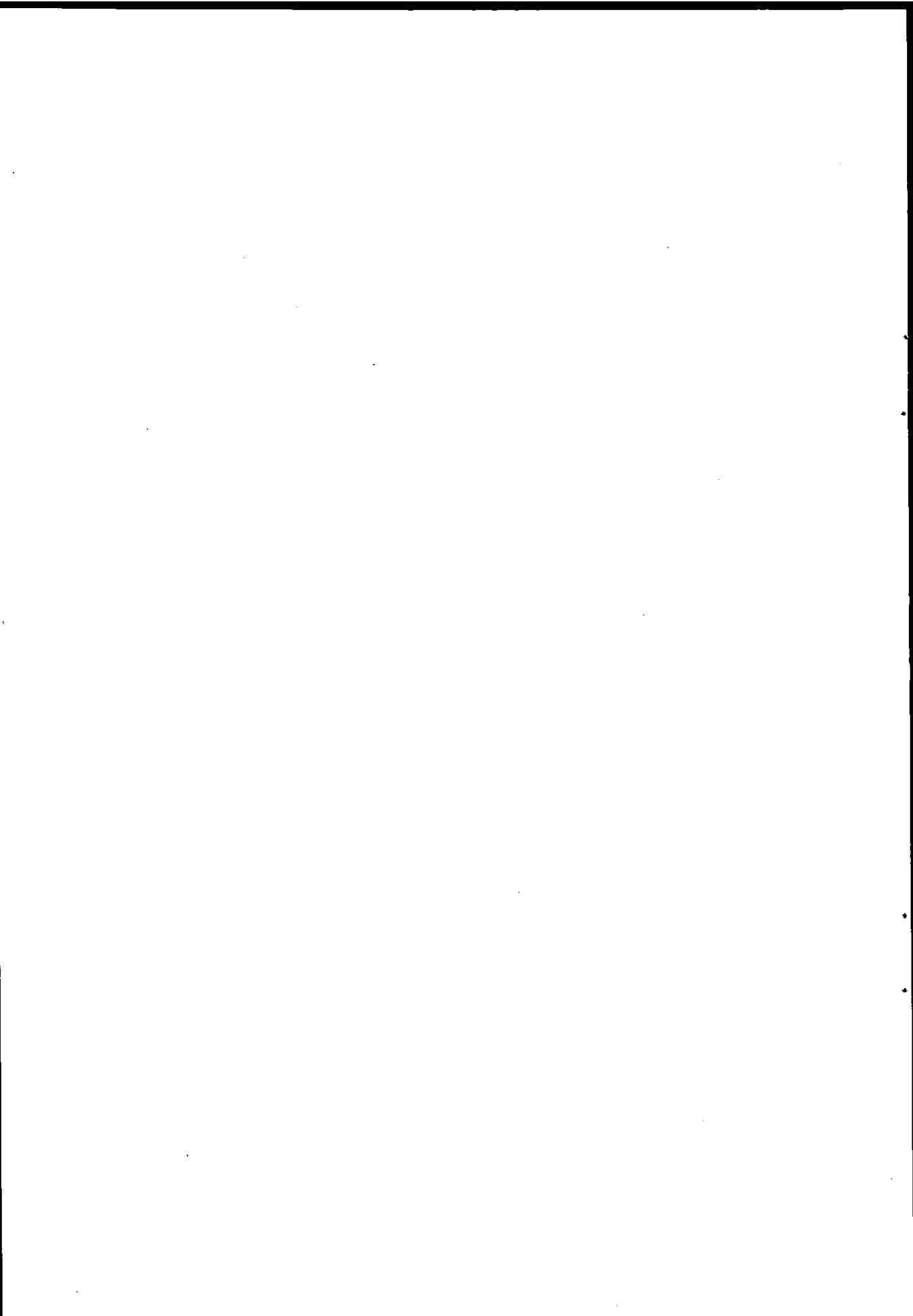
④については、ソース・データをそのままの形で編集する方法としては、現在でもLINEDを使用すれば可能である。この場合、ファイル名としては、DEMANDFL（このファイルは端末毎に割り当てられている）を使用しなければならないが、このファイルは分割型順編成であるのでメンバーとして、プログラムも含めて70個まで登録することができる（LINEDによって作成したファイルとTSS-FORTRANの関係については、TSS-FORTRANマニュアル（10ページ）を参照）。

⑤については、3元処理ということで、デマンド処理における磁気テープの使用は禁止している。これは磁気テープをセットするタイミングや他のジョブの磁気テープの使用状態等から操作が非常にむずかしくなること、デマンドでは長時間専有する可能性もあり、また、バッチ処理も多重処理をしている関係上複数のジョブが磁気テープを使用することなどから、磁気テープの台数が不足することなどによる。

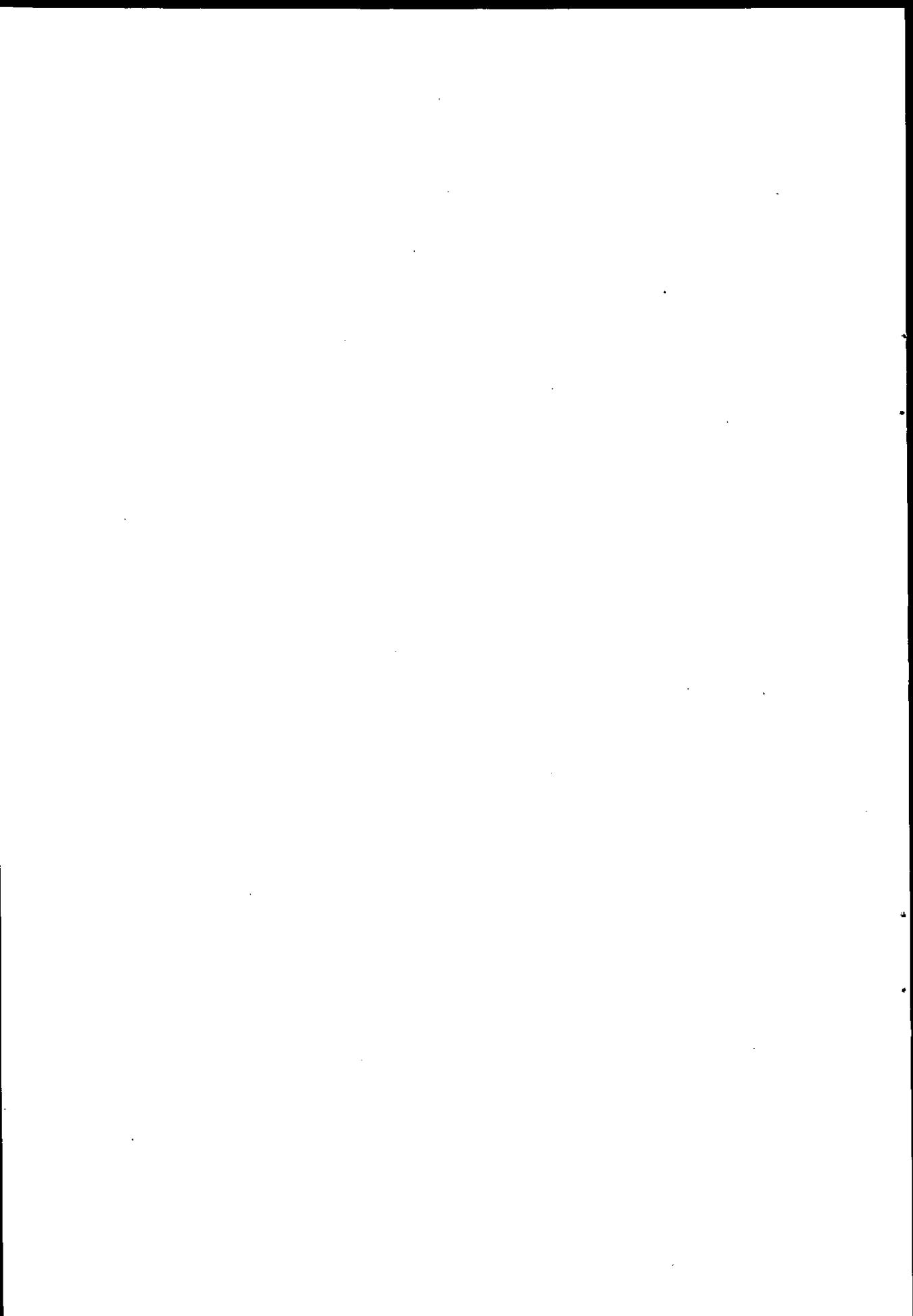
解決策としては、バッチ処理によって磁気テープを一旦大記憶ファイルに移しておくことが最も簡単な方法として考えられる。

⑥については、処理プログラムの不備であることが判明し、修正した。

⑦については、バッチ用のコンパイラ、リンクエディタを使用している関係上、デマンドとしては確かに不適当なメッセージが出力されることがあるが、これらについての修正は非常に困難である。



5. リモート・パッチ処理



5. リモート・バッチ処理

5.1 概 要

FACOM230-60 MONITOR-Vによるリモート・バッチ処理方式の応用実験は、昭和46年度から継続され、各種データの収集、システムの安定化、運用の効率化等を図ってきた。

実験当初は、ユニークな処理方式であるとされたリモート・バッチ処理方式も、現在ではかなり普及し、実用期に入ったようである。

MONITOR-Vにおけるリモート・バッチ処理は、デマンド用のタイプライタ端末、FACOM-R(RE)リモート・ターミナルのどちらかでも利用可能であるが、一般に入出力データがデマンド処理より多いため、カードリーダ付きのFACOM-R(RE)端末が使用される。

現在このFACOM-R(RE)端末は以下に示すように5台接続され、その利用状況は表5-1に示すとおりである。

本年度は、通商産業省・工業技術院機械技術研究所によって、リモート・バッチ処理用の旋削用自動プログラミング・システムMELTSが開発された。本稿はこのMELTSの概要および開発過程におけるリモート・バッチ処理方式の諸問題を中心に述べられている。

FACOM-R (RE)リモート・ターミナル設置場所

通商産業省・工業技術院機械技術研究所	1台
㈱東京繊維情報センター	1台
当財団	3台

5.2 利用状況

リモート・バッチ処理の利用状況は表 5 - 1 に示すとおりである。

センターにおけるリモート・バッチ処理の利用は、ソフトウェアの開発が主であるが、入力データの比較的少ない場合に用いられている。これは端末のカード・リーダが 100枚／分の速度のため、入力データが多い場合はバッチ処理を用いる。

出力は、センター出力、端末出力の両方が可能のため、出力量が多い場合はセンター出力を用いるようである。いずれにせよ、入力量が多量でない場合は、ユーザにとってターンアランドの速さがリモート・バッチ処理の魅力であろう。

表5-1 端末別月別リモート・バッチ利用件数

端末名 月	JIA (センタ)	JIB (センタ)	JIC (センタ)	JID (センタ)	JIE (センタ)	JIF (センタ)	MIT (通産省)	MPS (郵政省)	JSP (機振協)	MEL (技研)	TTI (織維)	計	平均
5	0	0	7	58	43	0	0	0	0	0	5	113	10.3
6	0	3	1	49	22	0	0	0	0	0	16	91	8.3
7	0	0	1	5	9	0	0	0	0	0	3	18	1.7
8	0	1	15	26	36	0	0	0	0	27	0	105	9.5
9	0	8	10	22	36	0	0	0	0	47	1	124	11.3
10	5	1	28	64	137	2	0	0	0	54	0	291	26.5
11	1	0	44	163	180	0	0	0	0	149	1	538	48.9
12	0	0	64	148	143	0	0	0	0	6	0	361	32.8
1	2	0	21	123	92	1	0	0	0	54	0	293	26.6
計	8	13	191	658	698	3	0	0	0	337	26	1,934	175.8
平均	0.9	1.4	21.2	73.1	77.6	0.3	0	0	0	37.4	2.9	214.9	

5.3 機械技術研究所におけるリモート・バッチ処理方式の利用

5.3.1 利用の目的と経緯

工業技術院機械技術研究所においてはその研究上の必要性から、日本情報処理開発センタのタイム・シェアリング・システム（FACOM230-60）のリモート・ターミナル（FACOM-RE）が昭和47年（1972年）7月末に設置され、同8月15日から稼動に入った。

すでにこのターミナルは、その使用の主目的である機械工場向けの一種のアプリケーション・プログラム・システムの開発や利用実験に活用されているが、昭和42年11月には、東京・晴海で開かれた日本・国際工作機械見本市において当会場にこのターミナルを臨時に移設して、そのプログラムの処理実演を参考展示するという試みも行った（図5-1）。このアプリケーション・プログラムは旋盤による加工、すなわち旋削加工用の自動プログラミング・システムであり、略称をMELTS（前記見本市当時のKKTSを含む）といい、図5-2に示すような数値制御（NC）旋盤が主な対象となるものである。



図5-1 東京国際工業機械見本市における様子
(47.11.15~26, 晴海会場にて)

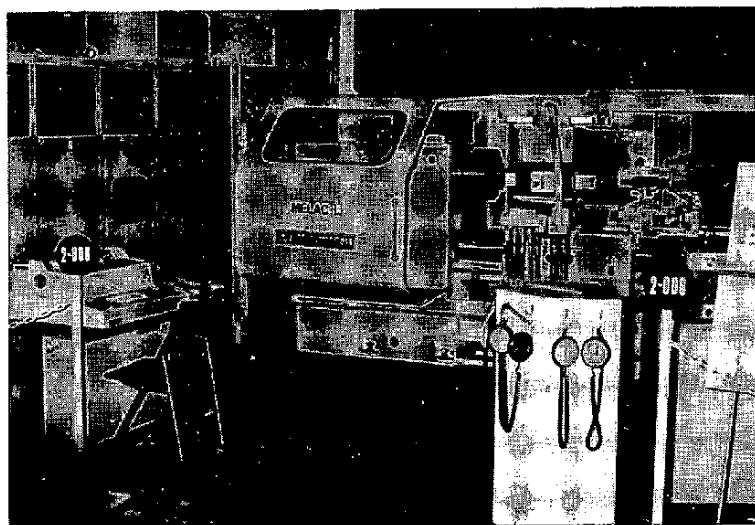


図 5-2 本システムの出力の対象となる旋盤

ここに、リモート・ターミナルの使用の状況を記すに当って、まずターミナルが必要と考えられるに至った経過をまとめる意味で、当研究所そのものや対象研究テーマの由来について記しておこう。

機械技術研究所 (Mechanical Engineering Laboratory, MEL) は、通商産業省・工業技術院に属し、機械工業に関するわが国唯一の国立研究機関として、機械工業の根底となる基礎的研究および新技術の開発研究を行なっている。旧名を機械試験所といい、昭和12年(1937年)に創立され、昭和23年(1948年)に工業技術庁(現在の工業技術院)設置に伴ないその所属となり、昭和46年(1971年)に機械技術研究所と改称して現在に至っている。

機械技術研究所の特色の一つは、機械を生むための機械である工作機械や加工技術に関する研究の伝統にあり、戦前からの一つの大きな軸となっている。それらに関連した自動化の研究関係で昭和30年(1955年)以降ぐらいからいくつかを拾ってみると図5-3のようになる。

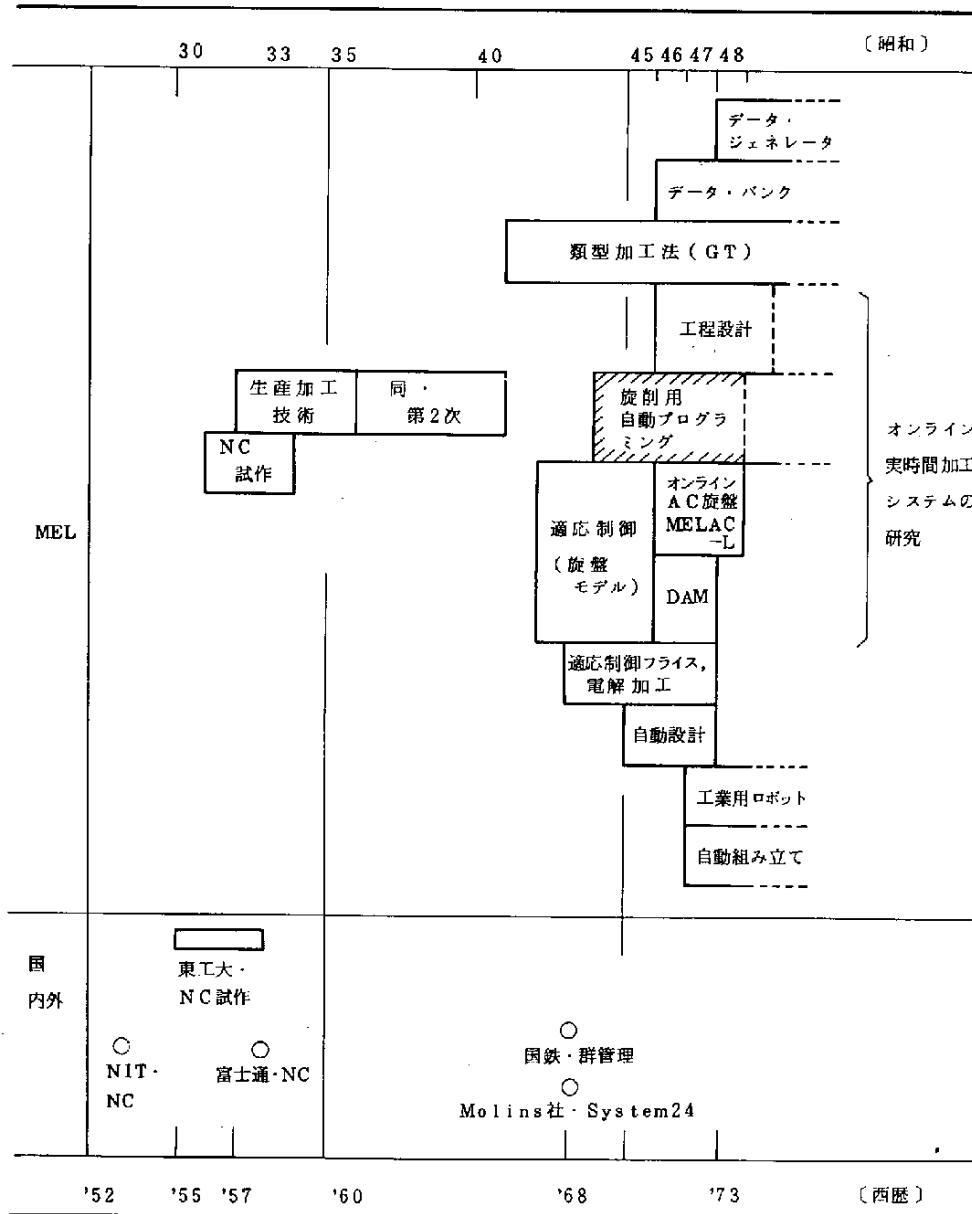


図 5-3 機械研における研究テーマの例

工作機械に対してその駆動の自動化とディジタル化とに貢献した数値制御は、1952年（昭和27年）にアメリカのMITから発表されたが、国内では昭和32年（1957年）に東京工業大学がNC旋盤を、また富士通がNCタレット・パンチプレスを発表した。機械研では昭和33年（1958年）にNCジグ中ぐり盤を試作している。

これとほぼ併行して、通産省の機械工業審議会の答申に応えて工業技術院の工業技術協議会に部会が設置され、学会や業界と連携を保ちつつ研究実施長期計画を策定すると同時に昭和32年から、生産加工技術に関する特別研究が所内に委員会を組織して開始されている。ここではまず、切削、研削、機械要素の加工、塑性加工について行なわれたが、ついで昭和36年からの第2次5ヶ年計画では、作業標準、新加工法、加工装置について進められた。

その後、窪田所長を中心としてわが国向きのグループ・テクノロジ（類型加工法）を探る研究が開始され、中小企業庁や機械振興協会などの協力によりKC-1, KK-1などの分類法が提案されてきた。

他方、加工の最適化のための新しい方向が、竹山（生産工学部）や当時の土井・物理制御部長（現・東大教授）を中心として昭和41年頃から検討され始めた。ここにおける一つの問題は、加工の最適化に対してコンピュータがどのように用いられるか、また、それまでに集積された多量の加工データをコンピュータによってどのように処理しうるか、ということであった。

もともと加工、とくに機械加工は、どの機械工場にも存在しうる基幹的な領域でありながら、殆んどの場合は工作機械の技術者や作業者の知識や熟練に頼っており、この加工のノウハウの処理の問題は機械工場の自動化にとって常に大きく立ちふさがり、その進歩が望まれるところであった。

さて、上記の検討の一つの帰結として、適応制御の研究が開始されたが、これは加工中の状態、たとえば切削力や加工精度などを検出器によって測定しながら、加工がなるべく最適になされるように、計算機流にいえばリアル・タイムに制御してゆくものである。この研究は昭和45年までに一段落し、その成果

は試作された適応制御旋盤 M E L A C - L として、やはり前記見本市に参考展示された。会場でリモート・ターミナルから出力された結果の切削実績もこの機械を用いて行なわれた。

他方、適応制御のように実際の検出器を用いて刻々の処理をしないで、あらかじめソフトウェアによって加工プロセスのモデルを作成しておき、加工に関する仕様が与えられれば、加工に先立ってその一部始終をモデルの上で推測して、最適な加工の仕方を求めてみる、という手法もあるはずである、ということから、旋削加工における荒削りの切削条件を求めるアルゴリズムが検討され、実際に当時、機械研で研究上の計算用に用いていた中型パラメトロン計算機でインプリメントして、昭和43年頃には答を出していた。

これは、前述のような加工のノウハウの自動処理という問題への一つのアプローチであった。この頃から N C 工作機械がわが国でも普及の方向へ向いつつあったが、N C は加工の手順が定まったあと紙テープを作り得ればそれを用いて工作機械の駆動の自動化に役立つというもので、むしろ高価な N C 機をより効率的に使うためにも、この加工のノウハウの蓄積と体系化という問題はより大きくクローズ・アップされることとなった。この研究の過程において、欧米におけるこのような方面的試み、とくに N C のための紙テープ（N C テープ）をより効率的に作成するためのプログラミング・システムの試みが伝えられるようになった。

昭和44年（1969年）からは機械研の特別研究として、加工のノウハウを含んだ加工用の自動プログラミング・システムを開発するべく、加工の多様性に鑑み、まず我国で最も普遍的な旋削加工に例をとり、数値制御プログラミングの研究という名称で発足した。ここにおいては加工のノウハウの体系化を進め、かつ一つの加工用自動プログラミング・システムとして、なるべく図面上の情報を直接的に記述出来る言語体系を作り、工場で使われている作業指導書（作業指示書、指示票などいろいろに呼ばれる）、N C テープ、プロッタ図面などを出力するソフトウェアのインプリメンテーションを、より本格的に進めることとなつた。

この研究に呼応して、昭和45年(1970年)には機械振興協会技術研究所によって斯界の学識経験者やエキスパートから成る専用プログラム研究委員会(委員長土井康弘東大教授)が設置され、機械研における研究成果、全国の機械工場を対象とした2度にわたる技術アンケート調査などをもとにアルゴリズム化の検討を重ねた。そして日本情報処理開発センタに設置のFACOM230-60を用いてインプリメンテーションを進めた。このバージョンでは、まず旋削加工におけるセンター作業(主として軸状の機械部品が対象)を対象としK-KTSと呼ばれた。計画当初にはシステムの規模や構成についていろいろの考え方があったが、(1)このような新しい試みについてはシステムを技巧的にコンパクトにすることも必要だが、まず本来の内容に力を注ぐこと、(2)機械工場における大型計算機の利用方式についても併せて検討していく、という観点などから、大型計算機を対象とすることとしたものである。

この研究関係の前後の経過を表5-2に示す。表にもある如く、昭和46年からは機械研においては、この自動プログラミングの研究は他の二つの研究、「自動生産システムのための工程設計」、「オンライン実時間旋削加工システム」と共に、「オンライン実時間加工システム」という特別研究となり、機械工場に対するコンピュータのより有機的な利用方法を検討することを目指した(図5-4)。

昭和47年の前半まではFACOM230-60はセンター・バッチで利用していたが、昭和47年8月からはリモート・バッチでも用いられるようになり、MELTS(MEL Turning System)の開発、利用ともリモート・ターミナルからも行なえるようになって現在に至っている。

表 5-2 旋削加工用自動プログラミング・システム MELTS 及び関連テーマの経緯

年	KKTS/MELTS 関係	関連研究テーマ
昭和41年(1966)	○加工の最適化とコンピュータの応用について検討を行なう。	
42 (1967)		○適応制御の研究を特研として開始し、MEL構内で、コンピュータと旋盤とを結合。
43 (1968)	○旋削・荒削りの切削条件自動決定プログラムを作成。	
44 (1969)	○「数値制御プログラミングの研究」として特研となる。	
45 (1970)	○日本情報処理開発センタのFACOM 230-60 を使い始める。	○機械振興協会技術研究所に研究委員会発足。
46 (1971)	○他の二つのテーマとともに、特研中項目「オンライン実時間加工システムの研究」にまとまる。	○同上終了。
47 (1972)	○日本情報処理開発センタ FACOM 230-60 のリモート・ターミナルをMEL内に設置(7月) ○東京・国際工作機械見本市に参考展示(11月。期間中、ターミナルを晴海会場へ設置変更)。	○データ通信適応制御マシニングセンタ加工システム「DAM」の公開実験(5月、MEL~神奈川県間。民間2社との共同研究)。 ○左記見本市に適応制御旋盤「MELAC-L」も併せて参考展示。
48 (予定)	○ターミナル側へプロッタを増設。 ○内容のレベル・アップ。 ○CIRP(国際生産加工技術会議、9月ユーゴスラビア)へ報告。	○「DAM」をMTDR(国際工作機械設計研究会議(10月、イギリス)へ報告。 ○電々公社武蔵野電気通信研究所と加工のソフトウェアに關し研究関係。

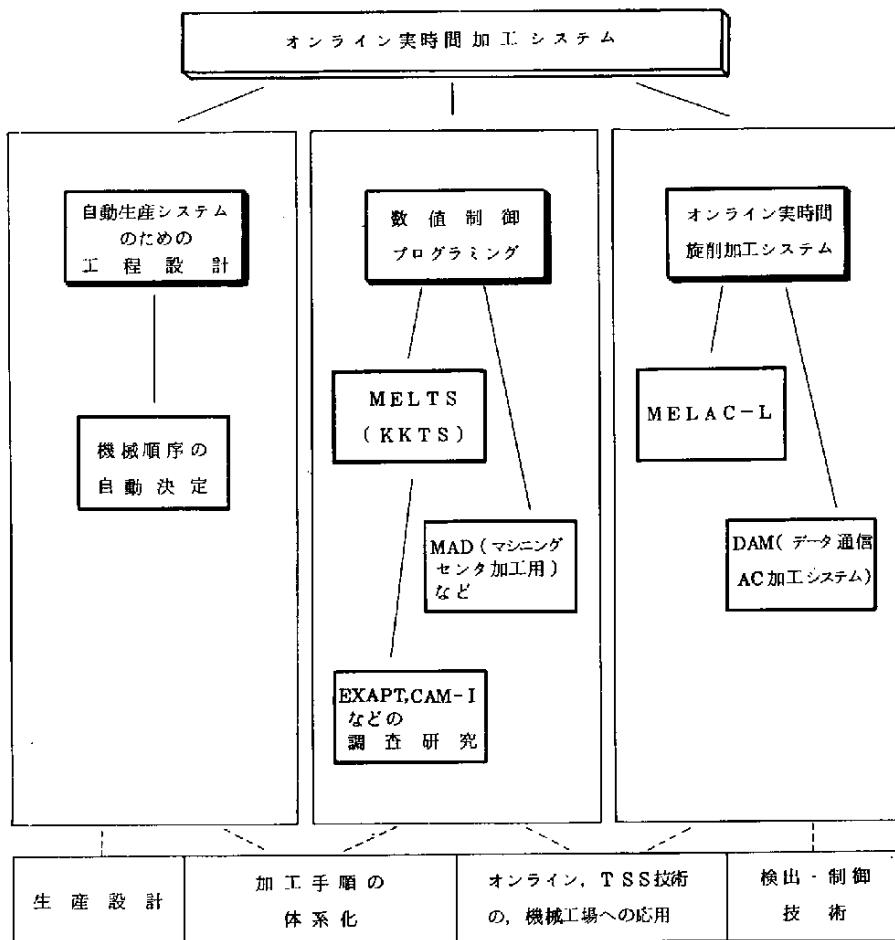


図5-4 オンライン実時間加工システムの研究内容

5.3.2 リモート・バッチ方式採用の理由

日本情報処理開発センターの機械室で FACOM 230-60 を利用しつつ、昭和 46 年度にはそのリモート・ターミナルの設置を計画した。その理由は次のごとくであった。

- (1) 加工用の自動プログラミング・システム開発上の必要性：当システムの研究の担当者、すなわち大型計算機の利用者の大半が勤務する MEL は、東京都杉並区井草 4 丁目にあり、ここから日本情報処理開発センターまでの所要時間は通常で 1 時間 10 分程度である（その地理的関係は図 5-5 のごとくである）。この地理的にやや遠隔であることが、他の研究テーマもかかえている担当者にとってかなりの負担となつた。

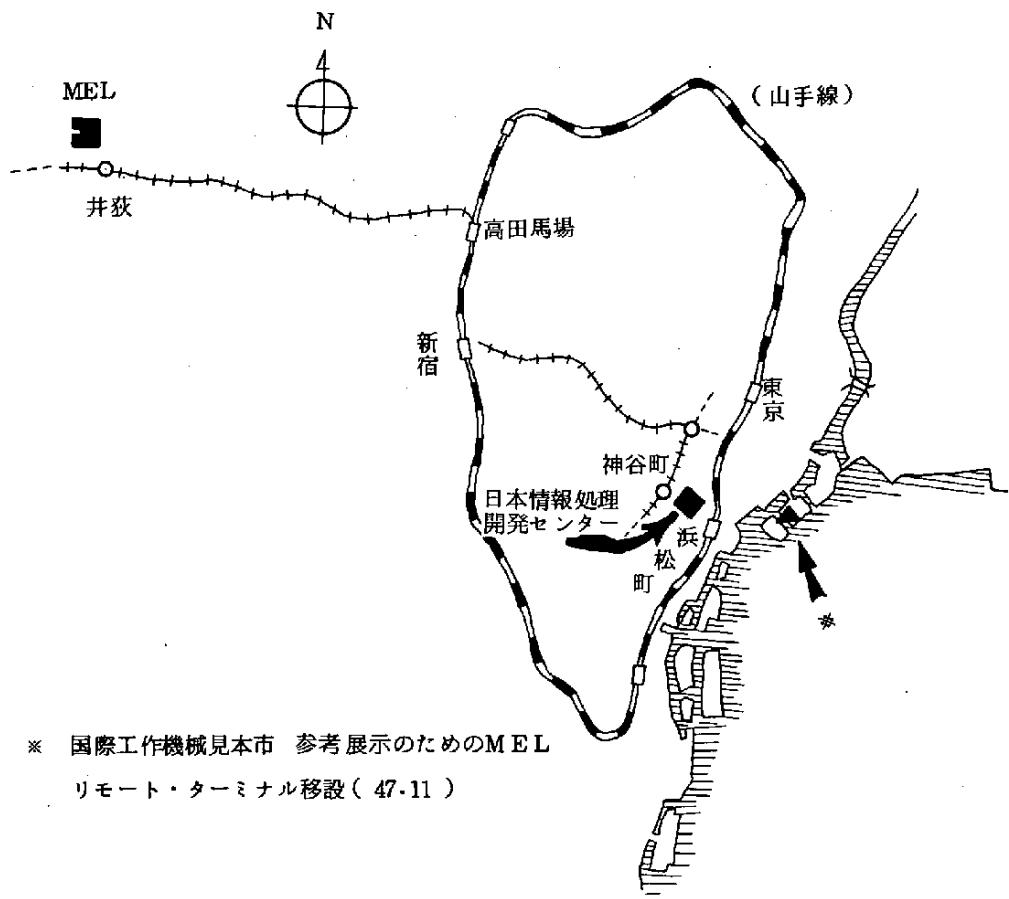


図5-5 日本情報処理開発センター～MEL間の地理的状況

もとより、かなりまとまったデバッグингやアップデートはセンタ・バッチで行なった方が効率がよいが、

- (a) サブルーチンのコンパイルや部分的なデバッグ、
- (b) MELに置いてある資料を参照しつつ行ないたいもの、

についてはMELに居たままで作業を行なうのが効率的である、という結論であった。

(2) 研究としての必要性：機械工業における自動化やコンピュータリゼーションというような境界領域的な分野は、とかくスムーズなアプローチがされにくい。関連するそれぞれの領域の技術や情報が円滑に交換され、種々の前向きの試みがなされて経験やデータが蓄積されてゆかなければ、思いつきでは

ない本格的な技術は開発されにくい。

最近のコンピュータ技術の目覚しい進歩は各分野に大きな影響を与えつつあるが、機械工業もその例外ではない。とくに大型コンピュータやそのオンライン技術、TSSなどは重要であり、しかもミニコンなど小型のものとも異って一寸した経験でマスターしうるものでもない。MELにおいては、表5-2にも記した「DAM」のようなモデルによってオンライン・システムの検討を行ってはいたが、より本格的に取り組む必要があると判断したものである。これによって機械工場向きオンライン・システムにおける利用体系、言語形式、技術データの扱いなどに関するデータをゆくゆくは得たいという考えである。

なお、これ迄世界各国で試みられた機械加工に関する、いわゆる自動プログラミング・システムの例をその内容的なレベルとともに表5-3に示した。

表5-3 自動プログラミング・システムの例

名 称	主たる対象	機能・形体	開 者	所有または開発	計 算 機	自動化	実 用
APT	一 般	多軸(3軸以上)の輪郭切削まで全機械加工	APTとは一般にAPTⅢまたはAPTⅣのことを指す	IITRI アメリカ	IBM, UNIVAC, CDC 国産大型機種など	◎	
2C,L	*	APTのサブセット、改良点もある	(2P,L),(2C)など穴加工や旋削加工対象もある	NEL イギリス	UNIVAC ICL		
IFAPTC	*	APTのサブセット	IFAPT Pなどもある	ADEPA フランス			
ADAPT	*	APTのサブセット	APTの言語体系に含まれる	IBM	IBM, TOSBACなど	◎	
MINIAPT	*	APTのサブセット(32KB)	沖電気でも同一名称で作製	Horn NC ドイツ	UNIVAC(西ドイツ)	◎	
EXAPT3	フライス加工	APT的	開発中	EXAPT協会 ドイツ	UNIVAC(?)		
FAPT	*	独自の言語	FAPT2を中心として、会話形式その他抜粋している	富士通ファナック	FACOM関係	◎	
HAPT	*	独自の言語	HAPTシリーズとして多種類のルーチンを開発している	日立精工 日立製作所	HITAC関係	◎	
PROFILE DATA	*	独自の言語	イギリスのみならずヨーロッパで広く使用されている	Ferranti イギリス	IBM, ICL, FP, CDC		
SPLIT	5軸のマシニングセンタ	独自の言語	OMNIMIL用として最適	Sundstrand アメリカ	UNIVAC	◎	
AUTOSPOT	穴 加 工	独自の言語	ボール盤などに適合	IBM	IBM	◎	
OKISPOT	"	独自の言語	冲電気			◎	
EXAPT 1	"	图形はAPT的、テクノロジ部あり	ドイツを中心としてヨーロッパでは実用化している	EXAPT協会	UNIVAC, GE, ICT, Siemens, IBMなど	◎	○
EXAPT 2	旋削加工	图形定義はAPT的、テクノロジ部あり	ドイツを中心として機能拡張中	EXAPT協会	UNIVAC, CDC, IBM, Siemensなど	◎	○
AUTOPIT	"	EXAPT 2と同一	EXAPT 2の前身	Pittler (西ドイツ)	IBM	◎	
AUTOPOL	"	フィックスドフォーム		IBM(西ドイツ)	IBM		
KOBRA	"	独自の言語		ノールウェイ工科大学	UNIVAC	◎	
GETURN	"	独自の言語	対話形式、オランダTNOで開発された	オランダTNO; GE	GE	◎	
CUT	"	独自の言語		Warner & Swasey	IBM		
UTURN	"	独自の言語	UCC APTに付加できる	UCC アメリカ		◎	
FMILL	自由曲面	3次元点群より自由曲面創成 APTと結合可能	— — —	Boeing	IBM, UNIVACなど	◎	
GEMESH	"	"		GE	IBM, GE, UNIVACなど	◎	
FMESH	"	3次元点群より自由曲面創成し加工処理可能		富士通	FACOM関係	◎	
HAPT-3D	"	"	HAMESHも自由曲面創成	日立精工	HITAC関係	◎	
研究開発の例							
MELTS	旋削加工	独自の言語	加工技術の自動化の追究	機械技術研究所	FACOM	◎	
STORK 2	"	独自の言語	同 上	東大工学部 機械振興協会 技術研究所		◎	
グラフィック・アート	*	独自の言語			HITAC		

* 一般的に2 $\frac{1}{2}$ 軸のフライス加工を主たる対象としているが、穴加工および旋削加工用にも利用しうる体系である。2次元空間の图形を取り扱うが、面(XY, YZおよびZX平面)を選択することにより立体を取り扱うことができる。

◎ 国内で実用化されている。

○ 国内でテスト的に使用されている。

◎ 加工技術に関する自動化を目指している。

5.3.3 機械研のリモート・ターミナルの構成

機械研のリモート・ターミナルの機器構成と、各々の大よその仕様は表5-4に示すとくである。いずれもターミナル・コンピュータとしてのFACOM-R-Eの標準的な機器のうちから、必要性と予算に応じて設置した。

後述のごとく研究の対象となるシステムが、入力はカード、出力はLPによるリスト、紙テープ及びプロッタ図であるので、カード・リーダ、ラインプリンター、低テープ・パンチャは必需品であった。紙テープ・リーダは、担当者等が別に従事しているテーマ（適応制御旋盤MELAC-L）等のために、R-FASPを空き時間に利用しうるようにすることも考えて取りつけた。プロッタは予算の都合上、初年度は必要に応じてセンタのプロッタで代用することとし、昭和48年度に設置する予定である。

図5-6はシステムの構成である。ただしセンタ例は、MELTSにおいて用いられる機器構成のみが画かれている。

なお、リモート・ターミナル側には将来、さらにオンライン・カード・パンチャおよびテレビ・ディスプレイ（出来ればキャラクタ・ディスプレイが希望

表5-4 機械技術研究所、リモート・ステーション機器構成及び仕様

構成機器	台数	仕様
中央演算処理装置 (FACOM R-E)	1	基本命令数 28種 サイクル・タイム 1.8マイクロ秒 内部記憶磁気コア8K語
回線制御アダプタ	1	伝送速度 1200 ポー用
タイプライタ装置	1	印字速度 600字/分 紙テープ用メカニカル・リーダ、同パンチャ付き
カード・リーダ	1	読み取り速度 100枚/分
ライン・プリンタ	1	印字速度 120~240行/分
紙テープ・パンチャ	1	50字/秒
紙テープ・リーダ	1	200字/秒

であるが、技術的に困難であるということなので)とを増設したいという希望を持っており、検討を始めたいと考えている。

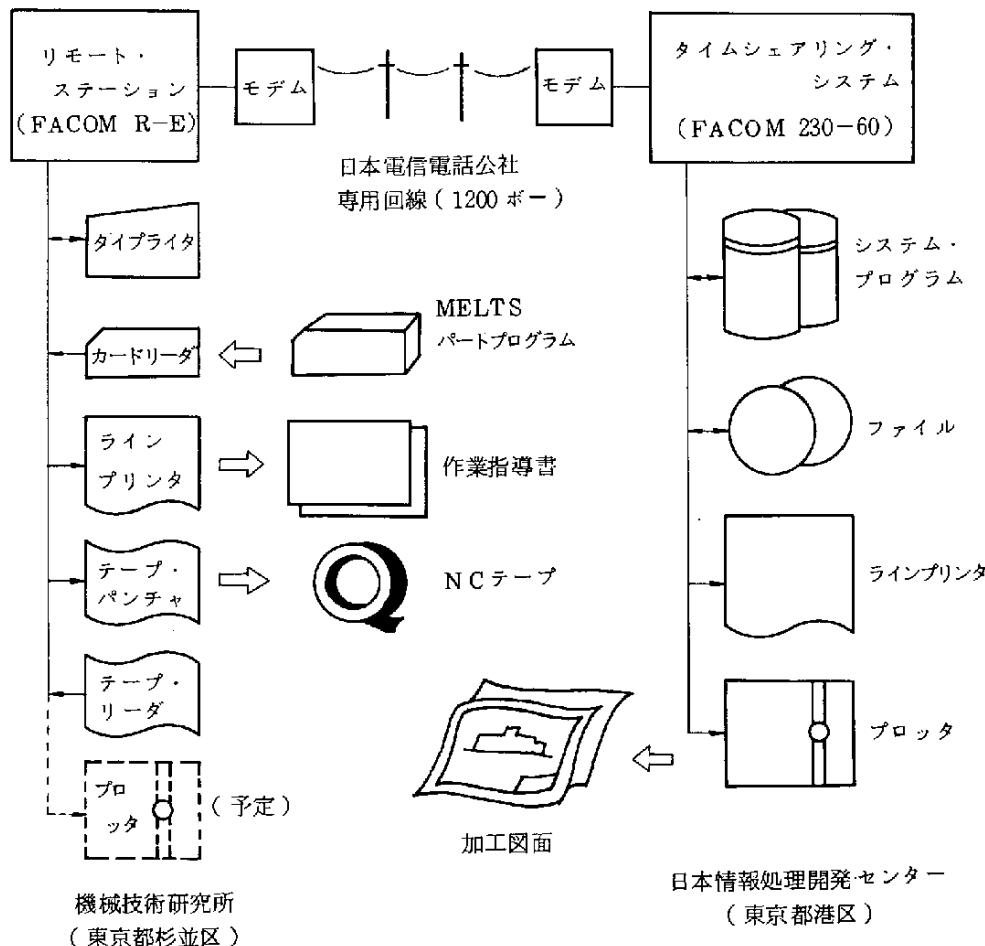


図 5-6 オンライン・システムの構成

5.3.4 旋削用自動プログラミング・システムMELTS

(1) MELTSの構成

図5-7にMELTSの全体の構成と、入力から出力までのデータの流れを示した。MELTSは、MELTS言語、システム・プログラムおよびデータ・ファイルから構成されている。その各々について簡単に説明しておこう。

a システム・プログラムとデータの流れ：トランスレータ，テクノロジー部およびポストプロセッサの3つの部門から構成されており，100%FORTRAN(IV)言語(FACOM 230-60)によって記述されている。MELTS言語によるパートプログラムはトランスレータにおいて，後加工(研削など)や寸法許容差を考慮した形状寸法に変換され，嵌め合い公差，表面あらさおよび基準面などの加工上必要なデータが計算される。形状を示す座標値やそれらに付随する諸データは，MELTSの心臓部であるテクノロジー部の処理を円滑に行うための，リング構造によるMELTSデータ構造に格納される。

テクノロジー部は後述するように9つのステージに大別されており，その各ステージにおいて順次，加工工程，切削条件(主軸回転数，送り速度およびきりこみ)，工具経路などが決定される。これらの処理はMELTSデータ構造を用いて行われる。切削の順序に従って求められた全工具の位置と，その工具が動くために必要なTコード，Fコード，SコードなどでCLデータが作られる。ポストプロセッサはCLデータより使用するNC旋盤用のNCテープを作る。なお，現場用および工具管理室用の作業指導書と，自動決定された工具経路を示すプロッタ図面が出力される。

b MELTS言語：熟練した加工技術者は，図面上の情報によって加工工程を決定する。この事から，製品の仕様がすべて指定されている図面上の情報をなるべく直接的で簡潔に記述することができ，システムのテクノロジー部に必要なデータを完全に供給することを目的として，MELTS言語は設計された。

c データ・ファイル：機械ファイル，材料ファイル，工具ファイルおよび作業ファイルの4グループから構成される。

(a) 機械ファイル：使用するNC旋盤の仕様を入れるファイルでテクノロジー部やポストプロセッサによってそのデータが用いられる。

(b) 材料ファイル：切削条件(f, v, d)を計算するステージで専ら使用される。機械技術研究所における旋削標準などの研究データを基礎と

して材料を材種別にグループわけして、工具寿命式、切削抵抗式などの係数として格納している。また特殊材料などでデータの数式化が行えない材料については、適正条件を直接ファイルできるように設計されている。

- (c) 工具ファイル：標準工具および手持工具は全て工具コードを付して登録する。刃部の形状、シャンク寸法、拘束条件および評価関数関係などのデータが格納され、工具決定のステージ等で用いられる。
- (d) 作業ファイル：ねじ切り作業や深穴のドリル加工など、蓄積データが少なくアルゴリズム化がむつかしい段階の作業について、経験的に行わ

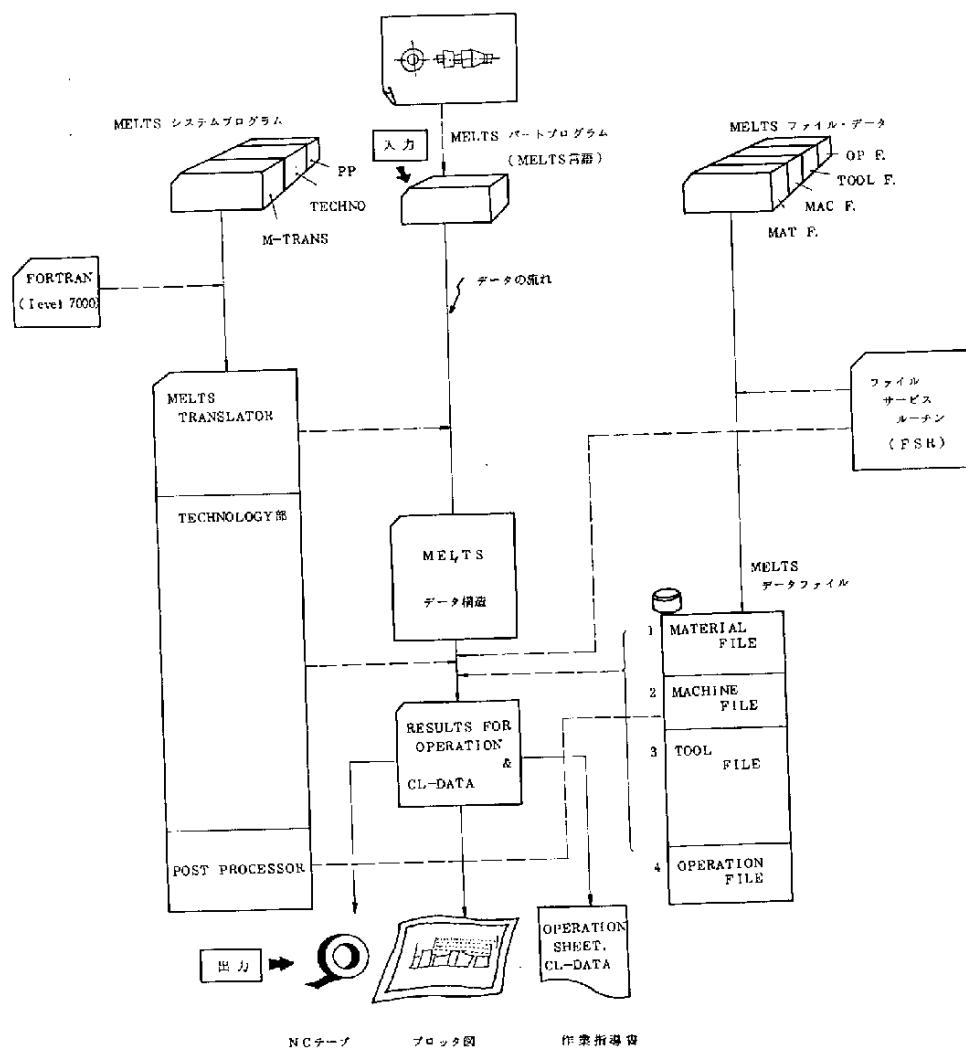


図 5-7 MELTS の構成

れているワーク・サイクルなどのデータが直接利用できるようにファイルされる。

4 グループのファイル・データはパートプログラムからの指定によって取り出される場合と、システム・プログラムが必要に応じてコアに読み込む場合がある。

(2) M E L T S の言語体系

図面上の情報をなるべく直接的で簡潔に記述できることを目的としてM E L T S 言語が設計されたことは前にも述べたが、さらに次の点についても留意した。

- カードのカラムに一さい無関係な形式（会話形式への拡張を考慮して）
- ボキャブラリ・ワードは旋削加工に馴染みのあるもの（現場の技術者に使用しやすい）
- 煩雑さを避けるために、特定の省略形を少なくし、全ボキャブラリ・ワードは頭から4文字までで識別する。

通常の旋削部品は図5-8に示す12個の基本的な図形に分解されるものとし、これらを“基本形状要素”と呼ぶ。素材および仕上がり形状は基本形状要素のつながりとして記述され、基本形状要素の1つに対応する、図面を構成する1線分を“セグメント”という。各セグメントは、基本形状要素を示すF, L, T, R, C, O, P, G, Q, D, BおよびSの記号に続く、1桁乃至3桁の数字によって表わされるセグメント名がつけられ、それに対応する図面上の諸データを用いて定義される。

M E L T S のパートプログラムは図5-9に示すように“素材形状の部”、“仕上がり形状の部”および“オペレーション部”的3つの部分から成り立っている。

各々の基本形状要素の定義文は2種類乃至9種類あり、加工の質に関する指定には、セグメントの定義文の中に記述する方法と、形状全体に対して指定する方法がある。寸法精度、アラインメント、表面あらさ、後加工について指示することができる。オペレーション部で使用するN C 旋盤、素材の

材質などを指定することによって、前もって作られているMELTS・データ・ファイルから必要なデータをコアに呼び出す手振りとする。図5-10に、MELTS言語によるパートプログラムの例を示した。

この例は旋盤作業のセンター作業の部品として形状の複雑度は普通程度であるが、MELTSへの入力であるパートプログラムでは約30ステートメント程度となる。パートプログラムの大きさは〔(素材のセグメント数)+(製品のセグメント数)+10〕の約1倍から1.2倍のステートメント数でおさまるだろう。MELTSは素材と製品の各々が約100個のセグメントから形成されている形状まで処理できるようになっている。

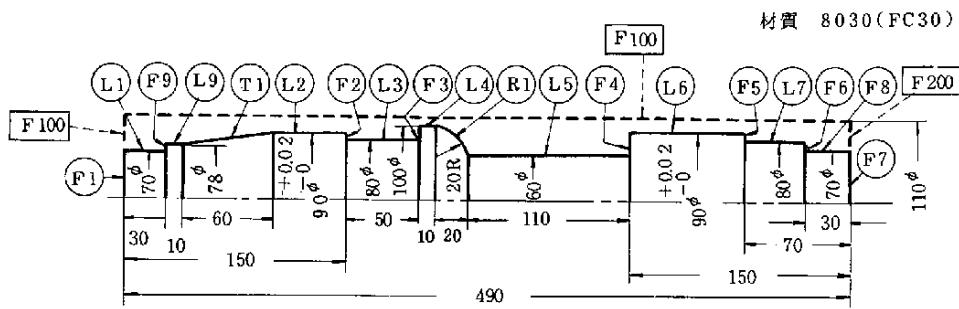
基本形状要素	シンボル	折曲する形状の例	補助線としきのシンボル
端面	F	L, T	AF
長手	L	—	AL
テーパー	T	~, ~T	AT
円弧	R	~, ~	AR
面取り	C	~, ~	
丸め	O	~, ~	
溝	G	U, V	
ぬすみ	Q	L, L	
ねじ	P	—	
総形	S	V	
中ぐり穴	B	—	
ドリル穴	D	—	

図5-8 基本形状要素

[MELTS statements]	説明
PARTNO = ...	パートプログラムの始り
BLANK	大見出し(素材形状の部のはじまり)
{ <u>LTOR</u> } *	中見出し(形状を記述していく向き)
RTOL	
{ <u>EXTERNAL</u> }	小見出し(外径または内径の指定)
INTERNAL	
s i ** = ...	形状の記述
:	
SPECIFICATION	大見出し(仕上がり形状の部のはじまり)
{ <u>LTOR</u> }	中見出し(形状を記述していく向き)
RTOL	
{ <u>EXTERNAL</u> }	小見出し(外径または内径の指定)
INTERNAL	
s i = ...	形状および加工の質の記述
:	
OPERATION	大見出し(作業条件の部の始り)
MACHINE = ...	使用する機械のファイル名の指定
MATERIAL = ...	" 素材 "
CLAMP = ...	" 保持方法の指定 "
:	加工条件の指定など(オプション)
END	パートプログラムの終り

- * { } の中のワードはいずれかが選択される。何もつかれない場合はアンダーラインを付したワードがあるものとみなされる。
- ** セグメント名を示す記号

図 5-9 MELTS のパートプログラムの構成



部品番号 KANKAN-13

```

@ ***** MELTS *****
@ KANKAN13 1972.11.19 @
@ ***** *****
@ PARTNO=KANKAN13
BLANK
LTOR

@ SPECIFICATION
LTOR
EXTERNAL
F1 =Z/10
L1 =DIA/70, LENG/30
F9 =Z/30
L9 =DIA/78, LENG/10
T1 =DIA1/78, DIA2/90, LENG/60
L2 =DIA/90, UT/0.02, LT/0, $  

LENG/50, TRIA/2
L2 =DIST/150, FROM/F1

L7 =DIA/80, LENG/40
F6 =DIST/-30, FROM/F7
L8 =DIA/70, LENG/30
F7 =Z/490
SURFACE=RA/100

@ OPERATION
MACHINE=MELAC
MATERIAL=8030, HB/160
CLAMP=PROCESS/1, DRIVER/SCEX
@ END

```

図 5 - 10 パートプログラムの例

(3) テクノロジー部の概要

M E L T S の概要を図 5-11に示した。テクノロジー部の 9 つのステージにおいて逐次加工の工程が自動決定され、出力ルーチンによって、保持法、ツーリングなどを指示する作業指導書をプリントし、N C テープを出力する。また指定により工具経路をプロットした図面も出力される。テクノロジー部は次の 9 つのステージを順次たどる。

- a パターン識別ステージ：セグメントのいくつかの連なりは、P, V, M または R のどれかの “パターン” を形成し、素材および仕上がり形状は、これらパターンの連なりによって特徴付けられ、一つの “パターン・タイプ” を有する。このステージは、対象とする部品がどのようなパターン・タイプに属するかを識別する。また形状の凹凸の大小関係等も把握する。
- b オペレーション・レベル・ステージ：切削においては一般に荒削り、中間仕上げおよび仕上げの 3 通りがあるが、これを 1 = [荒削り] ; 2 = [荒削り] + [中間仕上げ] , 3 = [荒削り] + [中間仕上げ削り] + [仕上げ削り] で表し、オペレーション・レベルと呼ぶ。この 3 段階のレベルはセグメント毎に決定される。図 5-11 の右側に例示するように、このステージでは寸法許容差、 L/D (部品の全長／部品の最小直径) および素材の熱処理の状態によって 3 つのレベルに区別される。セグメントレ 1 は直径方向の IT ナンバが小さい所から精度 (T) が良いクラスに属し、オペレーション・レベルは 3 となる。
- c クランピング・サイド・ステージ：クランピングの回数を確定し、第 1 工程では部品の左右いずれから切削をはじめるかを決定する。
- d 切削範囲ステージ：各工程 (チャッキング) の切削可能の範囲をきめる。
- e クランピング・ポジション・ステージ：パートプログラム内で指示された部品の保持方式により、チャッキングするものについては、機械ファイルのチャックの仕様を用いて加工物と爪との対応をつける。
- f 工具選択ステージ：加工に必要な全工具を決定する。すなわち加工物の形状、オペレーション・レベル、刃物台の制約などを考慮して、工具ファ

イルの中から適当な工具を選定する。

- g とりしろステージ：荒削りにおける工具経路決定の方針を定め、その工程に従って順次セグメント毎のとりしろを計算する。
- h 切削条件ステージ：(7)のステージで計算されたとりしろをオペレーション・レベルに従って、荒削り用、中間仕上げ用および仕上げ用に分割する。分割されたとりしろについて、素材の材質、形状、工具の材種および保持方式、機械の動力、送り速度および切削速度の制限などを斟酌して、きりこみ、トラバースの回数、送り速度および切削速度の組合せを求める。たとえば荒削りに際しては単位時間当たりの切削容積が最大となるような切削条件を求める。また中間仕上げおよび仕上げ削りについては表面あらさの指定が考慮されて、送り速度および切削速度が計算される。送り速度および切削速度については対応するコードが機械ファイルにより決定される。
- i 工具経路ステージ：(8)ステージで決定された切り込みおよびトラバースの回数に従って、荒削り、中間仕上げ、仕上げおよび溝入れなどの全工具経路を決定する。仕上がり形状と使用工具の干渉チェックを行いながら、荒削りから順次工具の位置を求め、その工具が動くために必要なTコード、Fコード、Sコードなどと共にCLデータを作る。
- j ポストプロセッサ：機械ファイル中の数値制御に関するデータを参照しながら、CLデータをNCテープのコードへ変換する。上述のテクノロジー部でかなりのことが処理されているため、ここでは対象とするNCの特殊なルールを付加するなどの他は、どのNCに対しても共通的な機能でよい。

(4) MELTS の出力

図5-12(a)は目で見るMELTSの入力および出力である。中間結果の主なものは、前述のテクノロジー部の各ステージの処理結果であり、作業指導書に出力しない詳細な結果もプリントする。次のバージョンではこれ等のプリントをパートプログラムにおいて選択できるように改良中である。

同図(b)は見本市会場にて、MELTSより出力されたNCテープで荒加工

を行った製品であり、同図(c)はその第1工程の工具経路を示すプロッタ図面である。

なおMELTSの出力したNCテープで切削している様子を図5-13に示した。

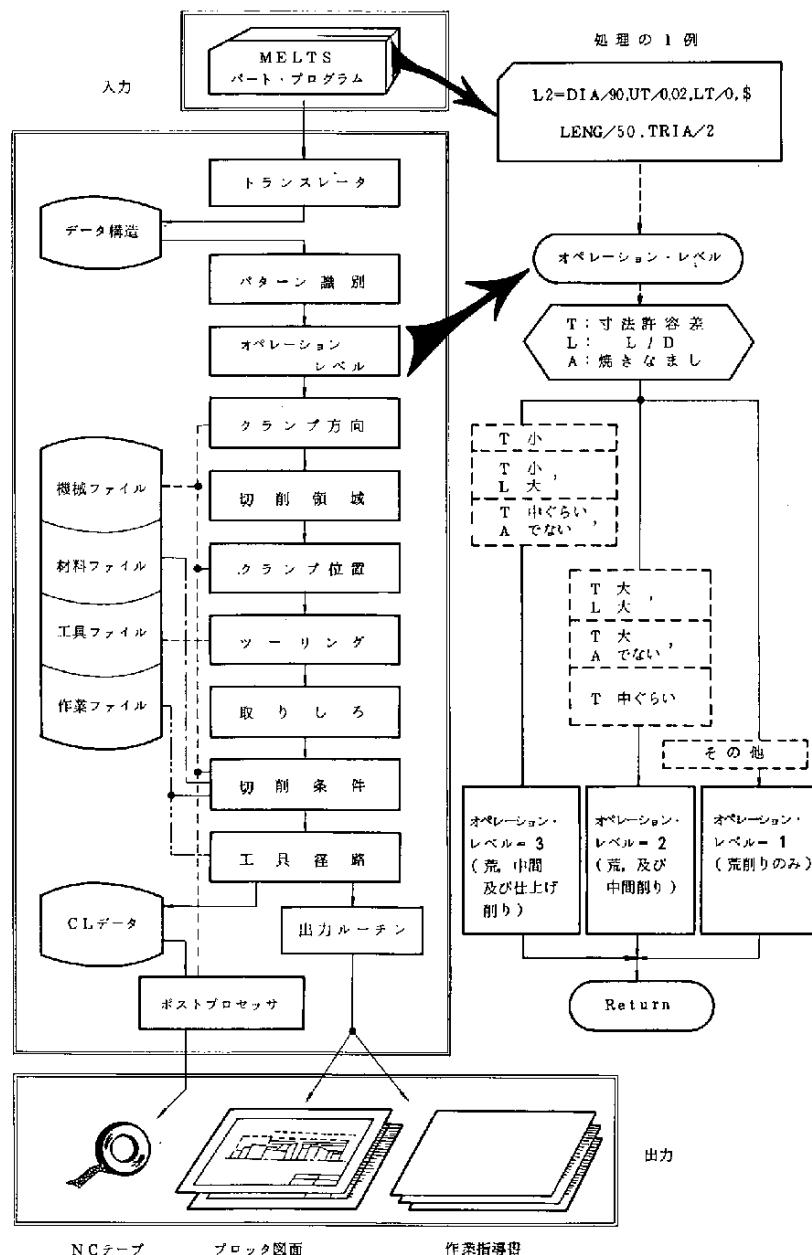
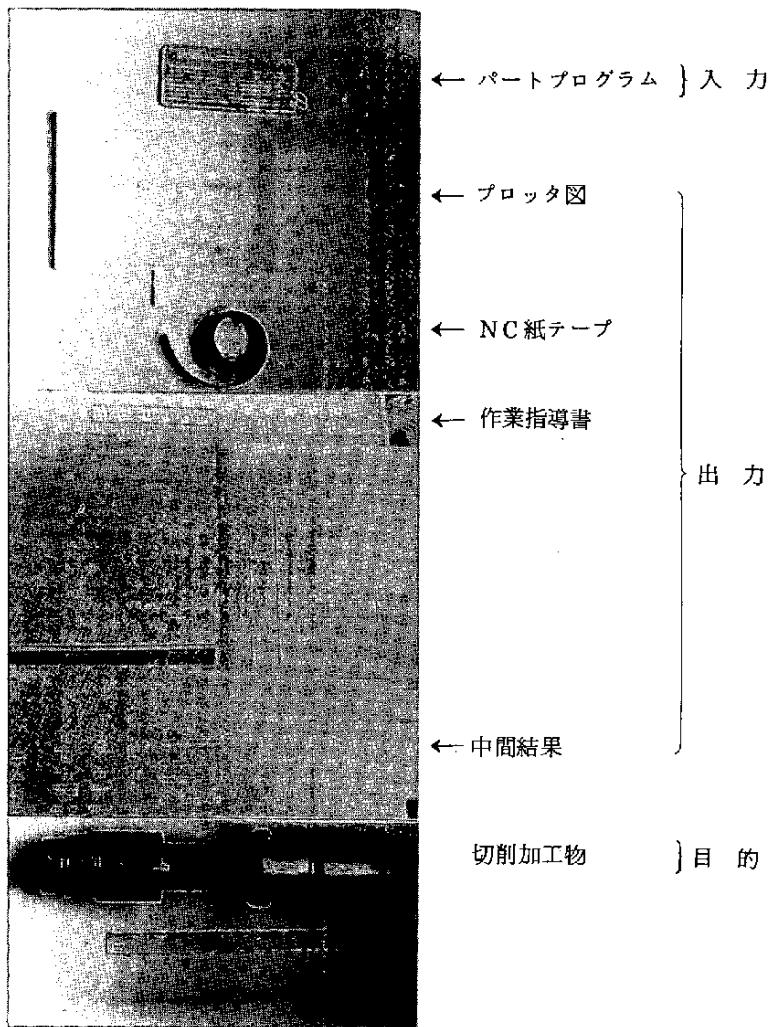
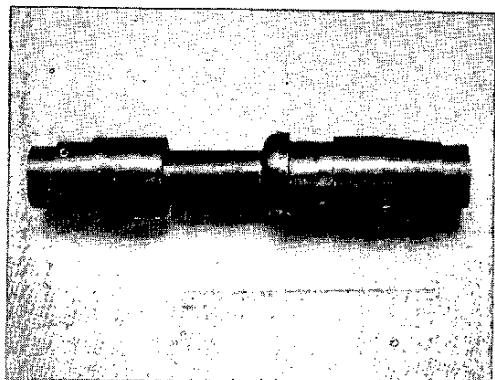


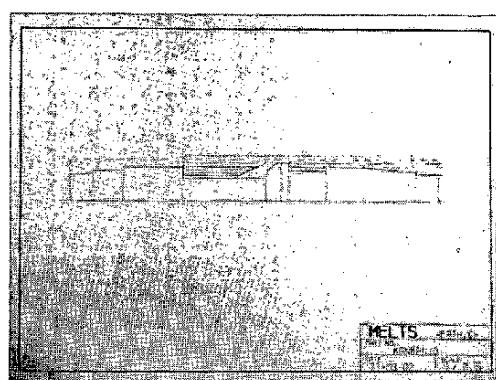
図5-11 MELTS概要



(a) パートプログラムから加工物まで

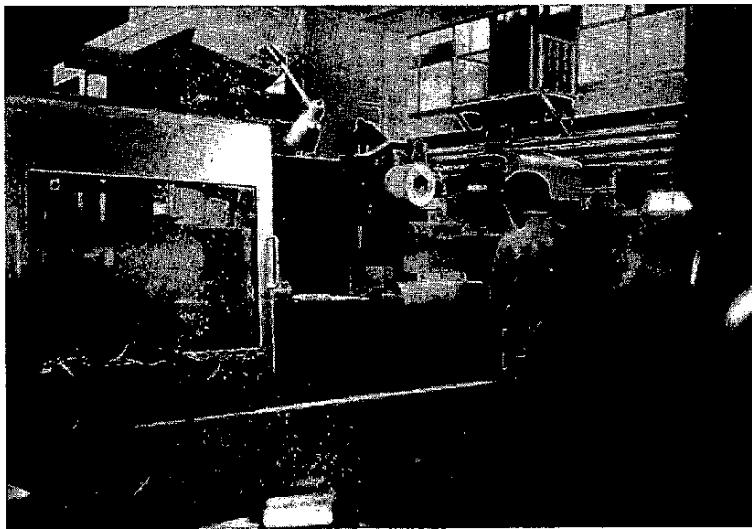


(b) 切削された製品（荒加工のみ）

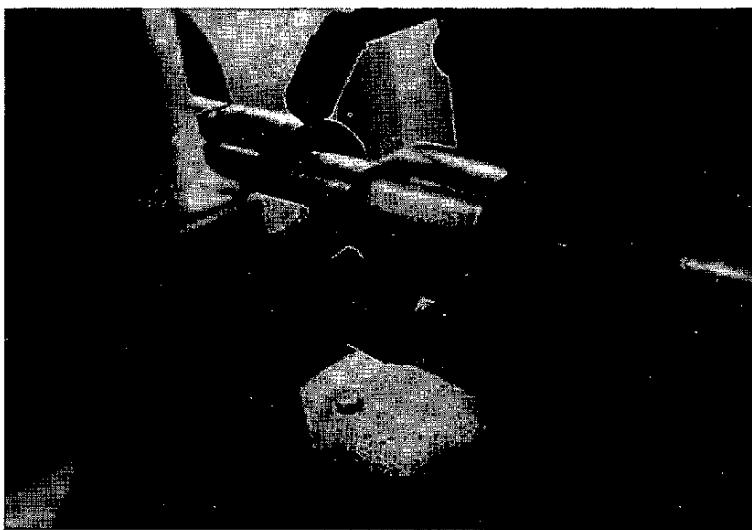


(c) 第1工程の工具径路

図 5-12 MELTS の入力および出力



(a) VNC 30による切削



(b) MELAC-Lによる切削

図5-13 MELTSの出力したNCテープによる加工の様子

5.3.5 インプリメンテーション

(1) 作業経過

MELTSのサブシステムであるセンター作業を対象としたKKTTSが第6回国際工作機械見本市に参考出品されたことは前に述べた通りである。その後このヴァージョン(Fヴァージョンと呼ぶことにする)はチャック作業を含めて処理できるように改良が開始されている。ここではFヴァージョンまでの作業経過を述べる。

図5-14に示したようにトランスレータ70K語、テクノロジー部はデータ領域を含めて116K語、ポストプロセッサ14K語でテクノロジー部とポストプロセッサは105個のサブルーチンから構成されている比較的大きいシステム・プログラムである。これらは延べ10人によってコーディングされた。本研究の中心グループが必ずしもプログラミングを専門としていない研究者であることと、各ステージのアルゴリズムの開発を重視した所から、プログラムの大きさに関してはあまり注意が払われていない。また所属する研究所も異なっているなどのため、共同作業を行いにくく、ステージ間のインターフェースに予想外の食い違いが生じたため、さらに応急用のプログラムが付加されるなど将来修正によってあきらかにサイズを小さくできる所もある。加工技術のノウハウの体系化にあたり、実験およびデータの分析を行ってアルゴリズムを開発し、流れ図からコーディングまでの作業を各研究者(主に4人)が行ったが、プログラミングに関しては必要に応じて専門のプログラマーに依頼できることを望ましいことを痛感した。

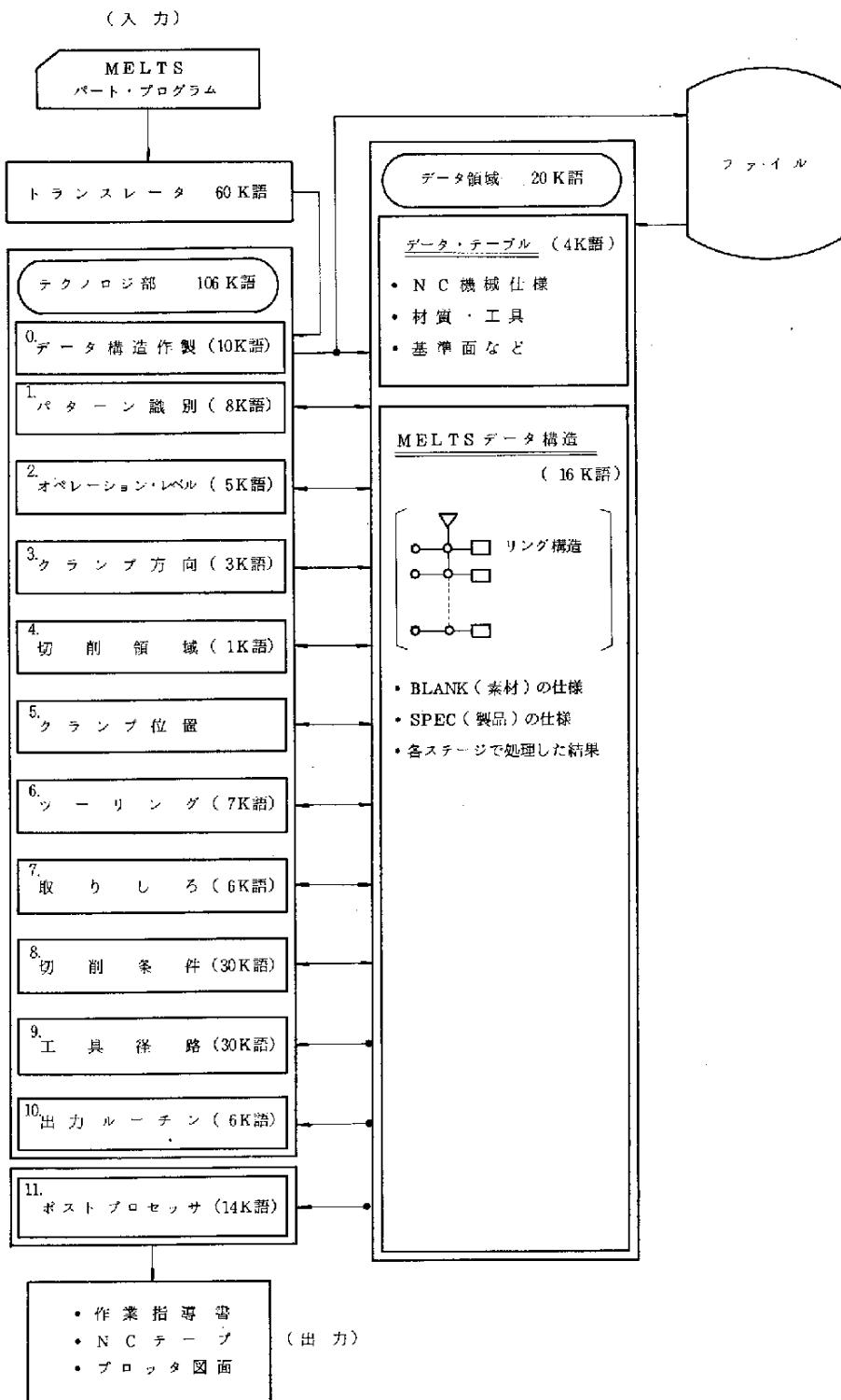


図 5-14 Fヴァージョンのデータ領域とプログラムの大きさ

(2) 実行およびプログラム修正

図5-15に示したようにFヴァージョンでは、プログラム・サイズが大きいためジョブを2つにわけた。JOB1は高速コアメモリ(HCM)において処理を行い、結果はディスクに出力される。テクノロジー部とポストプロセッサをJOB2として、低速コアメモリ(LCM)において処理を行った。LCMのため処理時間は長くなるが、プログラム・サイズの制限上止むを得なかった。このヴァージョンに手を加えないでセグメンテーションを行った場合最大のプログラム・サイズが図5-16に示すようになる。遠からずTS Sで利用可能なHCMの大きさに修正する必要がある。

現在NC機械その他のデータ・ファイルはまだ完備しておらず、データ・カードとなっており、ランの都度入力されている。

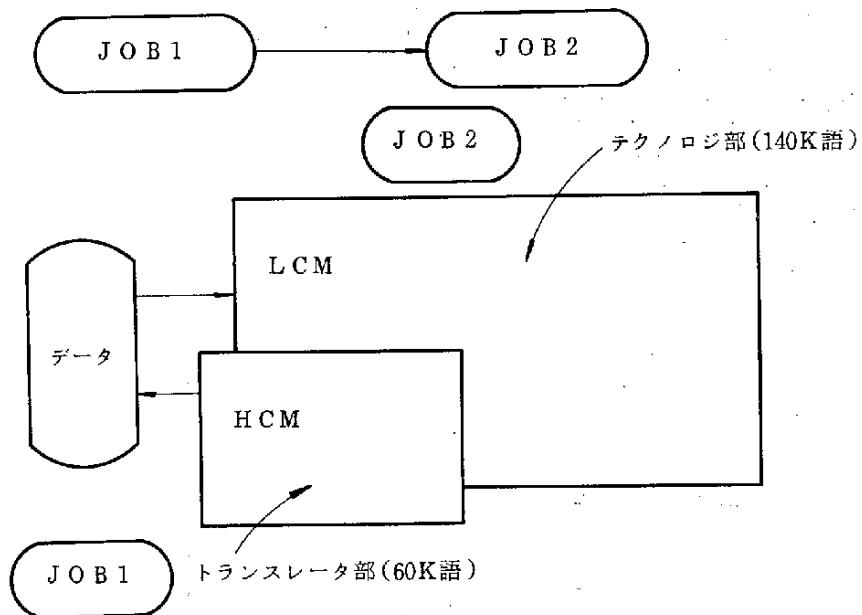


図5-15 Fヴァージョンの実行

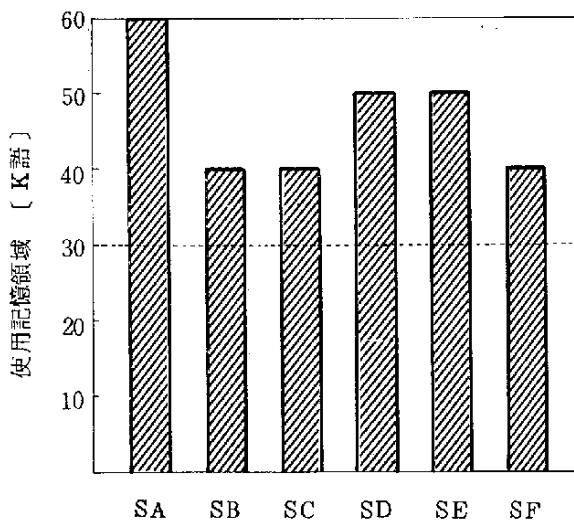


図 5-16 当面可能なセグメンテーションの大きさ

(3) デバッグギング

システム・プログラムは大別して 8 グループに分割してデバッグギングが行われた。リモート・バッチの端末が設置された当初においては、センターの諸条件によりディスクをプライベイトに占有することができなかつたこともあり、図 5-17 の(1)に示すように M T によりアップディットを行つた。グループ毎に M T 上の原始プログラムを修正し、テストランの都度コンパイルを行つてゐた。

ディスクが使用できるようになり、見本市を控えている所から、センターの御協力を得て大巾にディスクを借用して、リロケータブル (R B) 形式で図 5-17(2)に示す方式でデバッグを行つた。ディスクの使い方としては、図 5-18 の(1)に示すように各グループ単位でデバッグを行い、順次連結していく。F ヴァージョンにおいては、トランスレータは実行形式、テクノロジー部とポストプロセッサは実行形式と R B 形式の 2 本立を格納し、完全に結合されて 1 本となつたテクノロジー部とポストプロセッサの R B 形式のファイルは、デバッグ用である。これが今後のヴァージョン・アップにおいて基本的に使用される予定である。トランスレータについては、M E L T S の構想のもとに完全にコーディングしないことになる。なお、現在はとかく L P

への出力量が多く、これが多分にターミナルの使用効率を下げているので、同じジョブでも随時出力量を制御しうる手段を考えている。

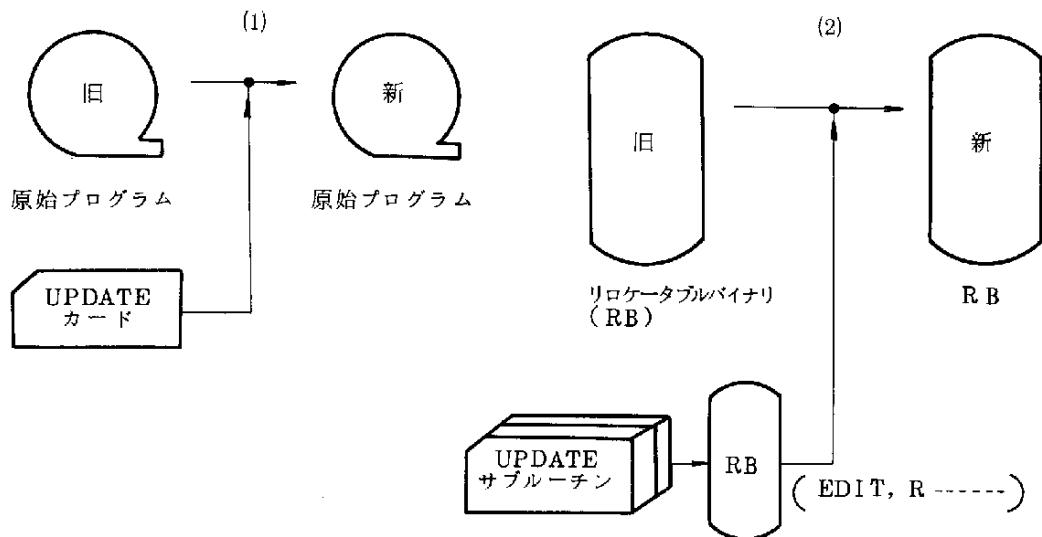
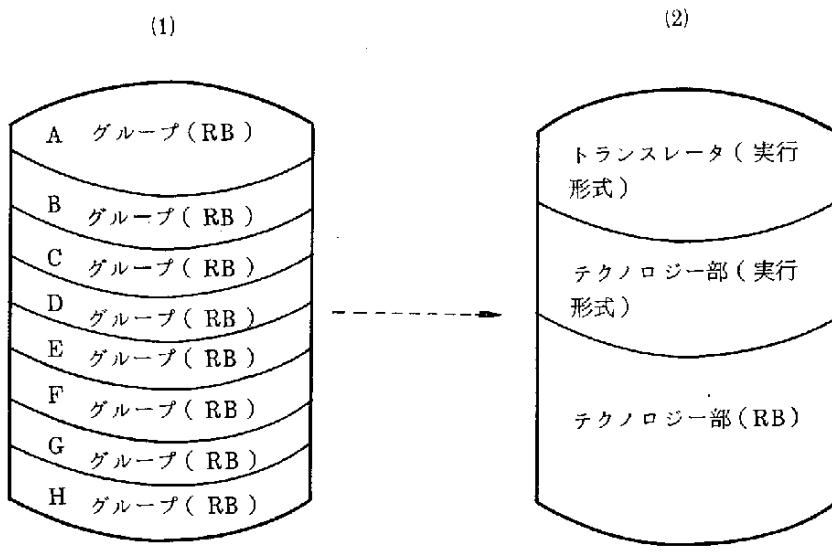


図 5-17 各ステージのデバッグの方法



47. 秋まで

F ヴァージョン

図 5-18 ディスクの使い方

5.3.6 リモート・ターミナルの利用結果

(1) データの背景

機械技術研究所のリモート・ターミナルの利用結果に関するいくつかのデータを、昭和47年8月15日～昭和48年1月31日間の使用実績について示す。この間の利用は全て、リモート・バッチ処理に限られている。

ここに示されるものの原データは、ジョブ件数にして180件余である。実際にはもっと多くのジョブが処理されているが、次の理由で全てのデータを扱かうことが出来なかった。

- a 未だ設置後日が浅いためとくに設置初期においては各種のテストや調査が行なわれ、その際、かなりのタイプライタ・メッセージやLPの出力リストが緊急に所外へ持出されざるを得なかつたこと。
- b 工作機械見本市での参考展示に際して、その移設時期や会期中の繁忙などのため、データの確保が完全に行かなかつたこと、など。

リモート・バッチとして扱かれたジョブの内容を類別すると大よそ次のごとくである。

① MELTSのインプリメンテーションに関して：

- (a) サブルーチンのコンパイル
- (b) サブルーチンのアップデート

これらについては、MTベースの場合と、ディスク・ベースの場合とがあるが、最近は殆んど後者となりつつある。

② MELTSの実行

- (a) 特定ステージの単独のラン
- (b) MELTS全体のラン
 - (i) 結果のチェックのため、出力専用のルーチンを適宜、さし込む場合がある。
 - (ii) センターのプロッタを用いる場合がある。

以下に、

① ターン・アラウンド・タイム（SYSIN終了からSYSOUT開始まで

の時間)

② 出力時間(SYSOUT開始から同終了までの時間)

③ コア・サイズ(K語)

について示す。これらは先に述べた理由により、逐ジョブ毎に対応はつけず、それぞれ単独に集計したものである。

なお処理されたジョブのうち、大きい部類に属するもの、及び小さいものの例を挙げると、表5-5のようになる。

表5-5 扱かれたジョブの例

	C P U 時間	C O R E 時間	C O R E サイズ
大きい部類 に属する ジョブの例	1' 29"	3' 10"	149 KW
	3' 29"	33' 14"	153 KW
	0' 41"	7' 35"	108 KW
	2' 11"	4' 48"	93 KW
小さい部類 に属する ジョブの例	14"	34' 53"	1 KW
	01"	1' 44"	15 KW
	01"	08"	14 KW
	03"	25"	15 KW
	01"	05"	18 KW

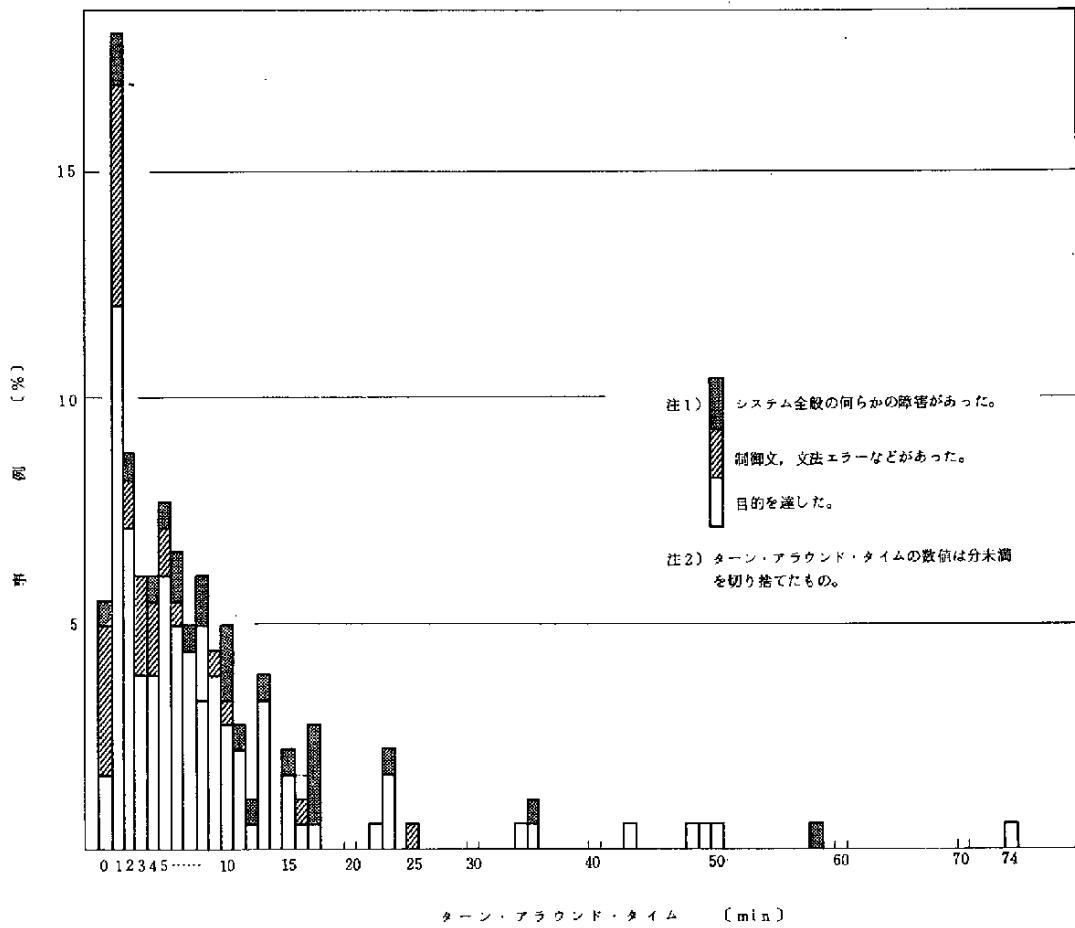
(2) ターン・アラウンド・タイム

図5-19に1ジョブ当たりのターン・アラウンド・タイムの分布を示した。このデータには、ユーザ側が制御文や文法のエラーを犯した場合も含まれている。なお直接本項には関係はないが、出力中にシステムに何らかの障害が発生して完全な結果を手に出来なかった場合も含まれている。

図5-20はこれを累積分布として見たもので、これより50%のジョブが6分以内に出力を始めることがわかる。

次に1日のうちのSYSIN時刻によって、ターン・アラウンド・タイムが違うかどうか見たものが、図5-21である。一般に午前中の方が早いように感じていたが、センターでのクローズドの前日受付分のジョブが処理されるためか、遅いこともしばしばある。むしろ、昼休み直後くらいの方が全般に早いような結果が出ている。

なお、通常のサービス時間以後のデータは見本市のために特にサービスをお願いしたときのもので、一、二の例外を除けば、大きなジョブを流すこと多かったにも拘わらず、殆んど10分以内に返って来たようである。



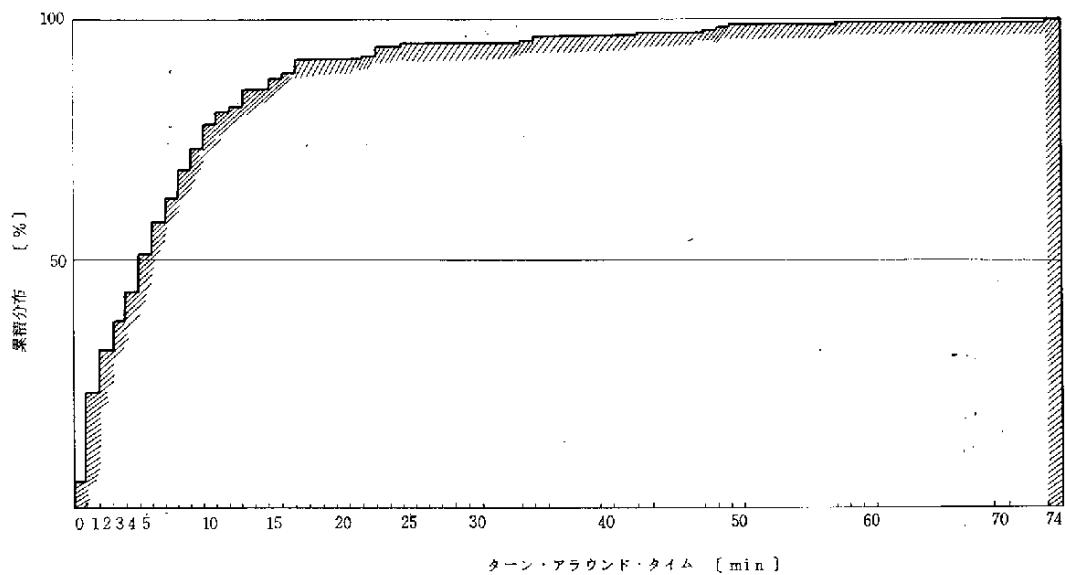


図 5-20 ターン・アラウンド・タイムの累積分布

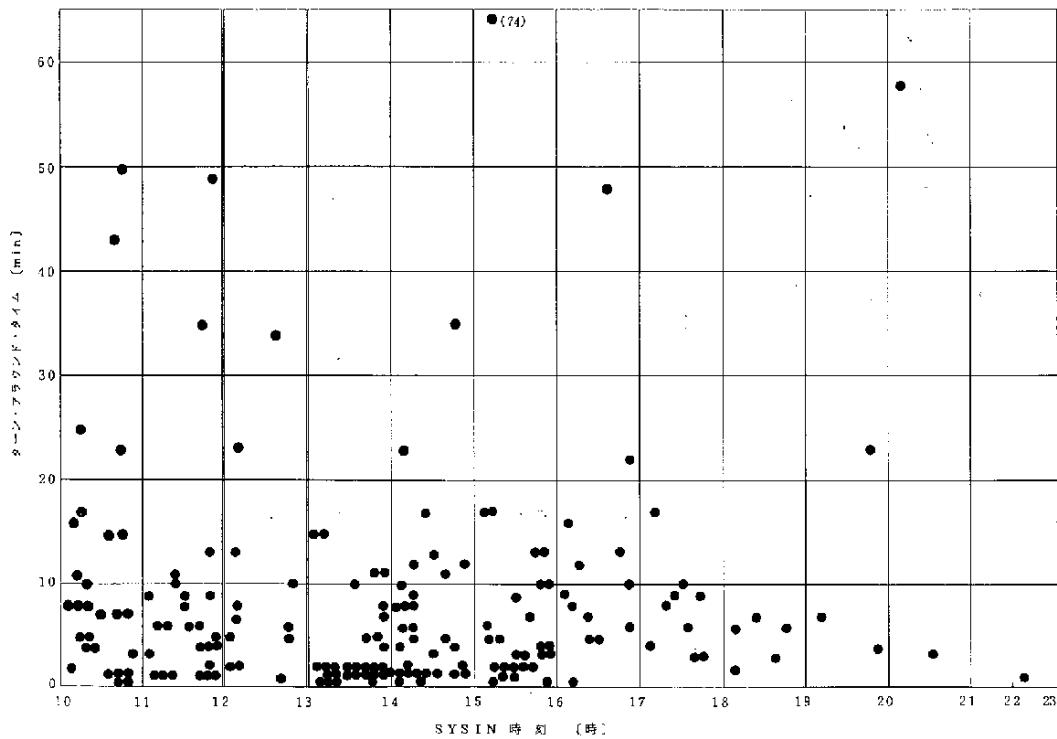


図 5-21 SYSIN時刻とターン・アラウンド・タイムとの関係

(3) 出力時間

図5-22は1ジョブ当たりのターミナルのLPへの出力時間である。MEL TSを実行させるときには、現在はチェック用に多量の出力を行なわせており、多いときでは200ページ程度となる。

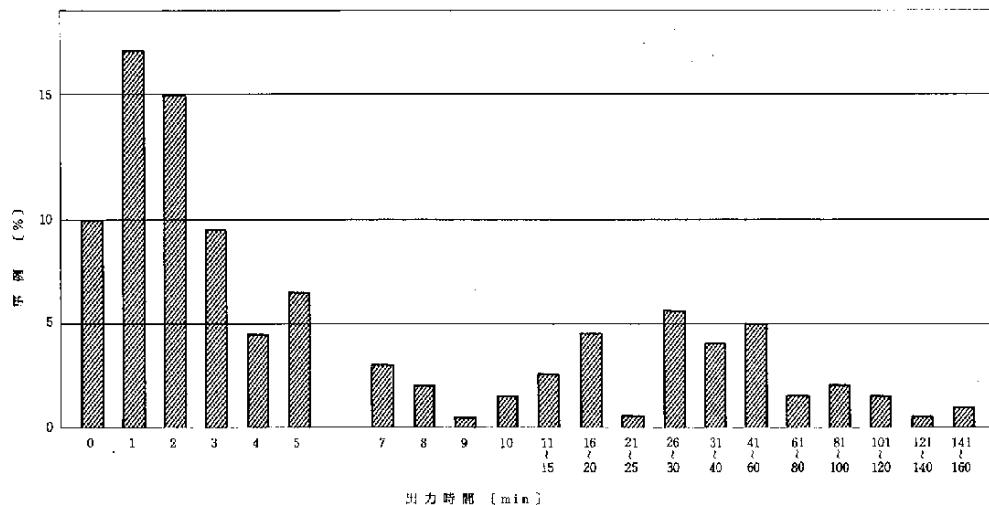


図5-22 リモート・ターミナルにおける出力時間

(4) コア・サイズ

1ジョブ当たりのコア・サイズの分布を図5-23に示した。

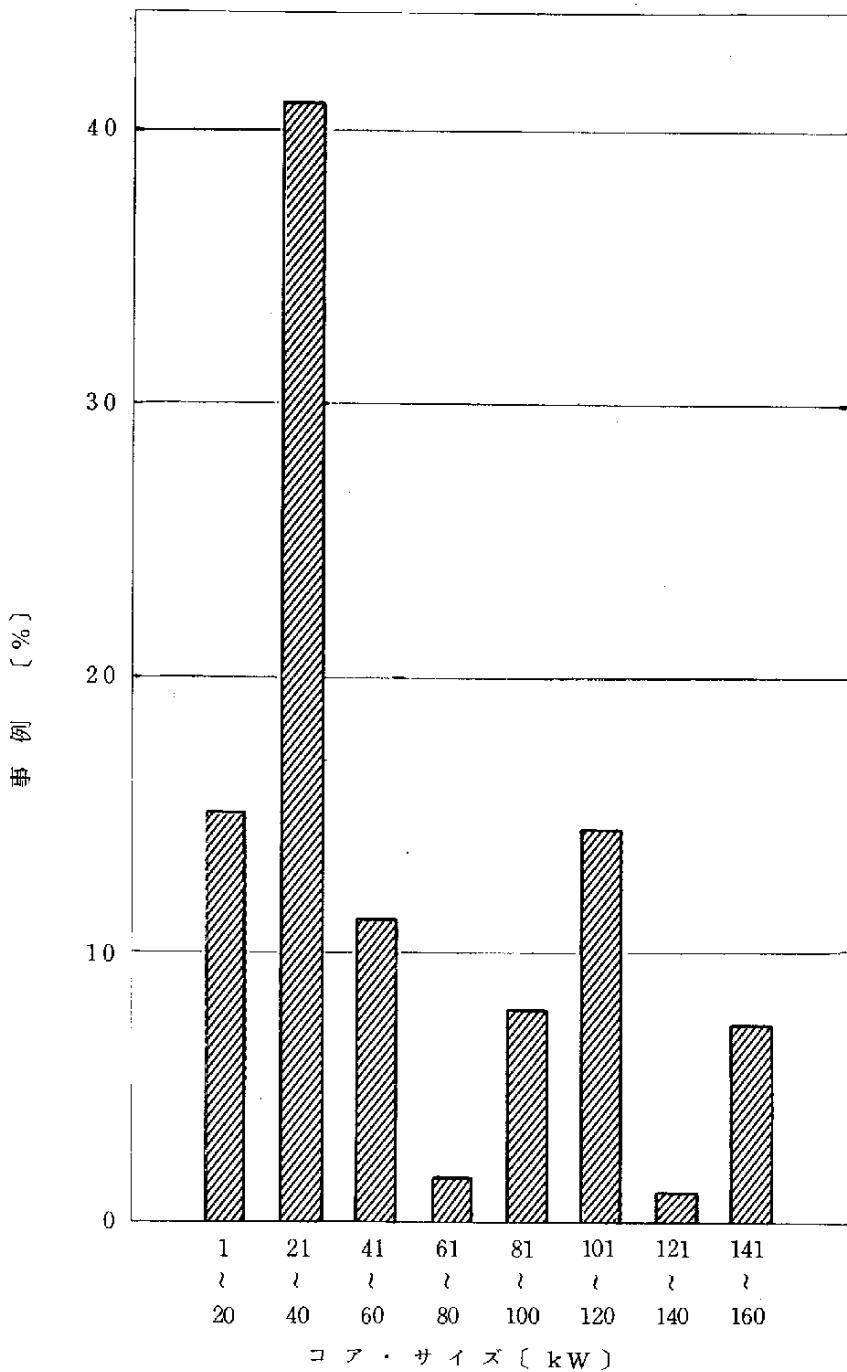


図 5-23 ジョブのコア・サイズ

5.3.7 リモート・バッチ処理方式を利用して

(1) 端末装置上の問題

7月末に当計算機室に搬入されてから、契約通り1週間の調整期間後ただちに単体として使用可能となり、8月15日からFACOM 230-60のリモート・ターミナルとして稼動に入って6ヶ月を経過した。期間を通して気付いた事を次に述べる。

- a ライン・プリンタ：初期において、“LP ERR”がおこったり、必要以上にページ送りをするなどの異常があったが、速やかにCEにより対策が講じられた。
- b カード・リーダ：期間の前半においては、“CR ERR”が起りやすく、ときには全くカードを読まなくなるなどの故障が目立った。現在でも1日のうちで電源投入後30分以内には“CR ERR”がおこる頻度が高いが、ほとんどすぐ正常に復している。
- c 出力ストップ^{*}：ラインプリンタにSYSOUT中に、突如プリントを停止し、コンソール・タイプライタには何もメッセージを出さない上に、タイプライタからのリクエストを受け付けない現象が起った。このとき回線制御プログラムはこわれている。この時センター側には“S ERR”がプリントされる。この原因はセンタ側と富士通の協力によって究明され、今年1月に対策が講じられ、その後この現象はまだ起っていない。ここでは詳細は省くがハード上の問題であった。
- d 紙テープ・パンチャーは、一度もTSSで異常なく稼動している。TSSには無関係だがタイプライタに付属している紙テープ読み取り部と、穿孔部の動作が初期において異常が多くあったが、現在は適切な処置によりスムーズに動いている。

(2) ソフトウェア・サイドの問題

処理件数の増大と見本市に出品するという期限付きの目標の存在により、

*端末側では“ウンスンエラー”と呼んでいた。

僅か半年余ではあるが種々の問題が発生し、かつ対策が立てられた。それらの主な問題は次のようなものである。

- a 端末側に I S O コードで紙テープを出力する場合の的確な手法がみつかるまでに、試行錯誤が繰り返された。これは次のジョブ制御文で指定できる。

例

```
      WRITE(8,900) (BUF(I), I=1, ICOUNT)  
900 FORMAT(100A1)
```

とプログラム中にあった場合

```
¥FD F08, SYSOUT=(REMOTE, 'DEV=PP'), FILE=(  
    TEMP, AAA), UNIT=F471K, VOL=WORK,  
    TMOD=9, RCDSIZE=136, BLKSIZE=136
```

- b 端末からセンタのMTを使用する場合に、file open しないことが時折り起った。これは全く同一条件のもとでオープンしたりしなかったりするという現象であったが、OS上の問題であることが判明し、解決をみた。
- c ラインプリンタのページの頭を揃えるためのセットの位置がなかなか安定しなかったが、現在安定している。
- d (1)の出力ストップの現象は、ソフト上の問題かハード上の問題か区別がつきにくい現象のために、多角的かつ総合的な数多くのテストを必要とした。この経験に照らして、TSSにおける原因不明の問題発生に対しては、センタ側の柔軟かつ統轄的な処置が大切であると考える。
- e カード・リーダに関するエラーが生じたとき、タイプライタからの指示で適切な処置（たとえばそのジョブをキャンセルするなど）がとれるとよい。現在は途中のカードを外して、¥ENDカードを読ませるが不便である。
- f ラインプリンタで出力中に、LPの操作ボタンによってプリントを中断出来ることが望ましい。現状では禁止事項となっている。
- g リモート・ターミナルを用いていると電話も非常によく使うものである

(モデムの電話とは別に)。たとえばセンタ側がダウンしたとき、処理が待たされているとき、など何らかのメッセージがタイプライタに出て来るに有難い。

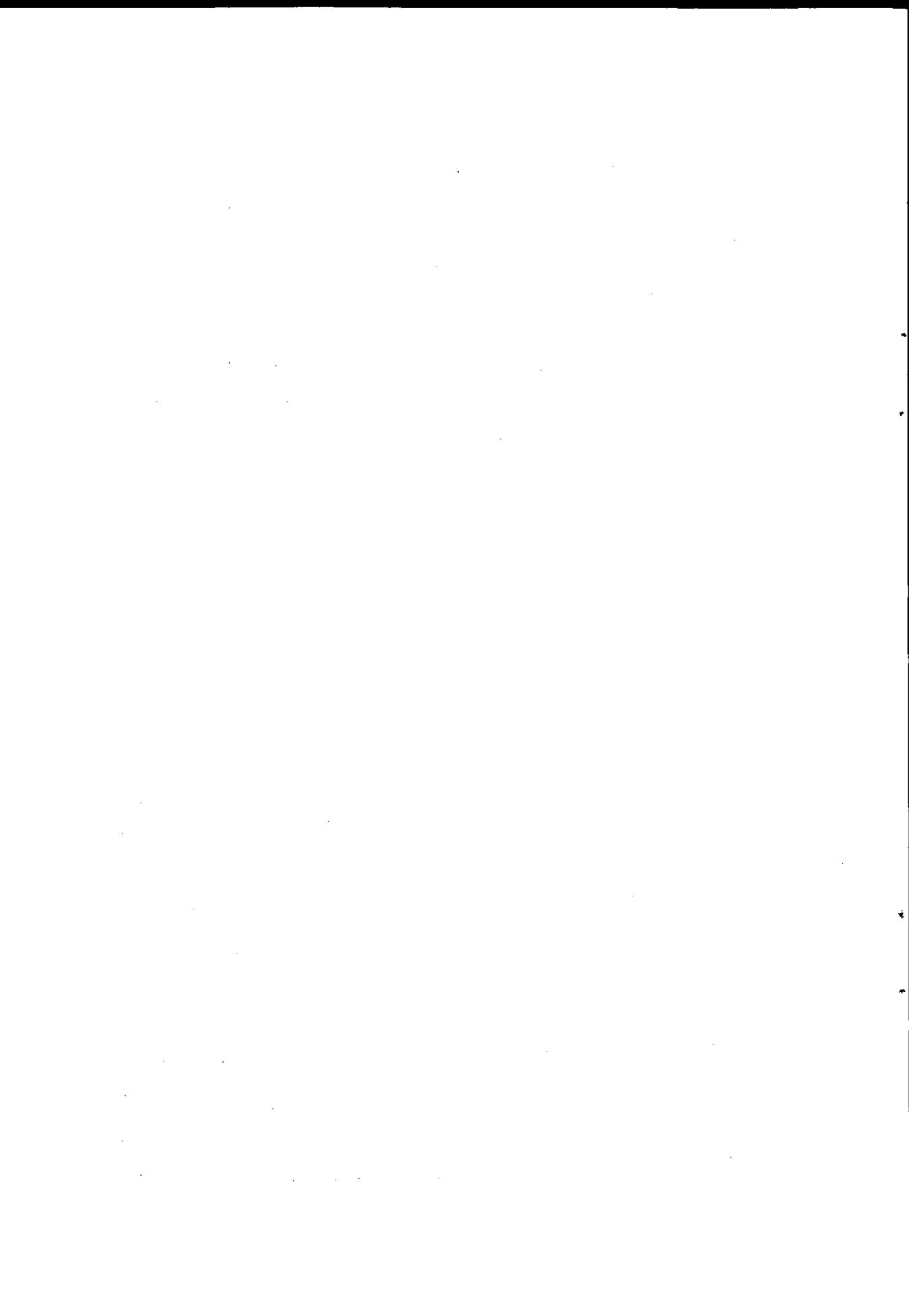
また、タイプライタのメッセージがでるときは、日付けも一緒に出るといい。

- h ついでながら、操作手引書がもっと懇切丁寧であれば有難いと思う。またジョブ・マクロについても幾通りかのメニューがあると良いと思う。
- i 第6節で記したようなデータのもとになる、ターミナルの利用状況に関する記録の確保、保存は重要なことであると常々考えている。しかし、コンピュータのユーザという立場にあるとき、その第1の目的はコンピュータによってしかるべき結果を得ることであり、どうしても目先きの過程は二の次になりがちであるので、相当な努力を払わないと、つい将来にとつて必要かも知れないデータも失なわれがちである。何かそれに関する効率的な上手なやり方があったら知りたいと考えている。

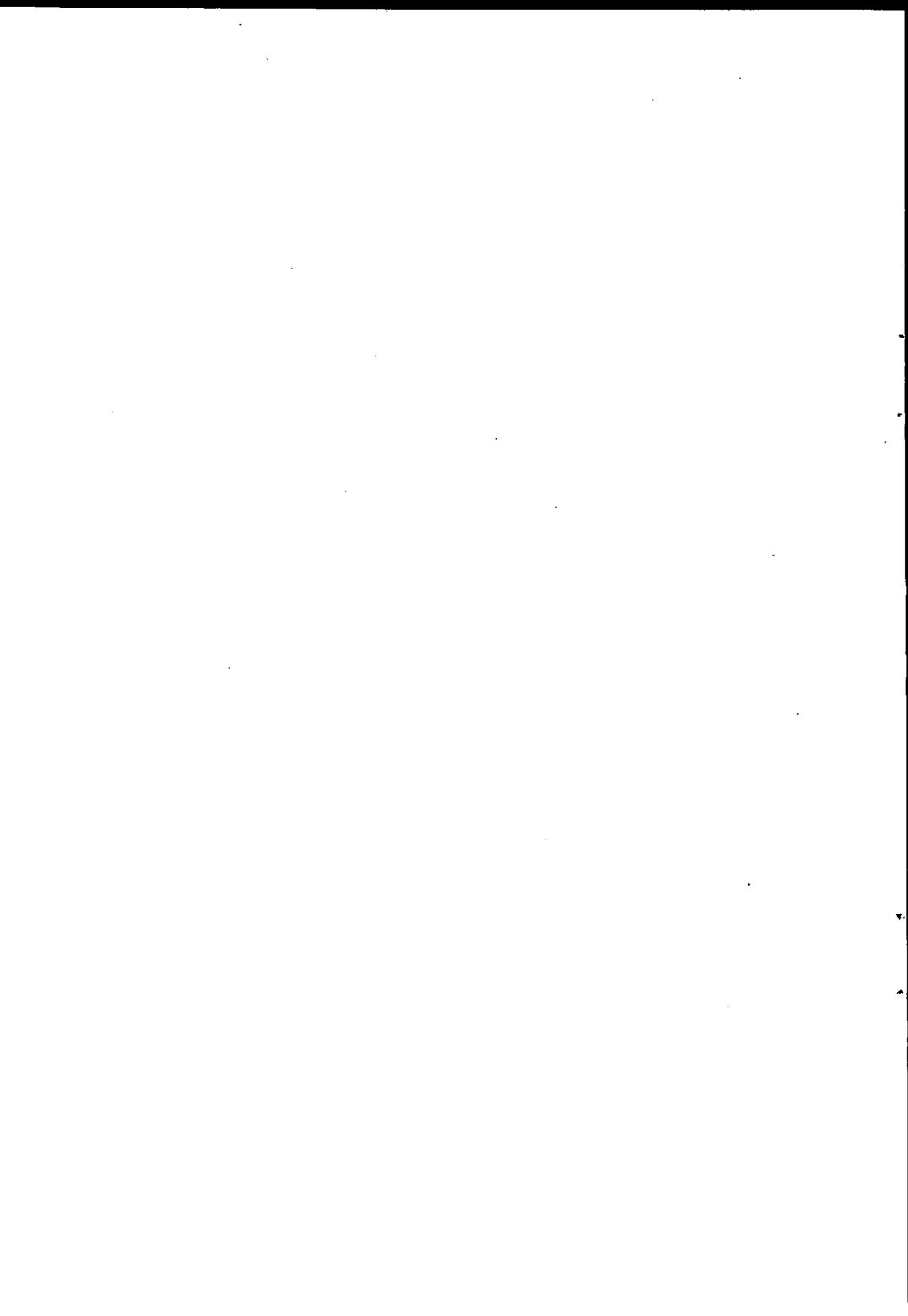
5.3.8 む す ひ

日本情報処理開発センタのFACOM230-60によるTSSリモート・ターミナルは、機械研にその一つを設置した未だ半年少々であるが、旋削加工用自動プログラミングの研究に対して有効に利用されている。その間、半月ほど晴海へ移設して工作機械見本市の参考展示に利用されたりもした。これまで、いくつかの技術的問題も発生したが、その都度解決への努力がなされ、全体として効率的な利用が果たされて来た。

今後もより活用して、機械工業向きの新しいシステムの研究・開発に役立てゆきたいと考えている。また本報告ではターミナルの利用実績に関するデータは、あまり充分なものを示せなかつたが、将来のためにも貴重な資料・情報となると思われる所以その蓄積にも努めたい。なお、当面はリモート・バッチ処理が主体であるが、デマンド処理も隨時、利用してゆきたいと考えている。



6. MONITOR-Vについて



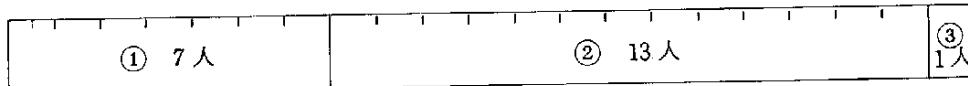
6 MONITOR-Vについて

外部端末に対する TSS サービスの実施にともない、完全クローズ制による MONITOR-V の 3 元処理を約 10 ヶ月運用してきたが、クローズ制に移行する際にはそれ迄オープン使用になれていたため、利用者から完全クローズに対しではかなりの反対もあったが、現在では極あたりまえのような状態になってきたようである。そこで現在のシステム及び運用方法等に対して、センターの使用者がどのような意見をもっているかアンケート調査を行なったのでその中の主なものについて掲載してみると以下のような結果となった。

なお、回答者は主な使用者 21 名である。

① MONITOR-V は卒直にいって使い易いですか。

1. 使い易い 2. 普通 3. 使いにくい



② 障害について、

- ハードウエア

1. 多すぎる 2. 普通 3. 少ない



- ソフトウエア

1. 多すぎる 2. 普通 3. 少ない



③ バッチ・ジョブで急行ジョブ、長時間ジョブ、CORE 使用量の大きいジョブといったようにジョブを区別して処理すべきだと思いますか。

1. YES 2. NO

① 10人	② 11人
-------	-------

④ 現在のような運用状態においてデマンド・ユーザに対してメモリの使用制限をすべきだと思いますか。

1. すべきだ 2. すべきでない 3. 場合による

① 14人	②	③ 4人	無回答 2人
-------	---	------	-----------

⑤ 制限すべきとすればどの位にすべきだと思いますか。

30K以下 2人	35K以下 3人	40K以下 2人	45K 以下 1人	65K以下 2人	無回答 11人
-------------	-------------	-------------	-----------------	-------------	---------

⑥ T S S サービス時間帯とバッチ処理時間帯を明確に分離すべきであると思いますか。

1. YES 2. NO

① 1人	② 19人	無回答
------	-------	-----

⑦ C O R E 容量、応答時間等からいってデマンド・ジョブの同時使用端末台数の制限をすべきだと思いますか(例えば、3端末以上の同時使用をソフト的に禁止するといったこと)。

1. すべきだ 2. すべきでない 3. わからない

① 1人	② 15人	③ 2人	無回答 3人
------	-------	------	--------

⑧ デマンドの優先権について、C O R E 優先権をバッチ処理よりも無条件に下げるべきだと思いますか。

1. YES 2. NO 3. わからない

① 4人	② 9人	③ 8人
------	------	------

⑨ 現在の運用形態は、

1. よい 2. 普通 3. 改善すべきだ

① 3人	② 14人	③ 3人	無回答
------	-------	------	-----

⑩ クローズ制に対しては、

1. 反対
2. やむを得ない
3. 方法を変えるべきだ
4. 今のままでよい

④ 21人

⑪ MONITOR-V の長所は、

この質問に対しては次のような回答があった。

- ・短所として指摘されるような点を除けば全般的によく出来ている。
- ・大体何んでも出来る)。
- ・機能が豊富である。
- ・オーブンシステムである(センタールーチン等センタ側で簡単に機能を追加出来る)。
- ・装置からの独立。
- ・機能の拡張性がある。

⑫ MONITOR-V の短所は、

- ・ジョブの優先権の扱いが原始的である(Multicsのような考え方を導入すべきである)。

この優先権については、スタテック的である、機械的であるといったように表現は異なるが最も多くの人が指摘している。

- ・デマンドジョブにおいても、Fileに対するFDカードをジョブ開始前に、マクロコマンドのパラメータという形で用意しなければならない。
- ・常駐部が大きすぎる。
- ・マクロコマンドの解釈あるいはデマンドジョブのイニシエートが遅すぎる。
- ・ロールイン／ロールアウトの対象がプログラム全体のため時間がかかる。
- ・デマンドジョブの優先権をマクロコマンド毎に指定できない。
- ・実行中のジョブに対して優先権の変更ができない。

- FACOM-R によるリモート・バッチ処理で PL/I や ALGOL のような特殊記号 (↑, ← 等) を含む言語を処理することができない。
- リエントラントの機能が使用できない（制限事項）。

各質問の集計結果をみると、全般的には現システムにはほぼ満足しているようである。予想外であったのが、問10のクローズ制に対する質問であり、完全クローズ制の実施にあたってはいろいろな問題があると予想されたが、実施後7ヶ月（このアンケートは11月に実施した）を経た現在では、回答者全員が今までよいと回答したことであろう。

これは、MONITOR-V が 3 元処理による完全クローズを実施できるオペレーティング・システムであることを証明するもので、全般的に現システムに満足しているという結果につながると思われる。

次に各質問について検討してみると、問1については、使いにくいと答えたのが1名で、他の20名は好意的な回答であった。問2は、障害に対する質問であるが、ハード、ソフト共“多すぎる”と“普通”が約半数を示している。これは、ハードについて言えば件数的には非常に多いと思われるが、その障害が集中的な傾向があるためと、最近は安定してきていていること、クローズ制のため直接マシンダウンに遭遇する機会が少ないとなどが“普通”という形になるのかも知れない。ソフトについては、やはり使用する方法によって異なるようである。すなわち、FORTRANやCOBOLといったコンパイラレベルの使用法であれば、さほど問題にはならない。特に障害が多いのは TSS 関係、データ管理関係であり、このような部分を使用する人は障害が多いと感ずるのである。表4-4(P73)のプログラム別利用状況からもわかるように、センターにおける業務はシステム・プログラムの開発が主であるため、アセンブラー(FASP)の利用頻度が非常に高い。問4に関しては、デマンド・ユーザのメモリ制限をすべきだという人が大部分である。問5では、40K以下が大部分であり制限するとなれば妥当なところであろう。実際問題としては制限するのは非常にむづかしい問題である。例えば、ある制限を越えたら強制的にそのジョブを打切ってしまうのか、警告を発するのか、打切ってしまうのであれば簡単であるが現在

のTSS-FORTRAN等では、ユーザはいったい自分がいまどの位のメモリを使っているのか全くわからないので、強制的に打切ることはできないであろう。問6については、TSSサービス時間帯とバッチ処理時間帯とを明確に分離すべきではないと答えた人が、ほぼ全員の19名である。また意見として分離する必要はないが、TSSだけの時間帯を2時間位設けてほしいというものがあった。しかし、定時間内では現在のバッチ処理件数からいって無理であろう。現在バッチの処理件数（クローズのみ）は、多い時で140～150件、平均的には120件前後である。7時間で120件だとすれば1時間当たり、17件以上ということになり、しかも3元処理ということを考慮すれば、定時間内にTSS専用時間帯を設けることはクローズ時間を延長しない限り無理なようである。問7のデマンド・ジョブの制限については、例えば九州大学では、メモリの関係から同時に使用できる端末の数を制限しているのでセンターの職員はどう反応するか質問してみたのであるが、大部分の人（15名）がすべきでないと答えている。問8は、CORE占有優先権を下げるとき、デマンドの応答が遅くなり、上げるとバッチが遅くなるといった問題があり、TSS利用者とバッチ利用者の意見のくい違うところである。ただ、デマンドが遅くなるということは、外部使用者のようにセンターの事情（ジョブの込み具合、バッチとデマンドとの関係等）を知らない人にとっては、使いものにならないといった印象を与えることになる。その点、バッチが多少遅れるのは、ジョブを依頼した人が直接機械に接していないのでデマンドほど気にならないであろうが、ジョブの依頼件数が多くればやはりターンアラウンド、スループット等の低下をまねき、前述のメモリの使用量とともに検討を要するところである。問9の現在の運用形態はの質問に対しては、17名が“よい”または“普通”であり、3名が改善すべきだと回答している。改善すべきだという意見には、メモリの使用制限を実施すべきであるという意見がその主な理由とされている。問10は、前述のとおり、全員が今までよいと答えている。問11、問12のMONITOR-Vの長所、短所にあげられたものは、いずれもうなづけるものであり、なかには長所が短所となり、短所が逆に長所にもなっているというものもあるようである。

短所の中で、最も問題とされるのは、何度もくり返えすようであるが、優先権の問題と、ロールイン／ロールアウトの問題である。

これは、ロールイン／ロールアウトが起らない程の C O R E 容量があれば問題はないのであるが、バッチの多重度 2 (リモート・バッチも含む) + デマンドの 3 元処理を行ない、しかも現在の制御プログラムの大きさを考えれば、それだけの C O R E 容量を持てるのは例外的であるといってよいであろう。もちろん、運用にあたってはその時の状態に応じてバッチの多重度を 1 ~ 4 位に変化させているが、多重度が 1 であると非常に能率が悪いようである。

実際の運用中には、当然各ジョブの C O R E 使用量等の関係があり、必ずそうなるというものでもないが、次のような現象がたびたび起り、その現象も理屈上は当然であるが、実際の利用面あるいはデマンド利用者の立場にたてば、もう少しどうにかならないものであろうかという声が聞かれた。

デマンド 2 つ、バッチ 2 つ位のジョブを実行中、C O R E 占有優先権がデマンドよりバッチが高い場合、デマンドが頻繁にロールイン／ロールアウトされるため (ロールイン／ロールアウトされることは当然であるがロールイン／ロールアウトの動作に時間がかかる)，端末利用者は 1 行入力あるいは 1 行出力するたびに数秒以上待たされる結果になる。この待たされるという現象は、何回かに 1 度という割合であれば、その待たされる時間が比較的長くてもさほど気にならないが、毎回であると利用者はデマンドを利用する気を失なう結果になる。これは、ロールイン／ロールアウトの時間を速くするか (ロールイン／ロールアウトの方法及びロールイン／ロールアウト用のファイルを高速化することが考えられる。後者については Edition-70 系から高速磁気ドラムが使用可能になる)，C O R E 占有優先権の高いバッチの方もロールアウトするようにすべきであろう。もちろん、現在でもデマンド・ジョブの数やその C O R E 使用量によっては、バッチ・ジョブもロールアウトされるが、そのような場合はこんどは逆に一旦ロールアウトされたバッチ・ジョブは、なかなかロールインされないといった状態になってしまことがある。

また、バッチ・ジョブの C O R E 占有優先権を下げて、デマンド・ジョブの

それと同じにした場合、デマンド・ジョブとバッチ・ジョブが同時に C O R E に入りきらない場合には、バッチ・ジョブがロールアウトされたままでデマンド・ジョブが終るまで何時間たってもロールインされないといった現象になる。

次のデータは、バッチの C O R E 占有優先権を変えて実験したものである。

実験結果—I, IIはそれぞれ次のような条件で実行させたものである。

ジョブ名	ジョブ種類	ジョブ優先権	実行優先権	CORE占有優先権	CORE使用量
DJIE016	デマンド	7	15	3	62K語
I REMOTE					
BBBBB					
DJIE016	リモート・バッチ	6	6	6	52K語
II REMOTE					
BBBBB					
DJIE016	ローカル・バッチ	5	5	5	52K語
II REMOTE					
BBBBB					

```

A 031045 J33 * 33353      STACK 19:04:23
A 03070 J33 * 33353      KAISI 19:04:27
A 100002 & 0220 NOT READY
A MAT C.002 SL 30333 C00002
RD MOIS ROCK C.002
33326 RLCUT-1 190452135 190452146
DJIE016 RLCUT-0 190452131 190452130
E27 JA
ZIKKO TYUU
RCU OUT TYUU DJIE016,REMOTE,33353
DJIE016 RLCUT-0 190604000 190605792
DJIE016 RLIN-0 190605645 190607512
DJIE016 RLIN-0 190607515 190609191
DJIE016 RLCUT-0 190609237 190610951
DJIE016 RLIN-0 190610964 190612690
DJIE016 RLCUT-0 190612728 190614405
DJIE016 RLIN-0 190614408 190615039
DJIE016 RLIN-0 190616129 190617808
DJIE016 RLIN-0 190617812 190619468
DJIE016 RLCUT-0 190619536 190621214
DJIE016 RLIN-0 190621217 190622934
DJIE016 RLCUT-0 190622972 190624701
E2 KILL REMOTE
DJIE016 RLIN-0 190624704 190626580
DJIE016 RLCUT-0 190626425 190628106
DJIE016 RLIN-0 190628109 190629785
DJIE016 RLCUT-0 190629831 190631557
DJIE016 RLIN-0 190631551 190633248
DJIE016 RLCUT-0 190633297 190634997
DJIE016 RLIN-0 190635001 190635676

```

実験結果—I

AC ADIS TICK C.001
 A1 7 31
 ZIKKO TYUU DJIE016, BBBB, REMOTE
 ROLL OUT TYUU REMOTE
 BB BBB
 BBBB RLIN-0 191059952 191101727
 DJIE016 RLIN-0 191101702 191103730
 DJIE016 RLIN-0 191103003 191106613
 REMOTE RLIN-0 191105022 191107007
 REMOTE RLIN-0 191107000 191108105
 DJIE016 RLIN-0 191108109 191110095
 DJIE015 RLIN-0 191110161 191112203
 BBBB RLIN-0 191111453 191113273
 BBBB RLIN-0 191113273 191115055
 DJIE015 RLIN-0 191115063 191117115
 REMOTE RLIN-0 191118537 191120337
 DJIE016 RLIN-0 191117315 191120532
 REMOTE RLIN-0 191120551 191121393
 DJIE016 RLIN-0 191121702 191123754
 DJIE016 RLIN-0 191123953 191125094
 BBBB RLIN-0 191125223 191127013
 BBBB RLIN-0 191127015 191128305
 DJIE015 RLIN-0 191123300 191130822
 DJIE016 RLIN-0 191130986 191134000
 REMOTE RLIN-0 191132203 191134175
 REMOTE RLIN-0 191134179 191135294
 DJIE016 RLIN-0 191135297 191137255
 DJIE016 RLIN-0 191137402 191139433
 BBBB RLIN-0 191138592 191140528
 BBBB RLIN-0 191140531 191142349
 DJIE016 RLIN-0 191142353 191144413
 REMOTE RLIN-0 191145298 191147773
 DJIE016 RLIN-0 191144623 191147093
 REMOTE RLIN-0 191157349 191158423
 DJIE016 RLIN-0 191158432 191200484
 DJIE016 RLIN-0 191201315 191203293
 BBBB RLIN-0 191202553 191204323
 BBBB RLIN-0 191214002 191215777
 DJIE015 RLIN-0 191215730 191217768
 REMOTE RLIN-0 191219750 191221498

実験結果-II

実験結果-Iをみると、CORE占有優先権の最も高いREMOTEがCORE内に居すわり、デマンド・ジョブのDJIE016だけがロールイン／ロールアウトされ、ローカル・バッチのBBBはロールアウトされたままで全くロールインされない。DJIE016だけが頻繁にロールイン／ロールアウトされるのは、CORE占有優先権から言えば当然であるがあまり頻繁に行なわれるため、前述のように端末利用者は1行入力あるいは1行出力のたびに待たされる結果となる。また、ローカル・バッチが全くロールインされないととも問

題ではなかろうか。このような場合、ある一定時間以上ロールアウトされていれば優先権を変更して、強制的にロールインするような方法がとれないものであろうか。

実験結果-IIは、CORE占有優先権をデマンド、リモート・バッチ、ローカル・バッチ全て同じにした場合のものである。これをみると、3種類のジョブが均等にロールイン／ロールアウトされており、ある面では非常に公平である。ただ、この場合デマンドが1つであるためこういう結果になったと思われ、デマンドが2～3個入った場合にはやはりバッチ・ジョブが全く動かないということになるかもしれない。

それゆえ、実効優先権をこれまでのCPU使用時間、CORE占有時間、CORE使用量、ロールアウトされていた時間、同時に動いている他のジョブ及びジョブの種類等を考慮して、もっと動的に変化させてほしいものである。

しかし、これは簡単なことではないと思われる所以、せめて補助的な手段として次のようなことは出来るようにしてほしいものである。

- ① ある一定時間以上ロールアウトされていたものは強制的にロールインする。
- ② 操作員が実行中のジョブの優先権の変更が出来ること。例えば、強制的にロールアウトさせることなども含めて。
- ③ 操作員が指定したジョブの実行を一時中断すること。

②、③については、次のような運用上の経験からである。あるバッチ・ジョブ(CORE使用量の相当大きなジョブであったが)を、実行中にデマンド・ジョブが1～2個入ってそれが交互にロールイン／ロールアウトを起す状態になった場合、そのままにしておくとロールイン／ロールアウトのために時間をとられ、バッチ・ジョブが終了するまでには20～30分位かかった。このジョブは、単独で実行させれば20～30秒で終了するものである。このような場合デマンド・ジョブを20～30秒止めておけば全体的には効率が良くなるであろう。デマンド・ジョブを一時止めるということは理屈上は暴論かも知れないが、本応用実験におけるような運用形態での運用経験からいうと前述のようにデマンド・ユーザにとってはたまに相当時間待たされても、毎回数秒以上ず

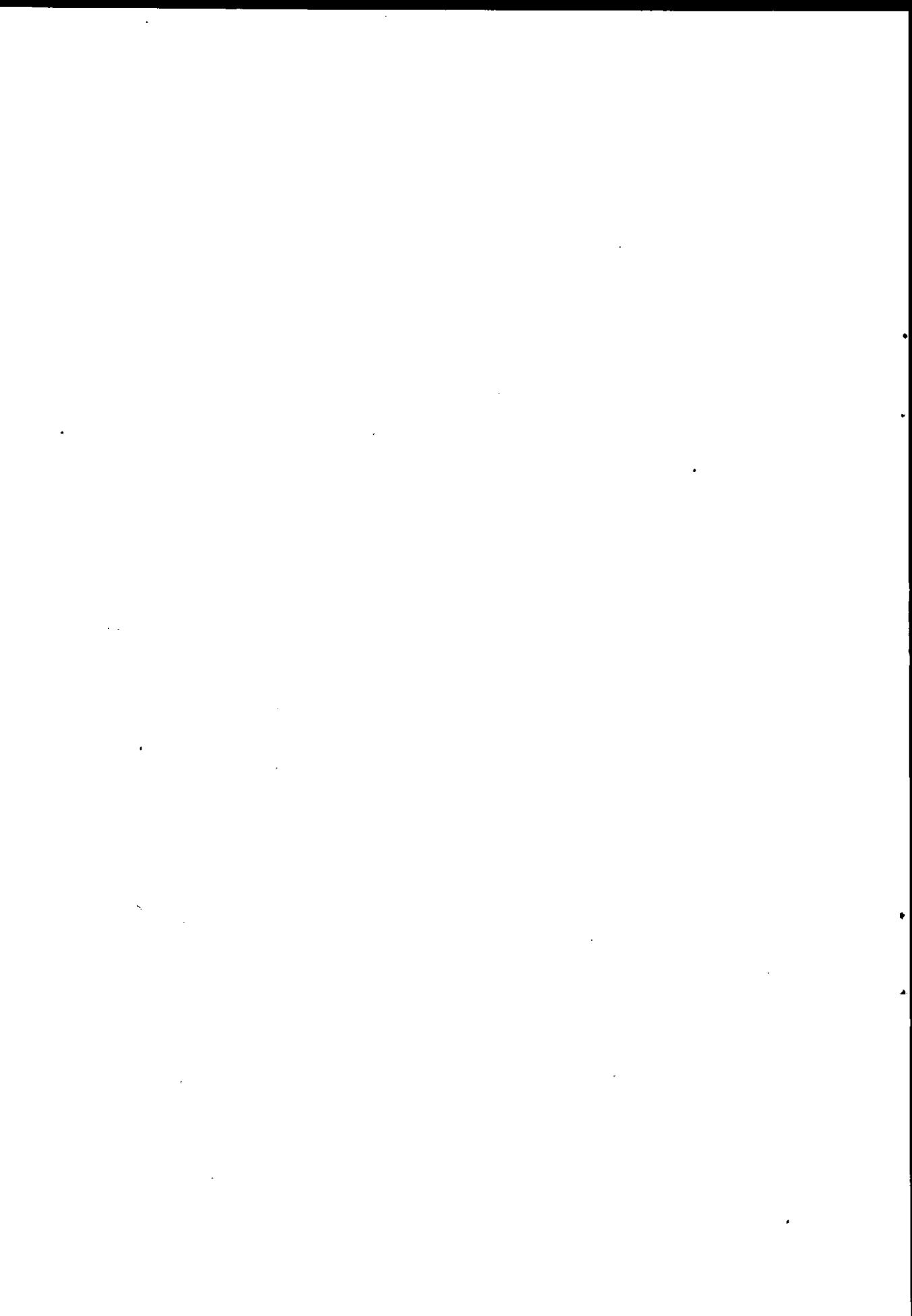
つ待たされるよりは心理的に楽である。

ただこのようなことは運用上の問題であり、希望であるが、C O R E 使用量の制限、多重度、C P U 使用時間等の規制及び時間帯の分離等である程度さけるられるであろうし、すべてを制御プログラムに押しつけるのは無理であろう。ここで述べたような点はこのような制限に関しては非常にゆるい状態についてあることを付記しておきたい。

また、もう一つの問題として、あげられたものにリモート・バッチによる、PL/I, ALGOL等の使用が出来ないというものがある。

利用出来ない理由は、それぞれの言語のなかにある特殊記号(「~, !等)が端末側から送れないためであるが、PL/I, ALGOL共コンパイラ言語であり、PL/I等はこれからは更に利用が期待されるものだけにせひリモート・バッチでの利用が可能になるよう望まれる。

7. 調 査



7 調査

本年度は下記の2項目について調査を行なった。

調査Ⅰ TSSユーザ調査。

調査Ⅱ 地方公共団体におけるTSSサービス利用の可能性についての調査。

7.1 調査 I

7.1.1 調査目的

本年度のタイム・シェアリング・システムの調査としては、昨年度に行なったセンタ側の現状調査とは異なった観点から、TSSに対する意見をとらえるため端局側の調査を行なった。

端局側がTSSに対して、どのような考え方を持ち、センタ側に対してどのような希望を持っているかなどを把握し、それを今後のタイム・シェアリング・システムおよび効率的運用へ反映させることを目的とした。

なお、調査内容は次のとおりである。

(1) 調査内容

イ. 端末装置導入の目的とその効果

ロ. 安定性

ハ. ソフトウェア

ニ. 端末装置の操作性

ホ. センタ側に対する要望

ヘ. その他

(2) 調査期間

イ. 期間

昭和47年11月27日～12月1日

(3) 対象先の端末装置とセントラルマシンの種類

項目 対象先	端末機の種類	伝送速度 (Bit/S)	セントラルマシン
A 大 学	A 1端局 FACOM-R (主にリモート・バッチ用に使用。ミニコンピュータ。)	1,200	FACOM230-60
	A 2端局 F-1591A (主にデマンド用に使用)	200	
	A 3端局 F-1592A (")	50	
B 大 学	B 1端局 F-1592A (主にデマンド用に使用)	50	FACOM230-60
	B 2端局 F-1592A (")	50	

7.1.2 端末装置導入の目的とその効果

今回の調査は大学関係の端局を対象としたが、これらの大学には日本学術会議の勧告によって、共同利用のための大型計算機が設置されており、各地に散在する大学の研究者のために TSS サービスが試験的に開始され、このサービス形態が当を得たものであるとして普及され現在に至っているわけである。

今回の調査対象となった各端局も、この趣旨にそって端末を導入し、①ターンアラウンド・タイムを短縮させること、②必要な時に必要なだけ大型計算機を使用することにより、デバッグ効率の向上を図ること、③遠隔地とセンタ間のカード、プリンタ用紙などの持ち運び労力および時間のロスによる非能率的な研究の改善を図ること、④TSS とバッチ・ジョブとの併用によるスループットの向上などをその導入目的として上げていた。

一方、その効果としては、各端局によって利用条件が違うため一概には言えず、各端局によって多少差異がある。しかし、共通していることは後述するように、センタ側の諸制限事項、安定度、ソフトウェアなどに不満な点をいくつか持ちながらも、上記の目的に対する効果は上っているとの意見を持っていた。

特に、ある端局では自分の所に小型コンピュータを設置するとしても、現在の研究室予算ではなかなか簡単にできないが、現在の TSS サービスを利用することになってかなりのジョブを消化し、研究遂行のためにかなりの成果があがっているという所があった。

また、センターにバッチ・ジョブの処理を依頼した場合、急行ジョブ (CPU タイム 1 分) で半日から 1 日、普通ジョブ (CPU タイム 5 分) で 1 日から 2 日のターンアラウンド・タイムを必要とし、繁忙期ではさらに延びることもあるが、リモート・バッチ処理では、ジョブを投入してから結果が打ち出されるまでに、20~40 分程度 (そのときのバックグラウンドのジョブの処理状況によって、これより早い場合も遅い場合もある) ですむため、このターンアラウンドの短さが一番魅力であるという感想を持っていた。

7.1.3 安定性

TSSの基本的な条件としてのハードウェア、ソフトウェア全般にわたる安定性について意見を聞いたが、要約すると次のような点が指摘された。

- (1) センタ側のハード・ソフトのシステム・ダウンが多発した時期があった。
- (2) リモート・バッチ用端末 FACOM-R で、カードリーダの不調が時々あった。
- (3) ファイルが読めない時があった。

端局側のセンタダウン（ハード・ソフトによるシステム・ダウン）に対する反応は予想以上に大きく、ある端局では、「初めの半年間は、センタ側の管理プログラムであるモニタの虫によるシステム・ダウンが多く、能率低下がはなはだしかった。この原因は、制御プログラムがローカル・バッチ処理と TSS を同時に制御するために、きわめて規模が大きいこと、十分なテストを経ないうちに使用されることおよびハードウェアの障害がときどき起ることなどによる。計算機と直接コンタクトしているという一種の優越感を持っている利用者は、システム・ダウンに何回も遭遇すると絶望感にとらわれてしまう。しかし、最近ではモニタおよびハードウェアの安定度が向上したため、研究活動に有効に利用されるようになっている」という意見を述べていた。

やはり、当財団で経験しているように TSS サービスの初期時、あるいはモニタのアップデート、ハードウェアの増設時に発生する初期障害の多発に遭遇しているようである。

端末装置の障害頻度についてはあまり苦情がなかったが、FACOM-R をリモート・バッチ用端末として使用している端局で、カードリーダのリードミスなどによるロスが目立ち、他には C P U のパリティエラーが初期時に多く発生したようである。カードリーダのリードミスについては、カードリーダの読み取り機構のチェック機能に欠点があるようと思われる。端末装置としての経済性からすると、このチェック機能を充実させることによって、端末装置のコストが高くなるのではあまりかんばしくない。かといって、読み込まれたジョブが、リードミスによるコンパイルエラーとして端末に出力されたのでは、それまで

に費やされた時間のロスなどを考慮するとこれまた非常にかんばしくない。この経済性と機能の充実度をどのように釣り合わせるかは、非常に難しい問題であるが、端末利用者としてはよりよい機能を備えた安価な端末装置を要求したいところであろう。

7.1.4 ソフトウェア

使用言語、コマンド、応答時間と処理時間などについて調査を行なった。

(1) 使用言語

リモート・バッチ処理を対象とする端局においては、普通のローカル・バッチ処理と同じ言語が使えるので問題はないとしており、特に利用度の多い言語としては、取り扱う業務の関係で FORTRAN や ALGOL が使用されていた。

デマンド処理を対象とする端局では、BACCUS が使用されていた。BACCUS を使用しているのは、これらの端局が会話型言語として BACCUS しかサービスを受けておらず、あとはファイルの作成、追加、変更を会話形式で行なう LINED プログラムがある程度であるからであろう。

この BACCUS のねらいは決して悪くはないのだが、従来 FORTRAN を使用してきたユーザの FORTRAN に対する要求は根強いものがあり、今回の調査先でもほとんどがこのことを指摘していた。ある端局の意見を借りると、「デマンド処理用言語としてもっと使い易いものが開発される必要がある。たとえば、会話型 FORTRAN ぐらいは備えられていることが望まれる。デマンド処理があまり利用されないのは、言語の使いにくさばかりでなく、新しい言語を覚えて積極的に使いこなそうとする利用者が少ないともよろう。それだけ FORTRAN は、『母国語』的存在になっているともいえるのではないか」といっている。

また、ある端局では BASIC が欲しいという所もあった。これもやはり、今まで使用してきた言語に対する要求の強さが感じられた。

いずれにしろ会話型 FORTRAN ぐらいはユーザとしては欲しいところであ

ろう。

(2) コマンド

現在のモニタに追加して欲しいコマンドについて調査したが、次のような意見が聞かれた。

- イ) リモート・バッチ処理で使用できる“KILL”コマンドが欲しい。
- ロ) リモート・バッチ処理、デマンド処理で使用できる“DISPLAY”コマンドが欲しい。

上記のイ)については、リモート・バッチ処理でジョブを処理中、何らかの理由でそのジョブを終らせてむだな計算機の使用をさけ、また、使用料金のむだ使いをさけるために、TSSにはぜひ必要なコマンドであると思われるが現在その機能がない。しかし、現在のMONITOR-VはEdition-60系列であるが、これよりもバージョンの高いEdition-70系列では、これが実現されるようである。

ロ)については、端末からインプットしたジョブの処理状態、すなわち、実行中であるのか実行待であるのかなどの表示を求めるコマンドで、これもTSSには欠かせないコマンドであると思われる。このコマンドについては、最近のMONITOR-V Edition-60系で実現されている。

(3) ライブラリ

数年前までのTSSのメリットとしては、単にコンピュータのリソースを比較的自由に安く利用することにあったが、最近のTSSに対するユーザの要求はこういった状態から一步進んで、どのようなライブラリが使えるかというアプリケーションサポートへと目的が変ってきたといわれている。

このような意味で、今回の調査でもライブラリに対する調査を行なったが、具体的な意見は聞かれなかった。これは、これらの端局のTSSサービスには科学技術計算、統計計算などの基本的なライブラリがある程度用意されているためか、あるいは、これらの端局が新たなライブラリを要求するほど高度で広範な業務をTSSで処理するまでに至っていないのか断定できない。

(4) 応答時間と処理時間

この問題に関しては、端末利用者のTSSに対する経験度でその感じ方も違い、また、そのTSSが持つ諸条件（COREの容量、ジョブの多重度；ホールアウトファイルをどの装置に割り当てるかなど）によっても違ってくるので、いちがいにはいえないところであるが、利用者の意見をまとめると次のようになろう。

リモート・バッチ処理については、結果が出力されるまでに予想よりはるかに処理時間を必要とした経験を1回から2回持っているが、全体として問題ないとしている。

デマンド処理については、ジョブの開設時に10秒から20秒ほど待たされるため、これが長く感じられるとしているが、他の応答についてはあまり問題にする端局はなかった。

7.1.5 端末装置の操作性

端末装置が使い易いか使い難いかは、ユーザにとって非常に重要な問題である。ユーザがそのTSSサービスを受けるのは端末装置を介してであり、その操作性によってタイム・シェアリング・システムの評価にも通じるといつても過言ではないが、今回の調査先では次のような意見が述べられた。

- (1) ステートメント入力後に  ボタンを押すが、これを1つのボタンですむようにできないか。（F1592A）
- (2) ミスタイルのほとんどが上中下段のシフトキーの押し違いにある。このシフトキーを改善できないか。
- (3) アンダーライン記号、マイナス記号、長音記号の3種類は区別がつきにくい。もっとわかりやすくできないものか。
- (4) ステートメント入力のたびに  を押さなければならないのは手数がかかりすぎる。 で代用できるようにならないか。（F1591A）

7.1.6 センタ側に対する要望

端末ユーザがセンタ側にいだく要望事項には、どのようなものがあるか、センタの事情を考慮に入れずに意見を述べてもらったが、次のような意見が聞かれた。

(1) 制限事項に関するもの

- イ. C P U の最大実行時間（5分）が少なすぎる。
- ロ. リモート・バッチで磁気テープを使わして欲しい。
- ハ. 共用ファイルの割り当て領域を増やして欲しい。
- ニ. サービス時間帯をもっと拡張して欲しい。（夜間の利用もできれば良い）
- ホ. 同時に多数の端局が端末を利用している場合、入力を拒否される場合がある。これは T S S の原則に反するのではないか。

(2) システム・ダウンに関するもの

- イ) ダウン時の表示をハード的あるいはソフト的な方法で端末に知らせて欲しい。（現在やっている所もあったが、表示時間がかかりすぎること）
- ロ) バック・アップのコンピュータがないため復旧が遅く感じる。もっと早く復旧させて欲しい。

7.1.7 その他の

今まで述べた以外に調査先で聞かれた意見としては次のようなものがあった。

(1) 端末装置について

- イ) 1,200 B P S の速度に合う端末装置となると C P U, コアメモリ, カードリーダ, ラインプリンタでかなりの金額になる。今後安価な端末装置が開発されることを望む。
- ロ) 1,200 B P S 用のFACOM-R 端末では、デマンド処理をする場合タイプライタを使用するが、入力については現在の能力（20字／秒）で十分である。出力についてはやはり速ければ速いほど良い。なお、50 B P S の端末では、デマンド・ジョブの入出力に対して能力不足が感じられる。

200 B P S 程度の端末がデマンド処理用に適していると思う。

(2) 通信回線について

- イ) TSSの使用に当っては、電々公社の専用通信回線を借用するわけであるが、回線借用料が距離に比例して高くなり維持が大変である。リモート・バッチ処理に対しては回線の速度が速いほど望ましい。しかし、経済性を考慮すると今以上に高速度になれば研究室レベルでは維持がむつかしくなる。1つの方法としては、いくつかの研究室が集まって共同利用的なりモートステーションを設置することが考えられる。
- ロ) 回線の安定性については、稼動開始当初心配していたが、これに関する限りほとんど問題はなかった。

(3) 端末装置の互換性について

技術の進歩に伴なって、より高性能なコンピュータ、よりパラエティーに富んだ端末装置、ますます高速な通信回線が開発されるであろう。従って、数年間で新機種と交代させシステムの機能を拡大したいと願う気持が当然生じてくる。これについては、センタも利用者も同じである。しかし、新機種を導入する際は、端末とセンタとのインターフェースを十分考慮すべきである。すなわち、ハードウェアおよびソフトウェアのインターフェースの互換性を確保するよう望みたい。センタが新機種になったとたんに端末が使用不可能になったのではこまる。

7.2 調査II

7.2.1 調査目的

地方公共団体が情報化社会の進展に対応するために、地域企業団体（特に中小企業）の近代化促進の一助としてコンピュータを利用した情報提供、情報処理および情報処理教育を行なうこととした情報処理センターの開設を計画する動きが各地でおこっている。

その種の計画、準備中の情報センターではコンピュータの利用方法について主につきの三点について調査研究を行なっている。

- レンタル、買取りによるコンピュータの導入、使用
- 他の団体、企業（主に計算センター等）のコンピュータの利用……全面委託、コンピュータの時間借り等
- 通信回線の利用による他の団体企業のコンピュータ利用または遠隔情報処理サービス（商用TSSサービス）の利用

以上のコンピュータの利用方法のうち、いずれが最適であるかは、計画または準備中の上記の情報処理センターの性格、規模および情報処理業務の内容等、種々の条件によってそれぞれ一長一短があり、いちがいに断言することが出来ない。しかしながら通信回線を利用したコンピュータの使用、いわゆる遠隔情報処理は比較的廉価の端末機を設置して中央の大型コンピュータの機能を隨時必要に応じて活用出来るという大きな利点があると考えられるが、現在までのところ小型コンピュータのレンタル、または買取りによる設置がその大半を占めている。

その理由の一つとして考えられるのが、現時点では地方に遠隔情報処理サービスを提供する企業、団体が少ないとあるが公衆通信回線の開放にともないその種のサービスを提供する企業、団体も地方都市に順次増加するものと考えられる。

上記のようなすう勢にかんがみ遠隔情報処理システムの利用分野、今後開発するソフトウェアの方向種類等を研究するため、中小企業に対する情報の提供、

情報処理業務をその主たる業務として現在コンピュータの利用方法を調査、検討中である代表的なセンターを想定して遠隔情報処理サービスの利用の可能性の調査を実施した。

7.2.2 調査対象センターの性格

調査対象センターはコンピュータを利用して県内の中小企業の経営の近代化合理化促進の一助として企業の必要とする外部情報（県内主要生産品、その生産会社、県内外の新技術）の提供、県内中小企業の動態管理、経営診断および情報処理サービスを行なう目的で、地方公共団体が設立した団体であり、現在コンピュータの使用方法（単独導入、他の団体・企業のコンピュータ使用等）等について調査研究を行なっている。

7.2.3 主な業務の概要

(1) 経営診断

経営診断は企業からの経営診断の申込み、診断の実施（予備診断—現場診断—総合調整—改善勧告）の手順で行なわれるが（図7-1参照）、コンピュータで処理を計画している部分は、取扱う情報が計数化、処理手順がパターン化されており、かつその処理時間を短縮する必要性またはメリットのある予備診断である。

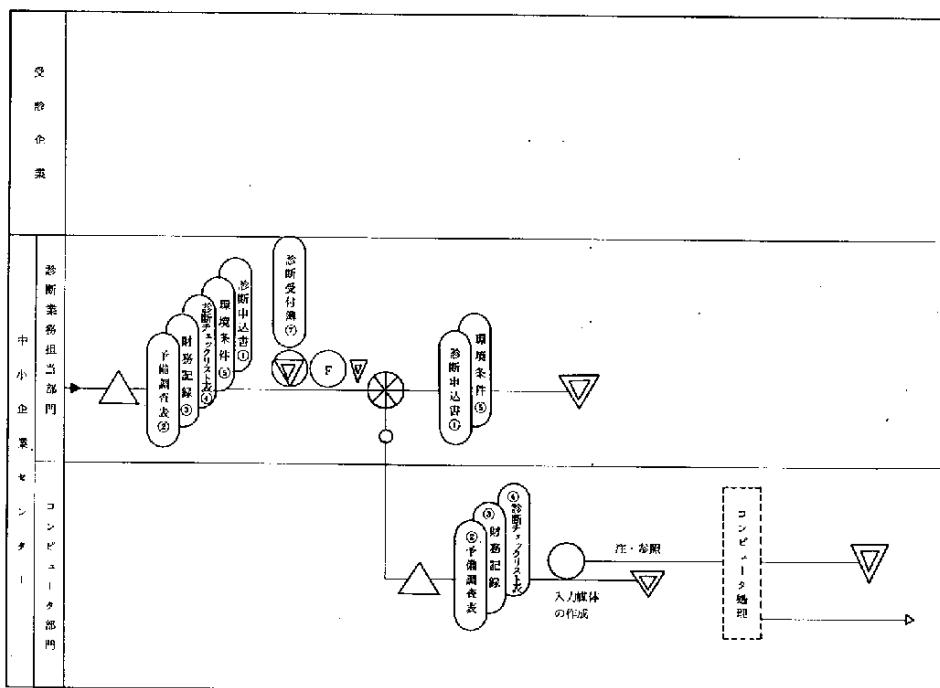
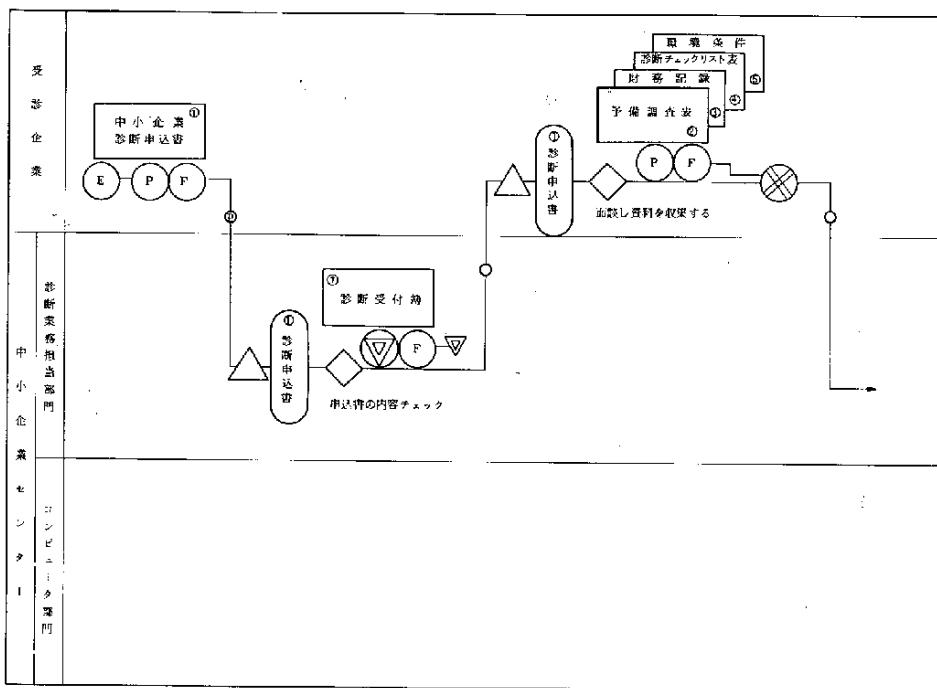
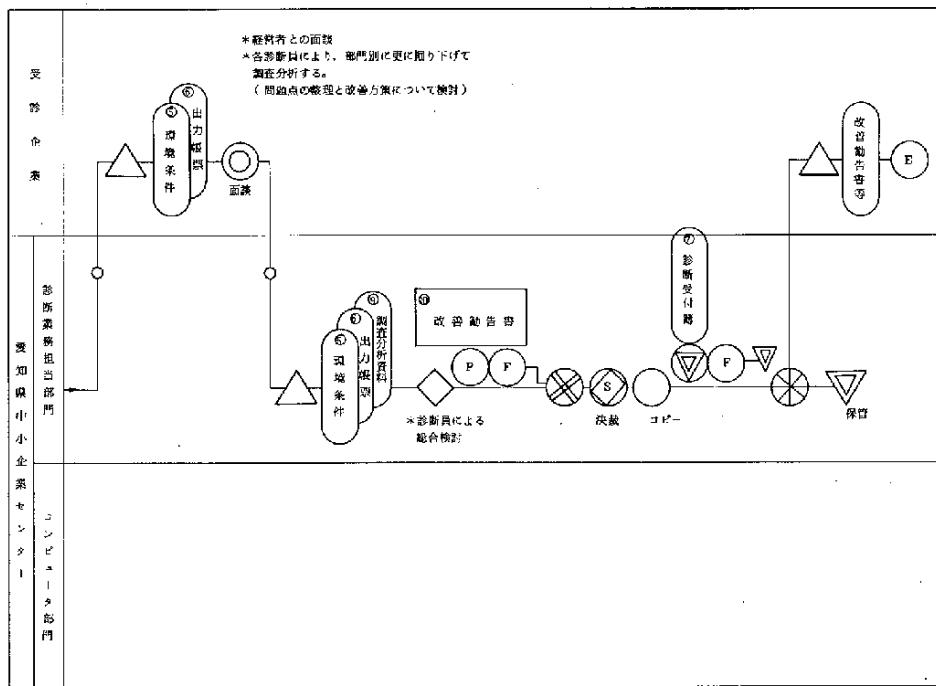
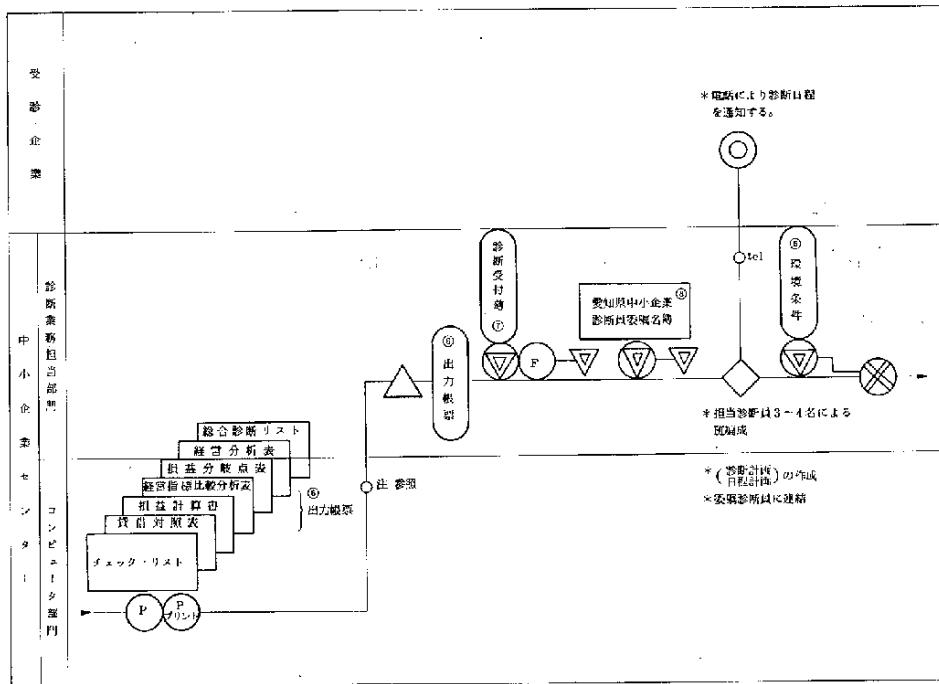


図 7-1 コンピュータを使用した診断業務の想定流れ図



注 診断業務においてコンピュータの利用方法については主にこの部分、即ちデータ、アウトプットの受けわたし方法、手段がデータの発生状況（受診企業の申込み状況）と関連して調査、研究部分である。

(2) 動態管理

本業務は県下1万万余件の中小企業の主に財務諸表等を収集して、その現状、動向を正確、かつ、迅速に把握し、地方公共団体、公共指導機関、中小企業診断担当部門および各中小企業経営者に経営近代化、合理化の指導および経営方針の決定等のための情報を提供することを目的としている。（図7-2参照）

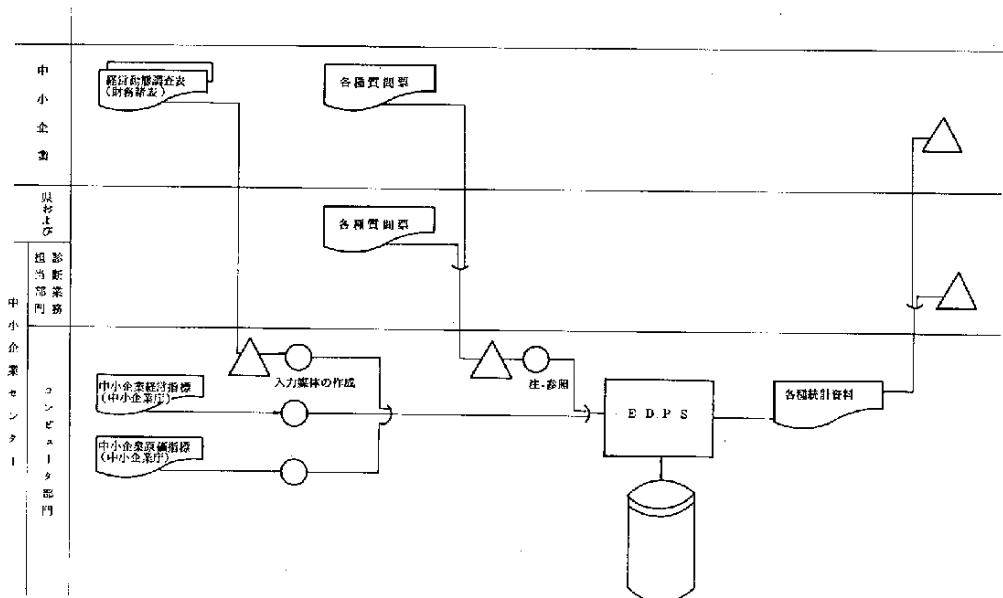


図7-2 コンピュータを使用した動態管理業務の想定流れ図

注 現時点では中小企業の経営者、管理者等が直接、中小企業センターの窓口、受付で各種の企業経営情報の提供を要求する場合が多い。従がって即時にその情報を提供することが望ましい。

(3) 情報検索提供業務

県内の生産品に関する情報、技術情報および公害情報等を蓄積して不特定

な中小企業等からの問合せに応じて情報を検索、提供する業務（図7-3参照）であるがコンピュータの利用によって情報提供の迅速化、その質的量的向上が期待される。

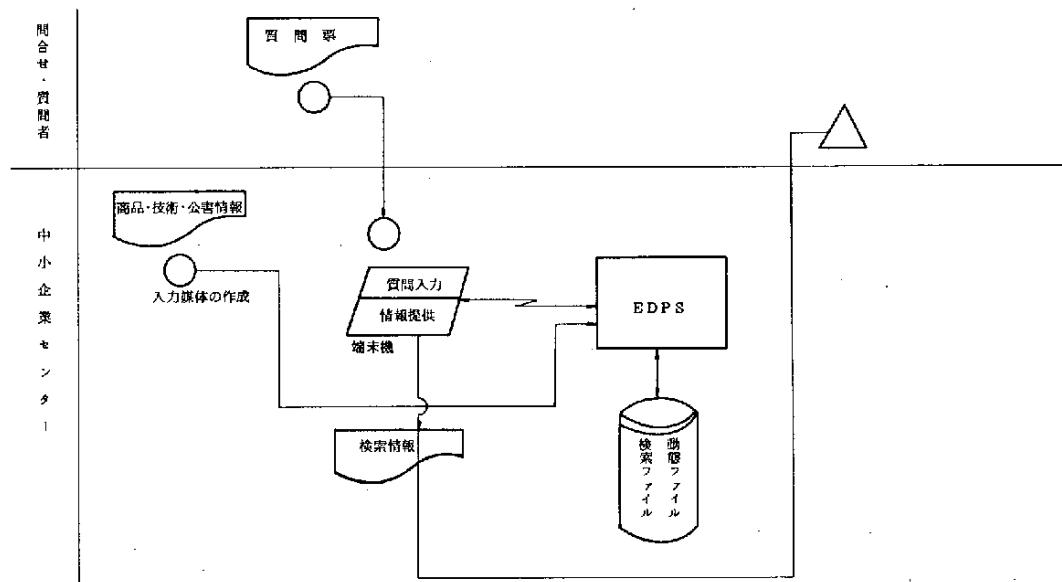


図7-3 コンピュータを使用した情報検索、提供業務の想定流れ図

7.2.4 コンピュータの利用方法の検討

コンピュータの利用方法としては、前述した如く①単独導入による場合、②計算センターに委託する場合、③通信回線を利用した遠隔情報処理システムによる場合等の3種類に分けることが出来るが、ここでは③によるコンピュータの利用方法を中心としてその利用形態、利用上の問題点等を検討する。

いうまでもなくコンピュータ・システムを考察する場合、適用業務はいかなるものであるかが、経済性と同時に重要な要件であり、その適用業務の目的、効果を十分に発揮できるシステムを設計しなければならない。

1) 適用業務の内容

前述した如く適用業務は、動態管理業務、経営診断業務、情報検索業務等の3種類である。

これらの業務をコンピュータ処理の面から考察すると三者三様の特性を持

っており異なっている。

すなわち動態管理業務については収集資料（入力データ）が膨大になるため簡単で正確な入力方式（装置）が要求され、また多量データの効率的なファイル方法（記憶媒体），あるいは高速な出力方法（装置）が望まれる。経営診断業務は前者に比較してデータ量が少ないためコンピュータ処理面での特性は顕著ではないが、場合によっては即時処理が要求される。

情報検索提供業務については相談者サービスの徹底を図る意味から、いつでも任意の時に相談者の質問に対する解答を即時に出力する必要があり、それに適した入力方法を、また高速に検索できるファイル構成および大量のデータをファイルできる記憶装置が必要となる。

以上のような特性を持つ3業務について整理すると次のようになる。

適用業務の内容（コンピュータ処理面）

業務名 検討項目	経営診断	動態管理	情報検索提供
ソースデータの形式	質問形式のデータおよび数字記入のデータの2通りがある	左と同じ	定型化された方法ではないケースが多い
入力データ量	1企業当たりのデータ件数は多いが、年間取扱い企業数はさほど多くない	1回の調査で約10,000企業であり、1企業当たり100データであるから1,000,000データとなり、非常に多い	入力データ量は少ない
出力データの形式	一覧表形式の出力帳票が適する	一覧表形式の出力帳票が適する	出力形式は定型化されない。カナ文字出力が主となる
出力データ量	あまり多くない	1時的に非常に多い	少ない
ファイル形式およびファイル・データ量	シーケンシャル・ファイルとする。データ量はあまり多くない。	シーケンシャル・ファイルおよび一部はランダム・アクセス・ファイルとする。データ量が多い	ランダム・アクセス・ファイルとする。データ量は多い
処理方法	一括処理あるいは即時処理	一括処理	即時処理
処理サイクル(緊急性)	データ発生時に随時行なう	年1，2回の処理であり、データ量も多いので長期的計画を立てて処理する	相談者は何時来るか分からぬため常に処理可能な状態にしており、データが発生したならば他の作業に割込んで処理する

2) 通信回線を利用した遠隔情報処理システムによるコンピュータ利用形態

情報センターにおいて「通信回線を利用した遠隔情報処理システムによるコンピュータの利用方法」によって当該適用業務を行なう場合の利用形態概念図(図7-4)は次のとおりである。

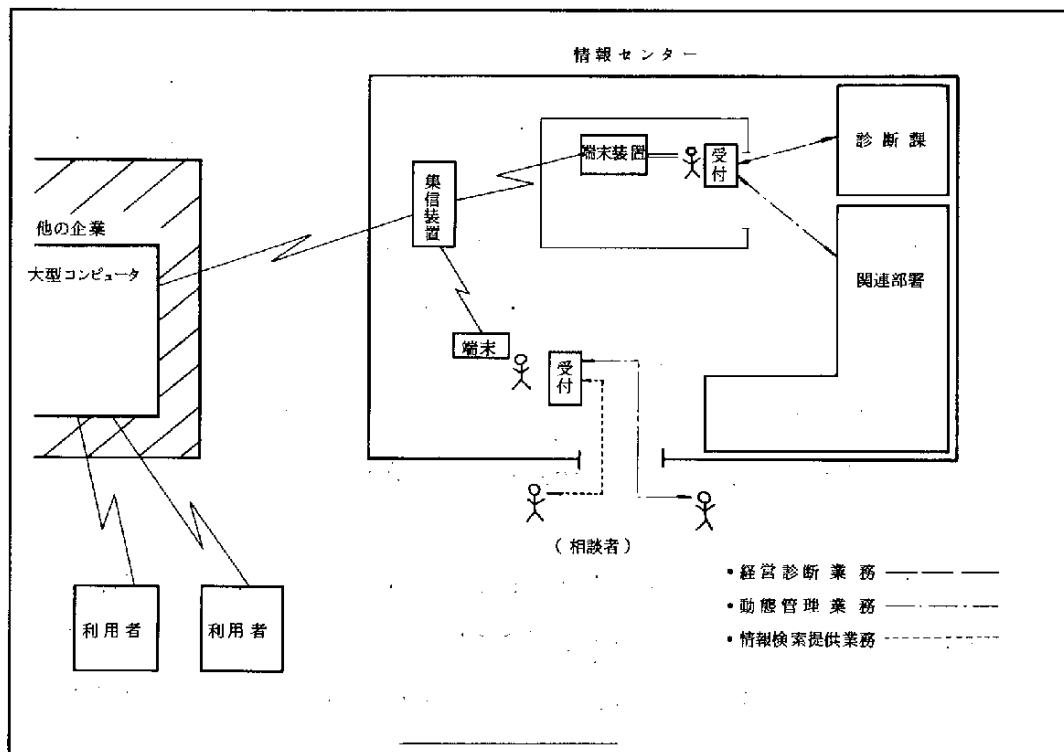


図7-4

3) 利用上の問題点

適用業務の内容を前提として、コンピュータ利用上の問題点を検討する。

(1) 適用業務について

外部の大型コンピュータとセンター内の端末装置を通信回線を介して接続し、入力データの送信、コンピュータ処理結果の受信等の順序で業務が遂行される訳であるが、適用業務の内容からみて、データ量とデータ伝送速度が処理効率の面で大きく影響してくれる。

経営診断業務および情報検索業務はデータ量は少なく、かつ、処理結果を出来るだけ早く出力する必要があり、この点から考えれば、この業務は

適している。特に情報検索提供業務については、大量のデータのファイルが必要であり、検索処理も高速性が要求されるため、大型のコンピュータが利用できることは、他のユーザも同時に使用することから多少のレスポンス・タイム（応答時間）がかかるとしても業務の目的効果は十分に期待できる。

動態管理業務については、10,000企業に及ぶ大量のデータを取り扱うため高速の入力装置および高速の通信回線が必要となる。前者については、当該装置の利用が不可能ではないが、不離一体である通信回線のデータ伝送速度に制限されるため、使用入出力装置も実質的には限定される。

現在利用できる通信回線は、2,400 ポー、1,200 ポー、200 ポーであり、最高の 2,400 ポー回線を使用しても決して満足できるものではない。

以上の如く業務の内容によって適、不適があり、これら 3 業務をコンピュータ化する場合十分検討する必要のある問題である。

(2) 融通性について

外部の大型コンピュータに用意されているユーティリティ・プロは一通り端末から使用でき、また端末装置についても高速なものは使用できないが、一応各種装置の使用が可能であり、この点からはある程度の融通性はある。

しかしながらコンピュータ使用時間帯については多少の制限があり、問題点である。

現在稼動している適用 TSS のサービス時間帯はセンターによって差違はあるが午前 8 時頃～午後 8 時頃までであり、これ以外の時間帯の使用はできない。

したがって、緊急の仕事が発生し、上記時間帯以外でのコンピュータ使用が必要になった場合には対処できず、他の手段によって解決しなければならない。

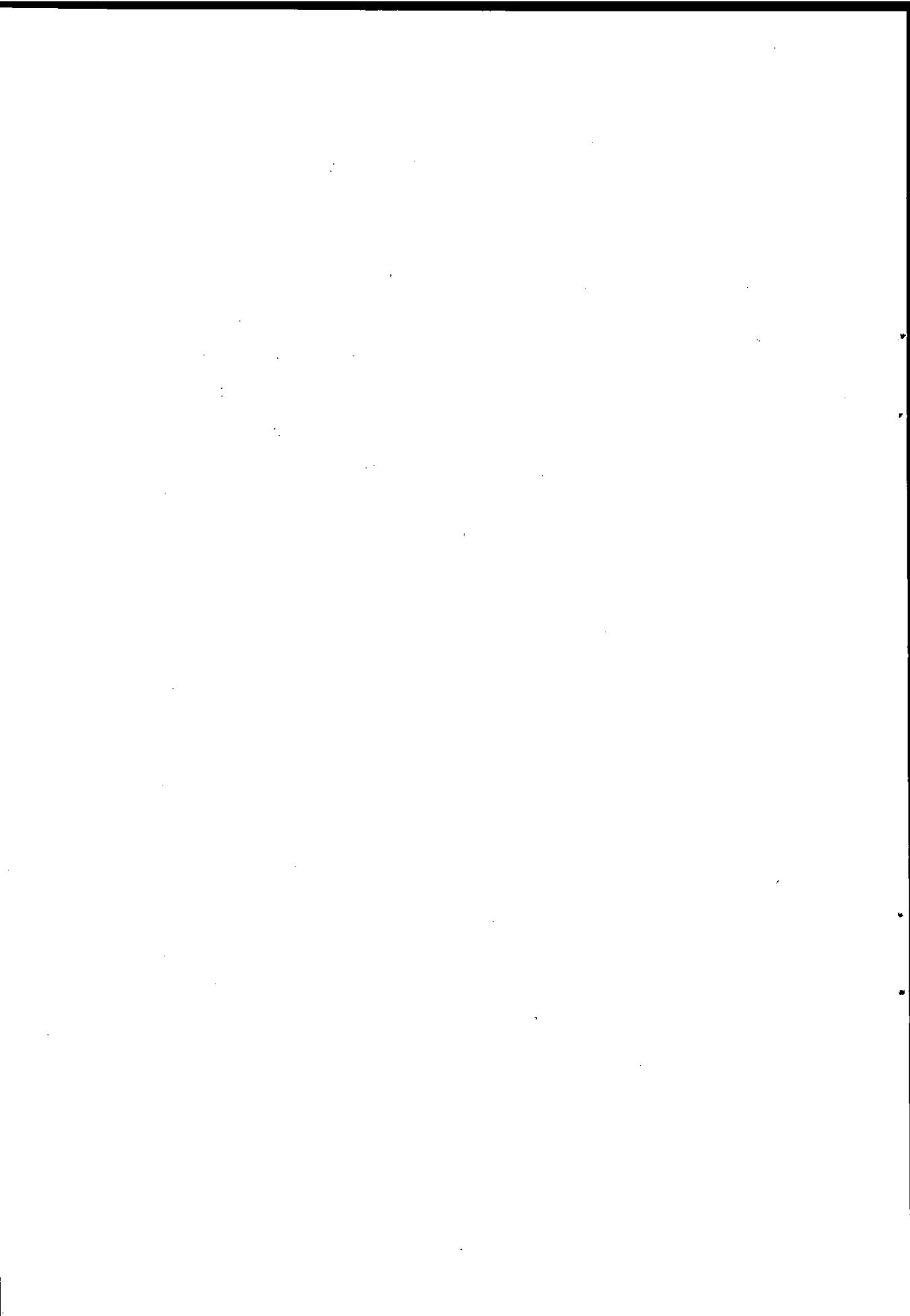
(3) 機密保持について

自社に設置した端末装置からデータを入力し、結果を出力する方式であ

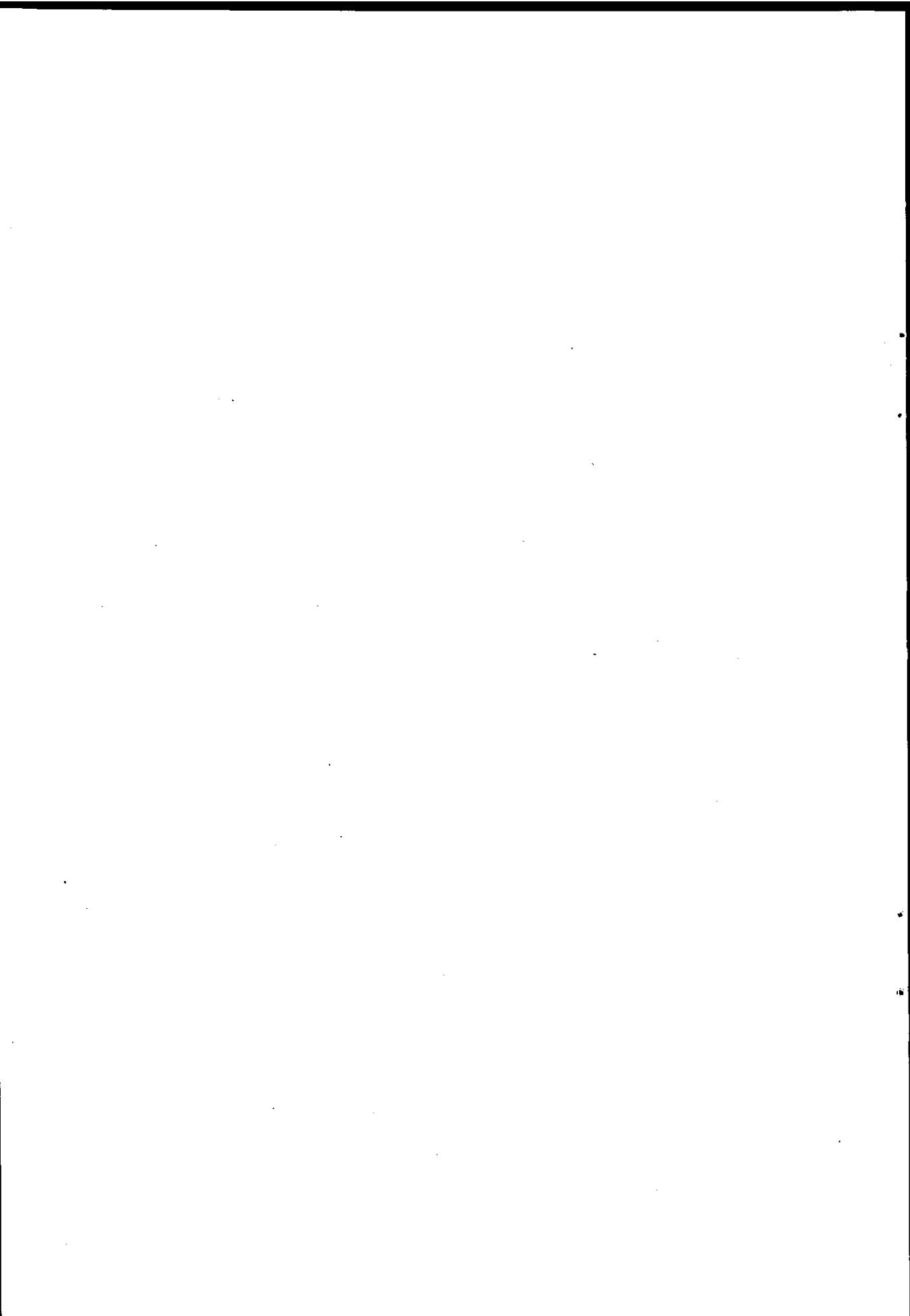
るからデータ（伝票、書類）を外部に持ち出すことがなくこの点からは機密漏えいの危険はない。

外部の大型コンピュータ側にファイルされているデータ、ユーザプロ等については共同利用している他ユーザに使用される危険があるが、大型コンピュータ側の管理運営上の問題であるが、通常他の端末装置からは使用できないようにシステムが設計されており機密漏えいの危険はない。

今回の適用業務の中で経営診断業務、動態管理業務については中小企業の経営内容（財務諸表）が処理データになるため、これが外部に漏れることは問題であり機密保守は遵守されなければならない。



8. プログラムの開発



8. プログラムの開発

プログラムの開発は、以下のような運営管理用の計算センター・ルーチンからオンライン用アプリケーションまで多岐にわたって行なった。

(1) 計算センター・ルーチンの作成

- 制御プログラムの一部として組込んだもの
- 会計ファイルから各種統計表を作成する処理プログラム

(2) CPL による科学計算用ライブラリーの作成

(3) TSS-FORTRANの\$REPLACE サブコマンドの新設

(4) オンライン文献検索システム JOLDOR-II の開発

計算センター・ルーチンについては、「3.2 計算センター・ルーチン」の項で述べたとおりであるので説明は省略する。

(2), (3), (4)について、その概要及び仕様を以下に示す。

8.1 CPL用科学計算ライブラリの開発

まえがき

本ライブラリーは、サブルーチン手続きあるいは関数手続きから成るもので、ユーザーが自由にメインプログラムの中に組み込むことができる。

手続きの引用は以下のようない形で行なわれる。

- (1) サブルーチンの呼び出し

CALL 入口名 (引数[,引数]………)

- (2) 関数の呼び出し

組込関数と同様に算術式の右辺の中 入口名 (引数)

で呼び出すことができる。

8.1.1 ライブラリーの特徴

- (1) 引数とパラメータ

引数とパラメータの属性は必ず一致させなければならない。各サブプログラムの引数の説明を参照（属性を省略としてあるのは省略時解釈の意）。

- (2) 配列の宣言

各サブプログラムはアレイの要素数を DECLARE ステートメントによって自由に定義できるように可変にしているので、引数の説明を参照して適宜定義すること。

- (3) 精 度

連立1次方程式、逆行列、固有値問題等データ数が多い場合や、代数方程式、微分方程式等対象が複雑な形をとりうる場合など丸めの誤差の集積が解におよぼす影響が大きいので、精度を高めるため内部演算が DECIMAL FLOAT で 20桁のプログラムを用意した。DMINV, DEIGENN, DSIMQ, DRTNIT, DRKG2, RUNGE, DSMPSN。

- (4) 行列演算

行列演算を目的とするサブプログラムを使用する場合次の点に注意するこ

1. 行列のディメンジョンは、すべて1次元でとられている。これは使用エリアをできるだけ有効に使う目的のためである。たとえば、対称行列を処理する場合は、普通 $N \times N$ のエリアが必要であるが、この場合は $N \times (N + 1)/2$ のエリアで済むようになっている。従って入力行列および出力行列の1次元アレイの要素数は、その型（一般行列、対称行列、対角行列のいずれか）によって次のようになる。

一般行列 $N \times M$

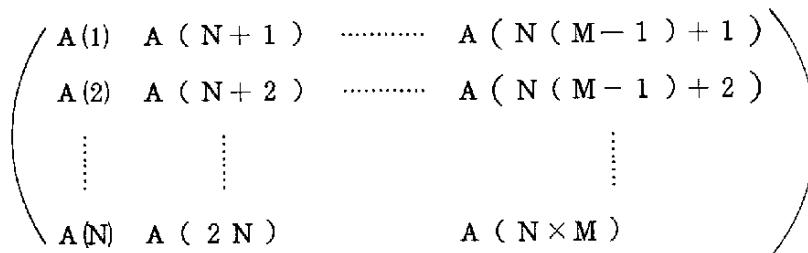
対称行列 $N \times (N + 1)/2$

対角行列 N

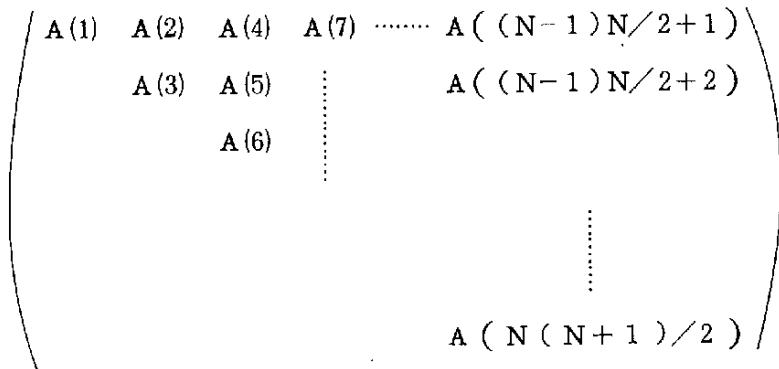
ただし $\begin{cases} N : 行列の行の数 \\ M : 行列の列の数 \end{cases}$

2. 入力行列の要素のデータを、その行列の型によって以下の番号づけで入力する。

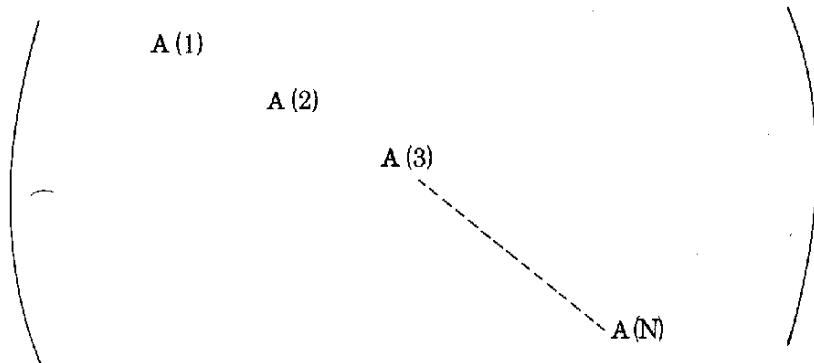
一般行列（全要素のデータを入力）.....



対称行列（上三角要素のデータだけ入力）.....



対角行列（対角要素のデータだけ入力）………



また出力行列の要素の値の出力も、その型によって 1 次元アレイ中、上に挙げた番号づけにストアされているものとして出力する。

3. 引数の説明の項で、行列の型を表わす数というのがでてくるが、これは、対象となる行列が一般行列なら 0、対称行列なら 1、対角行列なら 2 を入る。

[ライブライリー一覧表]

名	ライブラリーネーム	内 容	「ステップ数」
1	MTRA	行列の転置	17
2	MPRD	行列の積	43
3	LOC	2次元のアレイを1次元に換える	19
4	MCPY	行列のコピー	27
5	MINV	実数行列の逆行列を求める	96
6	DINV	同 上	96
7	POWER	固有値、固有ベクトルを求める (Power Exhaustion 法)	107
8	EIGENN	実対称行列の固有値、固有ベクトル (Jacobi 法)	113
9	DEIGENN	同 上	116
10	TRACE	与えられた固有値の中から、指定された値以上の固有値の個数	20
11	SIMQ	実係数連立1次方程式を解く (掃き出し法)	60
12	DSIMQ	同 上	61
13	SIMQG	同 上 (Gauss-Seidel 法)	62
14	POLRTN	実数係数代数方程式の実根、複素根を求める	125
15	RTMIT	非線型方程式の実根を求める (放物線逆補間法)	65
16	RTNIT	同 上 (Newton Raphson 法)	30
17	DRTNIT	同 上	33
18	ATKIN	補間 (Aitken-Lagrange 法)	104
19	LESQ	1次最小自乗法	42
20	RKG 2	1階常微分方程式 (Runge-Kutta-Gill 法)	32
21	DRKG 2	同 上	35
22	RKG 2	n元連立1階常微分方程式 (Runge-Kutta-Gill 法)	49
23	RUNGE	同 上	53
24	SMP SN	積分値を求める (Simpson の公式)	43
25	DSMP SN	同 上	45
26	QUADR	同 上 (2点公式, 3点公式, 4点公式, 5点公式)	30
27	GAUSS	正規乱数の発生 (中心極限定理)	28
28	NORMAL	同 上 (直接法)	25
29	RANDUN	一様乱数を求める	17
30	POISN	ポアソン乱数を求める	59
31	MULTR	回帰分析、重回帰分析、多項式回帰分析	57
32	TALLY	データの整理、総和、平均値、標準偏差、最大最小値	28
33	TTEST	母平均および平均値の差の検定のための統計量 t を求める	59
34	ORDER	相関係数行列を作る	45
35	PADD	多項式の和	41
36	BESFJN	第1種 Bessel 関数	70
37	BESFYN	第2種 Bessel 関数	123

〔(注) ステップ数とはライブラリーのソース・ステートメントの総数である。〕

8.1.2 各ライブラリーの説明

(1) MTRA

(1) 目的

与えられた行列 A の転置行列 $R = A^t$ を求める。

(2) 使用方法

CALL MTRA (A, R, N, M, MS);

(3) 引数の説明

(i) 入力

A ; BIN FLOAT(26) 入力行列。

N ; 省略 行列 A の行の数および行列 R の列の数。

M ; 省略 行列 A の行の数および行列 R の行の数。

MS ; 省略 行列 A の型を表わす数。

(ii) 出力

R ; BIN FLOAT(26) 行列 A の転置行列の入る出力行列。

(4) 使用されるサブプログラム

MCPY ([4] MCPY 参照)

LOC ([3] LOC 参照)

(5) 注意事項

(i) 各行列に対する 1 次元アレイの要素数、引数 MS、および行列要素の値の入力、出力に関しては

4.1-(4) 行列演算を参照。

(ii) 出力行列 R の型は、入力行列 A と同じである。

(6) 使用例

$$A = \begin{pmatrix} 5 & 8 & 7 & 10 \\ 9 & 2 & 4 & 6 \\ 13 & 8 & 2 & 16 \\ 5 & 9 & 7 & 8 \end{pmatrix}$$

の転置行列を求める。

```

? 0010  MMTRA: PROC OPTIONS(MAIN);
? 0020  DCL A(16) BIN,R(16) BIN;
? 0030  N=4;
? 0040  M=4;
? 0045  MS=0;
? 0060  A(1)=5;
? 0070  A(2)=9;
? 0080  A(3)=13;
? 0090  A(4)=5;
? 0100  A(5)=8;
? 0110  A(6)=2;
? 0120  A(7)=8;
? 0130  A(8)=9;
? 0140  A(9)=7;
? 0150  A(10)=4;
? 0160  A(11)=2;
? 0170  A(12)=7;
? 0180  A(13)=10;
? 0190  A(14)=6;
? 0200  A(15)=16;
? 0210  A(16)=8;
? 0220  CALL MTRA(A,R,N,M,MS);
? 0230  PUT EDIT(R)( SKIP,4 F(5));
? 0240  END MMTRA;
LAST LINE(END ST) 0700.00
? %RUN;

```

9	8	7	10
9	2	4	6
13	8	2	16
5	9	7	8

(注)

結果は以下のように印字されている。

A (1) A (2) A (4)

A (5) A (6) A (8)

.....
.....
.....

A (13) A (16)

[2] MPRD

(1) 目 的

与えられた 2 つの行列 A, B の積 R = A · B を求める。

(2) 使用方法

CALL MPRD (A, B, R, N, M, MSA, MSB, L) ;

(3) 引数の説明

(i) 入 力

A ; BIN FLOAT(26) 入力行列。

B ; BIN FLOAT(26) 入力行列。

N ; 省 略 行列 A, R の行の数。

M ; 省 略 行列 A の列の数。行列 B の行の数。

MSA ; 省 略 行列 A の型を表わす数。

MSB ; 省 略 行列 B の型を表わす数。

L ; 省 略 行列 B の列の数。

(ii) 出 力

R ; BIN FLOAT(26) 行列 A, B の積の入る出力行列。

(4) 使用されるサブプログラム

LOC ([3] LOC 参照)

(5) 注意事項

(i) 各行列に対する 1 次元アレイの要素数、引数 MSA, MSB および行列要素の値の入力、出力に関しては

4.1(4) 行列演算を参照

(ii) 出力行列 R の型は、入力行列 A および B の型により以下のようになる。

A	B	R
一般行列	一般行列	一般行列
一般行列	対称行列	一般行列
一般行列	対角行列	一般行列
対称行列	一般行列	一般行列
対称行列	対称行列	一般行列
対称行列	対角行列	一般行列

対角行列

一般行列

一般行列

対角行列

対称行列

一般行列

対角行列

対角行列

対角行列

(iii) 行列 R のエリアを、行列 A および行列 B と同じエリアに割当ててはならない。

(6) 使用例

$$A = \begin{pmatrix} 5 & 8 & 7 \\ 9 & 2 & 4 \\ 13 & 8 & 2 \end{pmatrix}$$

$$B = \begin{pmatrix} 6 & 13 & 7 \\ 1 & 4 & 8 \\ 10 & 3 & 2 \end{pmatrix}$$

で $A \times B$ を求める。

```
? 0010  MMPRD: .PROC OPTIONS(MAIN);
? 0020  DCL A(9) BIN FL0AT,B(9) BIN FL0AT,R(9) BIN FL0AT;
? 0030  N=3;
? 0040  M=3;
? 0060  MSA=0;
? 0070  MSB=0;
? 0080  L=3;
? 0100  A(1)=5;
? 0110  A(2)=9;
? 0120  A(3)=13;
? 0130  A(4)=8;
? 0140  A(5)=2;
? 0150  A(6)=8;
? 0160  A(7)=7;
? 0170  A(8)=4;
? 0180  A(9)=2;
? 0190  B(1)=6;
? 0200  B(2)=1;
? 0210  B(3)=10;
? 0220  B(4)=13;
? 0230  B(5)=4;
? 0240  B(6)=3;
? 0250  B(7)=7;
? 0260  B(8)=8;
? 0270  B(9)=2;
? 0280  CALL MPRD(A,B,P,N,M,MSA,MSB,L);
? 0290  PUT EDIT(R)(SKIP,3 F(20,8));
? 0300  FND MMPRD;
LAST LINE(FND ST)    0740.00
? %RUN:
```

1.08000001E+02	9.60000015E+01	1.06000001E+02
1.18000001E+02	1.37000001E+02	2.07000002E+02
1.13000001E+02	8.70000012E+01	1.59000003E+02

[3] LOC

(1) 目的

与えられた行列の 2 次元の番号付けを 1 次元の番号付けにかえる。

(2) 使用方法

CALL LOC (I, J, IR, N, M, MS);

(3) 引数の説明

(i) 入力

I	；	省略	ユーザの指定する要素の行番号。
J	；	省略	ユーザの指定する要素の列番号。
N	；	省略	与えられた行列の行の数。
M	；	省略	与えられた行列の列の数。
MS	；	省略	与えられた行列をストアする型を表わす数。

(ii) 出力

IR ; 省略 1 次元に直した場合の行列要素の番号。

(4) 使用されるサブプログラム

なし。

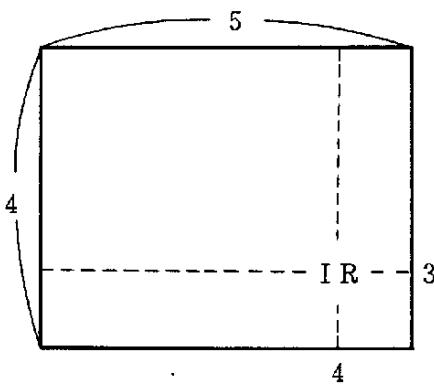
(5) 注意事項

(i) 各行列に対する 1 次元アレイの要素数, 引数 MS, および行列要素の値の入力, 出力に関しては,

4. 1 (4) 行列演算を参照。

(6) 使用例

大きさ (4, 5) の一般行列の第 3 行, 第 4 列の要素の 1 次元の番号付けを求める。



```

? 0010  ML6C: PROC OPTIONS(MAIN);
? 0020  I=3;
? 0030  J=4;
? 0040  N=4;
? 0050  M=5;
? 0060  MS=0;
? 0070  CALL L6C(I,J,IR,N,M,MS);
? 0080  PUT DATA(IR);
? 0090  END ML6C;
LAST LINE(FND ST) 0280.00
? %RUN;
IR=      15;

```

[4] MCPY

(1) 目的

与えられた行列のすべてをコピーする。

(2) 使用方法

CALL MCPY (A, R, N, M, MS) ;

(3) 引数の説明

(1) 入力

A ; BIN FLOAT(26) 入力行列。

N ; 行列 A, R の行の数。

M ; 行列 A, R の列の数。

MS ; 行列 A の型を表わす数。

(4.1(4) 行列演算参照)

(ii) 出 力

R : BIN FLOAT(26) 入力行列と同じ型の出力行列。

(4) 使用されるサブプログラム

LOC ([3] LOC 参照)

(5) 注意事項

(i) 各行列に対する 1 次元アレイの要素数, 引数 MS, および行列要素の値の入力, 出力に関しては, 4.1(4)行列演算参照。

(6) 使用例

4 行 4 列の行列をコピーする。

```
? 0010  MMCPY: PROC OPTIO(MS(MAIN));
? 0020  DCL A(16) BIN,
?           R(16) BIN;
? 0030  N,M=4;
? 0040  MS=0;
? 0050  GET LIST(A);
? 0060  CALL MCPY(A,R,N,M,MS);
? 0070  PUT FDIT((R(I) DD I=1 TD 16))(X(5),4F(6),SKIP);
? 0080  END MMCPY;
LAST LINE(END ST) 0360.00
? %RUN;
DATA 0050.00    A? 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
      1      2      3      4
      5      6      7      8
      9     10     11     12
     13     14     15     16
```

[5] MINV

(1) 目 的

実行列 A の逆行列 A^{-1} を掃き出し法により求める。

(2) 使用方法

CALL MINV (A, N, D, L, M);

(3) 引数の説明

(i) 入 力

A : BIN FLOAT(26) 要素数 ($N \times N$) の 1 次元アレイ。

		逆行列を求める行列(プログラム実行中書きかえられる)。
N	; 省略	Aの次数。
L	; 省略	要素数Nの1次元のワーク・エリア。
M	; 省略	要素数Nの1次元のワーク・エリア。

(ii) 出力

A	; BIN FLOAT (26)	要素数($N \times N$)の1次元アレイ。逆行列が求まる。
D	; BIN FLOAT (26)	入力行列Aの行列式の値が求まる。

(4) 使用されるサブプログラム

なし

(5) 注意事項

- (i) 行列式の値Dが0の場合、行列Aは特異行列であるため、逆行列は求まらない。
- (ii) 入力のAと出力のAは同じエリアである。

(6) 使用例

$$A = \begin{pmatrix} 2 & 4 & 3 \\ 4 & 30 & 3 \\ 6 & 5 & 5 \end{pmatrix}$$

の行列式及び逆行列を求める。

```

? 0010  MMINV: PROC OPTIONS(MAIN);
? 0020  DCL A(9) BIN FLOAT,L(3),M(3);
? 0030  DCL D BIN FLOAT;
? 0040  N=3;
? 0060  A(1)=2. ;
? 0070  A(2)=4. ;
? 0080  A(3)=6. ;
? 0090  A(4)=4. ;
? 0100  A(5)=30. ;
? 0110  A(6)=5. ;
? 0120  A(7)=3. ;
? 0130  A(8)=3. ;
? 0140  A(9)=5. ;
? 0150  CALL MINV(A,N,D,L,M);
? 0155  PUT SKIP(2);
? 0160  PUT DATA(0);
? 0170  PUT FDIT(A)(SKIP,3 F(20,8));
? 0175  PUT SKIP(5);
? 0180  END;
LAST LINE(END ST)    1150.00
? %RUN;

```

D=-2.1800000E+02;

-6.19266048E-01	9.17431145E-03	7.33944952E-01
2.29357773E-02	3.66972479E-02	-6.42201825E-02
3.57798166E-01	-2.75229360E-02	-2.01834857E-01

[6] DMINV

(1) 目的

実行列Aの逆行列A⁻¹を掃き出し法により求める。

(2) 使用方法

CALL DMINV (A, N, D, L, M);

(3) 引数の説明

A, Dの属性がDEC FLOAT(20)になる。

その他はMINVと同じ。 ([5] MINV 参照)

(4) 使用されるサブプログラム

な
し

(5) 注意事項

[5] MINV 参照。

(6) 使用例

$$A = \begin{pmatrix} 2 & 3 & 4 \\ 3 & 5 & 2 \\ 4 & 3 & 30 \end{pmatrix}$$

の行列式及び逆行列を求める。

```

? 0010  MDMINV: PROC OPTIONS(MAIN);
? 0020  DCL A(9) FLOAT(20),L(3),M(3);
? 0030  DCL D DEC FLOAT(20);
? 0040  N=3;
? 0060  A(1)=2.0;
? 0070  A(2)=3.0;
? 0080  A(3)=4.0;
? 0090  A(4)=3.0;
? 0100  A(5)=5.0;
? 0110  A(6)=3.0;
? 0120  A(7)=4.0;
? 0130  A(8)=2.0;
? 0140  A(9)=30.0;
? 0150  CALL DMINV(A,N,D,L,M);
? 0160  PUT SKIP(2);
? 0170  PUT DATA(0);
? 0175  PUT SKIP(2);
? 0180  PUT EDIT(A)(SKIP,2 F(30,19));
? 0185  PUT SKIP(5);
? 0190  END MDMINV;
LAST LINE(END ST)    1160.00
? %RUN;

```

$\theta = -2.00000000000000000000E+00;$

-7.199999999999972485F+01	4.0999999999999984297E+01
5.499999999999979070F+00	3.899999999999985134E+01
-2.199999999999991572F+01	-2.0999999999999988752F+00
6.9999999999999973441F+00	-3.9999999999999984927E+00
-4.0999999999999979877F-01	

(7) POWER

(1) 目的

実行列 A の固有値および固有ベクトルをパワー (Power) 法とイグザーション (Exhaustion) 法により求める。

(2) 使用方法

```
CALL POWER ( A, N, K, N1, M1, ITER, EPS, V, V1,
              V2, X2, XE, X, KK, IER );
```

(3) 引数の説明

(i) 入 力

A : BIN FLOAT(26) 要素数(N, N)の2次元アレイ。固有値、固有ベクトルを求める行列。

(実行中書きかえられる)

N ; 省略	Aの次数
K ; 省略	求める固有値の数。(絶対値最大のものからK個)
N1 ; 省略	メインプログラムにおけるAのディメンジョンの行の数。
M1 ; 省略	メインプログラムにおけるXのディメンジョンの行の数。
ITER ; 省略	最大繰り返し回数。ITER=0と入れるとサブプログラムでITER=200とおく。
EPS : BIN FLOAT(26)	解の相対誤差。EPS=0と入れるとサブプログラムでEPS= 10^{-5} とおく。
V : BIN FLOAT(26)	要素数Nの1次元アレイ。ワーク・エリア。
V1 : BIN FLOAT(26)	要素数Nの1次元アレイ。ワーク・エリア。
V2 : BIN FLOAT(26)	要素数Nの1次元アレイ。ワーク・エリア。
X2 : BIN FLOAT(26)	要素数Nの1次元アレイ。ワーク・エリア。

(ii) 出 力

XE : BIN FLOAT(26) 要素数Kの1次元アレイ。求ま

		った A の固有値。
X	: BIN FLOAT(26)	要素数(N, K) の 2 次元アレイ。 A の固有ベクトルが固有値に対応してカラムワイズに求まる。
K	: 省 略	求まった固有値の数。
KK	: 省 略	収束に関する情報を与えるコード。 KK 番目の固有値を求めるとき, EPS を収束判定基準として ITER 回繰り返したが収束しなかったため, EPS を 10 倍したこと示す。
IER	: 省 略	エラー・コード。 IER=0 正常に解が得られた場合。 IER=1 固有値が K 個求まつたが, K + 1 番目以降は EPS を収束判定基準として ITER 回繰り返したが収束しなかった場合。 (K は(2)出力を参照) IER=2 KK 番目の固有値を求めるとき EPS が 10 倍され, 固有値が求めたい数だけ求まった場合。 IER=3 計算途中でセロ・デ

イバインドを起こした
場合。

(4) 使用されるサブプログラム

なし

(5) 注意事項

- (i) 入力のKと出力のKは同じエリアである。
- (ii) EPS = 0.と入力された場合 EPS = 10^{-5} とおき計算を行なう。
- (iii) ITER=0と入力された場合 ITER=200とおき計算を行なう。
- (iv) IER=1のときK個まで固有値が求まり(Kは入力時の値と違う)
XEとXにはK個までの固有値と固有ベクトルがストアされている。
もしK=0であれば、1つの固有値も求まらなかつたことを示している。
- (v) IER=2のときKK番目の固有値を求めるときEPSが10倍され,
XEとXには求めたい数だけの固有値および固有ベクトルがストアさ
れている。なお(KK+1)番目以降の固有値を求めるときは10倍さ
れたEPSにより収束判定を行なう。
- (vi) IER=3のとき各引数には計算途中の値が入っている。

(6) 使用例

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 1 & 0 & 3 \end{pmatrix}$$

の固有値、固有ベクトルを3個すべて求める場合。

```

? 0010  MPower: PROC(MAINS);
? 0020  DCL A(3,3) BIN,X(3,3) BIN,XF(3),V(3) BIN;
? 0030  DCL V1(3) BIN,V2(3) BIN,X2(3) BIN;
? 0040  N1=3;
? 0050  M1=3;
? 0060  N=3;
? 0070  K=3;
? 0080  EPS=0.1E-03;
? 0090  ITER=0;
? 0100  A(1,1)=1.0;
? 0110  A(1,2)=1.0;
? 0120  A(1,3)=0.0;
? 0130  A(2,1)=0.0;
? 0140  A(2,2)=2.0;
? 0150  A(2,3)=0.0;
? 0160  A(3,1)=1.0;
? 0170  A(3,2)=0.0;
? 0180  A(3,3)=3.0;
? 0190  CALL PPOWER(A,N,K,N1,M1,ITER,EPS,V,V1,V2,XF,X,KK,IER);
? 0200  PUT SKIP(2);
? 0210  PUT DATA(XE);
? 0220  PUT FDIT(X)(SKIP,3 E(20,8));
? 0230  PUT DATA(K,KK,IER);
? 0235  PUT SKIP(5);
? 0240  END MPower;
LAST LINE(END ST) 1320.00
? XRUN;

```

```

XE(1)= 2.9994919E+00   XE(2)= 2.0001228E+00   XF(3)= 9.9999999E-01;
      3.38434215E-04      5.77057272E-01      -8.94449144E-01
      3.38434215E-04      5.77057272E-01      1.09358509E-04
      9.99999895E-01      -5.77935874E-01      4.47169736E-01
K=          3    KK=          0    IER=          0;

```

[8] EIGENN

(1) 目的

実対称行列の固有値および固有ベクトルをヤコビ (Jacobi) 法により求める。

(2) 使用方法

```
CALL EIGENN ( A, N, ITER, R, IC, N9 );
```

(3) 引数の説明

(i) 入力

A ; BIN FLOAT(26) 固有値、固有ベクトルを求める

			行列。要素数 $N \times (N+1)/2$ の 1 次元アレイ。カラムワイズに記 憶されている。
N	； 省 略	A	の次数。
N9	； 省 略		$N \times (N+1)/2$ 。
ITER	； 省 略		繰返し計算の最大繰返し回数。 ((5)(vii) 参照)
IC	； 省 略		求めたい結果に対する情報。 IC = 0 固有値と固有ベクトルを 求める。 IC = 1 固有値のみ求める。

(ii) 出 力

A	； BIN FLOAT(26)	要素数 $N \times (N+1)/2$ の 1 次元 アレイ。固有値が対角線上に大き いものから求まる。
R	； BIN FLOAT(26)	要素数 $(N \times N)$ の 1 次元アレイ。 固有ベクトルが固有値に対応して カラムワイズに求まる。
ITER	； 省 略	エラー・コード。 ITER=0 入力時の ITER 回繰返 したが、収束判定基準 を満たさない場合。 ITER ≠ 0 出力時の ITER 回で解 が得られた場合。

(4) 使用されるサブプログラム

な し

(5) 注意事項

- (i) 入力の A と出力の A は同じエリアである。

(ii) 行列 A には

$$\begin{pmatrix} A(1) & A(2) & A(4) \\ & A(3) & A(5) \\ & & A(6) \end{pmatrix} \quad \left| \begin{array}{c} \\ \\ \downarrow \end{array} \right.$$

とカラムワイズにストアされなければならない。

(iii) 固有値は $A(1), A(3), A(6), \dots$ に求まる。

(iv) 丸めの誤差の観点より $N \leq 50$ が望ましい。 (DEIGNN の場合は $N \leq 70$)

(v) 入力の ITER と出力の ITER は同じエリアである。

(vi) 固有ベクトルは固有値に対応してカラムワイズに求まる。

(vii) 収束判定基準は $\sqrt{2B} \times 10^{-6}/N$ である。ここで B は非対角線上の要素の 2乗和である。

(6) 使用例

$A = \begin{pmatrix} 2 & -1 & 0 \\ & 2 & -1 \\ & & 2 \end{pmatrix}$ の固有値、固有ベクトルを求める。

```
? 0010 MFIGENN: PROC OPTIONS(MAIN);
? 0020 DCL A(6) BIN,R(9) BIN;
? 0030 N=3;
? 0040 N9=N*(N+1)/2;
? 0060 ITER=20;
? 0070 IC=0;
? 0080 A(1)=2.;
? 0090 A(2)=-1.;
? 0100 A(3)=2;
? 0110 A(4)=0.;
? 0120 A(5)=-1.;
? 0130 A(6)=2;
? 0140 CALL FIGENN(A,N,ITER,R,IC,N9);
? 0150 PUT SKIP(2);
? 0160 PUT EDIT(A)(SKIP,3 F(20,9));
? 0170 PUT EDIT(R)(SKIP,3 F(20,8));
? 0180 PUT DATA(ITER);
? 0190 PUT SKIP(5);
? 0200 END MFIGENN;
LAST LINE(FND ST) 1340.00
? BRUN;
```

3.41421353E+00	9.09494727E-13	1.09999999E+00
4.19726146E-03	-4.59454230E-12	5.85736461E-01
5.00000044E-01	-7.07106798E-01	4.99999992E-01
-7.07106783E-01	-2.06666773E-03	7.07106798E-01
5.00000029E-01	7.07106783E-01	5.00000029E-01
ITER= 14;		

[9] DEIGENN

(1) 目的

実対称行列の固有値及び固有ベクトルを Jacobi 法により求める。

(2) 使用方法

CALL DEIGENN (A, N, ITER, R, IC, N9);

(3) 引数の説明

A, R の属性が DEC FLOAT(20) となり, その他は EIGENN と同じ。 ([8] EIGENN 参照)

(4) 使用されるサブプログラム

なし

(5) 注意事項

[8] EIGENN 参照。

(6) 使用例

$$A = \begin{pmatrix} 2 & -1 & 0 \\ & 2 & -1 \\ & & 2 \end{pmatrix}$$
 の固有値, 固有ベクトルを求める。

```

? 0010  MOEIGENN: PROC OPTIONS(MAIN);
? 0020  DCL A(6) FL6AT(20),R(9) FL6AT(20);
? 0030  N=3;
? 0040  N9=N*(N+1)/2;
? 0050  ITER=20;
? 0070  IC=0;
? 0080  A(1)=2. ;
? 0090  A(2)=-1. ;
? 0100  A(3)=2;
? 0110  A(4)=0. ;
? 0120  A(5)=-1. ;
? 0130  A(6)=2. ;
? 0140  CALL DEIGENN(A,N,ITER,R,IC,N9);
? 0150  PUT SKIP(2);
? 0160  PUT FDIT(A)(SKIP,2 F(30,19));
? 0170  PUT FCIT(R)(SKIP,2 F(30,19));
? 0180  PUT DATA(ITER);
? 0190  PUT SKIP(5);
? 0200  END DEIGENN;
LAST LINE(END ST) 1370.00
? XRUN;

```

3.4142138231229183272E+00	2.5797313615306464475E-12
2.000000985993689667E+00	4.2035848654465397138E-08
7.9821831648513741461E-11	5.8578645807498081000E-01
5.0000005007915177215E-01	-7.0710630784404240993E-01
4.999998759330974664E-01	-7.0710678144978029676E-01
-3.0205232157943157092E-08	7.0710681578345372364E-01
5.0000000778894356967E-01	7.0710679358654526522E-01
5.000000958185509603E-01	
ITER= 14;	

[10] TRACE

(1) 目 的

与えられた固有値 $\lambda_1, \lambda_2, \dots, \lambda_n$ の中から、指定された値以上の固有値の個数を見つけ出す。

(2) 使用方法

CALL TRACE (M, R, CON, K, D, MM);

(3) 引数の説明

(1) 入 力

M ; 省 略 固有値の個数。

MM ; 省 略 $M(M+1)/2$ 。

R ; BIN FLOAT(26) M個の固有値を下降順に入れる。
 要素の数は $(M(M+1)/2)$ の
 1次元アレイである。ここで R
 は次のように入力する。

$$R = \begin{pmatrix} R(1) & R(2) & R(4) \\ & R(3) & R(5) \\ & & R(6) \end{pmatrix}$$

$(M \times M)$ 行列の対角要素 $R(1)$,
 $R(3)$, $R(6)$, ……に固有値 λ_1 ,
 λ_2 , λ_3 , ……, λ_M をストアす
 る。ここで $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_M$ である。

CON ; BIN FLOAT(26) 基準値以上の固有値の個数を決
 めるために与える基準値。

(ii) 出 力

K ; 省 略 CON より大きな固有値の個数。

D ; BIN FLOAT(26) CON より大きな固有値の累積
 パーセントを D(1)から D(K)まで
 出力し, CON より小さい固有
 値はそのまま D(K+1)から
 D(M)までに出力する。すなわち

$$D(j) \begin{cases} = \sum_{k=1}^j \frac{\lambda_k}{M} : \lambda_k \geq CON, (j = 1, 2, \dots, K) \\ = \lambda_j : \lambda_j < CON, (j = K+1, \dots, M) \end{cases}$$

(4) 使用されるサブプログラム

な し

(5) 注意事項

K個の基準値(CON)以上の固有値に対する累積パーセントは

$$d_j = \sum_{i=1}^j \frac{\lambda_i}{M}$$

ただし

$$j = 1, 2, \dots, K$$

M : 全固有値の個数。

$$K \leq M$$

(6) 使用例

R(I)=-I, CON=-8.0, M=4 とする。

```
? 0010  MTRACE: PROC OPTIONS(MAIN);
? 0020  DCL R(10) BIN,D(10) BIN,CON BIN;
? 0030  M=4;
? 0040  MM=M*(M+1)/2;
? 0050  DO I=1 TO 10;
? 0060  R(I)=-I;
? 0070  END;
? 0080  CON=-8.0;
? 0090  CALL TRACE(M,R,CON,K,D,MM);
? 0100  PUT DATA(K);
? 0110  PUT FDIT(D)(SKIP,3 F(20,8));
? 0120  END MTRACE;
LAST LINE(END ST)    0330.00
? %RUN;
K=            3:
-2.50000003E-01      -1.00000001E+00      -2.50000003E+00
-1.00000001E+01      3.60351502E-01      0.00000000E-77
0.00000000E-77      0.00000000E-77      0.00000000E-77
1.47460903E-01
```

[11] SIMQ

(1) 目的

実係数連立1次方程式, $AX=B$ の解を掃き出し法により求める。

(2) 使用方法

```
CALL SIMQ ( A, B, N, IER );
```

(3) 引数の説明

(i) 入力

A ; BIN FLOAT(26) 要素数(N×N)の1次元アレイ。
 連立方程式の係数行列。(プログラム実行中書きかえられる)
 B ; BIN FLOAT(26) 要素数Nの1次元アレイ。
 定数項。

N ; 省略 Aの次数。

(ii) 出力

B ; BIN FLOAT(26) 要素数Nの1次元アレイ。解。
 入力のBと同じエリアである。

I ER ; 省略 エラー・コード。

I ER=0 正常に解が得られた場合。

I ER=1 行列が特異行列になり、解が得られなかった場合。

(4) 使用されるサブプログラム

なし

(5) 注意事項

- (i) I ER=1 の時Bには計算途中の結果が入っている。
- (ii) 入力のBと出力のBは同じエリアである。

(6) 使用例

連立方程式

$$\left\{ \begin{array}{l} 4x_1 - 5x_2 + 6x_3 = 1 \\ 10x_1 + 4x_2 + 9x_3 = 76 \\ x_1 - 10x_2 - 9x_3 = 16 \end{array} \right\}$$

を解く。

A(1), A(2), ……, A(9)には, 4, 10, 1, ……, 9, -9
 B(1), ……, B(3)には, 1, ……, 16と入れる。

```

? 0010  MSIMQ: PROC OPTIONS(MAIN);
? 0020  DCL A(9) BIM FLMT,B(3) BIM FLMT;
? 0030  N=3;
? 0050  A(1)=4;
? 0060  A(2)=10;
? 0070  A(3)=1;
? 0080  A(4)=-5;
? 0090  A(5)=4;
? 0100  A(6)=-10;
? 0110  A(7)=6;
? 0120  A(8)=9;
? 0130  A(9)=-9;
? 0140  B(1)=1;
? 0150  B(2)=76;
? 0160  B(3)=16;
? 0170  CALL SIMQ(A,B,N,IER);
? 0180  PUT DATA(B);
? 0185  PUT DATA(IER);
? 0190  END MSIMQ;
LAST LINE(END ST) 0800.00
? 3RUN:
B(1)= 1.0000000E+01      B(2)= 3.0000000E+00      B(3)=-4.0000000E+00;
IER=          0;

```

[12] DSIMQ

(1) 目的

実係数連立1次方程式、 $AX = B$ の解を掃き出し法により求める。

(2) 使用方法

CALL DSIMQ (A, B, N, IER) ;

(3) 引数の説明

A, B の属性が DEC FLOAT(20) になる。

その他は SIMQ と同じ。（[11] SIMQ 参照）

(4) 使用されるサブプログラム

なし

(5) 注意事項

[11] SIMQ 参照。

(6) 使用例

```

? 0010  MNSIMQ: PROC OPTIONS(MAIN);
? 0020  DCL A(9)  FLOAT(20),B(3) FLOAT(20);
? 0030  N=3;
? 0050  A(1)=4;
? 0060  A(2)=10;
? 0070  A(3)=1;
? 0080  A(4)=-5;
? 0090  A(5)=4;
? 0100  A(6)=-10;
? 0110  A(7)=6;
? 0120  A(8)=9;
? 0130  A(9)=-9;
? 0140  B(1)=1;
? 0150  B(2)=76;
? 0160  B(3)=16;
? 0170  CALL DSIMO(A,B,N,IFR);
? 0180  PUT EDIT(B)(SKIP,2 F(30,19));
? 0185  PUT DATA(IER);
? 0190  END MNSIMQ;
LAST LINE(END ST)    0810.00
? %RUN;

1.00000000000000000000E+01      3.00000000000000000000E+00
-4.00000000000000000000E+00
IER=          0;

```

(13) SIMQG

(1) 目的

実係数連立 1 次方程式, $AX = B$ の解をガウス・ザイデル (Gauss-Seidel) 法によって求める。

(2) 使用方法

```
CALL SIMQG ( A, B, N, M, N1, M1, EPS, ITER,
IER );
```

(3) 引数の説明

(i) 入力

A ; BIN FLOAT(26) 要素数(N, N)の2次元アレイ。連立方程式の係数行列。
(プログラム実行中書きかえられる)

B ; BIN FLOAT(26) 要素数(N, M)の2次元アレイ。

		連立方程式の右辺の定数行列。
N	; 省 略	Aの次数。50以下の数。
M	; 省 略	Bの列の数。
N1	; 省 略	メインプログラムにおけるAのディメンジョンの行の数。
M1	; 省 略	メインプログラムにおけるBのディメンジョンの行の数。
EPS	; BIN FLOAT(26)	解の相対誤差(5)注意事項(VI)参照)
ITER	; 省 略	最大繰り返し回数。
(ii) 出 力		
B	; BIN FLOAT(26)	要素数(N, M)の2次元アレイ。 連立方程式の解の行列。
IER	; 省 略	エラー・コード。 IER=0 正常に解が得られた場合。 IER=1 EPSを判定基準としてITER回繰り返したが収束しなかった場合。 IER=2 N>50の場合。

(4) 使用されるサブプログラム
な し

(5) 注意事項

- (i) $N \leq 50$ であること。
- (ii) IER = 1 のとき Bには入力した定数行列が入っている。
- (iii) IER = 2 のとき エラー・コード以外の引数はこのサブプログラムを呼び出したときの状態のままである。

(IV) 係数行列の対角要素の絶対値が他の要素の絶対値よりかなり大きいこと。

(V) 入力の B と出力の B は同じエリアである。

(VI) 収束は $\frac{\sum_{i=1}^N (x_i(n+1) - x_i(n))^2}{\sum_{i=1}^N (x_i(n+1))^2} \leq EPS$ かどうかで判定される。

($x_i(n)$, $x_i(n+1)$ は近似根)

(6) 使用例

$$\begin{cases} x_1 + x_4 = 5 \\ x_2 + x_4 = 6 \\ x_3 + x_4 = 7 \\ x_4 = 4 \end{cases} \quad \begin{cases} x_1 + x_4 = 2 \\ x_2 + x_4 = 2 \\ x_3 + x_4 = 2 \\ x_4 = 4 \end{cases} \quad \begin{cases} x_1 + x_4 = 2 \\ x_2 + x_4 = 2 \\ x_3 + x_4 = 2 \\ x_4 = 2 \end{cases}$$

とすると、

$A(1, 1), A(1, 2), \dots, A(4, 4)$ には $1, 0, 0, 1, \dots, 0, 1$

$B(1, 1), B(2, 1), B(3, 1), B(4, 1)$ には $5, 6, 7, 4, \dots$

$B(3, 1), B(3, 2), B(3, 3), B(3, 4)$ には $2, 2, 2, 2$

と与える。

```

? 0010  MSIMQG: PROC OPTIONS(MAIN);
? 0020  DCL A(4,4) BIN,B(4,3) 'BIN,EPS BIN,IX FIXED(12),X BIN;
? 0030  N=4;
? 0040  N1=4;
? 0050  M=3;
? 0060  IX=0;
? 0070  M1=3;
? 0080  EPS=1.0E-5;
? 0090  ITER=50;
? 0100  GET LIST(((A(I,J) DD J=1 TO N) DD I=1 TO N));
? 0110  GET LIST(((B(I,J) DD J=1 TO M) DD I=1 TO N));
? 0200  PUT EDIT(A)(SKIP,3 E(20,8));
? 0210  PUT EDIT(B)(SKIP,3 E(20,8));
? 0220  CALL SIMQG(A,B,N,M,N1,M1,EPS,ITER,IER);
? 0230  PUT DATA(IER);
? 0240  PUT EDIT(B)(SKIP,3 E(20,8));
? 0250  END MSIMQG;
LAST LINE(END ST)    0870.00
? $RUN:
DATA 0100.00      A? 1,0,0,1,0,1,0,1,0,0,1,1,0,0,0,1
DATA 0110.00      B? 5,2,2,6,2,2,7,2,2,4,1,2
               1.00000002E+00      0.00000000E-01      0.00000000E-01
               1.00000002E+00      0.00000000E-01      1.00000002E+00
               0.00000000E-01      1.00000002E+00      0.00000000E-01
               0.00000000E-01      1.00000002E+00      1.00000002E+00
               0.00000000E-01      0.00000000E-01      0.00000000E-01
               1.00000002E+00
               5.00000007E+00      .2.00000004E+00      2.00000004E+00
               6.00000008E+00      2.00000004E+00      2.00000004E+00
               7.00000010E+00      2.00000004E+00      2.00000004E+00
               4.00000009E+00      1.00000002E+00      2.00000004E+00
IER=          0:
               1.00000002E+00      1.00000002E+00      0.00000000E-01
               2.00000004E+00      1.00000002E+00      0.00000000E-01
               3.00000004E+00      1.00000002E+00      0.00000000E-01
               4.00000009E+00      1.00000002E+00      2.00000004E+00

```

(14) POLRTN

(1) 目的

与えられた 36 次までの実数係数の代数方程式 (n 次方程式) $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$ の実根、複素根をニュートン・ラブソン (Newton-Raphson) 法により求める。

(2) 使用方法

```
CALL POLRTN(XCOF, COF, N, ROOTR, ROOTI, IER,
```

N9) ;

(3) 引数の説明

(i) 入 力

XCOF : DEC FLOAT(20) 要素数(N+1)の1次元アレイ。

方程式の係数 a_0, a_1, \dots, a_n 。

次数の低い方から高い方へと入
れる。

(実行中書きかえられる)

N : 省 略 方程式の次数 n。

COF : DEC FLOAT(20) 要素数(N+1)の1次元アレイ。

ワーク・エリア。

N9 : 省 略 N + 1。

(ii) 出 力

ROOTR : DEC FLOAT(20) 要素数Nの1次元アレイ。方程
式の根の実数部。ROOTR(1)に
は1番目に求まった根の実数部
が入る。

ROOTI : DEC FLOAT(20) 要素数Nの1次元アレイ。方程
式の根の虚数部。ROOT I(1)に
は1番目に求まった根の虚数部
が入る。

IER : 省 略 エラー・コード。

IER=0 正常に解が求まつた
場合。

IER=1 $N < 1$ の場合。

IER=2 $N > 36$ の場合。

IER=3 5組の初期値につい
て絶対誤差 10^{-5} を

収束判定基準として
500回反復しても根
が求まらなかった場
合。

I ER=4 最高次の項の係数が
0の場合。

I ER=5 オーバ・フローが生
じた場合。

(4) 使用されるサブプログラム
な し

(5) 注意事項

- (i) $N \leq 36$ であること。
- (ii) 高次の方程式の場合、計算中オーバ・フローが起こりやすい。
- (iii) 最高次の項の係数は0でないこと。
- (iv) I ER = 1, 2, 4 のとき エラー・コード以外の引数はこのサブ
プログラムを呼び出したときの状態の
ままである。
- (v) I ER = 3, 5 のとき ROOTR, ROOTI には計算途中の結
果が入る。

(6) 使用例

$$x^2 - 4 = 0 \text{ の場合。}$$

```

? 0010  MPOLRTN: PROC OPTIONS(MAIN);
? 0020  DCL XCDF(3) FLOAT(20),CDF(3) FLOAT(20),ROOTR(2) FLOAT(20);
? 0030  DCL ROOTI(2) FLOAT(20);
? 0040  IER=1E-5;
? 0050  N=2;
? 0060  N9=N+1; ,
? 0070  XCDF(1)=-4.;
? 0080  XCDF(2)=0;
? 0090  XCDF(3)=1;
? 0100  CALL POLRTN(XCDF,CDF,N,ROOTR,ROOTI,IER,N9);
? 0110  PUT EDIT(ROOTR)(2 E(30,19));
? 0115  PUT EDIT(ROOTI)(2 E(30,19));
? 0120  PUT DATA(IER);
? 0130  END MPOLRTN;
LAST LINE(END ST) 1450.00
? %RUN:
      -2.00000000000000000000E+00      2.00000000000000000000E+00
      0.00000000000000000000E+00      0.00000000000000000000E+00
IER=           5;

```

(15) RTMIT

(1) 目的

非線型方程式 $f(x)=0$ の 1 つの実根を 2 つの初期値 x_l , x_r を与えて、
2 分法および放物線逆補間法により求める。

(2) 使用方法

```
CALL RTMIT(X, XL, XR, EPS, ITEND, IER);
```

(3) 引数の説明

(i) 入力

XL ; BIN FLOAT(26) 指定区間の下限値 x_l 。（実行
中こわされる）

XR ; BIN FLOAT(26) 指定区間の上限値 x_r 。（実行
中こわされる）

EPS ; BIN FLOAT(26) 最大繰返し回数。

ITEND ; 省略 収束判定基準。

(ii) 出力

X ; BIN FLOAT(26) 結果が求まる。

IER ; 省略 エラー・コード。

I ER=0 正常に解が得られた場合。

I ER=1 FUNCT(XL)・FUNCT(XR)>0 である。

I ER=2 EPS を収束判定基準として ITEND 回繰返
したが、2 分法が収束しなかった場合。

(4) 使用されるサブプログラム

FUNCT(X) ; ユーザ作成の関数型サブプログラム。

引数は次の通りである。

X ; 省略 独立変数X。

(5) 注意事項

(i) I ER=1 のとき X には $0.5(XL+XR)$ が入っている。

(ii) I ER=2 のとき X には ITEND 回繰返し計算された近似根が
入っている。

(iii) $f(XL) \cdot f(XR) \leq 0$ となるように XL と XR を与えなければな
らない。

(iv) ESP は $|x| > 1$ のときは相対誤差として、 $|x| \leq 1$ のときは絶
対誤差として使われる。(x は近似値)

(6) 使用例

$$f(x) = x^2 - 2$$

$$xr = 1.0$$

$$xl = -3.0 \text{ とすると}$$

```
? 0010 MRTMIT: PROC OPTIONS(MAIN);
? 0020 DCL X BIN,XL BIN,XR BIN,EPS BIN;
? 0030 XL=-3.0;
? 0040 XR=1.0;
? 0050 EPS=1.0E-5;
? 0060 ITEND=50;
? 0070 CALL RTMIT(X,XL,XR,EPS,ITEND,IER);
? 0080 PUT DATA(X,IER);
? 0090 FUNCT: PROC(X);
? 0100 RRETURN(X**2-2);
? 0110 END FUNCT;
? 0120 END MRTMIT;
LAST LINE(END ST) 0760.00
? %RUN;
X=-1.4142135E+00      IER=          0;
```

(16) RTNIT

(1) 目 的

非線型方程式 $f(x)=0$ の 1 つの実根を初期値 x_0 を与えてニュートン・ラプソン (Newton Raphson) 法により求める。

(2) 使用方法

```
CALL RTNIT(X, Y, EPS, ITEND, RTORD, IER);
```

(3) 引数の説明

(i) 入 力

Y : BIN FLOAT(26) 根 x の第 1 近似値 x_0 。

(プログラム実行中書きかえられる)

EPS : BIN FLOAT(26) 根の相対誤差。

ITEND : BIN FIXED(25,0) 最大繰返し回数。

RTORD : BIN FLORT(26) 根 x の重根度について何もわからない場合は 1. とおく。

(ii) 出 力

X : BIN FLOAT(26) 結果が求まる。

IER : エラー・コード。

IER=0 解が正常に得られた場合。

IER=1 EPS を収束判定基準として ITEND 回繰返したが収束しなかった場合。

IER=2 ITEND ≤ 0 または $f'(x)=0$ の場合。 (5)
注意事項(iii)参照)

(4) 使用されるサブプログラム

FUNCT(X, VAL, DER) : ユーザ作成のサブルーチン型サブプログラム。

引数は次の通りである。

X : 省 略 独立変数。

VAL	；	省略	$f(X)$
DER	；	省略	$f'(X)$

(5) 注意事項

- (i) IER=1 のとき Xには ITEND 回繰返し計算された近似根が入っている。
- (ii) IER=2 のとき Xには繰返し計算途中の近似根が入っている。
- (iii) ITEND ≤ 0 のときでも 1度は計算を行ない、収束した場合は IER = 0 となる。

(6) 使用例

$$f(X) = x^2$$

$x_0 = 1.0$ とする。

```
? 0010 MRTNIT: PROC OPTIONS(MAIN);
? 0020 DCL X BIN,Y BIN,EPS BIN,RTORD BIN;
? 0030 Y=1;
? 0040 EPS=1.0E-8;
? 0050 RTEND=1;
? 0060 ITEND=100;
? 0070 CALL RTNIT(X,Y,EPS,ITEND,RTORD,IER);
? 0075 PUT DATA(X,IER);
? 0080 FUNCT: PROC(X,VAL,DER);
? 0090 VAL=X**2-2;
? 0100 DER=X+X;
? 0110 RETURN;
? 0120 END FUNCT;
? 0130 END MRTNIT;
LAST LINE(END ST) 0430.00
? %RUN:
X= 1.4142135E+00      IER=          0;
```

[17] DRTNIT

(1) 目的

非線型方程式 $f(X)=0$ の 1 つの実根を 2 つの初期値 x_1, x_r を与えて、2 分法および放物線逆補間法により求める。

(2) 使用方法

CALL DRTNIT(X, Y, EPS, ITEND, RTORD, IER);

(3) 引数の説明

X, Y, EPS, RTORDの属性がDEC FLOAT(20)となる。

その他は、RTNITと同じ。(〔16〕RTNIT 参照)

(4) 使用されるサブプログラム

〔16〕RTNIT 参照。

(5) 注意事項

〔16〕RTNIT 参照。

(6) 使用例

$$f(X) = x^2 - 2$$

$x_0 = 1$ とする。

```
? 0010  MDRTNIT: PROC OPTIONS(MAIN);
? 0020  DCL X FLOAT(20),Y FLOAT(20),EPS FLOAT(20),
?           RTORD FLOAT(20);
? 0030  Y=1;
? 0040  EPS=1.0E-8;
? 0050  ITEND=100;
? 0060  RTORD=1;
? 0070  CALL DRNTNIT(X,Y,EPS,ITEND,RTORD,IER);
? 0080  PUT DATA(X,IER);
? 0090  FUNCT: PROC(X,VAL,DER);
? 0100  DCL X FLOAT(20),VAL FLOAT(20),DER FLOAT(20);
? 0110  VAL=X**2-2;
? 0120  DER=X+X;
? 0130  RETURN;
? 0140  END FUNCT;
? 0150  END MDRTNIT;
LAST LINE(END ST) 0490.00
? %RUN;
X = 1.4142135623730950512E+00    IER=          0:
```

〔18〕ATKIN

(1) 目的

n 個の変数値 x_i ($i = 1, 2, \dots, n$) に対する関数値 y_i ($i = 1, 2, \dots, n$) を用いて、 m 個の変数値 X_j ($j = 1, 2, \dots, m$) に対する関数値 $Y_j = f(X_j)$ ($j = 1, 2, \dots, m$) をエイトキン・ラグランジュ (Atkin-Lagrange) 法により補間する。

(2) 使用方法

```
CALL ATKIN(ARG, VAL, X, M, EPS, Y, WORK, IER1,
```

IER2, IER3);

(3) 引数の説明

(i) 入 力

ARG ; DEC FLOAT(20) 要素数Nの1次元アレイ。既知
関数値に対する変数値。 $I \neq J$
に対して、 $ARG(I) \neq ARG(J)$ と
する。

VAL ; DEC FLOAT(20) 要素数Nの1次元アレイ。 ARG
に対する関数値。 $V A L(I)$ と
 $ARG(I)$ は対応させて入力。

X ; DEC FLOAT(20) 要素数Mの1次元アレイ。求め
ようとする補間値の変数値。
ただし、 $ARGMIN - \Delta \leq X(I)$
 $\leq ARGMAX + \Delta$ を満足するも
の。

ここで、

$$ARGMIN = \min_I (ARG(I)),$$

$$ARGMAX = \max_I (ARG(I)),$$

$$\Delta = \min_I \{ ARG(I+1) - ARG(I) \} / 2.$$

N ; 省 略 既知点の個数。 $N > 2$ 。

M ; 省 略 求めようとする補間値の個数。

EPS ; DEC FLOAT(20) 解の絶対誤差。

WORK ; DEC FLOAT(20) 要素数Nの1次元アレイ。ワー
ク・エリア。

(ii) 出 力

Y ; DEC FLOAT(20) 要素数Mの1次元アレイ。Xに

		対する補間値。 $X(I)$ に $Y(I)$ が対応する。
I ER 1 ; 省 略		エラー・コード。
	I ER 1=0	正常に補間値が求められた場合。
	I ER 1=1	$J \neq J$ に対して, $\text{ARG}(I) = \text{ARG}(J)$ の場合。
I ER 2 ; 省 略		エラー・コード。
	I ER 2=0	正常に補間値が求められた場合。
	I ER 2=1	$\text{ARGMIN} - \triangle \leq X(I) \leq \text{ARGMAX} + \triangle$ にあてはまらない場合。
I ER 3 ; 省 略		エラー・コード。
	I ER 3=0	正常に補間値が求められた場合。
	I ER 3=1	補正量が増加しあじめ、所望の精度で得られなかった場合。
	I ER 3=3	$N \leq 2$ で精度のチェックができない場合、または補間値が所望の精度に到達する以前にエイトキン・ラグラ

ンジュ法の計算を
終了した場合。

(4) 使用されるサブプログラム

な し

(5) 注意事項

- (i) $f(x)$ は一価関数であること。
- (ii) ARG および VAL に入力したデータの順序は、演算実行中に変わる。
- (iii) 3 個のエラー・コードの値がすべて 0 でないとき、Y には次の値が入る。

I ER 1=1 ; $1 < J$ に対して ARG(I)=ARG(J) のとき、ARG(J) を取除いて計算した結果が入る。

ただし、取除了後の計算中に他の理由 (I ER 2 ,
I ER 3 の項に記した) によりエラーとなった場合、
Y にはそのエラーに対する値が入る。

I ER 2=1 ; Y(I)=-1 . E 75

I ER 3=1, 2 ; $N < 1$ のとき、このルーチンでは Y(I) に何も入れない。

$N = 1$ のとき、Y(I)=VAL(1)。

$N > 1$ のとき、Y(I) には計算結果が入る。

エラー・コードに 0 以外の値がセットされてもすべて X(I) について計算が続行される。

(6) 使用例

VAL に e^x の $x = 0.1$ から 0.1 ずつ 1.0 までの値を与えておき、 $e^{0.25}$,
 $e^{0.75}$ を求める。

```

? 0010  MATKIN: PROC OPTIONS(MAIN);
? 0020  DCL ARG(10) FLOAT(20),VAL(10) FLOAT(20),X(2) FLOAT(20);
? 0030  DCL Y(2) FLOAT(20),WORK(10) FLOAT(20),EPS FLOAT(20);
? 0040  N=10;
? 0050  M=2 ;
? 0060  EPS=1.E-5;
? 0070  Z=0;
? 0080  DO I=1 TO 10;
? 0090  Z=Z+0.1;
? 0100  ARG(I)=Z;
? 0110  VAL(I)=EXP(Z);
? 0120  END;
? 0130  X(1)=0.25;
? 0140  X(2)=0.75;
? 0150  CALL ATKIN(ARG,VAL,X,N,M,FPS,Y,WORK,IER1,IER2,IER3);
? 0160  PUT DATA(Y);
? 0165  PUT DATA(IER1,IER2,IER3);
? 0170  END MATKIN;
LAST LINE(END ST) 1270.00
? %RUN:
Y(1)= 1.2840255841220698845E+00      Y(2)= 2.1169998141704127593E+00;
IER1=          0      IER2=          0      IER3=          0;

```

(19) LESQ

(1) 目的

最小二乗法により係数を求める。

(2) 使用方法

CALL LESQ (AMN, BM, A, B, M1, N1, IER);

(3) 引数の説明

(i) 入力

AMN : BIN FLOAT(26) 要素数 ($M_1 \times N_1$) の 1 次元ア
レイ。第 1 列にはすべて 1, 2
列目以降には独立変数のデータ
がカラムワイズに記憶されてい
る。

BM : BIN FLOAT(26) 要素数 M_1 の 1 次元アレイ。従
属変数のデータ。

M1 : 省略 標本数 ($M_1 \geq N_1$)。

N1 : 省略 独立変数の数 + 1。

A ; BIN FLOAT(26) 要素数(M1×N1)の1次元アレイ。ワーク・エリア。

(ii) 出 力

B ; BIN FLOAT(26) 要素数N1の1次元アレイ。
係数が求まる。B(1)には定数項
が求まる。

I ER ; 省 略 エラー・コード。

I ER=0 正常に解が得られた
場合。

I ER=2 行列が特異行列にな
り解が得られなかっ
た場合。

I ER=2 M1<N1, M1=0,
N1=0 の場合。

(4) 使用されるサブプログラム

SIMQ ([11] SIMQ 参照)

(5) 注意事項

(i) M1 ≥ N1 でなければならない。

(ii) AMN にはデータを次の順序で入れなければならない。

AMN(1)=1, AMN(2)=1, …,
AMN(M1)=1

変数1に関するデータ AMN(M1+1), AMN(M1+2),
……, AMN(M1+M1)

変数2に関するデータ AMN(2M1+1), AMN(2M1
+2), ……, AMN(2M1+M1)

⋮ ⋮

変数(N1-1)に関するデータ AMN((N1-1)M1+1), AMN
((N1-1)M1+2), ……,

AMN (N1×M1)

(III) IER=1のときBには演算途中の値が入る。

(IV) IER=2のときBにはこのサブプログラムを呼び出したときの状態のままの値が入る。

(V) B(1)には定数項が求まる。

(6) 使用例

```
? 0010 MLESQ: PROC OPTIONS(MAIN);
? 0020 DCL AMN(15) BIN,BM(5) BIN,A(9) BIN,B(3) BIN;
? 0030 M1=5;
? 0040 N1=3;
? 0050 L1=M1*M1;
? 0060 AMN=1.0;
? 0070 MM=M1+1;
? 0080 GET LIST((AMN(K) D0 K=MM TD L1));
? 0090 GET LIST((BM(K) D0 K=1 TD M1));
? 0100 CALL LESQ(AMN,BM,A,B,M1,N1,IER);
? 0110 PUT SKIP DATA(S);
? 0120 PUT SKIP DATA(IER);
? 0130 END MLESQ;
LAST LINE(END ST) 1170.00
? %RUN;
DATA 0080.00      AMN? 2,4,3,4,30,3,6,2,5,3
DATA 0090.00      BM? 3,8,10,9,2
B(1)= 6.8626934E+00    B(2)=-1.8438420E-01    B(3)= 2.9552916E-01;
IER=          0;
```

[20] RKG 2

(1) 目的

与えられた1階常微分方程式 $\frac{dy}{dx} = f(x, y)$ を与えられた初期条件 (x_0, y_0) のもとに、きざみ幅 h でルンゲ・クッタ・ギル (Runge-Kutta-Gill) 法により解き、与えられた k に対し $h \times k$ ごとの各点における解を求める。

(2) 使用方法

```
CALL RKG2(H, XI, YI, K, N, VEC);
```

(3) 引数の説明

(i) 入力

H ; BIN FLOAT(26) 計算のきざみ幅。
 XI ; BIN FLOAT(26) 初期条件を与える点X。
 YI ; BIN FLOAT(26) 初期値Y。
 K ; 省略 解を格納する点から点までのき
ざみを個数K。
 N ; 省略 解を得る点の総数。

(ii) 出力

VEC ; BIN FLOAT(26) 要素数Nの1次元アレイ。
 間隔 $h \times K$ ごとの解。

(4) 使用されるサブプログラム

FUN(X, Y); ユーザ作成の関数型サブプログラム。

X ; 省略 微分方程式の独立変数X。
 Y ; 省略 微分方程式の従属変数Y。

(5) 注意事項

(i) VEC(1)には $X = X_0 + h \cdot k$ における解が入っている。

(6) 使用例

$\frac{dy}{dx} = \frac{3y}{1+x}$ を $y(0)=1$ のもとにきざみ幅を 0.1 とし $X = 1.0, 2.0, \dots, 10.0$ における解を求める。

```

? 0010  MRKG2: PROC OPTIONS(MAIN);
? 0020  DCL H BIN,XI BIN,YI BIN,VEC(10) BIN;
? 0030  H=0.1;
? 0040  K=10;
? 0050  XI=0.0;
? 0060  YI=1.0;
? 0070  N=10;
? 0080  CALL RKG2(H,XI,YI,K,N,VEC);
? 0090  PUT EDIT(VEC)(SKIP,3 E(20,8));
? 0110  FUN: PROC(X,Y);
? 0130  Z=3*Y/(1+X);
? 0140  RETURN(Z);
? 0150  END FUN;
? 0160  END MRKG2;
LAST LINE(END ST) 0490.00
? %RUN;

```

7.99969419E+00	2.69989073E+01	6.39973819E+01
1.24994870E+02	2.15991118E+02	3.42985875E+02
5.11978894E+02	7.28969953E+02	9.99958783E+02
1.33094530E+03		

[21] DRKG 2

(1) 目的

与えられた1階常微分方程式 $\frac{dy}{dx} = f(x, y)$ を与えられた初期条件 (x_0, y_0) のもとに、きざみ幅 h でルンゲ・クッタ・ギル (Runge-Kutta-Gill) 法により解き、与えられた k に対し $h \times k$ ごとの各点における解を求める。

(2) 使用方法

CALL DRKG 2 (H, XI, YI, K, N, VEC);

(3) 引数の説明

H, XI, YI, VEC の属性が DEC FLOAT(20) となり、その他は、
RKG 2 と同じ。 ([20] RKG 2 参照)

(4) 使用されるサブプログラム

[20] RKG 2 参照。ただし、引数 X, Y の属性は DEC FLOAT(20) になる。

(5) 注意事項

[20] RKG 2 参照。

(6) 使用例

$\frac{dy}{dx} = \frac{3y}{1+x}$ を $y(0)=1$ のもとにきざみ幅を 0.1 とし $X = 1, 2, \dots$

…， 10 における解を求める。

```
? 0010 MDRKG2: PROC OPTIONS(MAIN);
? 0020 DCL H FLAT(20),XI FLAT(20),YI FLAT(20);
? 0030 DCL VEC(10) FLAT(20);
? 0040 H=0.1;
? 0050 K=10;
? 0060 XI=0.0;
? 0070 YI=1.0;
? 0080 N=10;
? 0090 CALL DPKG2(H,XI,YI,K,N,VEC);
? 0100 PUT EDIT(VEC)(SKIP,2 F(30,19));
? 0110 FUM: PROC(X,Y);
? 0120 DCL X FLAT(20),Y FLAT(20);
? 0130 Z=3*Y/(1+X);
? 0140 RETURN(Z);
? 0150 END FUN;
? 0160 END MDRKG2;
LAST LINE(END ST) 0520.00
? %RUN;
```

7.9996932796947659905E+00	2.6998904210062078226E+01
6.3907377911812314906F+01	1.2499486524994796152E+02
2.1599111920100020000F+02	3.4298589148358120306E+02
5.1197893593760778376E+02	7.2897000373835968683E+02
9.0095884606501070000F+02	1.3309452219584149703E+03

[22] RKG 3

(1) 目 的

与えられた n 元連立 1 階常微分方程式

$$\frac{dy_1}{dx} = f_1(x, y_1, y_2, \dots, y_n)$$

$$\frac{dy_2}{dx} = f_2(x, y_1, y_2, \dots, y_n)$$

$$\frac{dy_n}{dx} = f_n(x, y_1, y_2, \dots, y_n)$$

を与えられた初期条件 $\begin{pmatrix} y_{10} \\ y_{20} \\ \vdots \\ y_{n0} \end{pmatrix}$ のもとにきざみ幅 h でルンゲ・ク

ッタ・ギル (Runge-Kutta-Gill) 法により解き, h ごとの各点における解を求める。

(2) 使用方法

CALL RKG3(XI, YI, Y4, H, NX, N, IER);

(3) 引数の説明

(i) 入 力

XI ; 省 略 初期値を与える点 x_0 。

YI ; BIN FLOAT(26) 要素数Nの1次元アレイ。初期
条件 $y_{10}, y_{20}, \dots, y_{n0}$ 。

H ; 省 略 計算のきざみ幅。

NX ; 省 略 解を得る点の総数。

N ; 省 略 方程式の元の数 n。

(ii) 出 力

Y4 ; BIN FLOAT(26) 要素数(N, NX)の2次元アレ
イ。 $Y4(1, I), Y4(2, I), \dots, Y4(N, I)$ には y_1, y_2, \dots, y_n の第1番目の解
が入る。

IER ; 省 略 エラー・コード。

IER=0 正常に解が得られた
場合。

IER=I 第I番目の解の計算
途中でオーバ・フロ
ーが起った場合。

(4) 使用されるサブプログラム

FUNC(K, X, Y) ; ユーザ作成の関数型サブプログラム。

引 数

K ; 省 略 何番目の方程式の右辺かを示す
数。

X ; 省 略 微分方程式の独立変数 x。

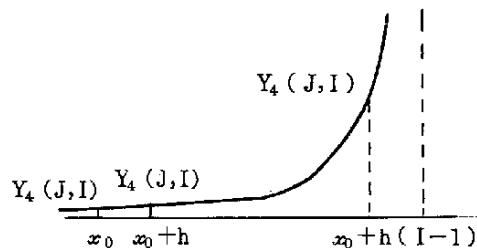
Y ; BIN FLOAT(26) 要素数Nの1次元アレイ。微分

方程式の従属変数 y_1, y_2, \dots

\dots, y_n 。

(5) 注意事項

- (i) IER = 1 のとき、オーバーフローが起こる 1 つ手前の点 $X = x_0 + h(I-1)$ までは解が求まり、それ以降は計算しない。すなわち $Y_4(J, 1), Y_4(J, 2), \dots, Y_4(J, I)$ には結果が入っている。
 $J = 1, 2, \dots, N$ 。図 15.5-1 参照。



- (ii) $Y_4(1, 1), Y_4(2, 1), \dots, Y_4(N, 1)$ には初期値 $y_{10}, y_{20}, \dots, y_{n0}$ が入っている。
(iii) 高階常微分方程式も連立微分方程式にして、このサブプログラムで解くことができる。

(6) 使用例

$y'' + y = 0$ を $y(0) = 0, y'(0) = 1$ のときにきざみ幅 0.1 で 20 点における解を求める場合。

$$y_1 = y, \quad y_2 = y' \quad \text{とおき}$$

$$\begin{cases} y_1' = y_2 \\ y_2' = -y_1 \end{cases} \quad \text{なる連立微分方程式に変形する。}$$

```

? 0010 MRKG3: PROC OPTIONS(MAIN);
? 0020 DCL YI(2) BIN,Y4 (2.20) BIN;
? 0030 XI=0.0;
? 0040 YI(1)=0.0;
? 0050 YI(2)=1.0;
? 0060 H=0.1;
? 0070 NX=20;
? 0080 N=2;
? 0090 CALL RKG3(XI,YI,Y4,H,NX,N,IER);
? 0100 PUT FDIT(((Y4(I,J) DO J=1 TO 20) DO I=1 TO 2))(SKIP,2 F(20,8));
? 0110 PUT DATA(IER);
? 0120 FUNC: PROC(K,X,Y);
? 0130 DCL Y(2) BIN;
? 0140 IK=2/K;
? 0150 Z=(-1)**IK*Y(IK);
? 0160 RETURN(Z);
? 0170 END FUN;
? 0180 END MRKG3;
LAST LINE(END ST) 0670.00
? %RUN;

```

0.00000000E+01	9.98333357E-02
1.98669165E-01	2.95519962E-01
3.89418028E-01	4.79425154E-01
5.64642041E-01	6.44217222E-01
7.17355603E-01	7.83326417E-01
8.41470509E-01	8.91206905E-01
9.32038679E-01	9.53557839E-01
9.85449443E-01	9.97494801E-01
9.99573543E-01	9.91664871E-01
9.73847851E-01	9.46300476E-01
1.00000002E+00	9.95004177E-01
9.80066597E-01	9.55336540E-01
9.21061098E-01	8.77582728E-01
8.25335845E-01	7.54842510E-01
6.96707129E-01	6.21610507E-01
5.40302962E-01	4.53596882E-01
3.62358629E-01	2.67499819E-01
1.69968243E-01	7.07384102E-02
-2.91982181E-02	-1.28843110E-01
-2.27200638E-01	-3.23288060E-01

IER= 0;

(23) RUNGE

(1) 目 的

与えられた n 元連立 1 階常微分方程式

$$\frac{dy_1}{dx} = f_1(x, y_1, y_2, \dots, y_n)$$

$$\frac{dy_2}{dx} = f_2(x, y_1, y_2, \dots, y_n)$$

$$\frac{dy_n}{dx} = f_n(x, y_1, y_2, \dots, y_n)$$

を与えられた初期条件 $\begin{pmatrix} y_{10} \\ y_{20} \\ \vdots \\ y_{n0} \end{pmatrix}$ のもとに、きざみ幅 h でルンゲ・クッタ・ギル (Runge-Kutta-Gill) 法により解き、与えられた k に対して $h \times k$ ごとの各点における解を求める。

(2) 使用方法

CALL RUNGE(Y, M, B, K, H);

(3) 引数の説明

(i) 入 力

H ; 省略	計算のきざみ幅。
K ; 省略	解を出力する点から点までのきざみの個数 k 。
M ; 省略	方程式の元の数 + 1。
B ; DEC FLOAT(20)	Y(1)の上限。印字は Y(1) > B になるまで続ける。

(ii) 出 力

Y ; DEC FLOAT(20)	要素数 M の一次元アレイ。間隔 $h \times k$ ごとの解。 Y(1) = X は独立変数。
F ; DEC FLOAT(20)	要素数 M の一次元アレイ。 $F_i = \frac{dY_i}{dY_1} (Y_1 = X, i = 1, \dots, M)$

(4) 使用されるサブプログラム

UHEN(Y, F) ; ユーザ作成サブプログラム。

方程式の右辺を定義する。

INSATU(Y, F) ; ユーザ作成サブプログラム。

結果を印字する。

(5) 注意事項

- (i) $F(1) = \frac{dY(1)}{dX(1)} = 1.0$ は RUNGE の中で定義されている。
- (ii) 最初 $Y(1) \sim Y(M)$ には初期値を与えておく。

(6) 使用例

$\frac{dy}{dx} = \frac{3y}{1+x}$ を $y(0)=1$ のもとにきざみ幅を 0.1 とし $X=1.0, 2.0 \dots$

…, 10.0 における解を求める場合。

```
? 0010  MAIN: PROC;
? 0020  DCL Y(2) DEC FLOAT(20);
? 0030  H=0.1;
? 0040  Y(1)=0;
? 0050  Y(2)=1;
? 0060  K=10;
? 0070  M=2;
? 0080  B=10;
? 0084  PUT SKIP(2);
? 0085  PUT LIST('           X                   Y                   DY/DX. ');
? 0086  PUT SKIP;
? 0090  CALL RUNGE(Y,M,B,K,H);
? 0100  UHEN: PROC(Y,F);
? 0110  DCL Y(2) DEC FLOAT(20),F(2) DEC FLOAT(20);
? 0120  F(2)=3*Y(2)/(1+Y(1));
? 0130  RETURN;
? 0140  END UHEN;
? 0150  INSATU: PROC(Y,F);
? 0160  DCL Y(2) DEC FLOAT(20),F(2) DEC FLOAT(20);
? 0170  PUT FDIT(Y(1),Y(2),F(2)) (F(10,2),2 E(20,7));
? 0180  RRETURN;
? 0190  END INSATU;
? 0210  END MAIN;
LAST LINE(END ST)    0750.00
? XRUN;
```

X	Y	DY/DX
0.00	1.0000000E+00	3.0000000E+00
0.99	7.9996932E+00	1.1999539E+01
1.99	2.6998903E+01	2.6998903E+01
2.99	6.3997376E+01	4.7998032E+01
3.99	1.2499485E+02	7.4995917E+01
4.99	2.1599111E+02	1.0799555E+02
5.99	3.4298538E+02	1.4699395E+02
6.99	5.1197892E+02	1.9199209E+02
7.99	7.2896998E+02	2.4298999E+02
8.99	9.9995381E+02	2.9998764E+02
9.99	1.3309451E+03	3.6298505E+02

(24) SMP SN

(1) 目的

与えられた関数 $f(x)$ の積分値 $Q = \int_a^b f(x) dx$ を、区間 $[a, b]$ においてシンプソン (Simpson) の公式により求める。

(2) 使用方法

CALL SMPSON(A, B, DEL, IMAX, QI1, Q, N, IER);

(3) 引数の説明

(i) 入力

A ; 省略 積分区間の下限値 a。

B ; 省略 積分区間の上限値 b。

DEL ; 省略 解の絶対誤差。((5), (i)参照)

IMAX ; DEC FIXED(5,0) きざみの回数。(IMAX ≥ 2)

(ii) 出力

QI1 ; 求められる積分値の 1 つ前の値
が入っている。

Q ; 積分値。

N ; DEC FIXED(5,0) 計算の結果得られるきざみの数。

IER ; DEC FIXED(1,0) エラー・コード。
IER=0 正常に解が求められた場合。

IER=1 A=B の場合。

IER=2 DEL=0 の場合。

IER=3 IMAX ≤ 1 の場合。

IER=4 DEL を収束判定基準として、最大繰返し回数まで繰返したが満たさなかった場合。

(4) 使用されるサブプログラム

F(X) ; ユーザが作成する関数型サブプログラム。

引数は次のとおりである。

X ; 省略 独立変数x。

(5) 注意事項

- 演算方法は、収束判定基準を満たすまで、つまり $|Q_n - Q_{n-1}| < |DEL \times Q_n|$ を満たすまで、きざみ幅を2等分していくシンプソンの公式を使用している。
- エラー・コード IER=1, IER=2, IER=3 がセットされたとき、 $Q1 = Q = 0.0$, $N = 0$ が出力される。IERに4がセットされたとき、 $Q1$, Q および N には、最大くり返し数まで計算した結果が出力される。

(6) 使用例

IMAX=100, DEL = 10^{-3} なるときの $\int_0^1 \frac{1}{1+x^2} dx$ を求める。

```
? 0010  MSMPSN: PROC OPTIONS(MAIN);
? 0020  NCL IMAX DEC FIXED(5,0),N DEC FIXED(5,0),
?           IER DEC FIXED(1,0);
? 0030  A=0.0;
? 0040  B=1.0;
? 0050  DEL=0.001;
? 0060  IMAX=100;
? 0070  CALL SMPSON(A,B,DEL,IMAX,S11,S,N,IER);
? 0075  PUT SKIP(2);
? 0080  PUT DATA(S11,S,N,IER);
? 0085  PUT SKIP(2);
? 0090  F:PROC(X);
? 0100  RETURN(1.0/(1.0+X*X));
? 0110  END F;
? 0115  PUT SKIP(5);
? 0120  FND MSPSN;
LAST LINE(END ST) 0560.00
? %RUN;
```

S11= 7.8539218E-01 S= 7.8539815E-01 N= 8 IER= 0;

[25] DSMPSON

(1) 目的

与えられた関数 $f(x)$ の積分値 $Q = \int_a^b f(x) dx$ を、区間 $[a, b]$ においてシンプソン (Simpson) の公式により求める。

(2) 使用方法

```
CALL DSMPSON(A, B, DEL, IMAX, Q11, Q, N, IER);
```

(3) 引数の説明

A, B, DEL, Q11, Q の属性は、DEC FLOAT(20)となり、
IMAX, N, IER の属性は省略となる。

その他は、SMPSONと同じ。（[24] SMPSON 参照）

(4) 使用されるサブプログラム

SMPSONと同じ。ただし、引数Xの属性は、DEC FLOAT(20)となる。

(5) 注意事項

[24] SMPSON 参照。

(6) 使用例

IMAX=100, DEL=10⁻⁶なるときの $\int_1^2 e^x dx$ を求める。

```
? 0010  MNSMPSON: PROC OPTIONS(MAIN);
? 0020  DCL A FLOAT(20),B FLOAT(20),DEL FLOAT(20),
?           SI1 FLOAT(20),S FLOAT(20);
? 0030  A=1.0;
? 0040  B=2.0;
? 0050  IMAX=100;
? 0060  DEL=0.1E-05;
? 0070  CALL DSMPSON(A,B,DEL,IMAX,SI1,S,"",IER);
? 0080  PUT SKIP(2);
? 0085  PUT DATA(SI1,S);
? 0090  PUT DATA(N,IER);
? 0095  PUT SKIP(5);
? 0100  F:PROC(X);
? 0110  DCL X FLOAT(20);
? 0120  RETURN(EXP(X));
? 0130  END F;
? 0140  END MNSMPSON;
LAST LINE(END ST) 0600.00
? MRUN;
```

```
SI1= 4.6707745794112161198E+00      S= 4.6707744065748331802E+00;
N=          32      IER=          0;
```

(26) QUADR

(1) 目的

等間隔に与えられたデータを結んだ直線、あるいは曲線を $f(x)$ としたときの積分値 S を 2 点公式、3 点公式、4 点公式、5 点公式により求める。

(2) 使用方法

```
CALL QUADR( Z, N, H, S, IER );
```

(3) 引数の説明

(i) 入力

Z ; BIN FLOAT(26) 1 次元アレイで与えられた入力データ。 $Z(K)$ ($K = 1, 2, \dots, N$) は K 番目の分点上の間数値である。

N ; 省略 データの数で Z の要素数を与える。

H ; BIN FLOAT(26) データの間隔。

(ii) 出力

S ; BIN FLOAT(26) 積分値

IER ; 省略 エラー・コード。

IER=0 正常に解が求められた場合。

IER=1 $N \leq 1$ である場合。

IER=2 $H \leq 0$ である場合。

(4) 使用されるサブプログラム

なし

(5) 注意事項

なし

(6) 使用例

N = 6, H = 0.01 のときの積分値を求める。

```
? 0010  MQUADR: PROC OPTIONS(MAIN):  
? 0020  DCL Z(6) BIN,H BIN,S BIN:  
? 0030  N=6:  
? 0040  H=0.01:  
? 0050  Z(1)=0.01:  
? 0060  Z(2)=0.02:  
? 0070  Z(3)=0.03:  
? 0080  Z(4)=0.04:  
? 0090  Z(5)=0.05:  
? 0100  Z(6)=0.06:  
? 0110  CALL QUADR(Z,N,H,S,IER):  
? 0120  PUT SKIP(2);  
? 0130  PUT DATA(S,IER);  
? 0140  PUT SKIP(5);  
? 0150  END MQUADR:  
LAST LINE(END ST) 0440.00  
? %RUN:
```

S = 1.7500001E-03 IER = 0:

[27] GAUSS

(1) 目 的

12 個の一様乱数を用い、中心極限定理によって 1 個の正規乱数を発生させる。

(2) 使用方法

CALL GAUSS (IX, S, AN, V);

(3) 引数の説明

(i) 入 力

IX : DEC FIXED(12) ($2^{35}-1$) 以下の整数。

S : BIN FLOAT(26) 正規乱数の標準偏差。

AN : BIN FLOAT(26) 正規乱数の平均値。

(ii) 出 力

V : BIN FLOAT(26) 1 個の正規乱数。

(4) 使用されるサブプログラム

RANDUN ([29] RANDUN 参照)

(5) 注意事項

- (i) 初期値 IX は桁数の大きな素数にとるとよい。
- (ii) 複数個の乱数を発生させる場合、(6)使用例に記載したように、1番目の乱数を発生させるときだけユーザが IX に初期値を入力し、2番目以降の乱数を求めるときには、前回の RANDUN ルーチンで計算された IX を初期値として使用する。ユーザは IX に数値を入力する必要はない。

(6) 使用例

平均値 0, 標準偏差 1 の正規分布に従う乱数を 100 個発生させる場合。

```
? 0010  MGAUSS: PROC OPTIONS(MAIN);
? 0020  DCL IX FIXED(12),S BIN,AM BIN,V BIN;
? 0025  PUT SKIP(2);
? 0030  DO I=1 TO 100;
? 0040  S=1.0;
? 0050  AN=0.0;
? 0060  CALL GAUSS(IX,S,AN,V);
? 0070  VD=V;
? 0080  PUT EDIT(VD)(E(21,8));
? 0090  END;
? 0100  PUT SKIP(5);
? 0110  END MGAUSS;
LAST LINE(END ST)    0390.00
? %RUN;
```

```
3.04056130E-01
-1.11313274E-01
-2.00752507E-02
-9.70569789E-01
7.68979996E-01
2.39062026E-01
2.86043867E+00
3.17852102E-01
6.06821268E-01
-1.02840456E+00
-4.33160953F-01
6.03772133E-01
1.35939737F+00
-8.18958342E-01
-7.67033040F-01
-1.06351837F+00
2.16776445E-01
4.10003624E-01
4.04785834E-01
-4.78675179E-01
1.51549175E-01
-3.17030958E-01
4.98952589E-01
9.33813780F-01
-1.40979714E+00
3.43274436E-01
-2.09658049E+00
-1.74114128E+00
-7.97844111E-01
```

-1.14477094E+00
9.56545680E-01
1.59645827E+00
-1.42891816E-01
4.05042849E-01
3.47748517E-02
-4.33401875E-01
-7.61112272E-01
-7.66341865E-01

|

(28) NORMAL

(1) 目的

直接法によって 1 個の正規乱数を発生させる。

(2) 使用方法

CALL NORMAL (IX, AVG, STD, Y);

(3) 引数の説明

(i) 入力

IX ; DEC FIXED(12) ($2^{35} - 1$) 以下の整数。

AVG ; BIN FLOAT(26) 正規乱数の平均値。

STD ; BIN FLOAT(26) 正規乱数の標準偏差。

(ii) 出力

V ; BIN FLOAT(26) 1 個の正規乱数。

(4) 使用されるサブプログラム

RANDUN ([29] RANDUN 参照)

(5) 注意事項

(i) 初期値 IX は桁数の大きな素数にとるとよい。

(ii) 複数個の乱数を発生させる場合、(6)使用例に記載したように、1番目の乱数を発生させるときだけユーザが IX に初期値を入力し、2番目以降の乱数を求めるときには、前回 RANDUN ルーチンで計算された IX を初期値として使用する。ユーザは IX に数値を入力する必要はない。

(6) 使用例

正規分布 $N(0, 1)$ に従う乱数を 10 個発生させる場合。

```

? 0010  MNORMAL: PROC OPTIONS(MAIN);
? 0020  DCL IX FIXED(12),AVG BIN,STD BIN,V BIN;
? 0030  IX=2753;
? 0035  AVG=0;
? 0036  STD=1;
? 0040  PUT SKIP(2);
? 0050  DO I=1 TO 10;
? 0060  CALL NORMAL(IX,AVG,STD,V);
? 0070  PUT DATA(IX,V);
? 0080  END;
? 0090  PUT SKIP(5);
? 0100  END MNORMAL;
LAST LINE(END ST) 0360.00
? %RUN;

```

IX=	1083120023	V=-2.2247927E+00;
IX=	198020223	V= 6.8627233E-01;
IX=	1649876105	V= 1.6069477E-01;
IX=	2081243161	V= 1.1080876E+00;
IX=	1056805157	V=-1.3351585E+00;
IX=	2098449360	V= 9.8186933E-01;
IX=	2052141476	V= 1.0917252E+00;
IX=	1241791480	V=-1.4553558E+00;
IX=	1520093892	V=-4.6621545E-01;
IX=	1251603966	V=-7.1870854E-01;

[29] RANDUN

(1) 目 的

1 個の整数を与えて区間 [0, 1] の実数型一様乱数, および区間 [0, 2^{35} , -1] の整数型一様乱数をそれぞれ 1 個発生させる。

(2) 使用方法

CALL RANDUN (IX, X);

(3) 引数の説明

(i) 入 力

IX : DEC FIXED(12) ($2^{35} - 1$) 以下の整数。

(ii) 出 力

IX : DEC FIXED(12) 区間 [0, $2^{35} - 1$] の整数型
一様乱数。

X : 省 略 区間 [0, 1] の実数型一様乱
数。

(4) 使用されるサブプログラム

なし

(5) 注意事項

- (i) 初期値 IX は桁数の大きな素数にとるとよい。
- (ii) 亂数列の最大周期は 2^{29} である。
- (iii) 複数個の乱数を発生させる場合、(6)使用例に記載したように、1番目の乱数を発生させるとときだけユーザが IX に初期値を入力し、2番目以降の乱数を求めるときには、前回計算された IX を初期値として使用する。ユーザは IX に数値を入力する必要はない。

(6) 使用例

区間 [0, 1] の 100 個の実数型一様乱数を発生させる場合。

```
? 0010  MRANDUN: PROC OPTIONS(MAIN);
? 0020  DCL X(100);
? 0030  DCL IX FIXED(12);
? 0040  DO I=1 TO 100;
? 0050  CALL RANDUN(IX,X(I));
? 0060  END;
? 0070  PUT SKIP(2);
? 0080  PUT EDIT((X(I)) 06 I=1 TO 100))(SKIP,3 E(20,8));
? 0090  END MRANDUN;
LAST LINE(FND ST)    0270.00
? %RUN;
```

2.59034551E-01	8.65920007E-01	5.32726705E-01
3.76251250E-01	1.31681039E-01	2.43741869E-01
5.98553687E-01	6.10984608E-01	3.20742592E-01
1.49231027E-01	4.52489562E-01	7.13648661E-01
8.20791378E-01	8.46981063E-01	2.93410502E-01
8.31269010E-01	5.41605845E-01	3.06935451E-01
2.42872849E-01	6.43765583E-01	7.53563135E-01
7.74685859E-01	1.38229794E-01	4.42779295E-01
3.12300376E-01	8.54841560E-01	4.61613737E-01
7.03015014E-01	9.02920752E-01	5.23924127E-01
4.63428184E-01	6.20044022E-01	6.59344919E-02
2.80694387E-01	4.29865568E-01	9.59798410E-01
2.28700924E-01	8.29968869E-01	3.31555023E-01
7.84985274E-01	1.50793001E-01	8.22598487E-01
2.82624304E-01	9.14278790E-01	9.18053790E-01
3.28652746E-01	5.72556138E-01	7.57259830E-01
5.22380426E-02	6.29123270E-01	1.10781220E-01
4.90559466E-01	7.77179449E-01	5.63926383E-01
1.72181051E-01	5.74143186E-01	7.70932495E-01
1.45767282E-01	4.42238688E-01	8.81614208E-01
1.14894047E-01	4.10871952E-02	8.13749760E-01
3.47219049E-01	3.89481127E-01	2.04414095E-01
9.55856777E-02	5.89801773E-01	2.01569860E-02
6.87405131E-02	1.84524051E-01	5.21843671E-01

1.13816024E-01	3.88488844E-01	1.70671816E-01
6.60297319E-01	2.27151751E-01	2.99032434E-01
2.86882251F-01	9.76815760F-01	5.28393805F-01
4.02712121F-01	3.50352488E-01	7.52291217F-01
4.15584988E-01	2.47883517F-02	6.03802189E-01
5.92425167F-01	9.53457921E-01	6.79161712E-01
5.79838648E-01	4.63014449E-02	5.50458148E-01
4.76875312F-01	9.32081252E-01	6.73981785E-01
9.23727303F-02	1.64027735E-02	2.13740034E-02
8.30788180E-01		

(30) POISN

(1) 目的

平均値 μ を与えて、1個のポアソン乱数を発生させる。

(2) 使用方法

CALL POISN(IX, AVG, NP);

(3) 引数の説明

(i) 入力

IX ; DEC FIXED(12) ($2^{35} - 1$) 以下の整数。

AVG ; BIN FLOAT(26) 平均値 μ 。

(ii) 出力

NP ; 省略 1個のポアソン乱数。

(4) 使用されるサブプログラム

RANDUN ([29] RANDUN 参照), NORMAL ([28] NORMAL 参照)

(5) 注意事項

(i) 初期値 IX は桁数の大きな素数にとるとよい。

(ii) 複数個の乱数を発生させる場合、(6)使用例に記載したように、1番目の乱数を発生させるときだけユーザが IX に初期値を入力し、2番目以降の乱数を求めるときには、前回 RANDUNで計算された IX を初期値として使用する。ユーザは IX に数値を入力する必要はない。

(iii) $\mu > 0$ であること。

(6) 使用例

平均 1, 6, 11, 16 ポアソン乱数を 10 個発生させる。

```
? 0010  MP0ISN: PROC OPTIONS(MAIN);
? 0020  DCL IX FIXED(12),AVG BIN;
? 0025  PUT SKIP(2);
? 0030  IX=110001;
? 0040  DD I=1 TD 16 BY 5;
? 0050  AVG=I;
? 0060  PUT DATA(IX,AVG);
? 0070  DD I=1 TD 10;
? 0080  CALL POISN(IX,AVG,NP);
? 0090  PUT DATA(NP);
? 0100  END;
? 0110  END;
? 0120  PUT SKIP(5);
? 0130  END MP0ISN;
LAST LINE(END ST) 0730.00
? XRUN;
```

IX=	110001	AVG= 1.000000E+00:
NP=	0:	
NP=	0:	
NP=	2:	
NP=	0:	
NP=	1:	
NP=	0:	
NP=	1:	
NP=	2:	
NP=	1:	
NP=	0:	
IX=	601060627	AVG= 6.000000E+00:
NP=	8:	
NP=	6:	
NP=	9:	
NP=	10:	
NP=	5:	
NP=	12:	
NP=	8:	
NP=	6:	
NP=	4:	
NP=	12:	
IX=	1612650228	AVG= 1.100000E+01:
NP=	12:	
NP=	8:	
NP=	10:	
NP=	13:	
NP=	14:	
NP=	8:	
NP=	12:	
NP=	8:	
NP=	10:	
NP=	11:	
IX=	516514865	AVG= 1.600000E+01:
NP=	20:	
NP=	14:	
NP=	9:	
NP=	13:	
NP=	15:	
NP=	16:	
NP=	16:	
NP=	11:	
NP=	13:	
NP=	12:	

(31) MULTR

(1) 目的

従属変数Yと、KK個の全独立変数Xから選びだしたK個の独立変数とが

$$Y = b_0 + b_1 x_1 + \dots + b_K x_K \dots \quad (31-1)$$

なる関係にあるとき、上式に対して回帰分析を行なう。

このサブルーチンは、一般に重回路分析、多項式回帰分析等に用いられる。

(2) 使用方法

```
CALL MULTR ( N, K, XBAR, STD, D, RX, RY,  
ISAVE, B, SB, T, ANS, K9 );
```

(3) 引数の説明

(i) 入力

N ; 省略 観測データの個数。

K ; 省略 回帰に必要な独立変数の個数
(1)目的(31-1)式参照)

XBAR ; BIN FLOAT(26) 全変数(K+1)個の各変数に対応
する平均値で、要素の数が(K
+ 1)個の1次元アレイ。

STD ; BIN FLOAT(26) (K+1)個の各変数に対応する
標準偏差で要素の数が(K+1)
個の1次元アレイ。

D ; BIN FLOAT(26) 全変数の各変数に対する平均値
からの偏差平方和で、要素の数
が(K+1)の1次元アレイ。

RX ; BIN FLOAT(26) 回帰に必要な独立変数(K個)
間の相関係数の逆行列(K×K)。
ただし、相関係数行列の各要素

は I SAVE に規定された独立変数の順序に従わなければならぬ。

RY : BIN FLOAT(26) 従属変数と回帰に必要な独立変数(K 個)との相関係数で、要素の数が K の 1 次元アレイ。ただし、 RY の各要素の順序は I SAVE に規定された独立変数の順序に従わなければならない。

K9 : 省略

K + 1

I SAVE : 省略

要素の数 K + 1 の 1 次元アレイ。独立変数に対応する番号を上昇順に入れる。ただし、 I SAVE (K + 1) には従属変数に対応する番号を入れる。

(ii) 出 力

B : BIN FLOAT(26) 回帰係数で要素の数が K の 1 次元アレイ。

SB : BIN FLOAT(26) 回帰係数の標準偏差で要素の数が K の 1 次元アレイ。

T : BIN FLOAT(26) 各回帰係数の計算された t 値で要素の数が K の 1 次元アレイ。

ANS : BIN FLOAT(26) 以下の結果を出力する。要素の数が 10 の 1 次元アレイ。

ANS(1) : 定数 ((31 - 1) 式の b_0 に相当する)

ANS(2) : 重相関係数(R)

ANS(3) : 残差の標準誤差(S)

ANS(4) ;	分散分析表における変動 X_1 , X_2 , ……, X_k に対する平方和 (SSAR)
ANS(5) ;	SSARの自由度K
ANS(6) ;	SSARの平均平方 (SSAR/K)
ANS(7) ;	残差平方和 (SSDR)
ANS(8) ;	SSDRの自由度 N-K-1
ANS(9) ;	残差分散 (残差の平均平方)
ANS(10) ;	F値。

(4) 使用されるサブプログラム

なし

(5) 注意事項

なし

```

? 0010 MMULTP: PROC OPTIONS(MAIN);
? 0020 DCL XBAR(5) BIN,STD(5) BIN,D(5) BIN;
? 0030 DCL S(4) BIN,SB(4) BIN,T(4) BIN,ANS(10) BIN;
? 0040 DCL RX(16) BIN,RY(4) BIN,ISAVE(5);
? 0050 K=4;
? 0060 K9=K+1;
? 0080 N=30;
? 0090 XBAR=1.0;
? 0100 STD=1.0;
? 0110 D=150.0;
? 0120 RX=-1.0;
? 0130 RX(1)=2.0;
? 0140 RX(3)=0.6;
? 0150 RX(6)=1.6;
? 0160 RX(7)=-0.6;
? 0170 RX(8)=0.5;
? 0180 RX(9)=0.6;
? 0190 RX(10)=-0.6;
? 0200 RX(11)=2.5;
? 0210 RX(12)=-2.0;
? 0220 RX(14)=0.5;
? 0230 RX(15)=-2.0;
? 0240 RX(16)=3.0;
? 0250 RY(1)=0.5;
? 0260 RY(2)=-0.2;
? 0270 RY(3)=4.0;
? 0280 RY(4)=1.0;
? 0290 DO I=1 TO 5;
? 0300 ISAVE(I)=I;
? 0310 FND;
? 0320 CALL MULTP(N,K,XBAR,STD,D,RX,RY,ISAVE,B,SB,T,ANS,K9);
? 0330 PUT EDIT((B(I),SB(I),T(I)) DO I=1 TO K))(SKIP,3 F(20,8));
? 0340 PUT EDIT(ANS)(SKIP,3 F(20,8));
? 0350 FND MMVLTR;
LAST LINE(FND ST) 0870.00
? ESRUN;

```

2.50000001E+00	1.52115749E+00	1.70922474E+00
-2.71999999E+00	1.35055646E+00	-1.99917005E+00
3.42000029E+00	1.70070575E+00	4.05088599E+00
-5.6000002E+00	1.36302932E+00	-3.00585638E+00
-1.70000031E+00	5.47028347E+00	1.31736101E+01
4.43860019E+03	4.00000009E+00	1.12215004E+03
-4.33860013E+03	2.50000003E+01	-1.73544004E+02
-6.46603360E+00		

(32) TALLY

(1) 目 的

第 j 番目の変数に対する第 i 番目の観測データを X_{ij} ($i = 1, 2, \dots, NO$, $j = 1, 2, \dots, NV$) とすると、各変数に対し、総和

$\left(\sum_{j=1}^{No} X_{ij} \right)$, 平均値 ($\bar{X}_{\cdot j}$), 標準偏差 (σ_j), 最小値 ($\min_i X_{ij}$) お

より最大値 ($\max_i X_{ij}$) を求める。

(2) 使用方法

```
CALL TALLY ( A, S, TOTAL, AVER, SD, VMIN,  
VMAX, NO, NV );
```

(3) 引数の説明

(i) 入力

A ; BIN FLOAT(26)

要素の数が (NO×NV) の 1 次元アレイ。データを変数ごとに、カラムワイズに入力する。

S ; BIN FLOAT(26)

要素の数が NO の 1 次元アレイ。各観測時点のデータを計算に寄与させるか否かを決定するためのキーを表わすベクトル。

NO ; 省略 観測データ数。

NV ; 省略 変数の個数

(ii) 出力

TOTAL ; BIN FLOAT(26)

要素の数が NV の 1 次元アレイ。各変数に対応する観測データの和。

AVER ; BIN FLOAT(26)

要素の数が NV の 1 次元アレイ。各変数に対応する観測データの平均値。

SD ; BIN FLOAT(26)

要素の数が NV の 1 次元アレイ。各変数に対応する観測データの標準偏差。

VMIN ; BIN FLOAT(26)

要素の数が NV の 1 次元アレイ。各変数に対応する

観測データの最小値。

VMAX ; BIN FLOAT(26)

要素の数が NV の 1 次元アレイ。各変数に対応する

観測データの最大値。

(4) 使用されるサブプログラム

なし

(5) 注意事項

なし

(6) 使用例

3 個の変数に対し、それぞれ 10 個の観測データがあるとき、各変数
に対し、総和、平均値、標準偏差、最小値および最大値を求める。ただ
し、観測データは、RANDUN で与える。

```

? 0010  MTALLY: PROC OPTIONS(MAIN);
? 0020  DCL IX FIXFD(12),X BIN;
? 0030  DCL A(30) BIN,S(10) BIN;
? 0040  DCL TOTAL(3) BIN,AVER(3) BIN;
? 0050  DCL SD(3) BIN,VMIN(3) BIN;
? 0060  DCL VMAX(3) BIN;
? 0070  IX=9;
? 0080  NG=10;
? 0090  NV=3;
? 0100  S=1;
? 0110  DO I=1 TO 30;
? 0120  CALL RANDUN(IX,X);
? 0130  A(I)=X;
? 0140  FND;
? 0150  PUT EDIT(A)(SKIP,3 F(20,9));
? 0170  CALL TALLY(A,S,TOTAL,AVER,SD,VMIN,VMAX,NG,NV);
? 0180  PUT EDIT(TOTAL)(SKIP,3 E(20,9));
? 0181  PUT EDIT(AVER)(SKIP,3 E(20,9));
? 0182  PUT EDIT(SD)(SKIP,3 E(20,9));
? 0185  PUT EDIT(VMIN)(SKIP,3 E(20,9));
? 0186  PUT EDIT(VMAX)(SKIP,3 E(20,9));
? 0190  END MTALLY;
LAST LINE(FND ST)    0560.00
? %RUN:

```

1.19237637E-01	7.15541347E-01	8.55662977E-01
6.37260702E-01	3.79548393E-01	2.22368050E-01
7.79683068E-01	6.48817896E-01	8.76539394E-01
5.16362011E-01	8.50155547E-01	3.45302186E-01
7.60980471E-01	9.01016026E-01	6.89753353E-01
7.45743960E-01	3.14819648E-01	9.65128103E-01
5.32620057E-01	3.86009536E-01	6.79470777E-01
8.35510790E-01	5.43619617E-01	2.86432221E-01
4.32179350E-01	5.86278364E-01	9.82781462E-02
5.14248600E-02	3.34011882E-01	3.04967343E-01
5.81102140E+00	6.49152897E+00	4.70217380E+00
5.81102147E-01	6.49152904E-01	4.70217378E-01
2.64248535E-01	2.39000055E-01	2.73727335E-01
1.19237637E-01	3.14819648E-01	5.14248609E-02
8.76539394E-01	9.65128108E-01	8.35510790E-01

[33] TTEST

(1) 目的

母平均および平均値の差の検定のための統計量 t を求める。

(2) 使用方法

CALL TTEST (A, NA, B, NB, NOP, NDF, ANS);

(3) 引数の説明

(i) 入 力

A ; BIN FLOAT (26) 要素の数が NA の 1 次元アレイ
で入力データ A_i を表わす。

NA ; 省 略 入力データ A_i の個数

B ; BIN FLOAT (26) 要素の数が NB の 1 次元アレイ
で入力データ B_i を表わす。

NB ; 省 略 入力データ B_i の個数

NOP ; 省 略

つぎに示す 4 つの仮説のうち、どの仮説に対し統計量 t を求めるかを指定する。 NOP の値に対応する仮説を以下に示す。

NOP=1…… B の母平均 = 与えられた値 A

(B の母分散は未知、このとき $NA = 1$ に
セット)

NOP=2…… B の母平均 = A の母平均。

(B の母分散 = A の母分散で未知の場合)

NOP=3…… B の母平均 = A の母平均。

(B の母分散 ≠ A の母分散で未知の場合)

NOP=4…… $\mu_d = 0$ (対応のある平均値の差の検定。)

ただし、 μ_d は A と B の互いに対応するもの
の差の平均。このとき $NA = NB$)

(ii) 出 力

ANS ; BIN FLOAT (26) 与えられた仮説に対する統計量

t_{ob}

NDF ; 省 略 計算された統計量 t の自由度 F。

(4) 使用されるサブプログラム

な し

(5) 注意事項

NA と NB は NOP = 1 の場合 (NA = 1) を除いて、 1 より大きくな
ければならない。また NOP = 4 の場合は NA = NB でなければならぬ。

NOP が 1, 2, 3, 4 以外の数値の場合は、自由度と統計量は計算さ
れない。このとき DF = ANS = 0

(6) 使用例

それぞれ 10 個のデータからなる標本 A と標本 B があり、互いに対応
するものの差の平均が 0 という確信がある場合、

仮説 : $\mu_d = 0$ を検定する。

入力データは GAUSS によって与える。

```
? 0010  MTTEST: PROC OPTIONS(MAIN);
? 0020  DCL A(10) BIN,B(10) BIN,Avg BIN,STD BIN;
? 0030  DCL V BIN,ANS BIN,IX FIXED(12);
? 0040  STD=0;
? 0050  AVG=1;
? 0060  NA=10;
? 0070  NB=10;
? 0080  NOP=4;
? 0090  DO I=1 TO NA;
? 0100  CALL GAUSS(IX,AVG,STD,V);
? 0110  A(I)=V;
? 0120  END;
? 0130  PUT EDIT(A)(SKIP,3 F(20,8));
? 0140  STDB=0;
? 0150  DO I=1 TO NB;
? 0160  CALL GAUSS(IX,AVG,STDB,V);
? 0170  B(I)=V;
? 0180  END;
? 0190  PUT EDIT(B)(SKIP,3 E(20,8));
? 0200  CALL TTEST(A,NA,B,NB,NOP,NDF,ANS);
? 0210  PUT DATA(NDF,ANS);
? 0220  END MTTEST;
LAST LINE(END ST)    1100.00
? %RUN:
-6.45438209E-01      2.32109788E-01      -1.47582473E+00
 6.54970183E-01      3.44575531E-01      -9.34534087E-02
-8.07739511E-01      7.63381138E-01      -2.84617431E-01
-3.36206801E-01
1.28425218F+00      -1.52211965F+00      -6.96466103E-01
-1.14423921F+00      2.13440421F-01      2.26490717F+00
1.26232590E+00      -7.85383716E-01      1.07318952E+00
7.14228764E-01
NDF=          9      ANS=  3.3516030E-01;
```

[34] ORDER

(1) 目 的

従属変数と独立変数とからなる変数間の相間係数行列から、従属変数といいくつかの独立変数とを選び出し、それらの変数間の相関係数行列を作る。

(2) 使用方法

```
DALL ORDER ( M, R, NDEP, K, ISAVE, RX, RY,  
MM, K9 ) ;
```

(3) 引数の説明

(i) 入 力

M ; 省略	全変数の数。
MM ; 省略	$M \times (M + 1) / 2$ 。
R ; BIN FLOAT(26)	相関係数行列で $(M+M)$ の対称行列の上部三角形の部分（対角要素も含む）をカラムワイズで入力する。要素の数 $M(M + 1) / 2$ の 1 次元アレイ。
NDEP ; 省略	相関係数行列の従属変数に対応する番号（例題参照）
K ; 省略	選び出される独立変数の数。
K9 ; 省略	$K + 1$ 。
ISAVE ; 省略	要素 $(K + 1)$ の 1 次元アレイ。 相関係数行列の選びだされる独立変数に対応する番号を上昇順に入れる。ただし、 ISAVE($K + 1$)番目には出力として NDEP の数字がは

いる。(例題参照)

(ii) 出力

RX : BIN FLOAT(26) 選ばれた独立変数間の($K \times K$)の相関係数行列をカラムワイズに出力する。

RY : BIN FLOAT(26) 従属変数と独立変数との相関係数ベクトル。

(例題) M 個の変数から成る相関係数行列の上部三角形部分を示すと図8-1のようになる。

変数に対応する番号	1	2	3	M
変 数	x_1	x_2	x_3	x_M
	$R(1)$	$R(2)$	$R(4)$	$R\left(1 + \frac{M(M-1)}{2}\right)$
		$R(3)$	$R(5)$	$R\left(2 + \frac{M(M-1)}{2}\right)$
			$R(6)$	$R\left(3 + \frac{M(M-1)}{2}\right)$
				
					$R\left(\frac{M(M+1)}{2}\right)$

図8-1 相関係数行列

そこで、 $M=5$ とすると、図8-1より

$$R = \begin{bmatrix} R(1) & R(2) & R(4) & R(7) & R(11) \\ & R(3) & R(5) & R(8) & R(12) \\ & & R(6) & R(9) & R(13) \\ & & & R(10) & R(14) \\ & & & & R(15) \end{bmatrix} \cdots \cdots (8-1)$$

と書ける(ただし、(8-1)式には上部三角形部分のみを表わしている)。

いま従属変数を x_1 、選びだされる独立変数を x_2, x_4, x_5 とする
ると、入力パラメータの作成はつきのとおりである。

M = 5

R = 相関係数行列の上部三角形部分をカラム・サイズに入れる

((8-1)式参照)

NDEP = 1 (従属変数に対応する番号)

K = 3 (独立変数の数)

ISAVE(1) = 2 (x_2 に対応する番号)

ISAVE(2) = 4 (x_4 に対応する番号)

ISAVE(3) = 5 (x_5 に対応する番号)

(4) 使用されるサブプログラム

なし

(5) 注意事項

なし

(6) 使用例

M = 5, NDEP = 1, K = 3 選び出す独立変数: x_1, x_2, x_3

と想定する。

```

? 0010  MORDER: PROC OPTIONS(MAIN);
? 0020  DCL R(15) BIN,TSAVE(4);
? 0030  DCL RX(9) BIN,RY(3) BIN;
? 0040  M=5;
? 0050  NDEP=1;
? 0060  K=3;
? 0070  ISAVF(1)=2;
? 0080  ISAVF(2)=4;
? 0090  ISAVF(3)=5;
? 0100  R(1)=0.1192376E+0;
? 0110  R(2)=0.715332E+0;
? 0120  R(3)=0.1474609E+0;
? 0130  R(4)=0.4428711E+0;
? 0140  R(5)=0.3295898E+0;
? 0150  R(6)=0.0392578E+0;
? 0160  R(7)=0.0697266E+0;
? 0170  R(8)=0.9111328E+0;
? 0180  R(9)=0.7353516E+0;
? 0190  R(10)=0.2070313E+0;
? 0200  R(11)=0.6210938E+0;
? 0210  R(12)=0.8642578E+0;
? 0220  R(13)=0.5947266E+0;
? 0230  R(14)=0.7861328E+0;
? 0240  R(15)=0.3603515E+0;
? 0250  MN=M*(M+1)/2;
? 260  K9=K+1;
? 0280  CALL MORDER(M,R,NDEP,K,TSAVE,RX,RY,MN,K9);
? 0290  PUT EDIT(RX)(SKIP,3 E(20,3));
? 0295  PUT EDIT(RY)(SKIP,3 E(20,3));
? 0300  END MORDER;
LAST LINE(END ST) 0500.00
? XRUN;

```

1.47460903E-01	9.11132812E-01	8.64257312E-01
9.11132812E-01	2.07031302E-01	7.85132812E-01
8.64257312E-01	7.86132312E-01	3.80351502E-01
7.15332016E-01		6.21093809E-01
9.69726607E-01		

[35] PADD

(1) 目 的

2つの多項式の和を求める。

すなわち、2つの多項式を $f(x)$, $g(x)$ とし求める結果を $h(x)$ とすると、

$$h(x) = f(x) + g(x) \text{ である。}$$

(2) 使用方法

```
CALL PADD ( Z, IDIMZ, X, IDIMX, Y, IDIMY );
```

(3) 引数の説明

(i) 入 力

X ; BIN FLOAT(26) 多用式 $f(x)$ の係数（次数の
 低いものから高いものへ）。
 要素数 IDIMX の 1 次元アレ
 イ。
 IDIMX; 省略 多項式 $f(x)$ の次数 + 1。
 Y ; BIN FLOAT(26) 多項式 $g(x)$ の係数（次数の
 低いものから高いものへ）。
 要素数 IDIMY の 1 次元アレ
 イ。
 IDIMY; 省略 多項式 $g(x)$ の次数 + 1。

(ii) 出力

Z ; BIN FLOAT(26) 結果 $h(x)$ の係数（次数の低
 いものから高いものへ）。要
 素数 IDIMZ の 1 次元アレイ。
 IDIMZ; 省略 $\max(IDIMX, IDIMY)$ 。

(4) 使用されるサブプログラム

なし

(5) 注意事項

なし

(6) 使用例

$$f(x) = 3 + 4x - x^2 + 2x^3$$

$$g(x) = 5 - 2x + 2x^2$$

の $f(x) + g(x)$ を求める。

```

? 0010 MPADD: PRMC OPTIGMS(MAIN);
? 0020 DCL X(4) BIN,Y(3) BIN,Z(4) BIN;
? 0030 IDIMX=4;
? 0040 IDIMY=3;
? 0045 IDIMZ=4;
? 0050 X(1)=3;
? 0060 X(2)=4;
? 0070 X(3)=-1;
? 0080 X(4)=2;
? 0090 Y(1)=5;
? 0100 Y(2)=-2;
? 0110 Y(3)=2;
? 0120 CALL PADD(Z, IDIMZ, X, IDIMX, Y, IDIMY);
? 0130 PUT FDIT(Z)(4 F(10));
? 0140 END MPADD;
LAST LINE(FND ST) 0340.00
? %RUN;

```

8 2 1 2

(36) BESFJN

(1) 目的

与えられた変数 x および次数 n に対する第1種ベッセル (Bessel) 関数 $J_n(x)$ の値を求める。

(2) 使用方法

$Y = \text{BESFJN}(X, N, IER)$; 組込関数と同様。

(3) 引数の説明

(i) 入力

X ; DEC FLOAT(20) 第1種ベッセル関数の変数 x 。

N ; 省略 第1種ベッセル関数の次数 n 。

(ii) 出力

IER ; 省略 エラー・コード。

$IER=0$ 正常に解が得られた場合。

$IER=1$ $N < 0$ の場合。

$IER=2$ $X \leq 0.0$ の場合。

$IER=3$ 相対誤差 10^{-6} の精度で得られなかつた場合。

I E R = 4 X の値に対して N の範囲が適当で
ない場合。（注意事項(iii)参照）

(4) 使用されるサブプログラム

なし

(5) 注意事項

(i) $N \geq 0$, $X > 0.0$ であること。

(ii) X と N との関係は次のようになっていること。これ以外の場合には
エラーとなり、I E R = 4 がセットされる。

$$0 < X \leq 15 \text{ のとき } N < 20 + 10X - \frac{X^2}{3}$$

$$X > 15 \text{ のとき } N < 90 - 0 + \frac{X}{2}$$

(iii) I E R = 1, 2, 4 のとき B E S F J N = 0.0 である。

(iv) I E R = 3 のとき B E S F J N には計算途中の結果が入る。

(6) 使用例

$J_1(1.5)$ を求める場合。

```
? 0010 MBESFJN: PROC OPTIONS(MAIN);
? 0020 DCL X FL0AT(20),Y FL0AT(20);
? 0030 N=1;
? 0040 X=1.5;
? 0050 Y=BESFJN(X,N,IER);
? 0060 PUT DATA(Y,IER);
? 0070 END BESFJN;
LAST LINE(END ST) .0760.00
? %RUN:
Y= 5.5793651938438415527E-01      IER=          0;
```

[37] B E S F Y N

(1) 目的

与えられた変数 x および次数 n に対する第 2 種ベッセル (Bessel)
関数 $Y_n(x)$ の値を求める。

(2) 使用方法

$Y = \text{BESFYN}(X, N, IER)$; 組込関数と同様。

(3) 引数の説明

(i) 入 力

X ; DEC FLOAT(20) 第2種ベッセル関数の変数 x。

N ; 省 略 第2種ベッセル関数の次数 n。

(ii) 出 力

IER ; 省 略 エラー・コード。

IER=0 正常に解が得られた場合。

IER=1 $N < 0$ の場合。

IER=2 $X \leq 0.0$ の場合。

(4) 使用されるサブプログラム

なし

(5) 注意事項

(i) $N \geq 0$, $X > 0.0$ であること。

(ii) $IER = 1, 2$ のとき $\text{BESFYN} = 0.0$ である。

(6) 使用例

$Y_1 (1.5)$ を求める場合。

```
? 0010 MBESFYN: PROC OPTIONS(MAIN);
? 0020 DCL X FLOAT(20).Y FLOAT(20);
? 0030 N=1;
? 0040 X=1.5;
? 0050 Y=BESFYN(X,N,IER);
? 0060 PUT DATA(Y,IER);
? 0070 END BESFYN;
LAST LINE(FND ST) 1250.00
? %RUN;
Y=-4.1230864077806472778E-01      IER=          0;
```

8.2 TSS-FORTRAN ¥REPLACEサブコマンドの新設

¥REPLACEサブコマンドは、あるステートメントの一部分を変更したいような場合、そのステートメントを全部タイプし直すのではなく、その変更部分だけを任意の文字列で置換える機能をもつものである。また、1ステートメントのみでなく、プログラム全体についても、1つの¥REPLACEコマンドで特定の変数名等を別の変数名にすべて置きかえることが出来るようにしたものである。

8.2.1 概 説

本コマンドはソースプログラムの入力中、あるいは¥UPDATEコマンドの修正情報として、部分文字列を指定した文字列で置き換え、ソースプログラムを修正することをその目的とする。

(1) コマンドの一般形

次に本サブコマンドの一般入力形式を示す。

¥REPLACE [ℓn_1 [, ℓn_2]], :XXXX:, :YYYY: (CR)

コマンド名の省略形は次のとおりである。

¥REP

解 説 :

現在処理中のプログラム単位に対し、

① ℓn_1 , ℓn_2 の両者が省略された場合。

そのプログラム内の全てのステートメントに対し、部分文字列
XXXX をYYYYで置き換える。

② ℓn_2 が省略され場合。

ℓn_1 のステートメントに対し、部分文字列XXXXをYYYYで置き換える。

③ 省略されなかった場合。

ℓn_1 から ℓn_2 までのステートメントに対して、部分文字列XXXXを
YYYYで置き換える。

(2) 他のコマンドとの関係

本コマンドは次の2つのコマンドのサブコマンドとして使用される。

¥INPUT

¥UPDATE

よって次のサブコマンドと同列である。

¥LIST

¥DELETE

¥RESEQ

(¥line number)

8.2.2 詳説

(1) 処理機能

本サブコマンドは大別して次の2つの処理機能から構成される。

*1. コマンドの Syntax Check

*2. 部分文字列の置換

以下にこれらについて解説する。

(a) コマンドの Syntax Check

*1. 次の形式以外は全てエラーとする。

$$\left\{ \begin{array}{l} \$REPLACE \\ \$REP \end{array} \right\} \hookrightarrow \left\{ \begin{array}{l} \text{空} \\ \ell n_1 \\ \ell n_1, \ell n_2 \end{array} \right\}, :XXX:, :YYY:$$

(コマンド名) (line No.) (文字列1)(文字列2)

*2. コマンド名について

本名は \$REPLACE とし、システムでは \$REP までをチェックする。

省略形は \$REP とする。

コマンド名の次には必ず一つ以上の空白(文字)をおく。

*3. line No. について

line No.は次の場合を許す。

① 空(指定なし)

② ℓn_1 の指定

③ ℓn_1 及び ℓn_2 の指定

ℓn_1 及び ℓn_2 は 5 柄以内の数字でなければならぬ。

$\ell n_1 \leq \ell n_2$ でなければならぬ。

line No. の次には必ずカンマ(,)を置かなければならぬ。

*4. 文字列 1 及び文字列 2 について

① 置換を行なう文字列を ' : ' で括って示す。

② ' : ' で括られた文字列は ' : ' を除く全ての文字を指定することが出来る。

③ 文字列 1 を文字列 2 で置き換える。

④ 文字列 1 と文字列 2 の間は区切記号 ',', を置く。

⑤ 文字列 2 以降の文字は全て無視する。

⑥ 文字列は最大 40 柄とする。

⑦ 本コマンドに対する継続は認めない。

⑧ 文字列 1 には空ストリングは認めないが、文字列 2 に対しては許す。

(b) 部分文字列の置き換え

*1. 置き換えの対象となるのは次のものである。

① ℓn_1 及び ℓn_2 の両者が指定された場合

$\ell n_1 \leq L n \leq \ell n_2 \dots (A \cdot 1)$

を満足する、その時点において処理対象となっているエレメント内の
line No. Ln に対し置き換えを行なう。

ただし、式(A・1)において、等号が成立しない場合は警告メッセージを出力する。

② ℓn_1 のみが指定された場合

現在処理対象となっているエレメント内において、 ℓn_1 で指定された
line No. のステートメントに対し、置き換えを行なう。

ただし、 ℓn_1 に対応する line No. が存在しない場合にはエラーメッ

セージを出力し、処理は行なえない。

又、 ln_1 で指定された line No. が *2 で示す制限に該当する場合はエラーメッセージを出力し、処理は行なわない。

③ ln_1 及び ln_2 が省略された場合

現在処理対象となっているエレメント内の全てのステートメントを置き換えの対象とする。

ただし、*2 で示す制限に該当するステートメントは置き換えの対象からはずす（エラーメッセージの出力も行なわない）。

ただし、現在処理対象となっているエレメント内にステートメントが存在しない場合にはエラーメッセージを出力する。

*2 置き換えの対象から除外するもの

① エレメント文

指定された line No. がない旨のエラーメッセージを出力し、処理を行なわない。

② SUBROUTINE 文及び FUNCTION 文

上記ステートメントに対する置き換えは出来ない旨のエラーメッセージを出力し、処理は行なわない。

③ コメント文

コメント文に対する置き換えは出来ない旨のエラーメッセージを出力し、処理は行なわない。

*3 文番号に対する置き換えは、次のチェックを行なう。

① 数字チェック

置き換え後の文字列が数字であるか否かをチェックする。

数字でない場合はエラーメッセージを出力し、文番号に対する置き換えるのみを行なわない。

② 長さチェック

置き換え後の文字列は数字であるが、その文字列の長さが、5 柄を越える場合にはエラーとし、エラーメッセージを出力して、その文番号に

対する置き換えのみを行なわない。

- *4. 置き換えの対象となる部分文字列が検出されなかった場合はエラーメッセージを出力する。
- *5. 継続行にまたがった文字列も置き換えの対象とする。
- *6. 置き換えを行なった後、継続行が 16 行を越える場合には、エラーメッセージを出力し、越えた部分は捨てられる。

(2) エラーメッセージ

エラー番号	レベル	メッセージ	備考
026	X	ET 826-X モジレツ ガ ナガスギル	
027	W	ET 827-W XXXXX コノ ステートメント ナンバー ノ オキカエ ハ デキナイ	
028	X	ET 828-X LINE-NUMBER ノ シティ ガ ムジュン シテイル	
029	X	ET 829-X :: ノ シティ ハ デキナイ	
030	X	ET 830-X SUBROUTINE ブン マタハ FUNCTION ブン ノ オキカエ ハ デキナイ	
031	X	ET 831-X LINE-NUMBER-1 ガ ミツカラナイ ムシ スル	
032	X	ET 832-X LINE-NUMBER-2 ガ ミツカラナイ ムシ スル	
033	X	ET 833-X ステートメント ガ 20 ギョウ オ コエタ	
034	W	ET 834-W ガイトウ スル モジ ガ ミツカラナイ	
035	X	ET 835-X コメント ブン ノ オキカエ ハ デキナイ	

3. 使用例

```
JOB KAISI.  
$INPUT MAIN  
00010 DIMENSION A(10,10)  
00020 DO 1 I=1,10  
00030 %REPLACE 10,:,:10::,:5:  
00030 %LIST,10,10  
    00010      DIMENSION A(10,5)  
00030 DO 2 J=1,10  
00040 A(I,J)=0  
ET310X エラーナイフン カル  
00050 %REP 10,:10:::100:  
00050 %REP 40,:I,J:::I,J:  
00050 %LIST  
    00010      DIMENSION A(100,5)  
    00020      DO 1 I=1,10  
    00030      DO 2 J=1,10  
    00040      A(I,J)=0  
00050 2 CONTINUE  
00060 1 CONTINUE  
00070 STOP  
00080 END  
00090 %LIST  
    00010      DIMENSION A(100,5)  
    00020      DO 1 I=1,10  
    00030      DO 2 J=1,10  
    00040      A(I,J)=0  
    00050 2 CONTINUE  
    00060 1 CONTINUE  
    00070 STOP  
    00080 END  
00090 %END  
$UPDATE MAIN  
>%REP ,:2,:,:5:  
>%REP 20,:10:::100:  
>%REP 困難,30  
ET815X ラメータ カル キヨシヨウ テキナイ  
>%REP 30,:10::,:5:  
>%UEND  
%LIST MAIN  
    00010      DIMENSION A(100,5)  
    00020      DO 1 I=1,100  
    00030      DO 5 J=1,5  
    00040      A(I,J)=0  
    00050 5 CONTINUE  
    00060 1 CONTINUE  
    00070 STOP  
    00080 END  
$STOP  
//TSUGI WA ? COMMAND OR 'BYE' ?      ZIKOKU=03.09 16:06:05  
JOB GWARI.  
/YOUR JOB-N0....DJID013
```

ソースプログラム入力中

UPDATEコマンド実行中

8.3 JOLDOR-IIの開発

8.3.1 概 要

当センターでは、1970年春にオンラインによる文献検索システム JOLDOR-I (JIPDEC ON LINE DOCUMENT RETRIEVAL) を開発し、FACOM230-60 TSSのもとでその実験を行ったが、このたび JOLDOR-I の改良システムである JOLDOR-II を開発したので、その概要を報告する。（JOLDOR-Iについては当センター報告書 43-S-005、および 44 年の第 10 回 プログラミング・シンポジウムの報告集に記されている）

JOLDOR-I の開発の狙いは次の 2 つに要約できる。

1. オンラインによる検索方式の開発および実験。
2. マン・マシン・インタラクションによる検索過程の効率向上に関する実験。

JOLDOR-I ではプロトタイプ的な色彩が強かったので、収容文献数も 1 ファイル当たり約 7000 件と、比較的少なく、初心者に対するガイダンスの機能にしても充分満足できたとは言い切れない。

そこで、約 2 年間の JOLDOR-I の利用経験を生かし、次のような改良及び修正を加えた JOLDOR-II システムを新たに開発し、実用化実験を進めている。

1. 1 ファイルに収容可能な文献数の制限解除。

これにより年間 3 万件、最大 10 万件程度は現機器構成で収容可能である。

2. マン・マシン・インタラクション機能の充実。

応答速度を低下させることなしに、ガイダンス機能、およびタイプ・ミスなどエラーに対する指示を、さらに充実させた。

3. 文献情報として、アブストラクトの追加、それにともなうコマンド体系の充実。

標題とキーワードでは文献の内容が把握できないため、文献のアブストラクトを追加している。

4. キーワードとして複合語の採用。

JOLDOR - I ではキーワードとして単語単位にしていたが、 JOLDOR - II では複合語を採用することにより検索効率の向上をはかっている。

5. タイプライタ端末のみ設置されているユーザーにも利用可能とする。

JOLDOR - I ではキャラクタ・ディスプレイからしか検索できなかったが、 JOLDOR - II ではタイプライタ端末からも検索可能にしている。

その他、ファイル方式および検索方式などの基本的な設計思想は JOLDOR - I に準じている。

8.3.2 オン・ライン文献検索の持つ意味

オンライン文献検索の基本的目標は、 man-machine interaction を有効に利用することである。 man-machine interaction とは、ユーザによる creation である。このことがもたらす意義は非常に大で、ユーザに対して想像以上の満足感を与える結果となる。

理想的なオンライン文献検索システムにおける man-machine interaction では論理展開の途中結果を自分で確認でき、検索ミス、操作ミスなどの誤りを恐れることなく試行錯誤で自由に論理展開できることである。（ユーザによる creation）

更にこれは、ユーザが検索あるいはコンピュータに対して全くの初心者であっても、ガイダンス機能が完備していれば全然支障はきたさない。

そこにはシステム側からの一方的押しつけは微塵も存在しない。

これから、文献検索システムは費用と設備が許す限りこのオンラインの方向に進むであろうと思われる。しかし、そこには当然 display のような man-machine interaction に適した装置が必要になってくる。

出力としては、マイクロ・フィルムを表示する装置、またそれを copy する装置等があれば、申し分ないと見える。

大量出力の問題には、センタへのバッチによる出力も必要となろう。

8.3.3 検索に対する JOLDOR-II の姿勢

JOLDOR-II の目標は man-machine interaction の充実と、検索効率の向上である。検索システムに対して最終的評価を下すのはユーザであり、man-machine interaction の充実とは、それに対するユーザの満足度が 1 つの評価基準となる。

この点では JOLDOR-II は JOLDOR-I の精神を引きつぐと共に、更に充実したガイダンス機能を盛り込み、検索に対して、特に予備知識を持たぬ一般ユーザに対しても応じられるように配慮した。

検索効率の向上という問題では、とかくファイル構造やアクセス法、あるいはより細かいプログラム技術の面のみが注目され勝ちであるが、一次情報から二次情報への indexing も重要なポイントになる。

本来、この indexing は著者の意図を正確に表現するように付けるものであって、解釈を加えるべきではない。それ故著者自ら indexer となるのが望ましい。このことによって、少くとも noise は減少するであろう。さいわいにも JOLDOR-II のデータは、そのほとんどが著者による indexing が行われている。

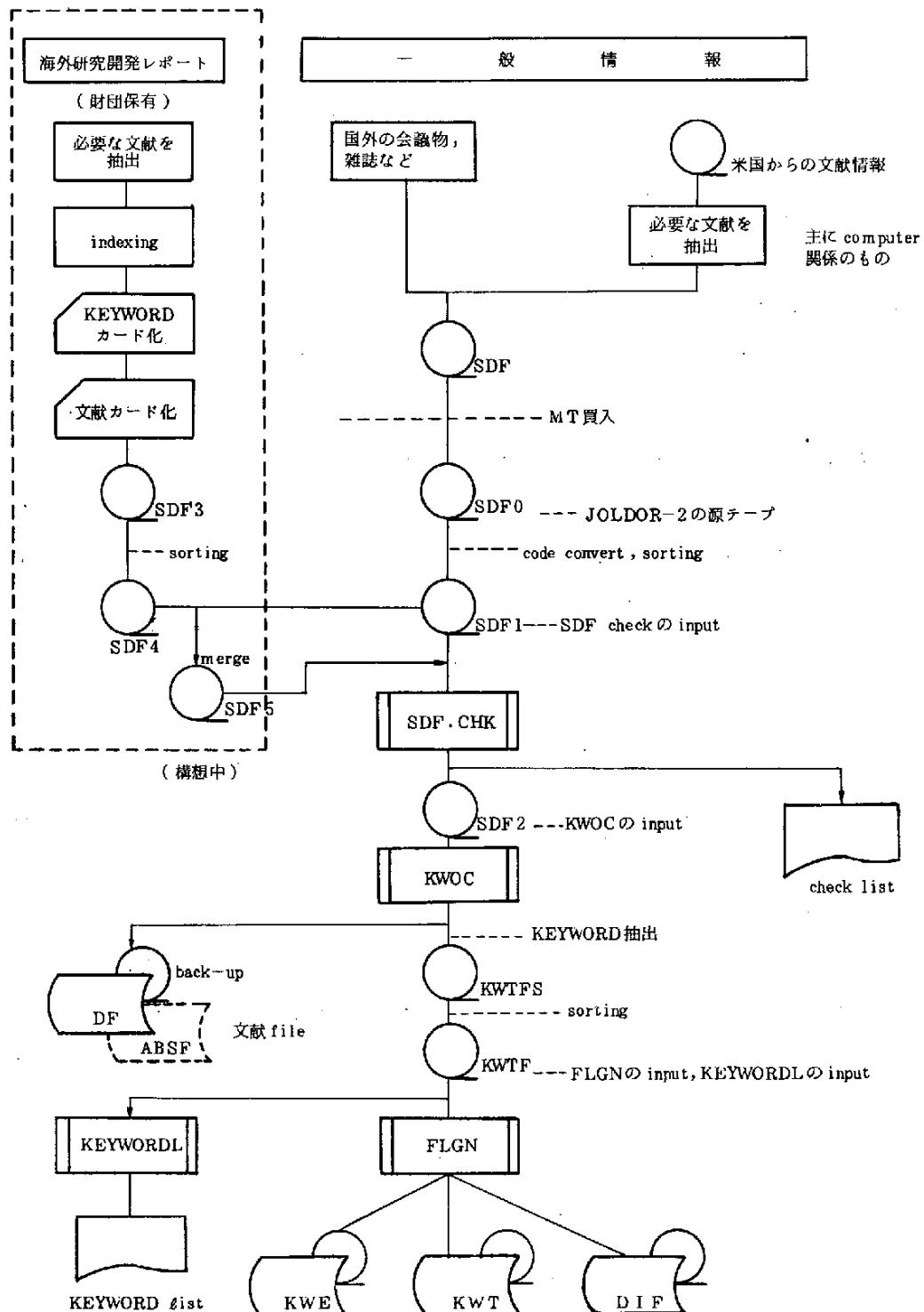
一般的に、再現率（正検索率）と適合率は反比例の関係にあり、多少の noise は増えても、検索洩れを減らそうとするのが世の趨勢である。理想（目標ではない）としては、再現率、適合率とも 1.0 の状態である。

$$\left\{ \begin{array}{l} \text{再現率} = (\text{検索された妥当な文献}) / (\text{貯蔵されている全妥当文献}) \\ \text{適合率} = (\text{検索された妥当な文献}) / (\text{検索された全文献}) \end{array} \right.$$

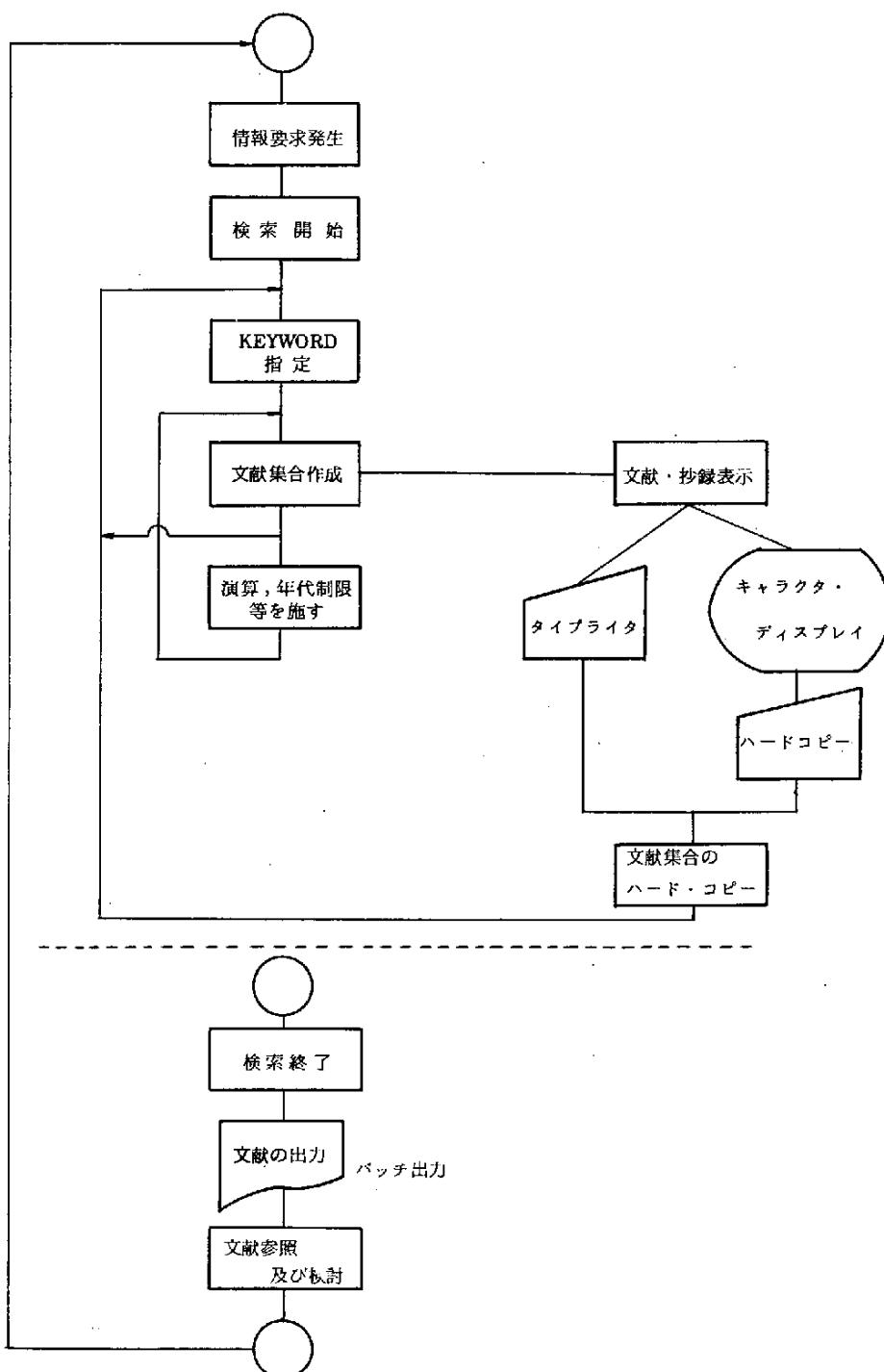
著者が indexer になることと、オンラインで検索することによって、検索洩れの主な原因である indexing ミスと検索質問式のミスは減少するであろう。これによって、適合率を下げることなしに、noise は減少する。

又、JOLDOR-II ではタイトルからの KEYWORD 採用を取り止め、ディスクリプタだけからとし、しかも複合語採用に踏み切ったので、技術的には、やゝ難しくなったが JOLDOR-I に比べて、noise および検索洩れが減少し、全体的に検索効率は上昇した。

8.3.4 情報蓄積経路



8.3.5 情報検索経路



8.3.6 システムの概要

(1) 最少機器構成等

- ① 中央処理装置 FACOM 230-60
- ② 磁気テープ装置 機種 F 603 F 7 Track 1台
9 Track 2台
- ③ 磁気ディスク装置 機種 F 471 K 7,294 Byte/Track
20 Track/Cyl
203 Cyl/Pack
1台～4台
- ④ ラインプリンタ 1,500 行/分(高速)
- ⑤ カード・リーダ 800 枚/分
- ⑥ 使用領域 HCM 20 kw
- ⑦ 端末機器
 - キャラクタ・ディスプレイ装置 F 6221 B 50 文字×20 行
 - タイプライタ端末(各種)
- ⑧ ソフトウェア FACOM 230-60 M-V
- ⑨ 使用言語 FASP, COBOL(, FORTRAN)
- ⑩ データ構成 磁気テープ 全部 固定長 ブロック化
磁気ディスク 一部 不定長
セグメント分割形式
- ⑪ 情報源 主に米国のコンピュータ関係の文献
- ⑫ データ・アップ・データ・サイフル 月1回(予定)
- ⑬ 蓄積文献数 約 30,000 件(年間)
- ⑭ 検索論理式 AND, OR, NOT
- ⑮ 蓄積 KEYWORD 数 約 30,000 種
- ⑯ KEYWORD. リスト提供
- ⑰ ファイル作成部分ステップ数 2,700

⑯ 検索部分ステップ数 6,900

(2) 検索時におけるファイル関連図及び検索例

ここでは、検索時に JOLDOR-II が参照するファイルを中心に各コマンドの動きを示すとともに、検索の流れを示す。

図 1 は、SELECT, COMBINE, LIMIT, DISPLAY, PRINT コマンドと各ファイル、テーブルとの関係を示している。

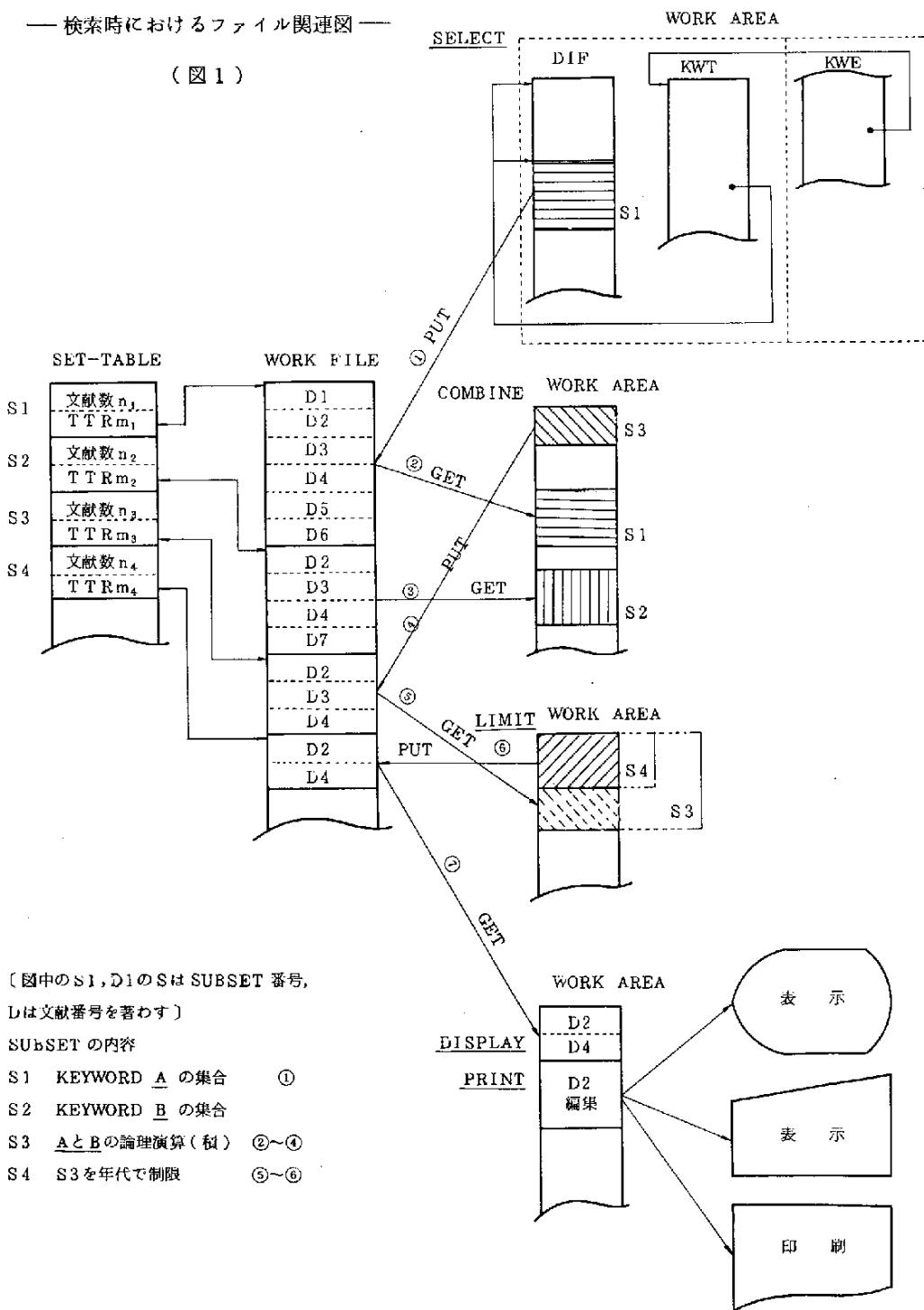
図 2 は、図 1 の流れに従って実際に検索した過程を示すハードコピーである。

図 3 は、図 2 の検索例による出力の一部である。

J O L D O R - II

—検索時におけるファイル関連図—

(図1)



前頁の流れ図に基づいた検索例

```
***D
? USER-ID...JIPDEC,03044348211
/ YOUR JOB-NO....DJIF021
MACRO BUN NYUURYOKU.
**JOL73Y
JOB KAISI.
WELCOME! THIS IS JOLDER-2 SYSTEM.
DO YOU KNOW HOW TO OPERATE THIS SYSTEM? (YES/NO)
/YES
* ENTER COMMAND *
/SELECT INFORMATION RETRIEVAL
* SET-NO 1 ; DOCUMENTS 321 *
* ENTER COMMAND *
/DISPLAY 1
```

TSSとの会話開始
利用者名、パスワード入力

JOLDOR-2マクロコマンド入力、会話開始

キーワード INFORMATION RETRIEVAL
指定

文献表示指定

(TITLE) REF-NO NDK720500361 DOC-NO 1
RAPID PRE-INDEXING BY MACHINE. TECHNICAL REPT.,.

(DESCRIPTOR)
ON-LINE SYSTEMS, SUBJECT INDEXING, AUTOMATIC, INFORMATION RETRIEVAL, INDEXES, ABSTRACTS, DICTIONARIES, COMPUTER PROGRAMS, INTREX PROJECT, AUTOMATIC INDEXING, SUBJECT INDEX TERMS,
(AUTHOR)
KAMPE WILLIAM R. II MASSACHUSETTS INST O

(BIBLIOGRAPHY)
REP-NO. AD-735 502 USGRDR 72 05
LANG. EN FLD. 05B VOL. NO. PP.70P* YR. 72

-315-

(図2)

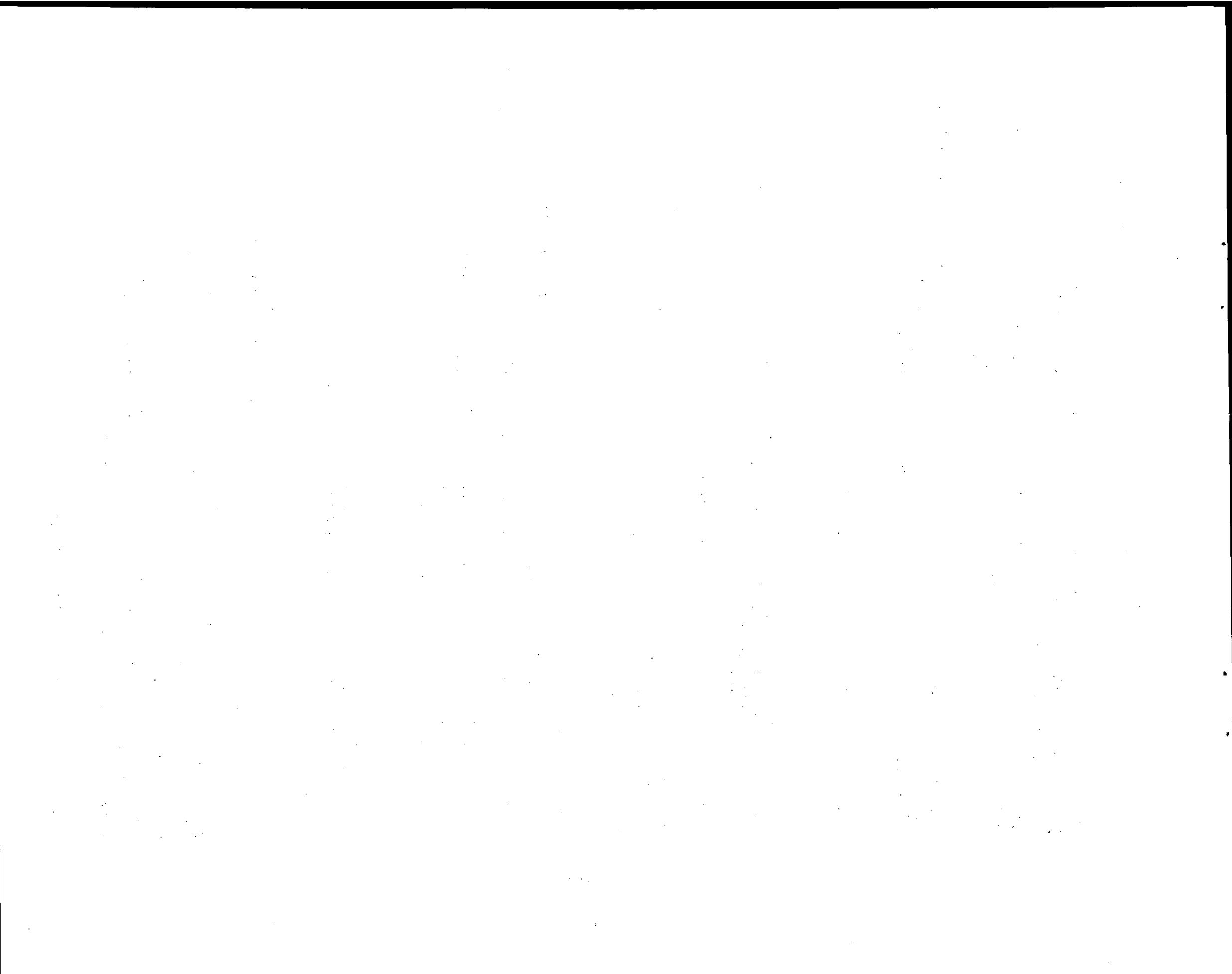
```
* ENTER COMMAND *
/SELECT ON-LINE SYSTEMS
* SET-NO 2 ; DOCUMENTS 29 *
* ENTER COMMAND *
/COMBINE 1*2 L
* SET-NO 3 ; DOCUMENTS 7 *
* ENTER COMMAND *
/LIMIT 3 72-73
* SET-NO 4 ; DOCUMENTS 4 *
* ENTER COMMAND *
/TEACH 4
```

キーワード ON-LINE SYSTEMS 指定
論理2項演算(積)
発行年代制限(72~73)
ガイダンス

* SET-NO 4 ; DOCUMENTS 4 *
* CONTAINS FOLLOWING SUBJECTS.(LIMITED 72-73)
(INFORMATION RETRIEVAL*ON-LINE SYSTEMS)

```
* ENTER COMMAND *
/PRINT 4
* ENTER COMMAND *
/*BYE
THANK YOU FOR USING JOLDER-2
//TSUGI WA ? COMMAND OR 'BYE' ? ZIKEKU=03.06 13:53
JOB CWARI.
/ YOUR JOB-NO....DJIF021
MACRO BUN NYUURYOKU.
**BYE
//CPU-TIME 00:00:01.774
ANATA WA DEMAND NO RIYOO O YAMETA.
ZIKEKU 13:53 NI OFF-LINE NI NATTA. SIYEEZIKAN CC:09:51
```

文献集合4のセンタ出力指定
TSSとの会話終了
使用時間(CPU)



SET-NO	<u>4</u>	文献集合番号	DOCUMENTS	
* CONTAINS FOLLOWING SUBJECTS. (LIMITED <u>72-73</u>) <u>(INFORMATION RETRIEVAL ON-LINE SYSTEMS)</u>			文献数	
指定キーワード (TITLE)	論理積	REF-NO <u>NDK720500361</u>	DOC-NO	<u>1</u>
		文献番号		文献順序

RAPID PRE-INDEXING BY MACHINE. TECHNICAL REPT.,.

(DESCRIPTOR)

ON-LINE SYSTEMS, SUBJECT INDEXING, AUTOMATIC, INFORMATION RETRIEVAL, INDEXES, ABSTRACTS, DICTIONARIES, COMPUTER PROGRAMS, INTREX PROJECT, AUTOMATIC INDEXING, SUBJECT INDEX TERMS.

(AUTHOR)

KAMPE WILLIAM R. II MASSACHUSETTS INST OF TECH CAMBRIDGE

(BIBLIOGRAPHY)

REPORT-NO.	AD-735 502 USGRDR 72 05	REF-NO	NDK720500607	DOC-NO	<u>2</u>
LANG.	EN FIELD 05B VOLUME NUMBER	PAGE	70P*	YEARS	<u>72</u>

発行年代

(TITLE)

MODIFICATIONS TO ON-LINE FILE MANIPULATION TECHNIQUES (MADAPS). FINAL TECHNICAL REPT. MAR 70-SEP 71.,.

(DESCRIPTOR)

ON-LINE SYSTEMS, REMOTE CONTROL, DATA PROCESSING, CONTROL EQUIPMENT, INFORMATION RETRIEVAL, INFORMATION RETRIEVAL, MANAGEMENT PLANNING, DIGITAL COMPUTERS, TIME SHARING, CATHODE RAY TUBE SCREENS,

(AUTHOR)

CIRA PAUL E. SYSTEM DEVELOPMENT CORP FALLS CHURCH

(BIBLIOGRAPHY)

REPORT-NO.	AD-735 751 USGRDR 72 05	REF-NO	NDK720500607	DOC-NO	<u>2</u>
LANG.	EN FIELD 09B VOLUME NUMBER	PAGE	54P	YEARS	<u>72</u>

(これはPRINT指定の出力結果の一部である)

(図3)

(3) コマンドの種類と機能

JOLDOR-IIで用いるコマンドは、以下の6種14個である。（下線は省略形）

a. FACOM 230-60 TSSとの会話用

① JOLDOR

② BYE

b. 文献集合を作成する為のもの（3個）

③ SELECT

④ COMBINE

⑤ LIMIT

c. 文献集合の表示・出力用（6個）

⑥ DISPLAY

⑦ NEXT

⑧ BACK

⑨ ABSTRACT

⑩ HARDCOPY

⑪ PRINT

d. 一連の検索作業を停止させるもの

⑫ ENDSEARCH

e. ガイダンス用

⑬ TEACH

f. 一連の検索作業を終了させるもの

⑭ BYE

以下、個々のコマンドについて、その機能を説明する。

① JOLDOR△{₅⁶}

端末装置とシステムをつなぐマクロ・コマンドであり、このとき、キャラクタ・ディスプレイを用いるかどうか意志表示する。このあとシステム側からメッセージが表示され、タイプライタあるいは、キャラクタ・ディ

スプレイでの検索が可能になる。

② BYE

TSSとの会話を終了させるコマンドであり、システム側から、使用時間、cpu time等が表示される。

③ SELECT \triangle keyword

指定した keyword が引用されている文献の集合を抽出する。システム側からは、文献集合の set-no 及び文献数が表示される。

④ COMBINE \triangle $n_1 \left\{ \begin{array}{c} * \\ + \\ - \end{array} \right\} n_2$

$n_1, n_2 : \text{set-no}$

既成 (SELECT, COMBINE, LIMITによる) の文献集合どうしに論理演算を施すものである。システム側からは、新しくできた文献集合の set-no 及び文献数が表示される。この文献集合に対しても、COMBINE を施すことが可能である。

⑤ LIMIT \triangle set-no $\triangle \left\{ \begin{array}{c} mn - nm \\ * - nm \\ mn - * \end{array} \right\}$

下限 上限
 $*(00)$ $*(99)$

既成 (SELECT, COMBINE, LIMITによる) の文献集合において、発行年代を制限するものである。システム側からは、新しくできた文献集合の set-no 及び文献数が表示される。この文献集合に対しても LIMIT を施すことが可能である。

⑥ DISPLAY \triangle set-no

既成 (SELECT, COMBINE, LIMITによる) の文献集合についている set-no を指定することにより、最初の文献内容を端末に表示する。

⑦ NEXT [\triangle n] n : 省略のとき 1

現在表示されている文献から n 文献進めた文献内容を端末に表示する。

⑧ BACK [\triangle n] n : 省略のとき 1

現在表示されている文献から n 文献戻した文献内容を端末に表示する。

⑨ ABSTRACT

現在表示されている文献の抄録を端末に表示するものであり、既に抄録が表示されているときは、次の抄録を表示する。

⑩ HARDCOPY

このコマンドは、キャラクタ・ディスプレイを用いているときに有効なコマンドで、現在表示されている文献内容あるいは抄録をそのままのイメージでタイプライタにコピーするものである。

⑪ PRINT set-no [, ABSTRACT]

既成（SELECT, COMBINE, LIMITによる）の文献集合のset-noを指定することにより、その集合全体の文献をジョブ終了後センタのライン・プリンタへバッチで出力させるものである。このとき抄録が要るかどうか指定することができる。

⑫ ENDSEARCH

一連の検索作業の停止を示すもので、今迄の検索で作成された情報は全て消去される。

再び検索を行いたいときには、SELECTコマンドを用いれば良い。

⑬ { TEACH
 { TEACH { command-name }
 { set-no } }

このコマンドは、3種のガイダンス機能を有し、JOLDOR-II全般的な説明、コマンドの使用方法、文献集合を構成するKEYWORDが表示される。

⑭ ¥BYE

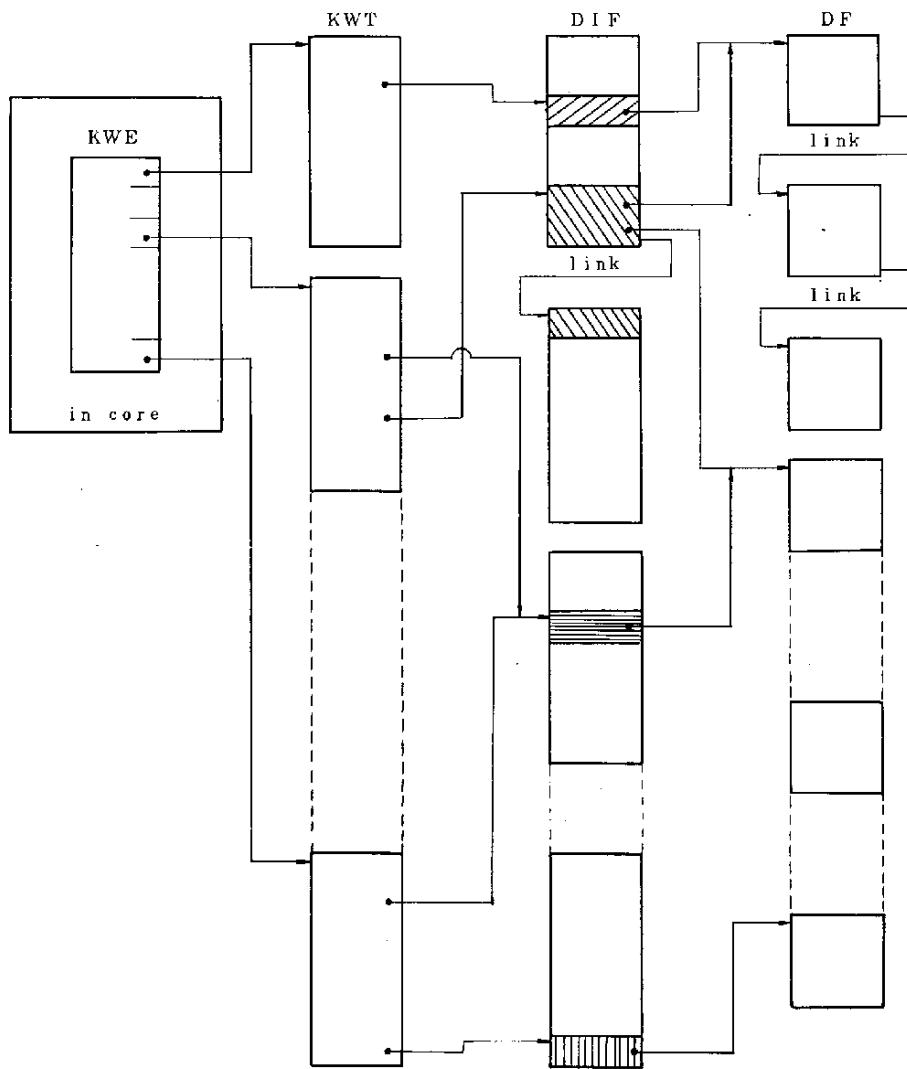
一連の検索作業の終了を示すものであり、JOLDOR-IIを用いての検索はここで終了する。

4) ファイル構成とその役割

a. ファイル構成

JOLDOR-IIシステムは、そのファイル構成に限ればJOLDOR-Iシステムと根本的に相違はない。ただ、DIFとDFは、それぞれのファイ

ル内でリンクしている。



(ファイル構成図)

各ファイルの説明を以下に記す。

KWE: Keyword Entry table (固定長, ブロック化)

指定された KEYWORD が KWT のどのブロックに入っているか調べる為のテーブルでコア・メモリに常駐している。内容は, KWT の TTR が入っている。

KWT: Keyword Table file (固定長)

KEYWORD の各キャラクタを節とする、リスト構造のテーブルで各KEYWORD の終りには、DIFの対応するブロックのTTR及び相対アドレスさらに文献数が入っている。

DIF: Document Index File (固定長)

DF の見出しになるファイルで、各KEYWORD に対応する文献の集合を表わしており、DF の TTR が入っている。

DF : Document source data File (不定長)

文献のソース・データで、タイトル、ディスクリプター、著者名、文献番号等文献書誌事項が入っている。

尚、このファイルは抄録ファイルも兼ねているので文献に抄録があれば、全て収容されている。従がって、このファイルは複数個のセグメントがリンクしている。

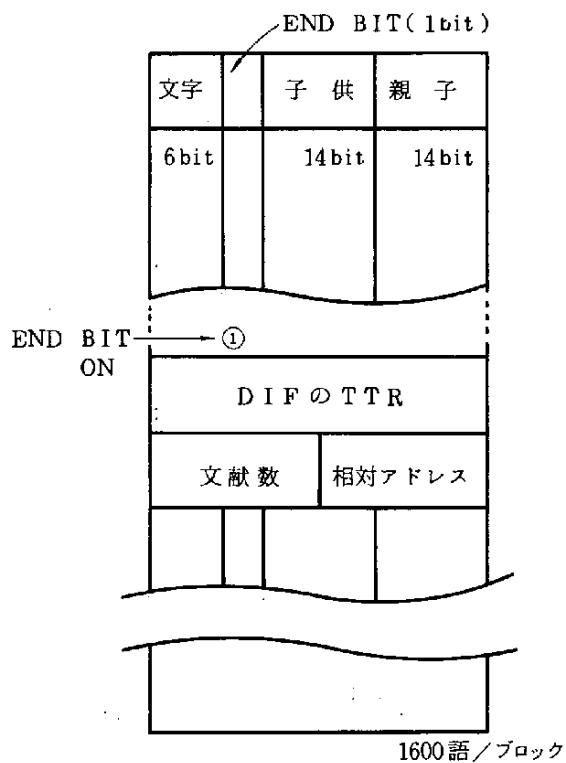
以下個々のファイルに関して具体的に述べる。

① KWE:

9語		1語
KEYWORD 1		TTR1
KEYWORD 2		TTR2

ここでいうKEYWORD とは、KWT 各ブロックに存在する先頭の KEYWORD であり、TTR はそのブロックのものである。このテーブルをサーチすることによって KWT のブロックが解る。

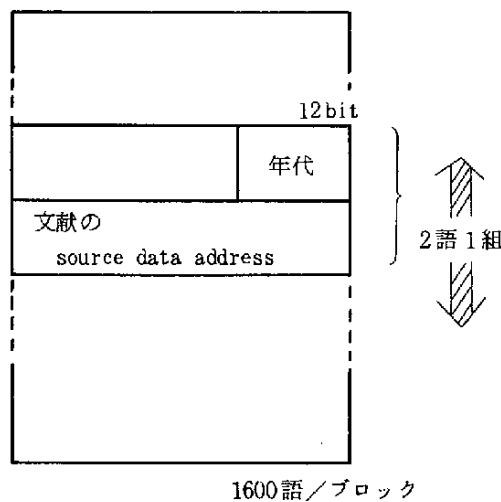
② KWT: このテーブルは前述のようにリスト構造になっており、KEYWORD の最後には、END BIT がオンになっている。さらにその後に、このKEYWORD を含む文献の見出し集合が存在



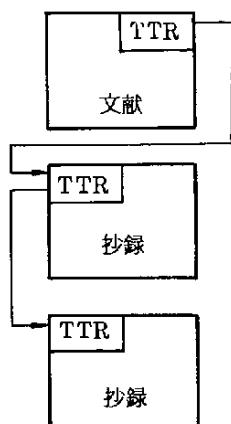
するアドレスが入っている。

DIFのTTRとは、文献見出し集合が存在するブロックのアドレスであり、相対アドレスとはそのブロック内のアドレスである。

- ③ DIF：年代には、文献の発行年代が入っておりLIMITコマンド使用の際、参照する。文献の source data address には、DFのTTRが入っている。このDIFは、KWTに入っている文献数の組($\times 2W$)が1つのグループとして並んでいる。



④ DF : 不定長最大 120 語 = 540 バイト

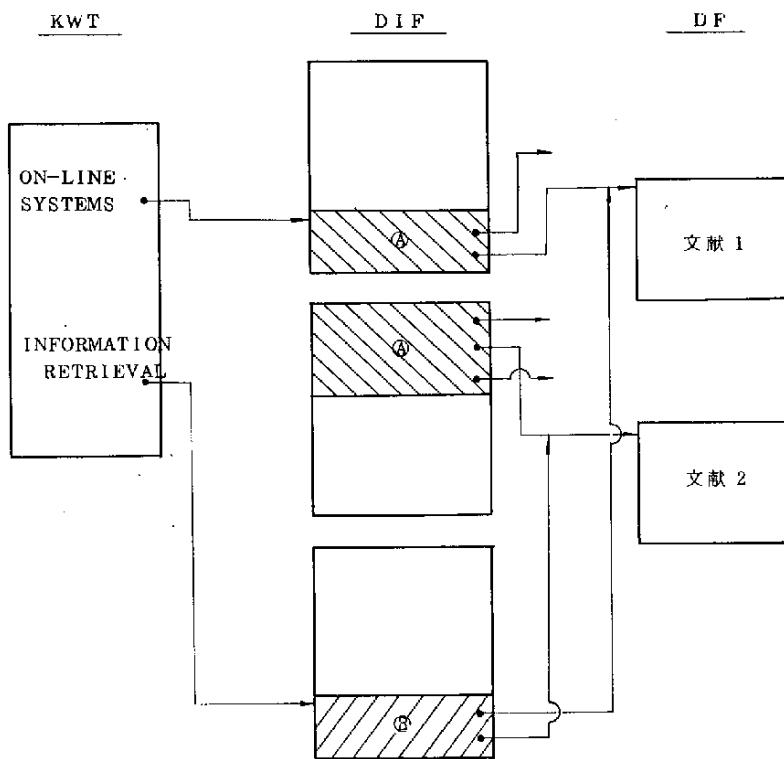


文献の TTR とは、この文献に抄録があるときにそのアドレスを示すものであり、抄録の TTR は、抄録が複数ブロック存在するときに用いるもので、その最終ブロックには 0 が入っている。

以上のファイル間の関係を以下に示す。

KEYWORD として

ON-LINE△SYSTEMS と INFORMATION△RETRIEVAL
両方含む文献を検索する場合。



A : ON-LINE \triangle SYSTEMS を KEYWORD にもつ文献見出し集合

B : INFORMATION \triangle RETRIEVAL を KEYWORD にもつ文献見
出し集合

論理演算の結果、B の内容と全く同じものが WORK file に作られて、
検索時に参照される。

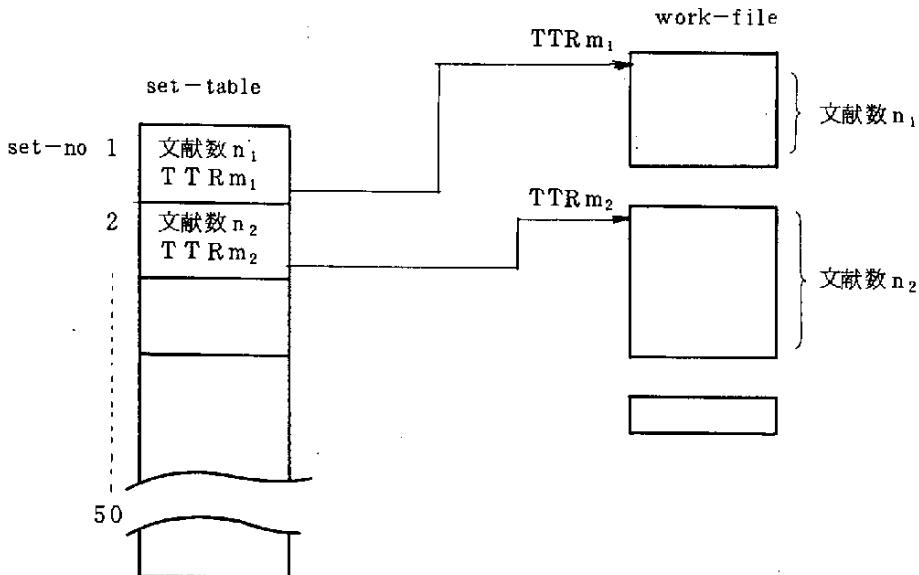
b. WORK FILE の役割

JOLDOR-IIでは、SELECT, COMBINE, LIMIT コマンドによって、次々と文献の見出しの集合をつくり出していく。既成の見出しの集合群は検索の途中で COMBINE, LIMIT, DISPLAY, PRINT コマンド等によって、再度参照される。従って、一度作成した集合は検索終了時迄保存しておく必要がある。これらの文献の見出しの集合群の保存に用いられるのが work file である。

1. work file の内容

内容そのものは、 DIF と全く同じである。

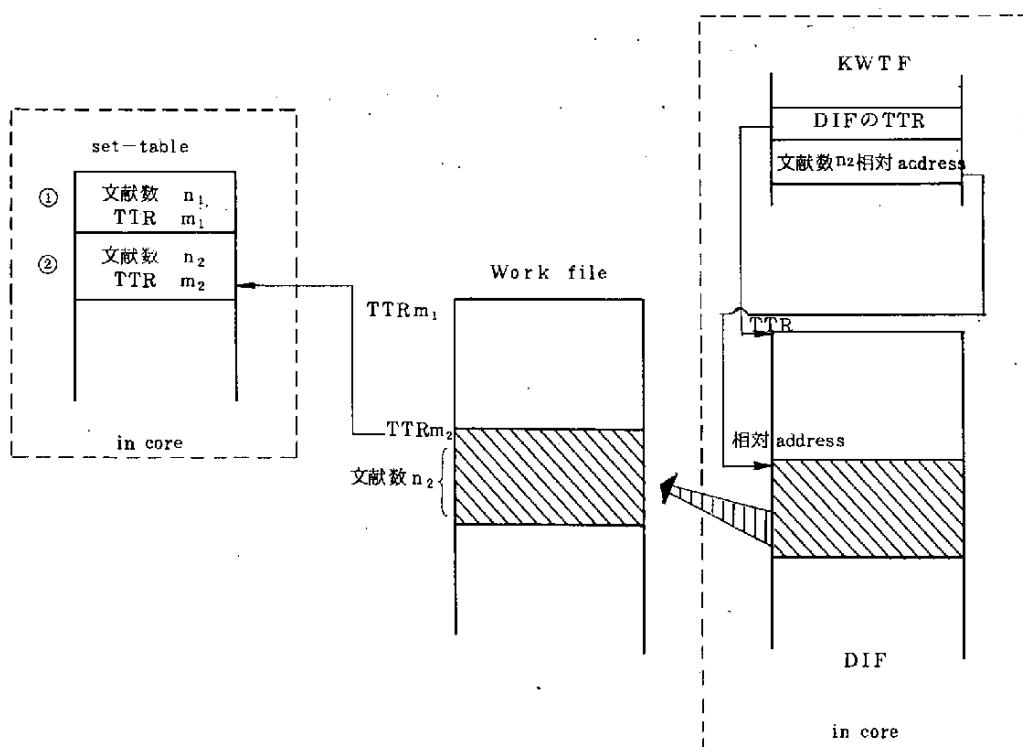
set-table (SELECT, COMBINE, LIMIT コマンドで作成された文献集合が、 work file のどこに書いてあるかを示す table) との関係を下に図示する。



c. 各コマンド実行時の work file の使われ方

1. SELECT コマンド

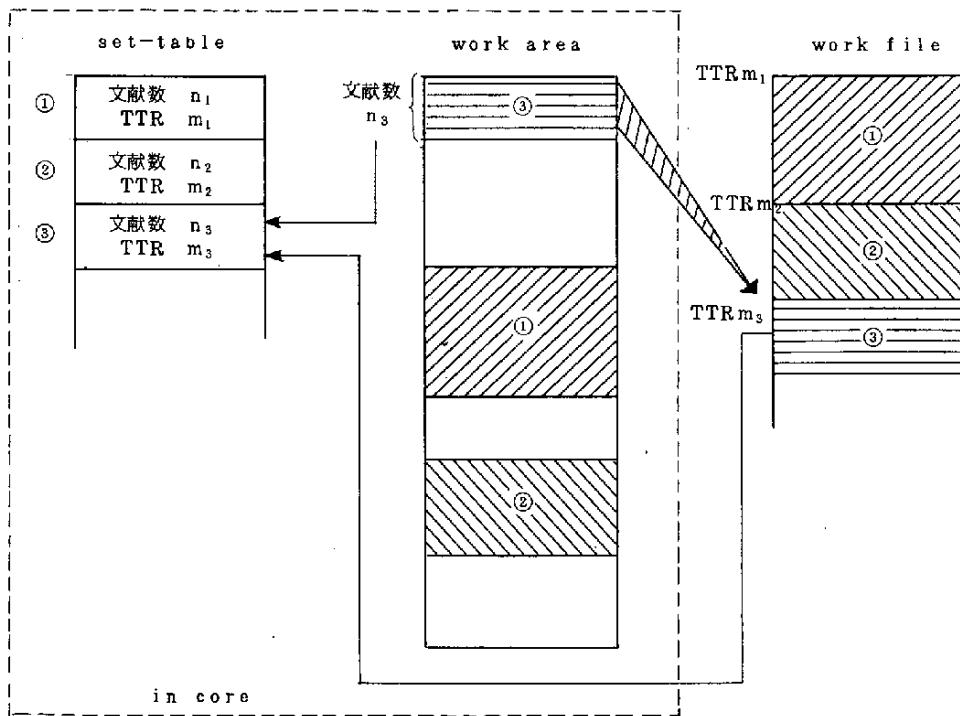
SELECT の処理は、指定された KEYWORD が、どの KWT にあるか、 KWE をサーチすることによって調べ、 KWT を読み込み DIF のアドレスを調べるとともに、 set-no と文献数を表示する。さらに、 DIF を読み込み該当見出し集合をそのまま work file に書き込むと同時に set-table へ文献数と work file のアドレスを登録する。



2. COMBINE コマンド

ex. COMBINE Δ 1 * 2

COMBINE の処理は、指定された文献集合に演算を施すもので、読み込みエリアは各ユーザ毎にある work area である。演算結果も work area に出来る。その文献数及び set-no を表示すると共に set-table へ文献数と work file のアドレスを登録する。

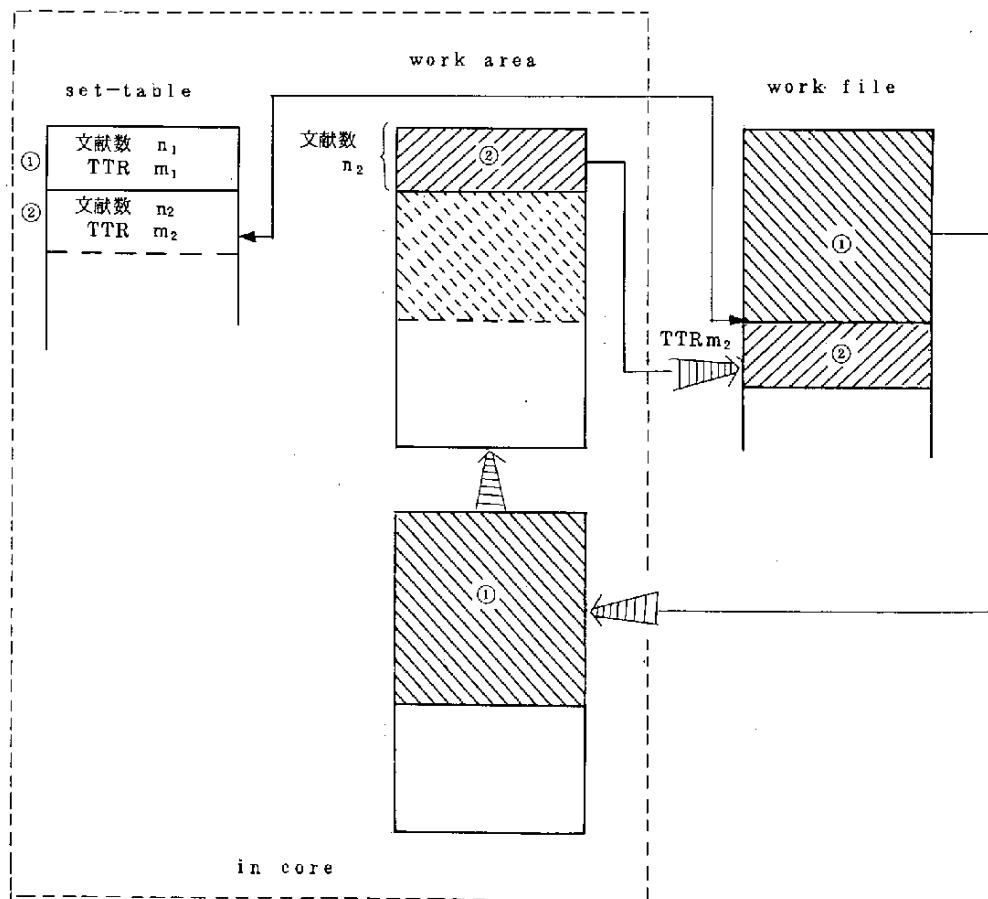


work area にある③が演算結果である。

3. LIMIT コマンド

LIMITの処理は、指定された文献集合に、発行年代の制限を加えるもので、読み込みエリアは各ユーザ毎にある work area である。結果も work area に出来る。その文献数及び set-no を表示すると共に、set-table へ文献数と work file のアドレスを登録する。

ex. LIMIT△1△69-72

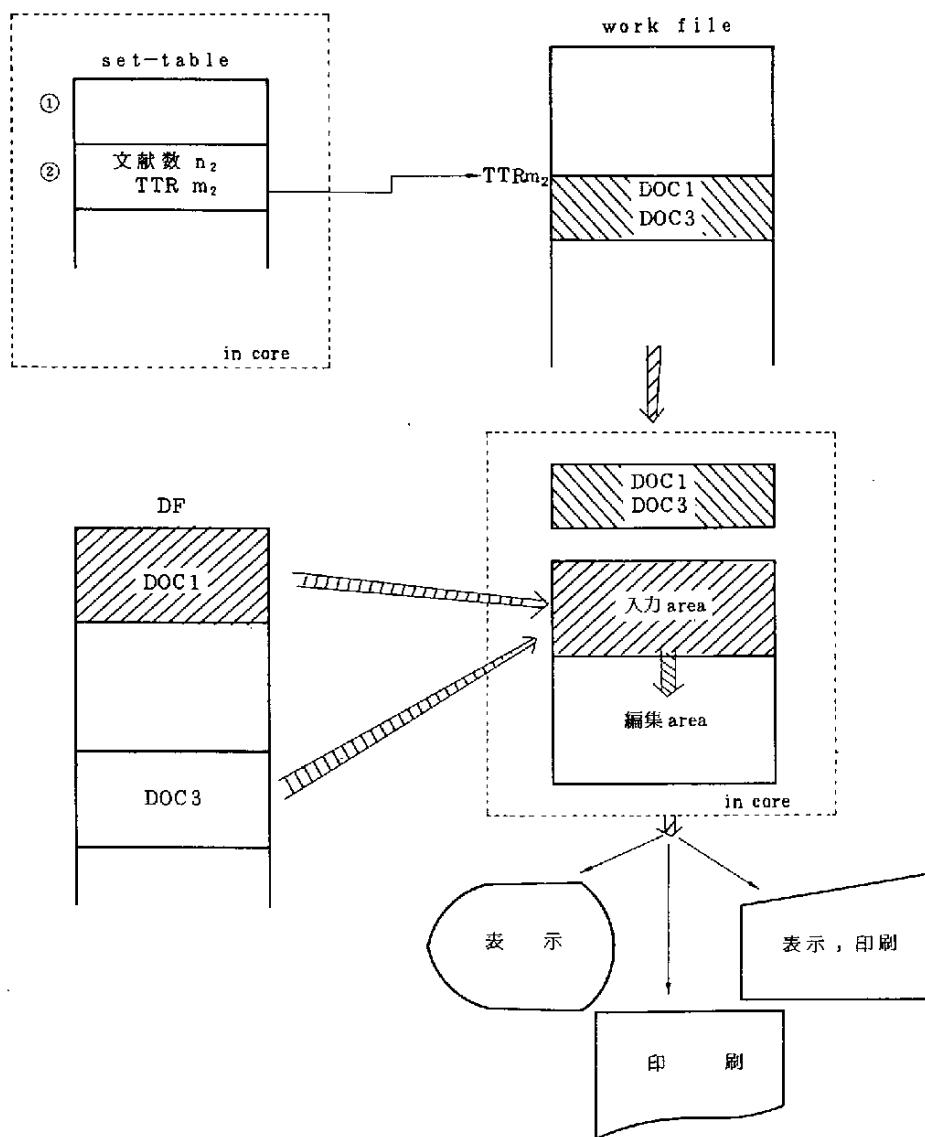


結果は、読み込んだ内容を上へつめるような形でできあがる。

4. DISPLAY, HARDCOPY, PRINT コマンド

これらの処理は、指定された文献集合を work file から、各ユーザ work area に読み込んで編集し表示、出力する。（NEXT, BACK, ABSTRACT もほぼ同様）

ex. DISPLAY Δ 2



5. ENDSEARCH コマンド

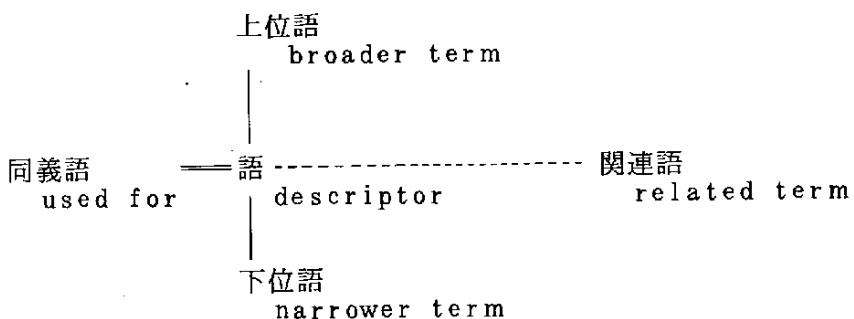
ENDSEARCH の処理は、一連の検索作業を停止させ、前述の set-table, work file 等をクリアして初期状態に戻す。これによってユーザは最初から検索できることになる。

6. このプログラムは、リエントラント構造になっており複数のユーザが单一のプログラムを参照することができ work file, work area はジョブ起動時にダイナミックに各ユーザ毎に与えられ、ジョブ終了後解放される。

8.3.7 今後の問題

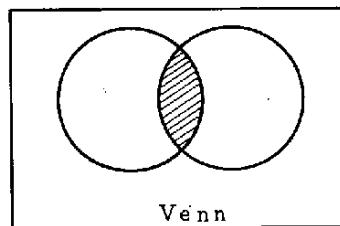
(1) シソーラス

検索効率の向上に欠くことのできないものにシソーラスがある。一般に以下ののような体型をもつ。



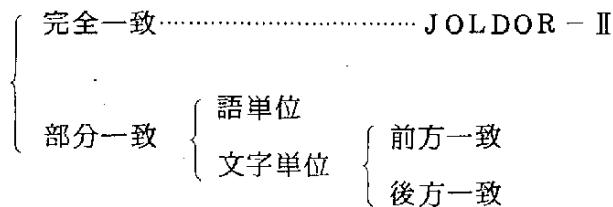
シソーラスの自動化は、かなり難解なもので、現在でも、その大部分が人労に頼っている。

JOLDOR-IIでは、JOLDOR-I時代に実験した関連語処理を参考にして関連語表を作成し、検索時における有効な手段としたい。また、この表を作成する際、重要なポイントである相関係数を求める式に、何れを採用するかが問題である。前の実験では、一番ポピュラーなVennの図より、そのオーバ・ラップの割合を求めていた。



(2) KEYWORD マッチング（語尾処理を含めて）

検索時におけるKEYWORD のマッチングは、現在完全一致を採用しているが、その他以下のものが考えられる。



仮に部分一致を採用するとすると、JOLDOR - II のKEYWORD の指定方法が複合語も含んでいるので、語単位の部分一致の方向で進めるべきである。

例えば、以下のようになる。

現在 KEYWORD; INFORMATION△RETRIEVAL

完全一致のみ採用

部分一致を採用した場合

KEYWORD; INFORMATION△RETRIEVAL

完全一致

部分一致

部分一致

何れか該当すれば採用

この部分一致を採用すると、再現率・適合率とも上昇が期待され、語尾処理の問題も幾分解決すると予想されるが noise の増加も充分考えられる。又、KEYWORD を単語に分割した場合、先頭から 5 ~ 6 文字が一致すれば、ほとんど同義であるという実験例もある。

(3) 検索方式

一般的に次の 3つが考えられる。

① 論理関係による方法 (AND, OR, NOT)

最も普及している検索方法で、ユーザが指定した KEYWORD によって作成された文献集合を、論理 2 項演算によって進める方法である。

JOLDOR - II もこれを採用し、今後もこれをベースにして進めていく予定である。

② 重みづけによる方法 (weighting)

各 KEYWORD , 著者, 分野, 年代等に重みをつけて、ある一定以上の重みだけのものを該当文献とする方法である。

③ Pre-search

あらかじめ、地域、分野、年代等で分類しておく方式で、データ量が膨大のときは非常に有効である。

JOLDOR - II には、直接関係ないが、特に MT ベースの検索システムのときは役立つであろう。

以上 3 ケースと性質が異なるが、次のようなものもある。

④ Historical Information

ユーザが最近の検索傾向を見たいときに役立つもので、KEYWORD の参照傾向、文献の参照傾向を定期的に配布したり、一部検索時に参照可能にするものである。

これを採用する場合ファイル構造の一部修正がある。

8.3.8 JOLDOR - II システムの検討及び反省

- (1) JOLDOR - II システムは、JOLDOR - I (1968 年開発開始) を前身としているが、今日 (1973.1) on-line 文献検索 system に対して、一応の区切りがついたと思われる所以歴史を振り返ると同時に、反省を含めて JOLDOR - II と JOLDOR - I との比較検討を行ってみる。

年 月

1968	6	JOLDOR - I 調査開始
	1970	2 JOLDOR - I 完成
	6	FACOM, TSS 運転開始と同時に, JOLDOR - I, TSS で運転開始
1971	11	
	?	関連語に関する実験
	12	
1972	4	JOLDOR - I 検討, データ再分析
	6	JOLDOR - II システム設計開始
	7	ファイル設計及び作成
	9	検索部分設計開始
	1973	1 JOLDOR - II 完成, 運転開始
	2	JOLDOR - IIIへの機能追加検討開始

(JOLDOR-II と JOLDOR-I との相違点)

項目	JOLDOR-II	JOLDOR-I
一度に指定できる KEYWORDの個数	1個	5個迄
KEYWORD	ディスクリプタからだけ採用。 複合語採用。	タイトル・ディスクリプタから 採用。語に分割。
蓄積KEYWORD 数	約 30,000	約 15,000
蓄積文献数	約 30,000 (年間)	約 7,000／ファイル
FILE関係	抄録ファイル追加	ナシ
検索装置	キャラクタ・ディスプレイ、タ イプライタ何れでも可能。	キャラクタ・ディスプレイのみ。
検索機能	ABSTRACT, LIMIT コマ ンド採用。 PRINT コマンドに抄録のオプ ション追加	ナシ ナシ
名称変更	HARDCOPY (旧TYPE)	
ガイダンス	完成 (TEACHコマンド)	一部起動
検索時における文 献集合の大きさ	無制限 (演算も可)	500位迄
割込み機能	追加	ナシ

この表で解るように、 JOLDOR-II と JOLDOR-I の大きな相違点は、
LIMIT コマンドと抄録ファイルの追加に伴う ABSTRACT コマンドの追
加、データ・ベースの容量の増大への対処及びタイプライタ端末装置単独で
も検索可能になったことである。

しかし、このタイプライタ端末装置での検索は、JOLDOR-I 当初の設計思想に大きな比重を占めている“検索時間の速さ”に背くような結果になる。オンライン文献検索の終極的目標が man-machine interaction であるといつてもその中枢をなしているのは、言うまでもなく、リアル・タイムである。従って、TSS 端末としてのキャラクタ・ディスプレイの開発が一日も早く望まれる。

(2) 反省事項

a. ファイル設計（大記憶）

文献集合の大きさ制限を撤廃したので、ファイル設計の一部修正を行った。

① KWE KWT のブロック数に比例して増えるが、現在 1 年分で約 300 レコードである。

無駄なスペースは皆無に近い。

② KWT 1 ブロック 1600 語 (1 トラック) で各トラックの無駄なスペースは $(1 - \frac{7200 - 52}{7294})$
最大 1.8 % である。

③ DIF 1 ブロック 1600 語 (1 トラック) で各トラックの無駄なスペースは $(1 - \frac{7200 - 45}{7294})$
最大 1.8 % である。

しかも DIF は、複数個のブロックがリンクされている。オンライン、アップディットを避けたことからスペース効率の大幅な上昇が得られた。

④ DF DF は ABSTRACT (抄録) ファイルをも含んでいて、しかも各セグメントがリンクしている。その上不定長である。各トラックの無駄なスペースは最大 7.4 % である。 $(\frac{540}{7294})$
KWT, DIF に比べて DF のスペース効率が著しく低いのは、POINT マクロとの関係で不定長でトラック・オーバ・ホールの機能がないためである。将来、この制限が解除されれば、

この値も減ると思われる。蓄積文献数が増加すればする程、その影響が大きい。文献数の増加によるスペース管理に影響があるのは、このDFだけである。

b. On-Line Processor

- ① LIMITコマンド採用に伴いDIFを一部修正した。
- ② 抄録ファイル追加に伴いABSTRACTコマンド採用。将来、抄録が増えたときには、抄録だけのNEXT BACKが必要となるかも知れない。
- ③ ガイダンス(TEACHコマンド)機能完備。
初心者に対する検索時の補助手段が充実した。
- ④ オンライン・アップディットを避けた理由は、一般にファイルの更新と検索効率は相矛盾する性質のものであり、この種のシステムでのデータベースの更新は、頻度的にもバッチで充分であり検索効率を犠牲にしてまでもオンライン化せねばならぬ性質ではないと判断したためである。
- ⑤ man-machine interactionに充分留意して進めてきたが、その限りでは目的は達せられたと思う。

c. JOLDOR-IIとMONITOR-Vとの関連

1. TSSとの兼ね合い

- ① 検索プログラム作成上の問題点

このプログラムは、オンラインで起動させるものであるが、デバッグは、システム起動時の会話部分を除いて全部バッチで行った。

それは、オンラインによるデバッグの効率が悪くなる恐れがあったからである。仮にオンラインでデバッグしたとすると、その時間、労力はバッチに比べて数倍に及んだと思われる。

尚、検索時における対話部分は、あらかじめカードにパンチしライン・プリンタへ結果を出力する方法を採用した。しかし、途中の論理展開に誤りが生じたとき、それ以降の処理が正常に行えず苦労した。

このような方法によれば、かなり入念なman-machine interactionを含むオンライン・システムであってもバッチ・モードでは

ば十分なディバッグができる。

2. ファイル・アクセス

File Generator から On-Line Processor までの作成を通じて FACOM-230-60 のデータ管理が非常に不安定であったので、その為のロス・タイムが相当あった。

① DF 作成に関して（不定長）

- i) 1つのファイルを2個の FCB で入出力 を分けて用いる際、 XTENDA, POINT マクロ、トラック・オーバー・フロー指定 等、少し条件が重なるとソフト・エラーになるので苦労した。
- ii) 最終的には、1個の FCB でアクセス時に FCB の中味を自分で書き換え、トラック・オーバー・フローを避けることによって一応解決した。

② WORK-FILE 作成に関して（不定長）

- i) 当初は2個の FCB で入出力を分けていたが、結局種々の障害により DF と同様 FCB は1個になった。
- ii) このWORK file は close する前に FORMAT WRITE , UPDATE WRITE, XTENDA MACRO, DIRECT READ を行うので DF よりも複雑になっているのが大きな原因の1つである。
- iii) ii) の 4つのマクロの兼合いがうまくいかなかったので、処置としては DF と同様にして処理した。

この1年間にレポート未提出のも含めると障害件数は9件に及んだ。大部分は、未だ修正されていない。

3. 利用上の問題点

現在 response time は、約 1 ~ 2 秒以内であるので '待たされる' という感じはない。しかし現在使用可能な端末（6台）が全部動き、しかもセンターのバッヂジョブが2個動いているときは約 4 ~ 5 秒となり、かなり遅くなる。参考迄に現在の slice time は 100 ミリ秒である。

又、タイプライタ端末に出力するとき各行ごとにロール・イン,ロール・アウトが行われている様なので、この為のロス・タイムも連続出力のときには、無視できない。現段階では、行間待時間は、平均1～2秒である。

検索時のコマンド指定誤りのリカバリーは充分チェックしてあるので問題はないが、ユーザによる端末の誤操作のリカバリーは難しく端末のハード的な問題も含んでおり、使いやすい端末機器の開発が望まれる。

誤操作のリカバリーはF1520M型端末が特に悪い。

マシン・ダウン(端末ダウンも含めて)によるリカバリーは現在何もその手段がない。少くとも次の様な機能が望まれる。ダウン時点のテーブル、ファイル関係をモニター側でセイブしてリストア時に再現出来る様にする。

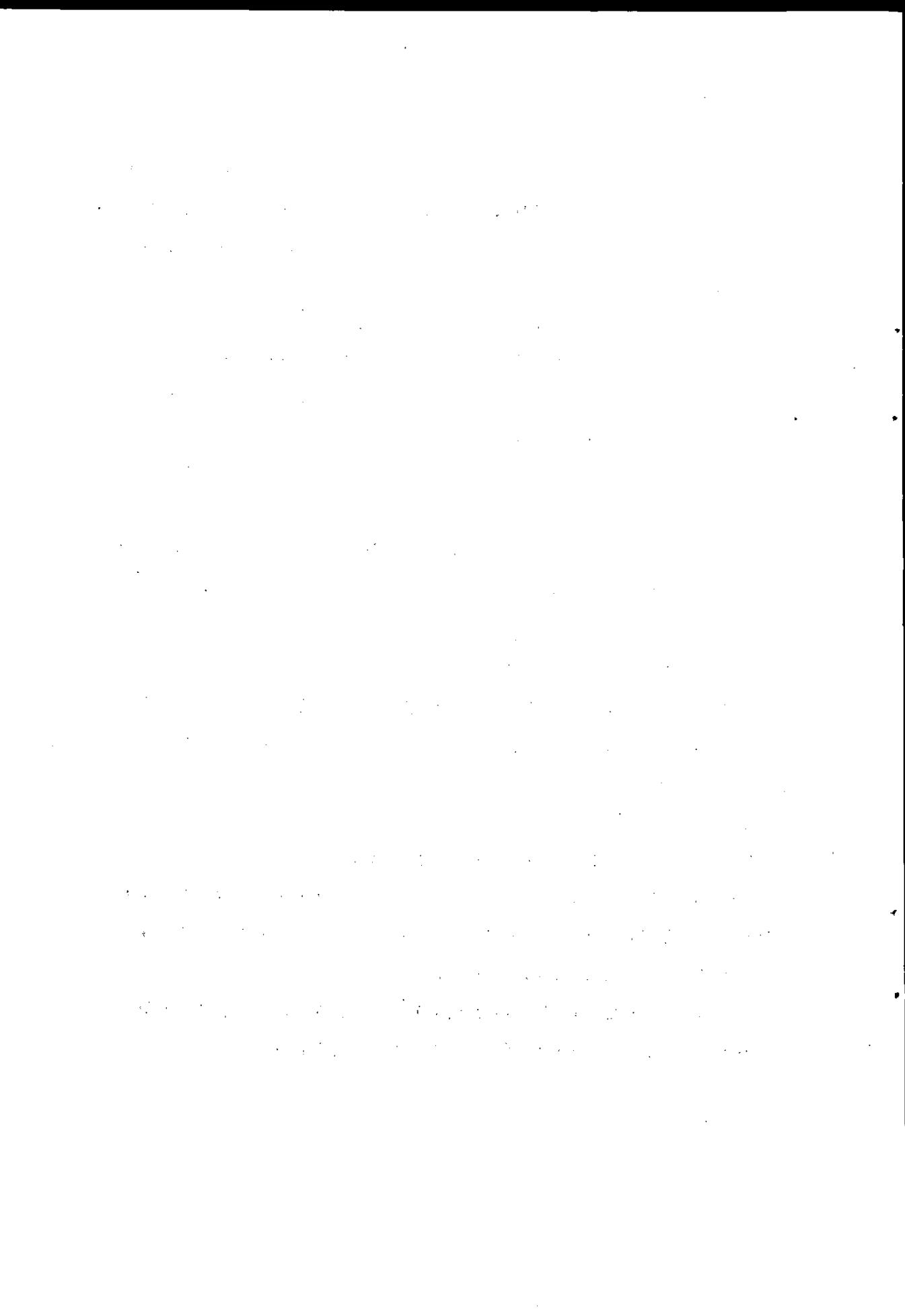
8.3.10 おわりに

足掛け6年(実動3年)に渡り、研究・実験を重ねてきた、このオンライン文献検索システムも一応区切りがつき、次はシソーラス・KEYWORDの語尾処理・部分matching、頻度範囲等の論理的な処理に関する研究に入るべきである。

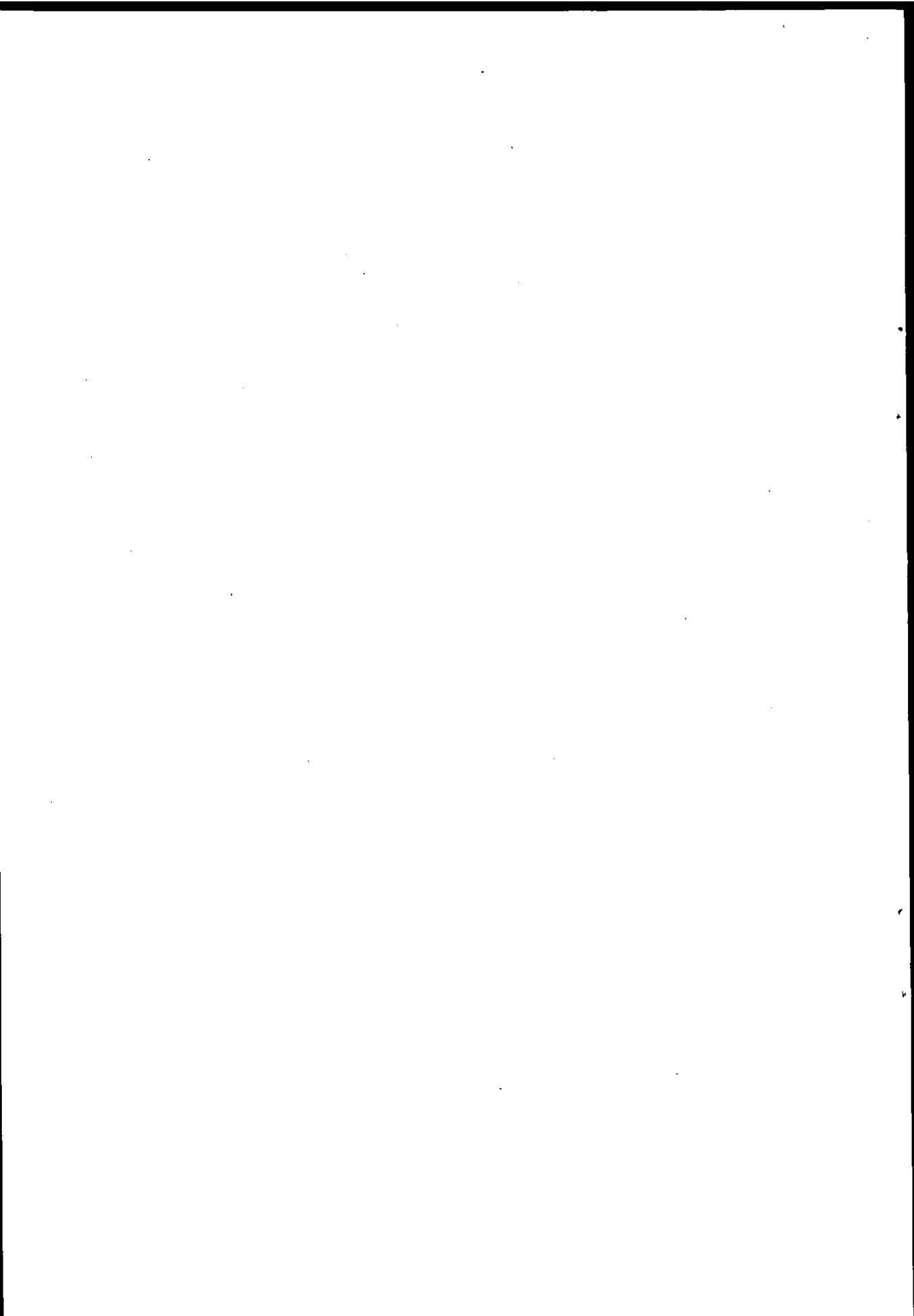
これらは、過去の実験を含めて、既に検討を進めている。

情報検索の新しい手法は実験的に生れることが多いあるが、理論的な解明も必要であり、現実的なサービスをする一方、他方では相当つっこんだ研究をしていく必要があり今後この方向に進みたい。

ここで我々設計者は、情報検索の原点に戻って情報検索の持つ意味、何故社会が情報検索を求めているのか深く検討すべきであると思う。



付録—I FACOM利用の手引書



付録一 I FACOM 利用の手引書

第 1 章 概 要

1. ハードウェアの構成

(本報告書の図 2-1 と重複するので省略)

2. ソフトウェアの構成

(本報告書の図 2-3 と重複するので省略)

3. ファイル割り付けおよび各種標準値

当センタの FACOM 230-60 のファイル割り付け、および各種標準値は下記のとおりです。

ファイル割付

項 目	機 番
1) MONITOR デバイス	C. 000, C. 002(代替用)
2) 処理プログラム	B. 000
3) ジョブ制御マクロファイル	B. 000
4) SYSIN(CR)	E. 220(Auto Mount)
5) SYSOUT(LP)	E. 230(Auto Mount)
6) 寸借ファイル	B. 002, B. 003, A. 106
7) WORKファイル	B. 002, B. 003, A. 106 D. 000~D. 006 A. 100 } A. 101 } デマンド用 A. 102 }
8) 共用ファイル	A. 103 } バッチ用 A. 104 }
	A. 105 リモートバッチ用
9) 持ち込みファイル	A. 107

標準値

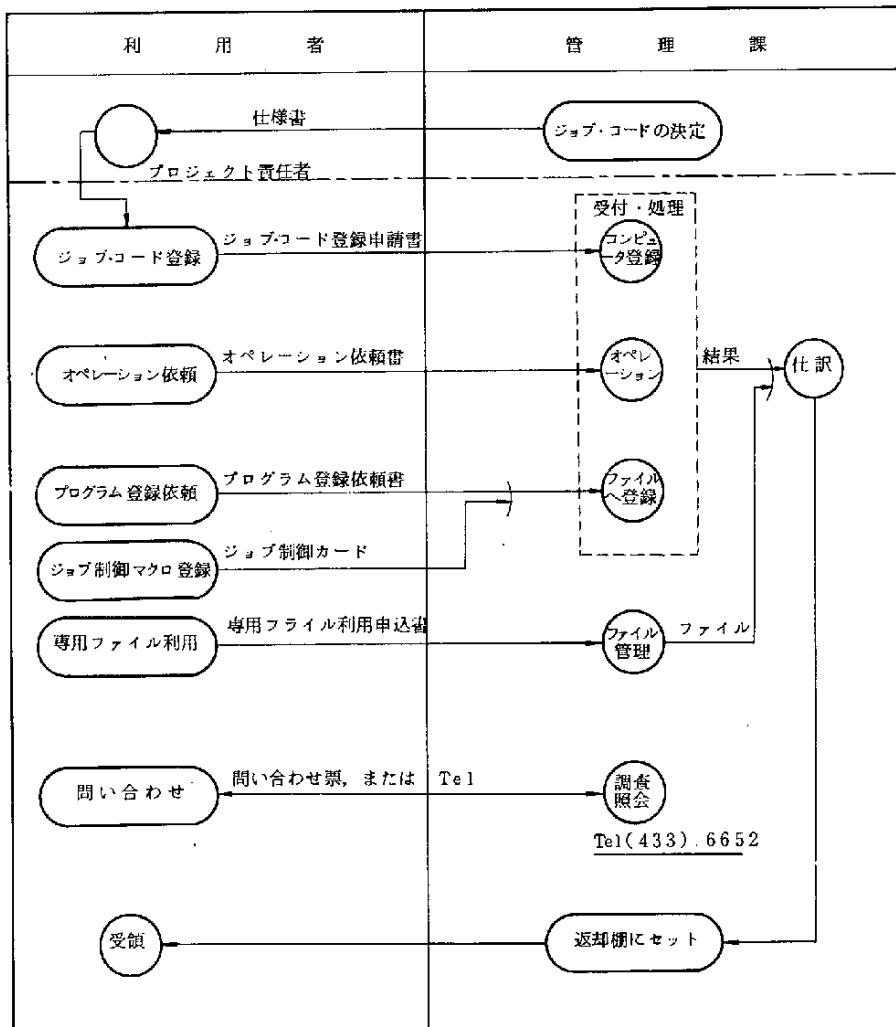
項目	標準値
1) CPU 占有時間	5分
2) CORE 占有時間	60分
3) SYSOUT 打ち切り行数	10,000行
4) SYSOUT 打ち切り頁数	200頁
5) SYSOUT 打ち切り枚数	10,000枚
6) 端末出力打ち切り行数	1,000行
7) 最大 CORE 使用数	40 KW 以内

第 2 章 クローズ制

1. 利用基準の概要

本項では、クローズ制を利用するためには必要な事項についての利用手順を説明します。

利用の手順の流れ図



2. ジョブ・コードの登録・閉鎖について

ジョブ・コードは、業務が発生するとその業務につけられるコードであり、管理課がコードを決定し業務の責任者に連絡いたします。

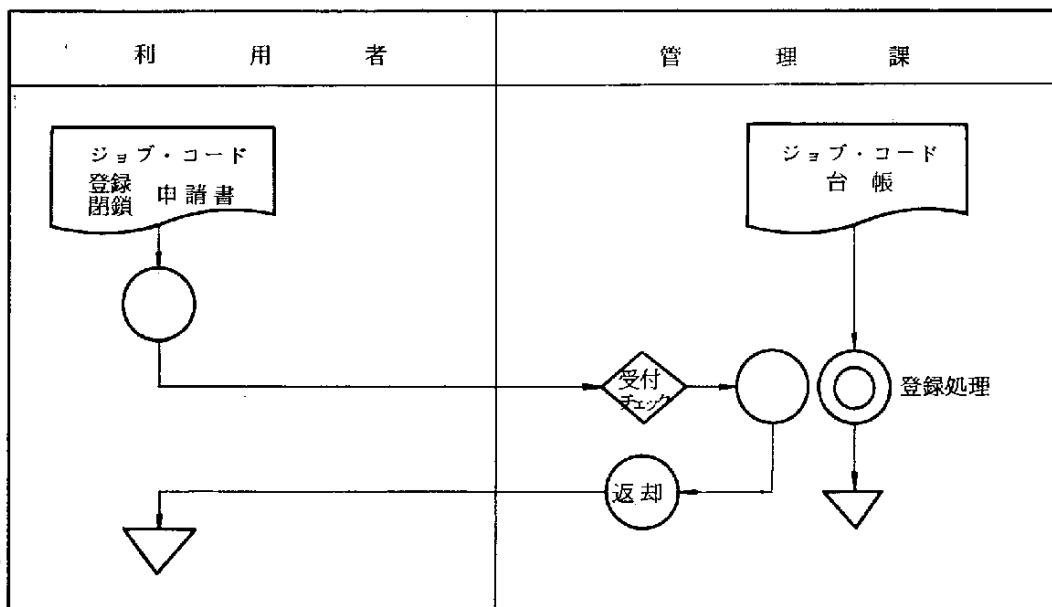
1) 登 録

業務の開始にあたり、電子計算機を利用する場合にはジョブ・コード登録申請書を管理課へ提出し、ジョブ・コードを登録しなければなりません。

2) 閉 鎖

業務が完了した場合は、直ちにジョブ・コード閉鎖申請書を管理課へ提出しなければなりません。なおジョブ・コードが閉鎖されると、このジョブ・コードでは電子計算機を利用することはできません。

利用の手順の流れは次のとおりです。



ジョブコード { 登録
閉鎖 } 申請書

年 月 日

利用責任者の所属		
利用責任者名		電話
業務名称		
ジョブコード		
使用言語		
業務内容の概要、その他		

※欄は記入不要です。

※受付者	
------	--

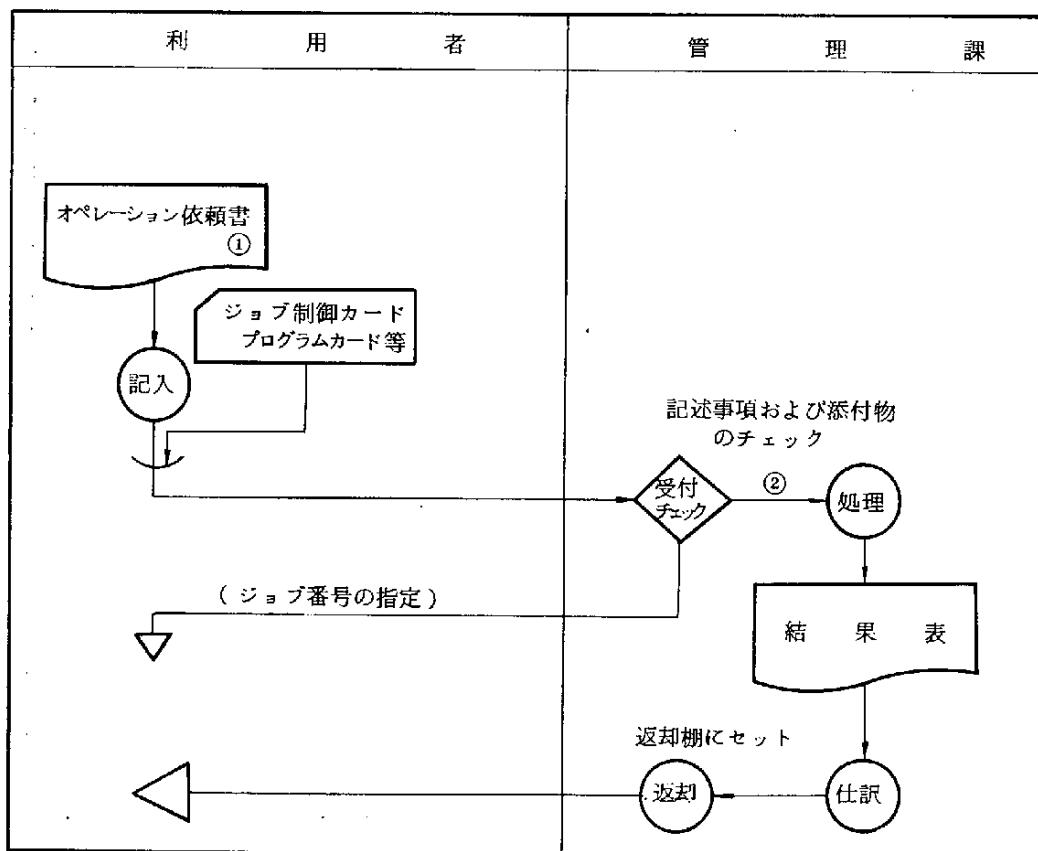
4. オペレーションの依頼について

オペレーションの依頼をする場合には次に述べる各項目が完全に満足していなければなりません。

- (1) オペレーション依頼書が完全に記述してあること。
 - (2) 持ち込み、または登録ファイルの指示のあるものは必ずそのファイルがあること。
 - (3) ジョブ制御言語（NO 文は不要）が必ず付加されていること。

(注) JOB文の登録番号はジョブ・コードを、利用者名は利用者氏名12文字を、必ず記入して下さい。

上記項目の1項目でも満足していない場合はクローズの対象とはなりません。
利用の手順の流れは次のとおりです。



使用機種名	F · N · H
-------	-----------

クローズ・オペレーション依頼書

月 日 Z番号

依頼者名			所 属	部 課	電 話	
ジョブ・コード			推 定 メモリー・サイズ	K	推 定 所用時間	分
持込みMT数	巻					
MT番号又は ラベル名 ボリューム通番		機番	機番	機番	機番	
ライン・プリンタ 出 力	汎用	P	推 定 出力ページ数 PAGE	カード・パンチ	枚	
	定型			紙 テ ー プ	単位	

処理概要(該当するものに○印を付けて下さい。その他のものは下空欄に詳細を書いて下さい)

1. COMPILE.
2. LINKAGE.
3. RUN.
4. INITIAL.

※ オペレーション報告欄

※受付者		※オペレータ名	
------	--	---------	--

5. ジョブ制御マクロの登録、修正、廃棄について

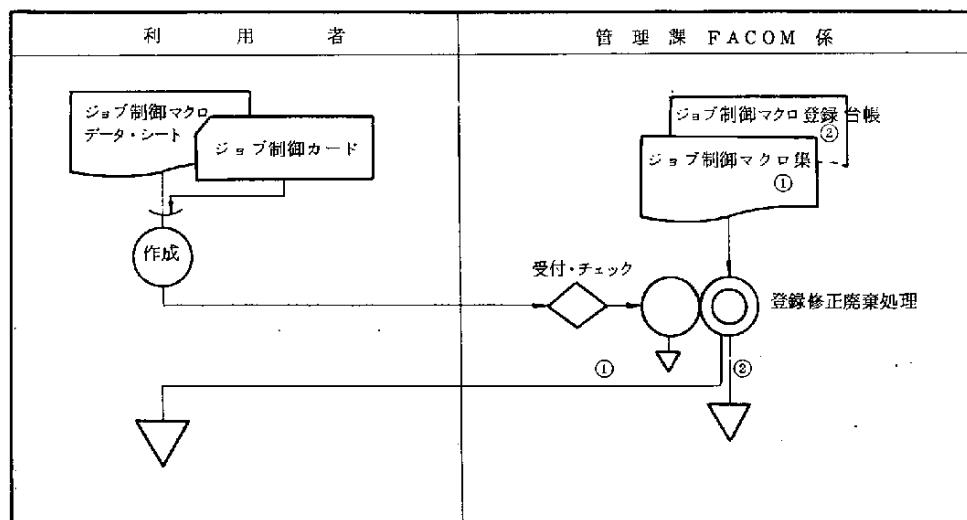
利用者はジョブ制御マクロの登録、修正、廃棄の申請を管理課 FACOM 係に行なうことができます。

登録：利用者はその使用に先立ってジョブ制御カードとジョブ制限マクロデータシートを管理課 FACOM 係に提出して下さい。

修正：登録されているジョブ制御マクロを修正する場合は、修正ジョブ制御カードとジョブ制御マクロデータシートを管理課 FACOM 係まで提出して下さい。

廃棄：登録されているジョブ制御マクロが必要でなくなった場合は、その内容を管理課 FACOM 係に提出して下さい。

なお標準ジョブ制御マクロはジョブ制御マクロ集に掲載し、関係部所に配布いたします。



DATA SHEET

6. プログラム登録について

FACOM利用者の作成したジョブはジョブ登録依頼書等を管理課FACOM係に申請することによりセンター・ファイルに登録することができます。これは、定期的にプロダクションを行なう必要があるジョブについて管理課FACOM係に一括管理を依頼する場合です。

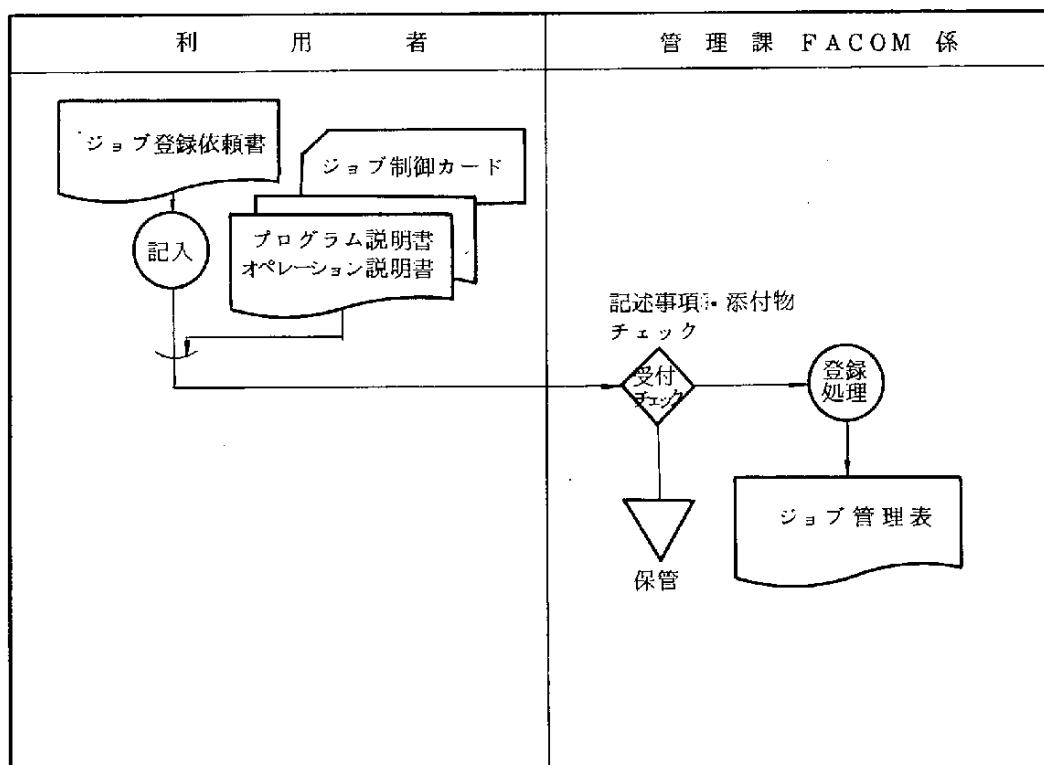
ただし、登録するためには次の条件を満足していなければなりません。

(1) デバックが完全に終っているもの

(2) プログラム説明書、オペレーション説明書がそろっているもの

上記条件の1項でも満足していない場合はこの対象にはなりません。

利用の手順の流れは次のとおりです。



プログラム登録依頼書

年 月 日

*欄は記入不要です。

※受付者		登録者	
※登録年月日			
※登録ファイルボリューム番			

7. 専用ファイルの利用について

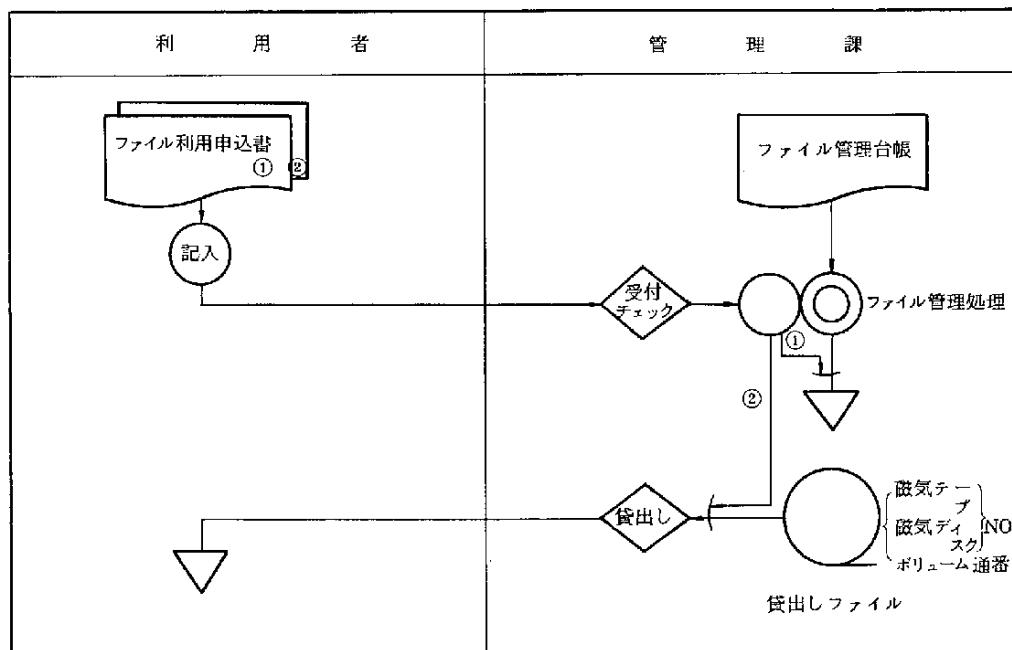
多量のプログラムやデータをファイルするためには専用のディスク・パックおよび磁気テープを利用することができます。

専用ファイルの利用を希望する場合は、利用に先だって専用ファイル利用申込書に必要事項を記述し、管理課に提出し承認を得て下さい。

なお、専用ファイルはジョブ依頼の都度、計算機室の受付に持参する方法と管理課 FACOM 係に保管を依頼しておく方法とがあります。

この指示は専用ファイル利用申込書にそのいずれかを記述して下さい。

利用の手順の流れは次のとおりです。



専用ファイル利用申込書

年 月 日

※欄は記入不要です。

※受付者	
※担当者	

8. クローズ制における制限および要望事項について

T. S. S の運用サービス時間帯のクローズ制において次のような制限事項および要望事項があります。

1) 制限事項

- (1) 各種の優先権の変更は許さない
- (2) 磁気テープ、磁気ディスクのフィジカル・ユニットの指定は許さない

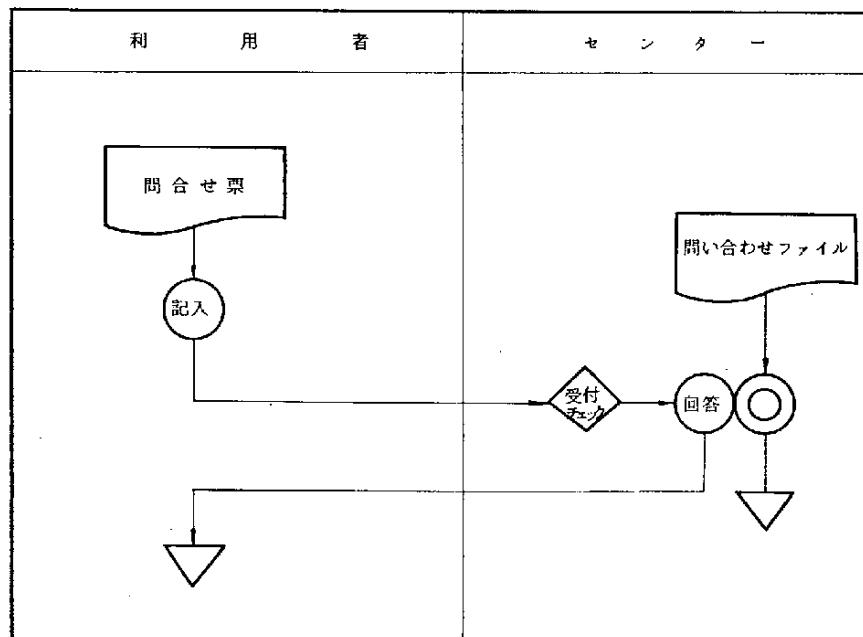
2) 要望事項

- (1) プリントファイルについては SYSOUT 指定が望ましい
- (2) コア・メモリーの使用量は 32 KW(HCM) 以内が望ましい
- (3) JOB 制御マクロの使用が望ましい
- (4) CPU の使用時間は 5 分以内が望ましい

9. 問い合わせについて

クローズ処理依頼者が計算処理結果に対しての疑義を生じた場合、また、センターに対して運用上の問題、その他いろいろな意見やお気づきの点がある場合などについて問合せ票に必要事項を記入の上、提出して下さい。なお、電話による問い合わせもできます。

利用の手順の流れは次のとおりです。



※

▲

問合せ票

年 月 日

社名 _____

所属 _____

氏名 _____

受付番号※	受付者※	種類	方法
		1.質問 2.要望 3.その他	1.来所 2.書面 3.電話

(質問内容) :

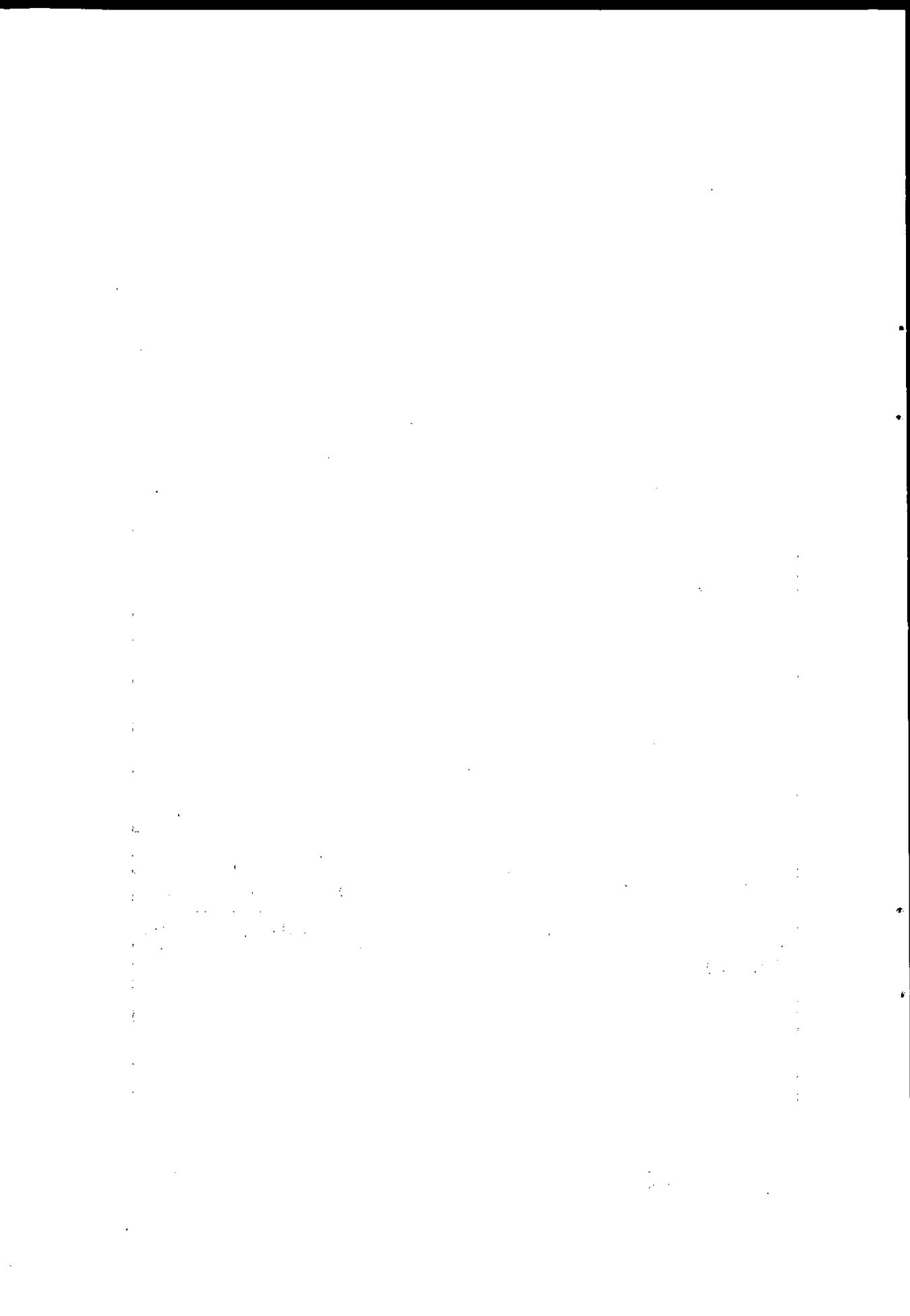
書類返送

要・否

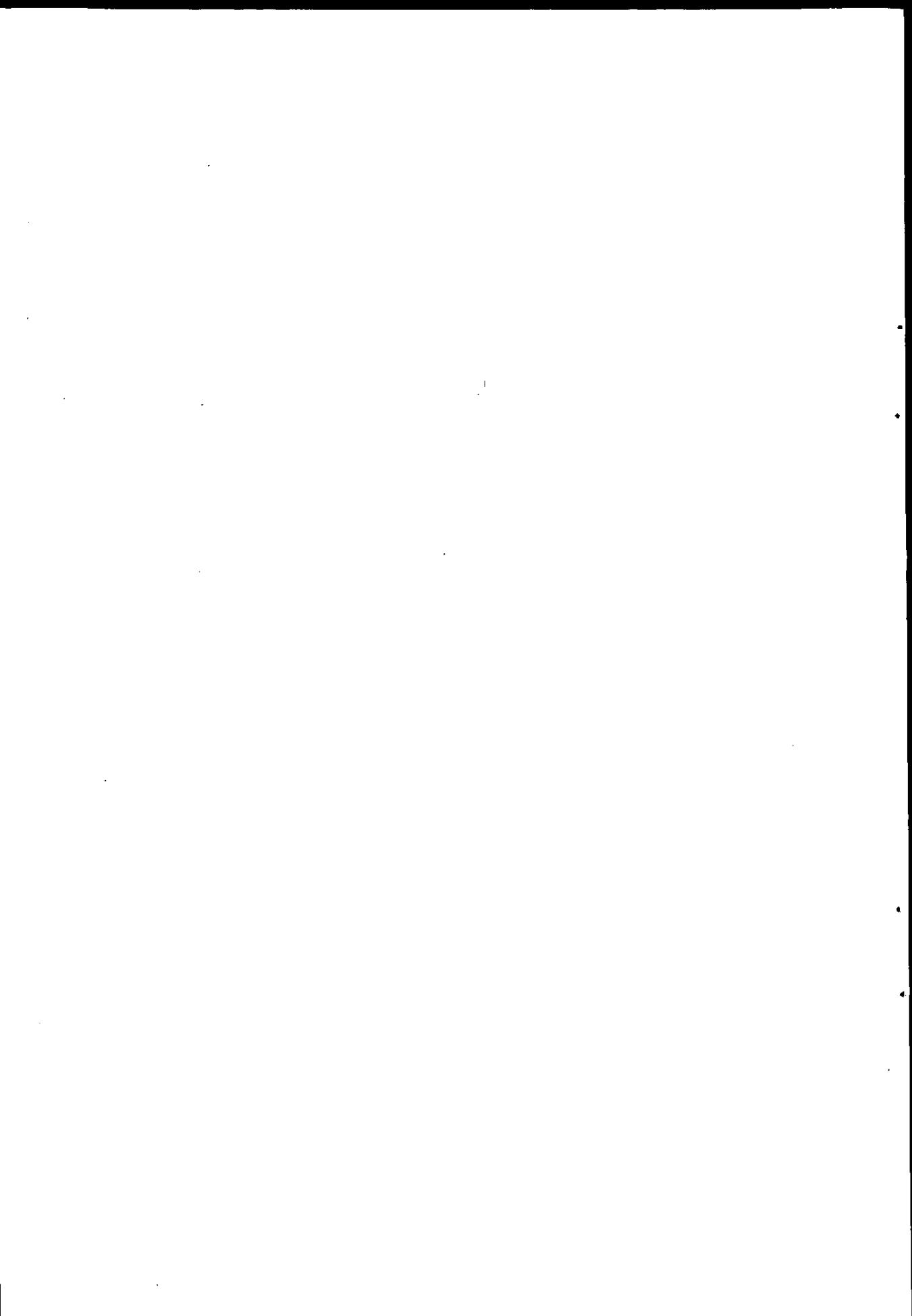
電話番号	回答年月日※	回答者名※	回答区分※
	年 月 日		1.解決 2.検討 3.打切

(解答) :

※欄は記入不要です。



付録-II ジョブ制御マクロ登録集



付録一Ⅱ

JOB 制御マクロ登録集

1 制御文の書き方

1.1 標準入力形式

ジョブ制御文は、図1のような内訳をもつ 80欄カード形式（標準入力形式）に記入してシステムに投入される。

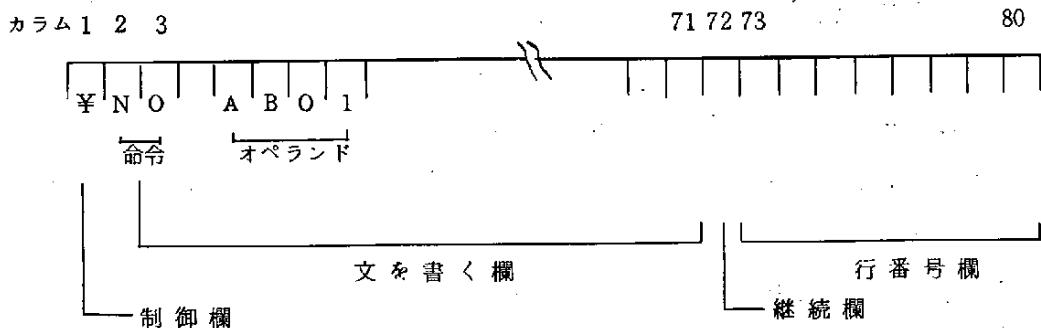


図 1

説明

制御欄（第1カラム）

文字¥を記入してジョブ制御文であることを示す。

文を書く欄（第2-71カラム）

命令とオペランドを書く。

継続欄（第72カラム）

文が次の行に継続するとき、空白以外の1文字を記入して継続していることを示す。

行番号欄（第73-第80カラム）

任意のものを記入してよい。

1.2 ジョブ制御マクロの呼び出し

ジョブ制御マクロの呼び出し形は次のとおりである。

	命 令	オペランド
¥	ジョブ制御マクロ名	実パラメータの並び

ジョブ制御マクロ名

・ジョブ制御マクロ登録ユーティリティで登録された名前。

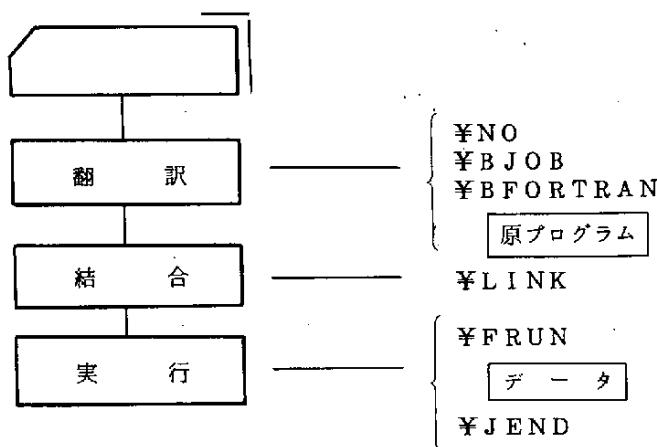
実パラメータの並び

実パラメータの並びとは、位置パラメータおよびキー・パラメータの並びである。位置パラメータとは、左から数えた位置によって、その意味が定まるものであり、キー・パラメータは、キーによってその意味が定まるものであり、これを書く順序については、位置パラメータに先行してはならない。実パラメータを指定しなければ、登録の際、決めた仮パラメータの値がとられる。

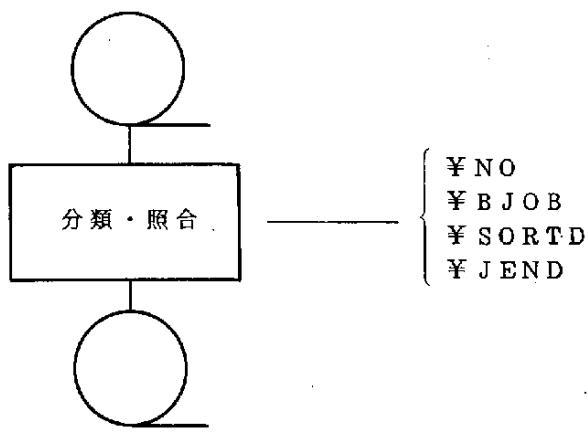
命令とオペランドは1つ以上の空白をあける。

1.3 ジョブ制御カードの並び

1. FORTRAN 翻訳 結合 実行 (3ステップ/1ジョブ)



2. SORT・MERGE (1ステップ／1ジョブ)



2 ジョブ制御マクロ

2.1 BJOB

(1) 機能

ローカル・バッチ・ジョブの全体的性格を表わす。

(2) 形式

¥BJOB 登録番号, 利用者名,

$\left\{ \left(\begin{array}{c} \text{条件下限}, \text{条件上限} \\ \hline 1, 511 \end{array} \right) \right\}$, $\left\{ \left(\begin{array}{c} \text{処理時間} \\ \hline 5.00 \end{array} \right) \right\}$,

$\left\{ \left(\begin{array}{c} \text{磁心時間} \\ \hline 60.00 \end{array} \right) \right\}$, $\left\{ \left(\begin{array}{c} \text{ページ数} \\ \hline 200 \end{array} \right) \right\}$, $\left\{ \left(\begin{array}{c} \text{行数} \\ \hline 10000 \end{array} \right) \right\}$,

$\left\{ \left(\begin{array}{c} \text{カード枚数} \\ \hline 10000 \end{array} \right) \right\}$, $\left\{ \left(\begin{array}{c} \text{ERROR} \\ \text{LIST} \\ \text{DETAIL} \end{array} \right) \right\}$

- (3) パラメータの説明 <> 内はキー
- = 第1パラメータ = <REG>
この位置に管理課から指定されたジョブ・コードを書く。このパラメータは必ず書くこと。
 - = 第2パラメータ = <USER>
この位置に利用者の名前を、英字や数字や点から成る12字以内の文字の列で書く。このパラメータは必ず書くこと。
 - = 第3パラメータ = <COND>
このジョブを構成する各ジョブ・ステップの実行が完了した時点で、後続のジョブ・ステップの実行を続行するか打切るかの判定条件を与える。
条件を満たせば後続のジョブ・ステップを実行しない。

条件下限	0～511までの符号のない整数。 *は0と等価
条件上限	0～511までの符号のない整数。 *は511と等価

条件を満たすとは

(条件下限 ≤ 完了コード ≤ 条件上限)

省略されたときは条件下限1、条件上限511とする。

- = 第4パラメータ = <CPU>
ジョブを遂行するのに必要な処理装置使用時間の予想値を書く。

処理装置使用時間	予想される計算時間を、 m m m m · s s の形式で書く。 mは分、sは秒を表わす。
----------	--

省略されたときは5.00 すなわち5分間である。

- = 第5パラメータ = <CORE>
ジョブを遂行するのに必要な磁心記憶占有時間の予想値を書く。

第4パラメータと同形式で書く。

省略されたときは 60.00 すなわち 1 時間である。

= 第6パラメータ = < PAGE >

システム出力としてラインプリンターで印刷する予想ページ数を、 5 桁以内の符号のない整数で書く。

省略されたときは 200 ページとする。

= 第7パラメータ = < LINE >

システム出力としてラインプリンターで印刷する予想行数を、 5 桁以内の符号のない整数で書く。

省略されたときは 10000 行とする。

= 第8パラメータ = < CARD >

システム出力として、カード穿孔機で穿孔する予想カード枚数を、 5 桁以内の符号のない整数で書く。

省略されたときは 10000 枚とする。

= 第9パラメータ = < ML >

ジョブ制御文を、どの程度詳しく出力すべきかを書く。

ERROR	入力したジョブ制御文にエラーがあった場合、またはモニタが処理プログラムを異常終了させた場合に、制御文をリストする。正常の場合にはリストしない。
LIST	入力したジョブ制御文をそのままリストする。ジョブ制御マクロの展開形はリストしない。
DETAIL	入力したジョブ制御文、更に、それがジョブ制御マクロ文であればその展開形もリストする。

いずれの場合も、システム入力データそのものはリストされない。

省略されたときは LIST とする。

(4) 例

¥BJOB 0512, KANRIKA,, 10.00,,, , DETAIL

登録番号 0512
利用者名 KANRIKA
処理時間 10分
メッセージ DETAIL

2.2 D JOB

(1) 機能

デマンド・ジョブの全体的性格を表わす。

(2) 形式

¥DJOB 登録番号 , 利用者名

(3) パラメータの説明

= 第1パラメータ =

この位置に管理課から指定されたジョブ・コードを書く。このパラメータは必ず書くこと。

= 第2パラメータ =

この位置に利用者の名前を、英字や数字や点から成る12字以内の文字の列で書く。このパラメータは必ず書くこと。

注意：バッチ・ジョブでは使用してはならない。

2.3 R JOB

(1) 機能

リモート・バッチ・ジョブの全体的性格を表わす。

(2) 形式

¥RJOB 登録番号 , 利用者名 ,

$\left[\left\{ \begin{array}{c} \text{条件下限, 条件上限} \\ \hline 1,511 \end{array} \right\} \right] , \left[\left\{ \begin{array}{c} \text{処理時間} \\ \hline 5.00 \end{array} \right\} \right] ,$

$\left[\left\{ \begin{array}{c} \text{磁心時間} \\ \hline 60.00 \end{array} \right\} \right] , \left[\left\{ \begin{array}{c} \text{ページ数} \\ \hline 200 \end{array} \right\} \right] , \left[\left\{ \begin{array}{c} \text{行数} \\ \hline 10000 \end{array} \right\} \right] ,$

$\left[\begin{array}{c} \text{カード枚数} \\ \hline 10000 \end{array} \right] , \left[\begin{array}{c} \text{端末打切り行数} \\ \hline 1000 \end{array} \right] ,$

$\left[\begin{array}{c} \text{ERROR} \\ \text{LIST} \\ \hline \text{DETAIL} \end{array} \right]$

(3) パラメータの説明 <> 内はキー

= 第1パラメータ = <REG>

この位置に管理課から指定されたジョブ・コードを書く。このパラメータは必ず書くこと。

= 第2パラメータ = <USER>

この位置に利用者の名前を、英字や数字や点から成る 12字以内の文字の列で書く。このパラメータは必ず書くこと。

= 第3パラメータ = <COND>

このジョブを構成する各ジョブ・ステップの実行が完了した時点で、後続のジョブ・ステップの実行を続行するか打切るかの判定条件を与える。条件を満たせば後続のジョブ・ステップを実行しない。

条件下限	0～511までの符号のない整数。 *は0と等価
条件上限	0～511までの符号のない整数。 *は511と等価

条件を満たすとは

(条件下限 ≤ 完了コード ≤ 条件上限)

省略されたときは条件下限 1, 条件上限 511とする。

= 第4パラメータ = <CPU>

ジョブを遂行するのに必要な処理装置使用時間の予想値を書く。

処理装置使用時間	予想される計算時間を、 m m m m · s s の形式で書く。 mは分、 sは秒を表わす。
----------	---

省略されたときは 5.00 すなわち 5 分間である。

= 第 5 パラメータ = < CORE >

ジョブを遂行するのに必要な磁心記憶占有時間の予想値を書く。

第 4 パラメータと同形式で書く。

省略されたときは 60.00 すなわち 1 時間である。

= 第 6 パラメータ = < PAGE >

システム出力としてラインプリンターで印刷する予想ページ数を、 5 行以内の符号のない整数で書く。

省略されたときは 200 ページとする。

= 第 7 パラメータ = < LINE >

システム出力としてラインプリンターで印刷する予想行数を、 5 行以内の符号のない整数で書く。

省略されたときは 10000 行とする。

= 第 8 パラメータ = < CARD >

システム出力としてカード穿孔機で穿孔する予想カード枚数を、 5 行以内の符号のない整数で書く。

省略されたときは 10000 枚とする。

= 第 9 パラメータ = < TLINE >

リモート・バッチ・ジョブの場合、システム出力として端末で印刷する予想行数を、 5 行以内の符号のない整数で指定する。

省略されたときは 1000 行とする。

= 第 10 パラメータ = < ML >

ジョブ制御文のリストを、どの程度詳しく出力すべきかを指定する。

ERROR	入力したジョブ制御文にエラーがあった場合、またはモニタが処理プログラムを異常終了させた場合に、制御文をリストする。正常の場合にはリストしない。
LIST	入力したジョブ制御文をそのままリストする。 ジョブ制御マクロの展開形はリストしない。
DETAIL	入力したジョブ制御文、更に、それがジョブ制御マクロ文であればその展開形もリストする。

いずれの場合も、システム入力データそのものはリストされない。

省略されたときはLISTとする。

(4) 例

¥RJOB 0512, KANRIKA, , , , , 2000

登録番号 0512

利用者名 KANRIKA

端末打切り行数 2000行

2.4 ALGOL

(1) 機能

ALGOLのコンパイル。

(2) 形式

¥ALGOL [{ 制御情報 }] , [{ ジョブ・ステップ名 }] ,
 [{ LCM }] , [{ ALGOL }] ,
 [{ PRINT }] , [{ REMOTE }] , [{ DIRECT }]

- (3) パラメータの説明 <> 内はキー
- = 第1パラメータ = <PARAM>
コンパイラーに伝えるべきパラメータ情報を記号列として書く。詳しくは別表1を参照。
省略されたときは LCM とする。
 - = 第2パラメータ = <STEP>
このジョブ・ステップにつけるべき名前である。同じジョブ内で同名のジョブ・ステップがあつてはならない。
省略されたときはALGOLとする。
 - = 第3パラメータ = <SYSOUT>
結果をジョブ終了後、システム出力装置から出力すべきことを指定する。

オプション	
PRINT	システム出力装置がセンター側のラインプリンターであることを示す。
REMOTE	システム出力装置が端末であることを示す。

- 省略されたときはPRINTとする。
- = 第4パラメータ = <DIRECT>
結果を直接センター側ラインプリンターに出力することを指定する。
省略されたときは指定なしとする。

(4) 例

¥ALGOL NOLIST, SEQ,, REMOTE

制御情報 NOLIST, SEQ

システム出力 端末

注意

第3パラメータと第4パラメータを同時に指定したときは、第4パラメータを有効とする。

2.5 BFORTTRAN

(1) 機能

FORTRAN のコンパイル。

(2) 形式

¥BFORTTRAN

$\left[\begin{array}{c} \text{制御情報} \\ \text{OPT} \end{array} \right] , \left[\begin{array}{c} \text{ジョブ・ステップ名} \\ \text{FORTRAN} \end{array} \right]$

$\left[\begin{array}{c} \text{PRINT} \\ \text{REMOTE} \end{array} \right] , (\text{DIRECT})$

(3) パラメータの説明 <> 内はキー

= 第1パラメータ = <PARAM>

コンパイラに伝えるべきパラメータ情報を記号列として書ぐ。

詳しくは別表2を参照。

省略されたときは OPT とする。

= 第2パラメータ = <STEP>

このジョブ・ステップにつけるべき名前である。同じジョブ内で同名のジョブ・ステップがあってはならない。

省略されたときは FORTRAN とする。

= 第3パラメータ = <SYSOUT>

結果をジョブ終了後、システム出力装置から出力すべきことを指定する。

オプション	
PRINT	システム出力装置がセンター側のライン プリンターであることを示す。
REMOTE	システム出力装置が端末であることを示す。

省略されたときは PRINT とする。

= 第4パラメータ =

< DIRECT >

結果を直接センター側ラインプリンターに出力することを指定する。

省略されたときは指定なしとする。

(4) 例

¥BFORTRAN OPT,MAP , , , DIRECT

制御情報 OPT,MAP

システム出力 直接指定

注意

第3パラメータと第4パラメータを同時に指定したときは第4パラメータを有効とする。

2.6 COBOL

(1) 機能

COBOLのコンパイル。

(2) 形式

¥COBOL [制御情報] , [ジョブ・ステップ名]
[LIST₁] , [COBOL] ,

[PRINT] , [DIRECT]
[REMOTE] ,

(3) パラメータの説明

<> 内はキー

= 第1パラメータ =

< PARAM >

コンパイラに伝えるべきパラメータ情報を記号列として書く。詳しくは別表3を参照。

省略されたときはLIST₁とする。

= 第2パラメータ =

< STEP >

このジョブ・ステップにつけるべき名前である。同じジョブ内で同名の

ジョブ・ステップがあってはならない。

省略されたときは COBOL とする。

= 第3パラメータ = <SYSOUT>

結果をジョブ終了後、システム出力装置から出力すべきことを指定する。

オプション	
PRINT	システム出力装置がセンター側のラインプリンターであることを示す。
REMOTE	システム出力装置が端末であることを示す。

省略されたときは PRINT とする。

= 第4パラメータ = <DIRECT>

結果を直接センター側ラインプリンターに出力することを指定する。

省略されたときは指定なしとする。

(4) 例

¥COBOL LIST, TRCHK

制御情報 LIST, TRCHK

注 意

第3パラメータと第4パラメータを同時に指定したときは、第4パラメータを有効とする。

2.7 FASP

(1) 機能

FASP のアセンブル。

(2) 形 式

¥FASP $\left[\begin{array}{c} \text{制御情報} \\ \text{LIST, NOSEQ} \end{array} \right]$, $\left[\begin{array}{c} \text{ジョブ・ステップ名} \end{array} \right]$,
 $\left[\begin{array}{c} \text{PRINT} \\ \text{REMOTE} \end{array} \right]$, (DIRECT), (PRGMAC),

$\left[\begin{array}{l} [LIBMAC] \end{array} \right], [ファイル名], \left[\begin{array}{l} \left\{ \begin{array}{l} \text{トラック数} \\ 20 \end{array} \right\} \end{array} \right],$

$\left[\begin{array}{l} \left\{ \begin{array}{l} \text{ボリューム通番} \\ \underline{2} \end{array} \right\} \end{array} \right]$

- (3) パラメータの説明 <> 内はキー
- = 第1パラメータ = < PAPAM >
 - ・アセンブラーに伝えるべきパラメータ情報を記号列として書く。詳しくは別表4を参照。
 - 省略されたときはLIST, NOSEQとする。
 - = 第2パラメータ = < STEP >
 - このジョブ・ステップにつけるべき名前である。同じジョブ内で同名のジョブ・ステップがあってはならない。
 - 省略されたときはFASPとする。
 - = 第3パラメータ = < SYSOUT >
 - 結果をジョブ終了後、システム出力装置から出力すべきことを指定する。

オプション	意味
PRINT	システム出力装置がセンター側のラインプリンターであることを示す。
REMOTE	システム出力装置が端末であることを示す。

- 省略されたときはPRINTとする。
- = 第4パラメータ = < DIRECT >
 - 結果を直接センター側ラインプリンターに出力することを指定する。
 - 省略されたときは指定なしとする。
 - = 第5パラメータ = < PRGMAC >
 - PRGMACファイルをつかうときに指定する。
 - 省略されたときは指定なしとする。

- = 第6パラメータ = <LIBMAC>
LIBMACファイルをつかうときに指定する。
省略されたときは指定なしとする。
- = 第7パラメータ = <FLNAME>
PRGMACあるいはLIBMACをつかうときは、必ずファイル名を指定しなければならない。ファイル名は、英字や数字や点から成る44字以内の文字の列でなければならない。
- = 第8パラメータ = <TRK>
PRGMACファイルを指定したときは、ファイル容量を書く。
省略されたときは、20トラックとする。
- = 第9パラメータ = <VOL>
PRGMACファイルあるいはLIBMACファイルの存在する大記憶装置のボリューム通番を指定する。
省略されたときは、2とする。

(4) 例

¥FASP,,, PRGMAC,, P·FL
 PRGMAC 使 用
 ファイル名 P·FL
 ト ラッ ク 数 20 ト ラッ ク

注 意

第3パラメータと第4パラメータを同時に指定したときは、第4パラメータを有効とする。

2.8 PL1

(1) 機 能

PL1 のコンパイル

(2) 形 式

¥PL1 $\left[\begin{array}{l} \text{制御情報} \\ \text{LCM=1} \end{array} \right]$, $\left[\begin{array}{l} \text{ジョブ・ステップ名} \\ \text{PL}_1 \end{array} \right]$

$\left[\begin{array}{l} \text{PRINT} \\ \hline \text{REMOTE} \end{array} \right] , \quad (\text{DIRECT})$

- (3) パラメータの説明 <> 内はキー
- = 第1パラメータ = <PARAM>
コンパイラーに伝えるべきパラメータ情報を記号列として書く。詳しくは別表5を参照。
省略されたときはLCM=1とする。
 - = 第2パラメータ = <STEP>
このジョブ・ステップにつけるべき名前である。同じジョブ内で同名のジョブ・ステップがあってはならない。
省略されたときはPL₁とする。
 - = 第3パラメータ = <SYSOUT>
結果をジョブ終了後、システム出力装置から出力すべきことを指定する。

オプション	意味
PRINT	システム出力装置がセンター側のラインプリンターであることを示す。
REMOTE	システム出力装置が端末であることを示す。

- 省略されたときはPRINTとする。
- = 第4パラメータ = <DIRDCT>
結果を直接センター側ラインプリンターに出力することを指定する。
省略されたときは指定なしとする。

(4) 例

¥PL ₁	NOLIST, FLAGZ
制御情報	NOLIST FLAGZ
注 意	

第3パラメータと第4パラメータを同時に指定したときは、第4パラメ

ータを有効とする。

2.9 LINK

(1) 機能

コンパイルした結果（相対形式プログラム）を集め、更に必要ならばライブラリーから適当な相対形式プログラムをつけ加えて、一つの実行可能なプログラムを作る。

(2) 形式

LINK [{ プログラム名 }] , [エントリーネーム] ,
[{ ジョブ・ステップ名 }] , [{ PRINT }]
[{ REMOTE }]
[{ ▼ 制御情報 ▼ }] , [{ NEW }]
[{ OLD }] , [{ CATLG }] ,
[{ TEMP }]
[{ ファイル名 }] , [{ ボリューム通番 }] ,
[{ E B }] , [{ 2 }] ,
[{ PASS }] , [{ 記憶領域 }]
[{ KEEP }] , [{ HCM }]
[{ DELETE }] , [{ SORT }] , [{ CATLG }]
[{ UNCATLG }]

(3) パラメータの説明

<>内はキー

= 第1パラメータ =

<PROG>

新しくできる実行形式プログラムに対して名前を与える。

英文字で始まる8文字以下の英字、数字又は点であること。

省略されたときはEXEPROMとする。

= 第 2 パラメータ = < ELM >

実行開始点を指示する。このパラメータはFORTRAN以外の言語では必ず指定する。しかしFORTRANにおいても特にエレメント文を入れたときは、その名前を書く。

省略されたときはFIMAINとする。

= 第 3 パラメータ = < STEP >

このジョブ・ステップにつけるべき名前である。同じジョブ内で同名のジョブ・ステップがあってはならない。

省略されたときはLINKとする。

= 第 4 パラメータ = < SYSOUT >

結果をジョブ終了後、システム出力装置から出力すべきことを指定する。

オプション	意味
PRINT	システム出力装置がセンター側のラインプリンターであることを示す。
REMOTE	システム出力装置が端末であることを示す。

省略されたときはPRINTとする。

= 第 5 パラメータ = < PARAM >

結合編集プログラム(LIED)に伝えるべきパラメータ情報を記号列として書く。詳しくは別表6を参照。

省略されたときは、MAP, LCM, WA=1とする。

= 第 6 パラメータ = < FLTYP E >

実行形式プログラムが入るファイル(EXELIBファイル)に関するものである。

オプション	意味
NEW	このジョブ・ステップではじめて作成するファイルである。
OLD	すでに作成済のファイルである。
CATLG	すでに作成済で、登録されているファイルである。
TEMP	このジョブの中だけで意味をもつ一時的なファイルである。

省略されたときは TEMP とする。

= 第 7 パラメータ = <FLNAME>

EXELIB ファイルに付ける名前である。

NEW, OLD, CATLG の場合

ファイル名は、英字や数字や点から成る 44 字以内の文字の列で表わす。

TEMP の場合

ファイル名は、英字や数字や点から成る 12 字以内の文字の列で表わす。

しかし TEMP の場合、書かれたファイル名は、実際のファイルの名前とはならない。

省略されたときは EB とする。

= 第 8 パラメータ = <VOL>

実行形式プログラムが格納されているファイルが存在する大記憶装置のボリューム通番を指定する。

省略されたときは 2 とする。

= 第 9 パラメータ = <DISP>

このパラメータは、ボリュームの取付けや取外し、ファイル保存や登録に関する指定を行なう。

オプション	意味
PASS	同じジョブの中の後続のジョブ・ステップからも、このファイルが利用されることを示す。
KEEP	このジョブ・ステップ終了時にこのファイルを保存する。
DELETE	このジョブ・ステップ終了時に、このファイルを消去する。
CATLG	このジョブ・ステップ終了時に、このファイルをファイル登録簿に登録し、かつファイルの保存手続きをとる。
UNCATLG	このジョブ・ステップ終了時に、このファイルをファイル登録簿から抹消する。ファイルそのものは消去されず、そのまま保存される。

省略されたときはPASSとする。

= 第10パラメータ = < SORT >

COBOLでソート・ルーチンを組込むときに指定する。

省略されたときは指定なしとする。

= 第11パラメータ = < DOMAIN >

記憶領域を定義する。パラメータを複数個書くときは、両端をカッコでかこむ。

例 (HCM, RW)

オプション	意味
HCM	この記憶領域が HCM にロードされるべきことを定義する。
LCM	この記憶領域が LCM にロードされるべきことを定義する。
RWX	読み書き実行自由な記憶領域であることを示す。
RW	データ用領域であることを示す。読み書き可能である。
RX	実行手続き用領域であることを示す。読み実行可能。

省略されたときは HCM とする。

(4) 例

¥LINK PRG1, PROG, , , NEW, P. LIB, , KEEP

プログラム名 PRG1

エントリー名 PROG

ファイル名 P. LIB (NEW, KEEP)

2.10 FRUN

(1) 機能

FORTRANの実行

(2) 形式

¥FRUN $\left[\begin{array}{l} \{\text{プログラム名}\} \\ \{\text{EXEPRO}\} \end{array} \right]$, $\left[\begin{array}{l} \{\text{ジョブ・ステップ名}\} \\ \{\text{FRUN}\} \end{array} \right]$,
 $\left[\begin{array}{l} \{\text{PRINT}\} \\ \{\text{REMOTE}\} \end{array} \right]$, $\{\text{DIRECT}\}$,
 $\{\text{PLOTTER}\}$, $\{\text{PUNCH}\}$.

- (3) パラメータの説明
- <>内はキー
- = 第1パラメータ = **<PROG>**
 実行プログラム名を書く。英字で始まる8文字以内の英字、数字又は点であること。
 LINKマクロの第1パラメータを指定したときは、その名前を書くが、省略したときは、このパラメータは省略する。
 省略されたときはEXEPROとする。
- = 第2パラメータ = **<STEP>**
 このジョブ・ステップにつけるべき名前である。同じジョブ内で同名のジョブ・ステップがあつてはならない。
 省略されたときは、FRUNとする。
- = 第3パラメータ = **<SYSOUT>**
 結果をジョブ終了後、システム出力装置から出力すべきことを指定する。
- | オプション | 意 | 味 |
|--------|----------------------------------|---|
| PRINT | システム出力装置がセンター側のラインプリンターであることを示す。 | |
| REMOTE | システム出力装置が端末であることを示す。 | |
- 省略されたときはPRINTとする。
- = 第4パラメータ = **<DIRECT>**
 結果を直接センター側ラインプリンターに出力することを指定する。
 省略されたときは指定なしとする。
- = 第5パラメータ = **<PLOTTER>**
 X-Yプロッターを使用するときに指定する。
 省略されたときは指定なしとする。
- = 第6パラメータ = **<PUNCH>**
 カード・パンチを使用するときに指定する。
 省略されたときは指定なしとする。

(4) 例

¥FRUN , , , PLOTTER

X-Yプロッター使用

¥FRUN PRG₁ , , , DIRECT

プログラム名 PRG₁

システム出力 直接指定

注 意

第3パラメータと第4パラメータを同時に指定したときは、第4パラメータを有効とする。

2.11 PRUN

(1) 機 能

PL₁ の実行

(2) 形 式

¥PRUN $\left[\begin{array}{l} \text{プログラム名} \\ \hline \underline{\text{EXEPRO}} \end{array} \right]$, $\left[\begin{array}{l} \text{ジョブ・ステップ名} \\ \hline \underline{\text{PRUN}} \end{array} \right]$,

$\left[\begin{array}{l} \underline{\text{PRINT}} \\ \hline \underline{\text{REMOTE}} \end{array} \right]$, DIRECT

(3) パラメータの説明 <>内はキー

= 第1パラメータ = <PROG>

実行プログラム名を書く。英字で始まる8文字以内の英字、数字又は点であること。

LINKマクロの第1パラメータを指定したときは、その名前を書くが、省略したときは、このパラメータは省略する。

省略されたときはEXEPROとする。

= 第2パラメータ = <STEP>

このジョブ・ステップにつけるべき名前である。同じジョブ内で同名の

ジョブ・ステップがあつてはならない。

省略されたときは、PRUNとする。

= 第3パラメータ =

<SYSOUT>

オプション	意 味
PRINT	システム出力装置がセンター側のラインプリンターであることを示す。
REMOTE	システム出力装置が端末であることを示す。

省略されたときはPRINTとする。

= 第4パラメータ =

<DIRECT>

結果を直接センター側ラインプリンターに出力することを指定する。

省略されたときは指定なしとする。

(4) 例

¥PRUN , PL. 1 , , REMOTE

ジョブ・ステップ名 PL. 1

システム出力 REMOTE

注 意

第3パラメータと第4パラメータを同時に指定したときは、第4パラメータを有効とする。

2.12 SORTD

(1) 機 能

作業用ファイルとして大記憶装置を使用してソートを行なう。

(2) 形 式

¥SORTD (ソート・キー) , $\left[\begin{array}{c} CH \\ ZD \\ PD \\ FI \\ FL \\ LG \end{array} \right]$, サイズ ,

レコード長，ブロック長， $\left[\begin{array}{l} \text{ジョブ・ステップ名} \\ \hline \text{SORT・DA} \end{array} \right]$,

$\left[\begin{array}{l} \text{PRINT} \\ \hline \text{REMOTE} \end{array} \right]$, ファイル名-1, $\left[\begin{array}{l} \text{機 番-1} \\ \hline \text{C. 001} \end{array} \right]$,

$\left[\begin{array}{l} \text{ボリューム通番-1} \\ \hline \text{C00001} \end{array} \right]$, ファイル名-2,

$\left[\begin{array}{l} \text{機 番-2} \\ \hline \text{C. 002} \end{array} \right]$, $\left[\begin{array}{l} \text{ボリューム通番-2} \\ \hline \text{C00002} \end{array} \right]$, $\left[\begin{array}{l} 8 \\ \hline 9 \end{array} \right]$,

$\left[\begin{array}{l} \text{T A} \\ \hline \text{D A} \end{array} \right]$, $\left[\left(\begin{array}{l} \text{HCM} \\ \hline \text{LCM} \end{array} \right), n \right)$, (WAYn=ON)

(3) パラメータの説明 <>内はキー

= 第1パラメータ = < KEY >

分類項目を指定する。指定の形式は次のものがある。

1. (P , I , C , O ,)
2. (P , I , O ,)

P	<p>分類項目：</p> <p>分類項目の始まる位置を記述する。</p> <p>記述はバイト、バイト・ビット又は・ビットとする。</p> <p>レコード上の最初のビットは1から始まる。</p> <p>バイト上の最初のビットは1から始まる。</p>
I	<p>長さ：</p> <p>分類項目の長さを記述する。</p> <p>記述はバイト、バイト・ビット又は・ビットとする。</p>
C	<p>比較方法：</p> <p>詳細は第2パラメータを参照。</p>
O	<p>正順、逆順：</p> <p>分類項目について正順に並べるか、逆順に並べるかを指定する。</p> <p>A 正順（小から大）</p> <p>D 逆順（大から小）</p>

尚、すべての分類項目の比較方法が同じであれば2の形式で書く。この場合は必ず第2パラメータを指定すること。

= 第2パラメータ = <FORMAT>

分類項目の比較方法を指定する。分類項目の比較方法がすべて同じであればこのオペランドで、データ形式を指定できる。第1パラメータの1の書き方をしたときはこのパラメータを指定してはならない。

CH	9ビット文字	(符号なし)
ZD	ゾーン型10進数	(符号あり)
PD	パック型10進数	(符号あり)
FI	固定小数点数	(符号あり)
FL	浮動小数点数	(符号あり)
LG	論理データ	(符号なし)

文字、論理データを除いて符号付代数的比較を行なう。

- = 第3パラメータ = < SIZE >
分類が行なわれる入力ファイルのレコード総数を記述する。レコード総数は、実数または概数で指定され、実数の場合は数字の前にCをつける。実数で指定した場合、実際に入力されたレコード数とのチェックが行なわれる。
- = 第4パラメータ = < RCD >
1レコードの大きさをバイト数で記述。この場合の値は、2語の整数倍でなければならない。
- = 第5パラメータ = < BLK >
1ブロックの大きさをバイト数で記述。
- = 第6パラメータ = < STEP >
このジョブ・ステップにつける名前である。同じジョブ内で同名のジョブ・ステップがあってはならない。
省略されたときは SORT. DA とする。
- = 第7パラメータ = < SYSOUT >
結果をジョブ終了後、システム出力装置から出力すべきことを指定する。

オプション	意味
PRINT	システム出力装置がセンター側の ラインプリンターであることを示す。
REMOTE	システム出力装置が端末であるこ とを示す。

省略されたときは PRINT とする。

- = 第8パラメータ = < F LNAME₁ >
入力ファイル名を与える。詳細は LINK の第7パラメータを参照。
- = 第9パラメータ = < UN I T₁ >
入力ファイルが取り付けてある装置種類又は、機番を指定する。

機 番

磁気テープ装置： D. 000～D. 006

磁気ディスク・パック装置： B. 002～B. 007

種 類

磁気テープ装置： F603F

磁気ディスク・パック装置： F471K

省略されたときはC. 001とする。

= 第10パラメータ = <VOL₁>

入力ファイルのボリューム通番を指定する。

省略されたときはC00001とする。

= 第11パラメータ = <FLNAME₂>

出力ファイルのファイル名を与える。詳細はLINKの第7パラメータ
を参照。

= 第12パラメータ = <UNIT₂>

出力ファイルが取り付けてある装置種類又は、機番を指定する。詳しく
は第9パラメータを参照。

= 第13パラメータ = <VOL₂>

出力ファイルのボリューム通番を指定する。

省略されたときはD00002とする。

= 第14パラメータ = <TMOD>

転送モードを指定する。

8 : 8ビットモード

9 : 9ビットモード

省略されたときは9ビットモード

= 第15パラメータ = <DEV'D>

入出力ファイルの種類を指定する。

TA: 磁気テープ装置

DA: 磁気ディスク・パック装置

= 第 16 パラメータ =

<PARAM>

作業用領域の種類、大きさを指定する。

n は符号のない整数である。

(HCM, n) は作業用領域として HCM (高速磁心記憶装置) を使用する。大きさは $n \times 1024$ 語である。

(LCM, n) は作業用領域として LCM (大容量磁心記憶装置) を使用する。大きさは、 $n \times 32768$ 語である。

省略されたときは (LCM, 1) を指定。

= 第 17 パラメータ =

<WAY_n>

ソートの作業用ファイルの個数を指定する。

n は 4 ~ 16 までの数。

省略されたときは 3 個とする。

(4) 例

¥S O R T D (1 , 5 , A , 10 , 2 , A) , CH , 5000 , 80 , 1200 , , ,
DATA₁ , , , DATA₂ , , , , , WAY₅ = ON

キー 1 , 5 , A , 10 , 2 , A

比較方法 CH (文字)

サイズ 約 5000 件

レコード長 80 衔

ブロック長 1200 衔

入力ファイル名 DATA₁

出力ファイル名 DATA₂

WAY 数 作業用ファイル数 5 個

¥S O R T D (3 , 4 , CH , A , 7.2 , 7.5 , LG , D) , , C 200 , 160 ,
800 , , , IN , F 471K , B 00003 , OUT , F 471K ,
B 00002 , 8 , DA , (HCM , 20)

キー 3 , 4 , CH , A , 7.2 , 7.5 , LG , D

・ サイズ	実数 200 件
レコード長	160 柄
ブロック長	800 柄
入力ファイル名	IN
機 番	F 471K (磁気ディスク・パック)
ボリューム通番	B 00003
出力ファイル名	OUT
機 番	F 471K
ボリューム通番	B 00002
転送モード	8 ビット
種 類	磁気ディスク・パック装置
作業用領域と大きさ	HCM 20Kw

2.13 SORTM

(1) 機 能

作業用ファイルとして磁気テープ装置を使用してソートを行なう。

(2) 形 式

$\$SORTM$ (ソート・キー), $\left[\begin{array}{c} CH \\ ZD \\ PD \\ FI \\ FL \\ LG \end{array} \right]$, レコード長

ブロック長, ファイル名-1, ファイル名-2,

$\left[\begin{array}{c} 8 \\ 9 \end{array} \right]$, $\left[\begin{array}{c} \{\text{ジョブ・ステップ名}\} \\ \text{SORT. MT} \end{array} \right]$, $\left[\left(\begin{array}{c} HCM \\ LCM \end{array} \right), n \right]$,

[WAY_n = ON]

(3) パラメータの説明

各パラメータの詳しい内容は SORTD を参照。

注 意

このマクロは入出力ファイルおよび作業用ファイルは総て磁気テープを使用している。それぞれの機番は、

入力ファイル	D. 001
出力ファイル	D. 002
作業用ファイル	D. 002~D. 006
最高WAY数	5 (省略時 3 WAY)

(4) 例

¥SORTM (5 , 1 , A , 1 , 2 , D) , CH , 100 , 5000 , ABC , DEF

キー	5 , 1 , A , 1 , 2 , D
比較方法	CH (文字)
レコード長	100 衔
ブロック長	5000 衔
入力ファイル名	ABC
出力ファイル名	DEF

2.14 SAVE

(1) 機 能

磁気ディスク・パック装置に存在するファイルを磁気テープに転送する。

(2) 形 式

¥SAVE ファイル名 , ボリューム通番 , $\left[\begin{array}{c} \text{機 番} \\ \hline 0 \end{array} \right]$,
 $\left[\begin{array}{c} \text{NEW} \\ \hline \text{OLD} \end{array} \right]$, $\left[\begin{array}{c} \text{ジョブ・ステップ名} \\ \hline \text{SAVE} \end{array} \right]$

- (3) パラメータの説明
- <> 内はキー
- = 第1パラメータ = <FLNAME>
転送したいファイルのファイル名を書く。このファイル名は出力側のファイル名にもなる。
 - = 第2パラメータ = <VOL>
転送したいファイルが存在する磁気ディスク・パックのボリューム通番を指定する。
 - = 第3パラメータ = <UNIT>
磁気テープの装着機番を0~6までの数で指定する。
省略されたときは0とする。
 - = 第4パラメータ = <MT>

オプション	意味
NEW	テープ・ラベルの直後からデータを書出す。
OLD	テープ上の最終データの後へ新データを書込む。

省略されたときはNEWとする。

- = 第5パラメータ = <STEP>
このジョブ・ステップにつけるべき名前である。同じジョブ内で同名のジョブ・ステップがあってはならない。
- 省略されたときはSAVEとする。

(4) 例

```
¥SAVE DATAFL, 3, , OLD
      ファイル名      DATAFL
      ボリューム通番   3
      MT指定          OLD
```

2.15 MOVE

(1) 機能

磁気テープファイルを磁気ディスク・パックに転送する。

(2) 形式

¥MOVE ファイル名，ボリューム通番，
[{ ト ラ ッ ク 数 }]
[{ 4 0 }]
[{ 機 番 }] , [{ ジ ョ ブ ・ ス テ ッ プ 名 }]
[{ 0 }] , [{ MOVE }]

(3) パラメータの説明 <>内はキー

= 第1パラメータ = <FLNAME>

転送したいファイルのファイル名を書く。このファイル名は出力側のファイル名にもなる。

= 第2パラメータ = <VOL>

転送先となる磁気ディスク・パック装置のボリューム通番を指定する。

= 第3パラメータ = <TRK>

出力側の領域をトラック数で指定する。また使われなかった部分は処理終了時に開放する。

省略されたときは40トラックとする。

= 第4パラメータ = <UNIT>

入力となる磁気テープの装置機番を0～6までの数で指定する。

省略されたときは0とする。

= 第5パラメータ =

このジョブ・ステップにつけるべき名前である。同じジョブ内で同名のジョブ・ステップがあってはならない。

省略されたときはMOVEとする。

(4) 例

¥MOVE DATAFL, 7, 50

ファイル名 DATAFL

ボリューム通番 7

トラック数 50 トラック

3. 制御情報

3.1 ALGOL制御情報

下線のあるものは省略時にとられるオプション

制御情報	意味
<u>L I S T</u>	コンパイル処理のエラーメッセージ以外の諸リストをとるか否かを当該ソースエレメントの先頭のコントロールカードのリスト指定で与えることができる。
<u>N O L I S T</u>	このジョブ・ステップに属するエラーメッセージ以外のリストはすべてカットされる。
<u>S E Q</u>	ソース・カードの識別番号のチェックをする。
<u>N O S E Q</u>	" をしない。
<u>M A R K E D</u>	2文字以上の区切り記号に判別記号<▼>をつけた形の原プログラムに指定される。
<u>M K L E S S</u>	これがないときの指定である。
<u>E L T Y P E</u>	原プログラムがELタイプで穿孔されている。
<u>H T Y P E</u>	" がHタイプで穿孔されている。
<u>L O N G</u>	原プログラム中に宣言された単精度の変数が長精度で宣言されたものとみなされる。
<u>S H O R T</u>	この変数を単精度のままにする。
<u>C C</u>	主記憶装置を使ってコンパイルーションを行なう。
<u>F C</u>	コンパイルは最初からファイルを使う。
<u>H C M</u>	作業領域を高度磁心記憶装置に割付ける。
<u>L C M</u>	" を大容量磁心記憶装置に割付ける。
C A R D N O = <符号のない整数>	ソース・ファイルに含まれるソースエレメントのうち枚数の最も多いものに対して、その枚数を指定する。この場合そのエレメントのコントロール・カードの枚数も含む。指定がなければC A R D N O = 200とみなされる。
E R R = <符号のない整数>	原プログラムの打切りエラーの個数を指定する。指定のないときはE R R = 50とみなされる。
エレメントセレクト	<p>書き方 $E = (\begin{matrix} \text{名前} \\ \text{名前} - \text{名前} \end{matrix} , \begin{matrix} \text{名前} \\ \text{名前} - \text{名前} \end{matrix}) \dots \dots)$</p> <p>機能 磁気テープ上にL I B Eの原プログラムE D I T機能を用いてファイルされた原プログラムの名前(エレメント名)を指定して指定されたエレメントだけ翻訳処理を行なうときに用いる。 名前一名前は両者のエレメント名を含むこの範囲にあるすべてのエレメントを翻訳する。</p>
L P V = <符号のない整数>	コンパイルのリストでページ制御をこの値(単位インチ)を1ページとして取扱う。2以下はダメ。

別表 1

3.2 FORTRAN制御情報

下線のあるものは省略時にとられるオプション

制御情報	意 味
<u>L I S T</u>	翻訳時の原プログラムの印刷を行なう。
N O L I S T	" の印刷を行なわない。
M A P	目的プログラムの記憶場所の詳細な情報を出力する。
S E Q	原プログラムの順序の照合を行なう。
O P T	目的プログラムのOPTIMIZE(最適化)をする。 なお、OPTIMIZEとは 1. 共通部分の除去を行なう。 2. LOOP中のLOOPに関係のないLOOPを外に出す。 3. レジスタを効果的に用いる。 OPTIMIZEしたときは、デバック文はつかえない。
L C M	翻訳時の作業領域を大容量磁心記憶装置に確保する。
C A R D = n	翻訳するカード枚数を与える。
C O M P = n	" エレメント数を与える。
A R E A = n	作業領域の大きさを指示する。 H C Mの場合 n × 1024語。 L C Mの場合 n × 32768語。
L I N E = n	翻訳時に使用する印刷用紙一頁に印刷できる行数を指示する。
E L M = (n ₁ , n ₂ , n)	L I B Eで作成したファイルにある原プログラムの翻訳を行なう場合、翻訳する原プログラムのあるメンバー名を指定する。指定できるメンバー名の数は最大 10 である。
F A S P	目的プログラムをFASP言語の形式で出力する。

別表 2

3.3 COBOL 制御情報

下線のあるものは省略時にとられるオプション

制御情報	意味
L I S T 0	エラーメッセージのみ印刷する。
L I S T 1	L I S T ₀ と原プログラムリストを印刷する。
L I S T 2	L I S T ₁ と記号対応表を印刷する。
L I S T 3	L I S T ₂ と目的プログラムリストを印刷する。
<u>S E Q</u>	原プログラムのシーケンスを調べる。
N O S E Q	" を調べない。
P U N C H	目的プログラムをカードに穿孔する。
<u>N O P U N C H</u>	" に穿孔しない。
<u>S P A C E 0</u>	ベタ打ちリストにする。
S P A C E 1	1行スペース置いてリストする。
S P A C E 2	2行スペース置いてリストする。
C S I Z E = H C M _n	$n \times 1024$ 語を作業領域として H C M にとる。
T R C H K	O C C U R S 句で取られたテーブルの反復回数をえたテーブルを参照しようとしたときにエラーメッセージをだす。
E = (メンバー名 1, ……, メンバー名 - n)	L I B E 形式でのメンバー名を指定するとメンバー名を翻訳する。
	メンバー名の最大個数は 10ヶまでである。

別表 3

3.4 FASP 制御情報

下線のあるものは省略時にとられるオプション

制御情報	意味
<u>L I S T</u>	アセンブル・リストを印刷する。
<u>N O L I S T</u>	" を印刷しない。
<u>P U N C H</u>	アセンブルしてできる相対形式プログラム単位をカードに穿孔する。
<u>N O P U N C H</u>	カードに穿孔しない。
<u>D E B U G</u>	アセンブルしてできる相対形式プログラム単位をデバック・モードで作り出す。
<u>N O D E B U G</u>	デバック・モードでは作らない。
<u>S E Q</u>	アセンブルすべき原プログラムのプログラムカードの順序を検査する。
<u>N O S E Q</u>	カードの順序は検査しない。
E = (プログラム単位名 , ……)	ファイルの中に 2つ以上のプログラム単位が含まれているとき, どのプログラムをアセンブルするかを指示する。
E = (* , プログラム単位名)	ファイルの最初から, かかれたプログラム単位名までの全プログラム単位をアセンブルする。
E = (プログラム単位名 , *)	かかれたプログラム単位名から, そのファイルの終りまでアセンブルする。
E = (プログラム単位名 1 - プログラム単位名 2)	プログラム単位名 1 からプログラム単位名 2 までの間にあるプログラム単位をアセンブルする。

3.5 PL₁ 制御情報

下線のあるものは省略時にとられるオプション

制御情報	意味
<u>SOURCE</u>	原プログラムを印刷する。
<u>NOSOURCE</u>	" を印刷しない。
<u>LIST</u>	目的プログラムを、 F A S P形式で印刷する。
<u>NOLIST</u>	" " 印刷しない。
<u>DICT</u>	目的プログラムの辞書部分を印刷する。
<u>NODICT</u>	" " を印刷しない。
<u>ATR</u>	原プログラム中で使われた各名標について、 つぎの事項を印刷する。 ○ その名標を宣言しているステートメント番号 ○ 名標を含む、 すべての構造体名標 ○ 名標に与えられた主要な属性 ○ それを引用しているすべてのステートメント番号 上記の事項を印刷しない。
<u>NO ATR</u>	目的プログラムを相対形式プログラム単位の形でカード上に出力する。
<u>DECK</u>	目的プログラムをカード上に出力しない。 目的プログラムを、 大記憶装置に出力する。 " " に出力しない。
<u>NODECK</u>	言語用の文字セットが 48字セットである。
<u>OBJECT</u>	言語用の文字セットが 60字セットである。
<u>NO OBJECT</u>	すべての診断メッセージを印刷する。
<u>CHAR 48</u>	診断メッセージのうち、 エラーの程度がX, Zであるものをすべて印刷する。
<u>CHAR 60</u>	診断メッセージのうち、 エラーの程度がZであるものを印刷する。 実行時のONコンディションで、 ステートメント番号を印刷する。 実行時のONコンディションで、 ステートメント番号を印刷しない。
<u>FLAGW</u>	
<u>FLAGX</u>	
<u>FLAGZ</u>	
<u>STM</u>	
<u>NO STM</u>	
MARGIN = (m m , n n)	条件 m m , n n は 2桁までの 10進数かつ, $1 \leq m m \leq n n \leq 80$ 原プログラムが、 入力媒体上のどの位置にあるかを指定する。 省略時 MARGIN = (2 , 72) nは 10進数 実行時の動向割りつけ用ストレジの大きさ(語数)を指定する。 単位 1024語。 省略時 WORK = 1
WORK = n	nは 10進数で、 $0 \leq n \leq 8$ である。 コンパイル時の作業域を LCM 上に作る。 単位は 32768語である。 省略時 LCM = 0
LCM = n	

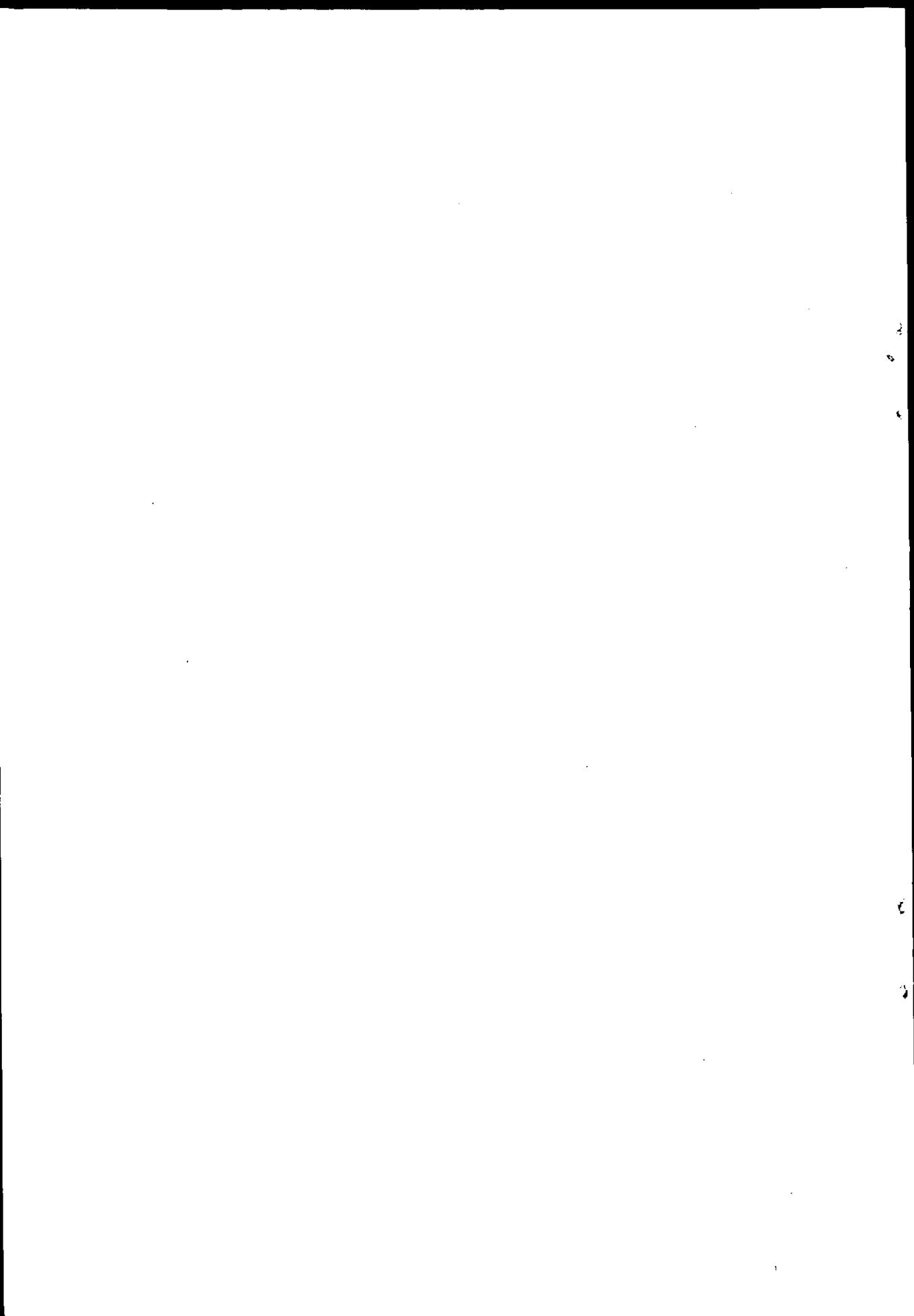
別表 5

3.6 LIED制御情報

制御情報	意味
D	実行形式プログラムがデバックモードで作成されることを指示する。
MAP	ラインプリンタへの出力として、番地割付け結果を出す。
CROSS	記号の相互参照関係を出す。
TREE	オーバレイ構造をなす記憶領域について、その木構造を、直観して、わかるように印刷すべきことを指示する。
WA = n	LEDの作業用領域を、LEDで定めた領域の大きさで不足する場合に指定する。 nは10進数で、単位は1024語である。
LCM	LEDの作業領域として、LCMを使用することを指示する。 このときWAで指定するnの単位は32K語である。
NOLIST	システム出力を出さないことを指示する。
CLIMIT = m	実行形式プログラムのステップ数がmステップを超えると重傷の誤りとする。

別表 6

付録-III 会計情報処理ルーチンの出力例



付録一Ⅲ 会計情報処理ルーチンの出力例

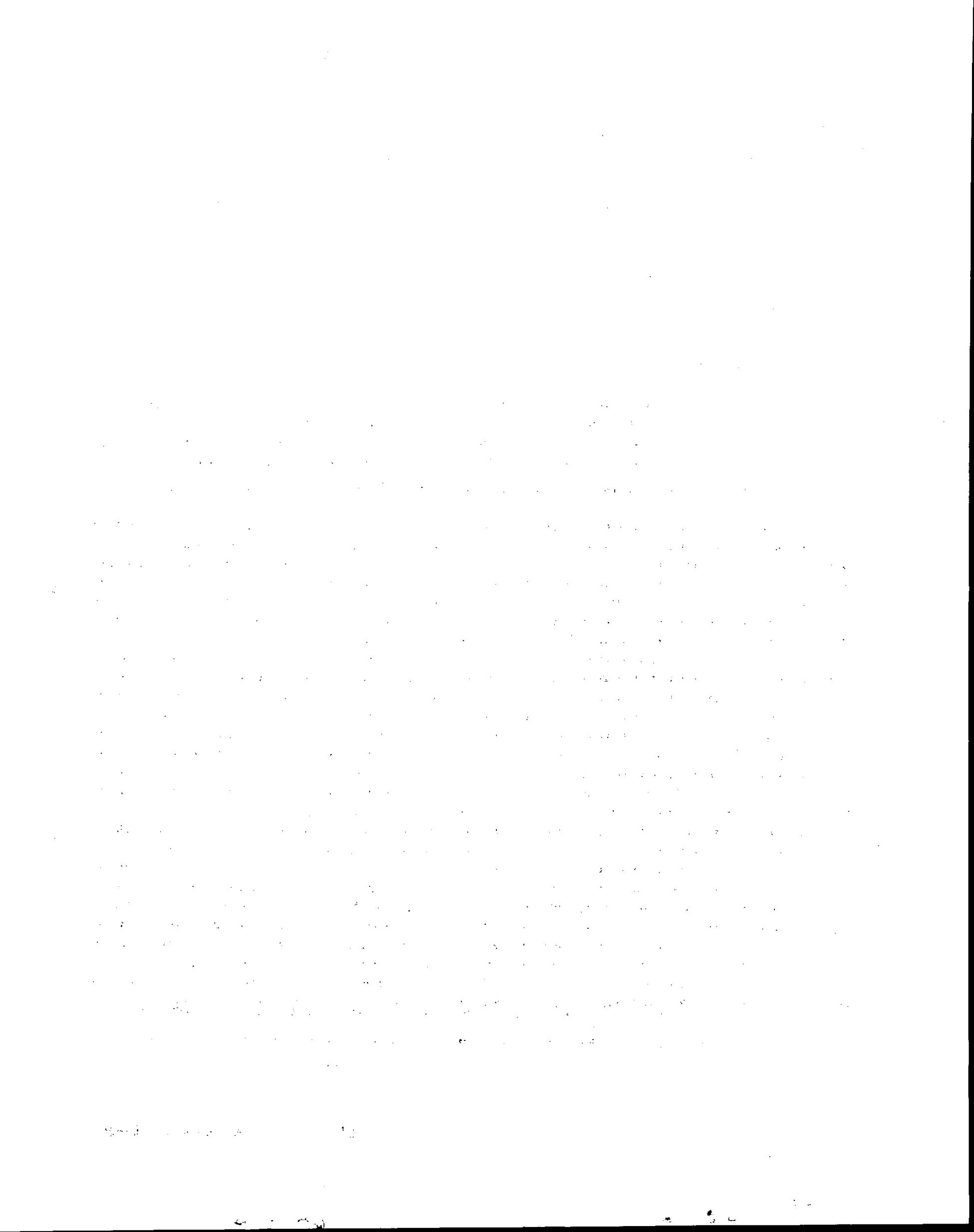
ジョブ毎の会計情報

*** JIPDEC FACOM 230-60 SYSTEM=D ACCOUNTING LIST *** - DAILY (03.05) & DETAIL - 73.03.06 PAGE 4

SEQ	JOB_NO	USER NAME	PROJ ECT#	PROG NAME	PRIORITY	NET CPU	TIME	CORE (KW)	CARD IN	PRINTER LINE	SYSIN PAGE	J_O_B START	O_N_E START	B_N_D FINSH	U_S_E KND	JOB CODE	EROR MULTI	BATCH	ROUT TIME			
76	DJSP014	KM	4940	DEL.	7	3	15	0' 8.290	0:12' 7	45	4	0	27	2 13:50	13:50	14:4	0:15	D	NORM	0	1'59	
77	Z047	HAMANAKA	0304	FORT	5	5	5	0' 8.450	0: 1'27	36	62	0	120	7 14:	2 14:	3 14: 5	0: 3	L	NORM	2	0'29	
78	Z045	KOKEN	0512	FASP	5	5	5	0'57.799	0: 3'27	41	68	0	9135	187	13:59	14: 0	14: 6	0: 7	L	NORM	1	2'56
79	Z049	IWAHORI	0201	PL1	5	5	5	0' 7.436	0: 0'29	76	128	0	342	9 14:	9 14:	9 14:10	0: 1	L	NORM	1	0' 2	
80	Z048	IWAHORI	0201	PL1	5	5	5	0'20.709	0: 0'43	79	363	0	847	18 14:	9 14:10	14:10	0: 1	L	NORM	1	0' 0	
81	Z036	KAWAMORI	0152	FORT	5	5	5	0' 7.713	0: 0'34	35	20	0	92	7 14:14	14:15	14:15	0: 1	L	NORM	1	0' 0	
82	DJIE008	MUROI	3005	DEL.	7	3	15	0'31.672	1: 9' 6	46	4	0	27	2 13:	4 13:	4 14:18	1:15	D	NORM	0	5'24	
83	Z050	KONGOJI	0512	FORT	5	5	5	0' 4.616	0: 0'16	36	274	0	300	9 14:22	14:23	14:23	0: 1	L	NORM	1	0' 4	
84	Z052	MUROI	3005	FORT	5	5	5	0' 7.254	0: 0'21	36	519	0	868	30 14:35	14:36	14:37	0: 1	L	NORM	2	0' 2	
85	DNTT013	KAGETORA	4910	DEL.	7	3	15	0'24.427	1: 0'55	46	4	0	27	2 13:33	13:33	14:39	1: 6	D	NORM	0	4'42	
86	Z051	TAKEMA	0301	COBO	5	5	5	2' 2.476	0: 9'37	29	1167	0	5138	105 14:28	14:31	14:46	0:15	L	NORM	1	4' 5	
87	Z053	MUROI	3005	FORT	5	5	5	0'21.32	0: 0'56	35	553	0	755	22 14:52	14:53	14:54	0: 2	L	NORM	1	0' 0	
88	Z056	KAWAMORI	0152	FORT	5	5	5	0' 7.486	0: 0'34	35	21	0	92	7 15:	0 15:	0 15: 1	0: 1	L	NORM	1	0' 0	
89	DJIF017	MUROI	3005	DEL.	7	3	15	0' 2.413	0: 2'37	43	4	0	27	2 15:	2 15:	2 15: 5	0: 4	D	NORM	0	0'31	
90	Z054	MUROI	3005	FORT	5	5	5	0' 7.477	0: 0'23	36	519	0	814	30 15:	5 15:	6 15: 7	0: 1	L	NORM	1	0' 2	
91	DJID016	KOKEN	0512	DEL.	7	3	15	0' 3.446	0:27'15	48	4	0	36	2 14:41	14:41	15: 9	0:29	D	ABOR	0	0'31	
92	Z064	MUROI	3005	FORT	5	5	5	0' 7.449	0: 0'34	36	520	0	808	29 15:11	15:13	15:13	0: 1	L	NORM	2	0' 3	
93	Z057	MUROI	3005	FORT	5	5	5	3'12. 2	0: 4'26	18	556	0	971	31 15:	9 15:10	15:15	0: 5	L	NORM	1	0' 0	
94	DJID016	KOKEN	0512	DEL.	7	3	15	0' 2.712	0: 8'46	47	4	0	36	2 15:	9 15:	9 15:19	0:10	D	ABOR	0	0'30	
95	Z066	MUROI	3005	FORT	5	5	5	0' 7.696	0: 0'27	36	521	0	810	30 15:18	15:19	15:20	0: 1	L	NORM	1	0' 4	
96	Z055	KAWAMORI	0152	FORT	5	5	5	0' 7.659	0: 0'35	35	22	0	96	8 15:21	15:22	15:23	0: 1	L	NORM	1	0' 0	
97	DJID016	KOKEN	0512	DEL.	7	3	15	0' 2.575	0: 4'18	46	4	0	36	2 15:19	15:19	15:24	0: 6	D	ABOR	0	0'32	
98	DJID016	KOKEN	0512	DEL.	7	3	15	0' 2.520	0: 3' 1	45	4	0	36	2 15:25	15:25	15:29	0: 4	D	ABOR	0	0'30	
99	Z058	KAWAMORI	0152	FORT	5	5	5	0' 7.701	0: 0'39	35	22	0	95	8 15:27	15:28	15:29	0: 1	L	NORM	1	0' 0	
100	Z059	MUROI	3005	FORT	5	5	5	0'21.719	0: 2' 4	32	522	0	1012	36 15:35	15:36	15:39	0: 3	L	NORM	2	0' 0	

*** JIPDEC FACOM 230-60 SYSTEM=D ACCOUNTING LIST *** - DAILY (03.05) & SUMMARY - 73.03.06 PAGE 7

THE NUMBER OF JOB			CPU TIME (HH:MM'SS)			CORE TIME (HH:MM'SS)					
DEMAND	REMOTE	LOCAL	TOTAL	DEMAND	REMOTE	LOCAL	TOTAL	DEMAND	REMOTE	LOCAL	TOTAL
41	0	89	130	0: 6'30	0: 0' 0	0:39'30	0:46' 1	19:13'10	0: 0' 0	2:56'33	22: 9'44
32%	0%	68%	100%	14%	0%	86%	100%	87%	0%	13%	100%

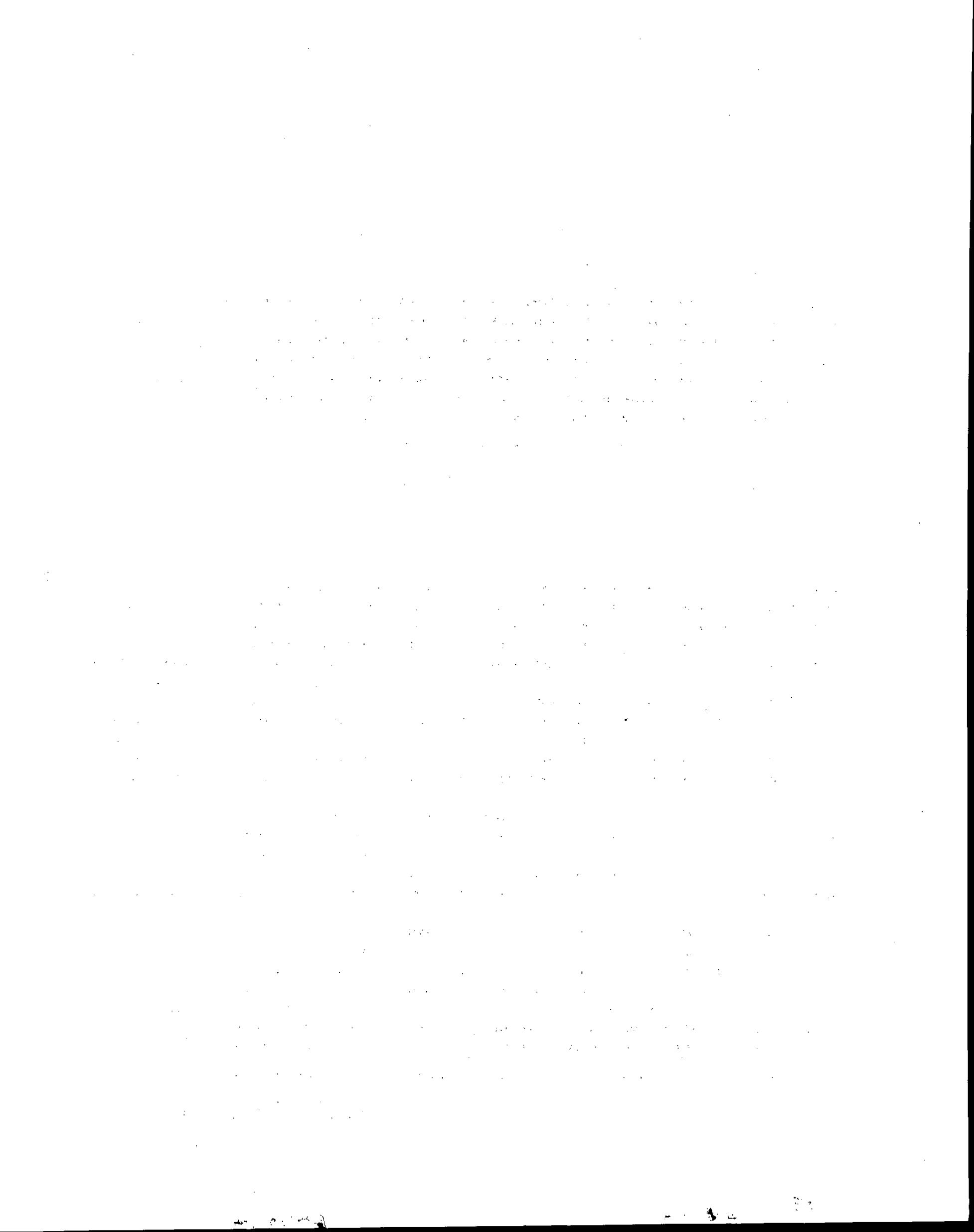


ジョブ種類別会計情報集計表

*** JIPDEC FACOM 230-60 SYSTEM-D ACCOUNTING LIST ***								- 1973.02 -			PAGE	13	
DATE	JOB KIND	CPU TIME	CORE TIME	CARD		PRINTER		ERROR CODE		JOB	JOB STEP		
		(HH:MM:SS,MS)	(HH:MM:SS,MS)	IN	OUT	LINE	PAGE	NORM	JCEC	ABNORM	NUMBER	NUMBER	USE TIME
25	LOCAL (CLOSE)	0: 0' 0.000	0: 0' 0.000	0	0	0	0	0	0	0	0	0: 0' 0	
	LOCAL	0: 1'53.477	0:20'35.442	5424	0	5977	148	2	1	2	5	16	0:34'47
	REMOTE	0: 0' 0.000	0: 0' 0.000	0	0	0	0	0	0	0	0 (0)	0	0: 0' 0
	DEMAND	0: 2'15.067	0:40'43.574	28	0	3608	93	4	0	3	7 (0)	7	0:42'39
	TOTAL	0: 4' 8.544	1: 1'19.016	5452	0	9585	241	6	1	5	12	29	1:17'26
26	LOCAL (CLOSE)	1:38'54.450	3:44'24.215	27040	0	65883	1902	43	4	30	77	184	9:33'25
	LOCAL	0:12' 3.372	1:48'58.198	3383	0	37324	996	48	5	8	61	81	3:12' 1
	REMOTE	0: 0' 0.000	0: 0' 0.000	20	0	34	3	0	3	0	3 (0)	0	0: 0' 0
	DEMAND	0: 0'57.694	5:51' 8.085	120	0	16453	391	16	3	11	30 (1)	49	6:24' 8
	TOTAL	1:51'55.516	11:24'30.498	30563	0	119694	3292	107	15	49	171	314	19: 9'34
27	LOCAL (CLOSE)	2:13'37.082	4:26'44.319	30079	0	69219	1722	29	3	24	56	159	7:26'30
	LOCAL	0: 5'34.153	0:23' 3.563	2746	0	4327	133	9	0	1	10	14	0:33'56
	REMOTE	0: C' 5.676	0: 0'20.875	612	0	127	10	1	3	1	5 (1)	2	0: 1' 4
	DEMAND	0: 4'47.573	6:41'19.909	84	0	9198	223	18	1	2	21 (3)	34	6:54'23
	TOTAL	2:24' 4.484	11:31'28.666	33521	0	82871	2088	57	7	28	92	209	14:55'53
28	LOCAL (CLOSE)	1:48' 6.521	5:39'57.750	36336	0	115178	2892	36	4	32	72	231	11:39'42
	LOCAL	0: 4'32.614	0:58'57.604	1064	0	5283	251	31	0	0	31	40	1:40'18
	REMOTE	0: 0'38.342	0: 3'25.063	1410	0	1115	40	3	1	4	8 (5)	9	0:10'49
	DEMAND	0: 4'34.932	10:52'57.628	164	0	3472	166	32	1	8	41 (3)	75	12:59'29
	TOTAL	1:57'52.409	17:35'18.045	38974	0	125048	3349	102	6	44	152	355	26:24'18

1398

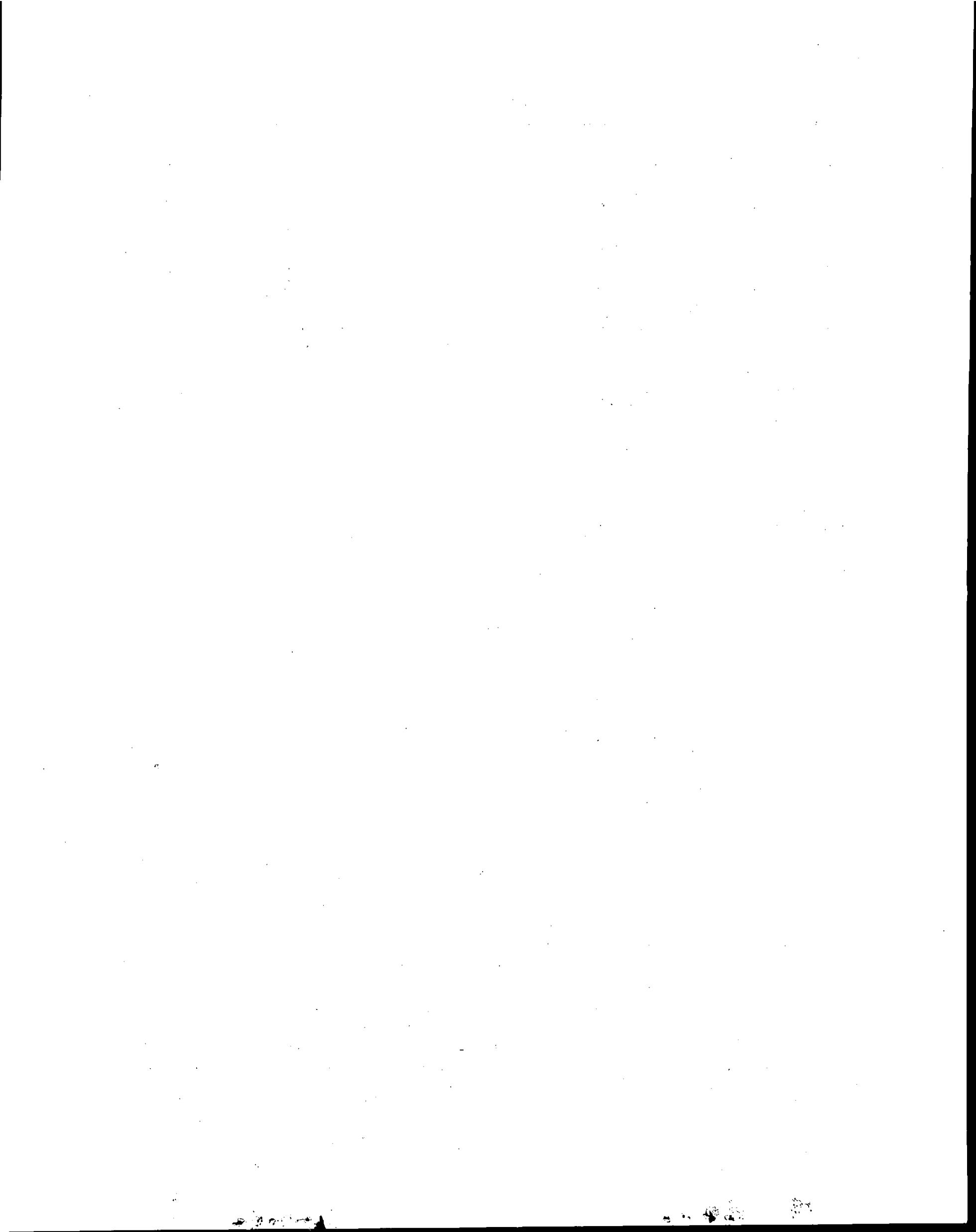
*** JIPDEC FACOM 230-60 SYSTEM-D ACCOUNTING LIST ***								- 1973.02 -			PAGE	14	
DATE	JOB KIND	CPU TIME	CORE TIME	CARD		PRINTER		ERROR CODE		JOB	JOB STEP		
		(HH:MM:SS,MS)	(HH:MM:SS,MS)	IN	OUT	LINE	PAGE	NORM	JCEC	ABNORM	NUMBER	NUMBER	USE TIME
TOTAL	LOCAL (CLOSE)	29:45'15.308	RR:53' 3.303	709987	0	1763078	43363	734	73	644	1451	3866	202: 2'20
	LOCAL	4: 2'40.907	23:54'36.913	108852	0	310478	9000	411	61	127	599	870	57:55'40
	REMOTE	1: 8'53.551	6:28'37.054	20984	0	107883	2787	124	35	45	204 (26)	377	15: 4'11
	DEMAND	5:13'27.070	247:26'19.439	4144	0	86572	3710	648	128	260	1036 (69)	1370	270:34'18
	TOTAL	40:10'16.536	366:42'36.709	843967	0	2268011	58860	1917	297	1076	3290	6483	545:36'29



端末別会計情報集計表

*** JIPDEC FACOM 230-60 SYSTEM-E ACCOUNTING LIST ***								1973. 2			PAGE- 24		
DATE	TERMINAL	KENSU	C P U (HH:MM'SS.MS)	C O R E (HH:MM'SS.MS)	SYSIN	TYPE	(CENTER) PAGE	LINE	NORM	ABNORM	JCER		
27	JIA DT221	4 4	0: 3'42.300	3:55'17.481	-	16	0	8	52	4	0 0	3:57	
	JIB DT211	8 6	0: 0'15.724	0:38' 8.752	32	12	196	8922	5	1 2	0:41		
	JIC FR- 1	3 3	0: 0' 1.632	0:22' 3.321	12	0	6	39	3	0 0	0:22		
	JID FR- 2	0 0	0: 0' 0.000	0: 0' 0.000	0	0	0	0	0	0 0	0: 0		
	JIE FR- 3	5 12	0: 0'30.475	0:42'40.339	20	0	9	118	4	0 1	0:49		
	JIF JP1500	2 1	0: 0' 0.163	0: 0' 5.207	19	24	6	99	1	0 1	0: 0		
	JSP KISIN	1 3	0: 0' 8.179	0:13'36.081	4	0	2	27	1	0 0	0:15		
	MEL KIKAI	1 1	0: 0' 5.513	0: 0'15.668	585	629	2	14	0	1 0	0: 1		
	MIT TUSAN	0 0	0: 0' 0.000	0: 0' 0.000	0	0	0	0	0	0 0	0: 0		
	MPS YUSEI	0 0	0: 0' 0.000	0: 0' 0.000	0	0	0	0	0	0 0	0: 0		
	NTT DENDEN	2 6	0: 0' 9.263	0:49'33.935	8	0	4	54	1	1 0	0:51		
	TTI SEN I	0 0	0: 0' 0.000	0: 0' 0.000	0	0	0	0	0	0 0	0: 0		
*		26 36	0: 4'53.249	6:41'40.784	696	665	233	9325	19	3 4	6:56		
28	JIA DT221	4 4	0: 0'25.713	3:23'30.796	16	0	8	55	2	2 0	3:25		
	JIB DT211	17 17	0: 0'29.132	1:46'41.782	68	0	119	2919	14	3 0	1:54		
	JIC FR- 1	6 16	0: 0'35.474	0:14' 9.990	24	0	12	148	6	0 0	0:21		
	JID FR- 2	6 15	0: 0'24.023	1: 2'27.764	142	24	36	1134	5	0 1	1:33		
	JIE FR- 3	7 17	0: 2'26.854	3:27'51.006	28	0	13	161	2	4 1	4: 5		
	JIF JP1500	1 1	0: 0' 0.075	0: 0' 8.941	9	17	2	15	0	1 0	0: 1		
	JSP KISIN	2 6	0: 0'18.363	0:54'17.492	8	0	4	54	2	0 0	1: 7		
	MEL KIKAI	5 5	0: 0'25.081	0: 2' 0.931	1275	3357	10	74	2	3 0	0: 7		
	MIT TUSAN	1 3	0: 0' 8.559	0: 5'13.989	4	0	2	27	1	0 0	0:35		
	MPS YUSEI	0 0	0: 0' 0.000	0: 0' 0.000	0	0	0	0	0	0 0	0: 0		
	NTT DENDEN	0 0	0: 0' 0.000	0: 0' 0.000	0	0	0	0	0	0 0	0: 0		
	TTI SEN I	0 0	0: 0' 0.000	0: 0' 0.000	0	0	0	0	0	0 0	0: 0		
*		49 84	0: 5'13.274	10:56'22.691	1574	3398	206	4587	34	13 2	13: 8		

*** JIPDEC FACOM 230-60 SYSTEM-E ACCOUNTING LIST ***								1973. 2			PAGE- 25		
DATE	TERMINAL	KENSU	C P U (HH:MM'SS.MS)	C O R E (HH:MM'SS.MS)	SYSIN	TYPE	(CENTER) PAGE	LINE	NORM	ABNORM	JCER		
***	JIA DT221	293 300	2:43'36.774	83:39' 2.189	1193	66	622	6955	160	97 36	88:11		
	JIB DT211	280 338	1: 3'26.904	56:11'19.806	1120	33	1511	41582	160	90 30	65:12		
	JIC FR- 1	58 89	0:21'22.207	9:35' 2.662	1458	142	219	5576	46	11 1	12:30		
	JID FR- 2	142 223	0:25'52.893	12:27'27.224	3523	1058	825	26389	105	22 15	17:50		
	JIE FR- 3	316 550	1:23'43.703	30:39'15.673	13941	2098	2603	91160	213	58 45	41: 9		
	JIF JP1500	56 54	0: 0'39.059	5:34'37.260	245	74	507	18502	28	16 12	7: 3		
	JSP KISIN	17 45	0: 3'27.270	9: 9'37.009	68	0	32	421	15	0 2	10:48		
	MEL KIKAI	26 34	0:10'46.363	0:26'18.382	3372	21519	87	2717	6	12 8	1:41		
	MIT TUSAN	1 3	0: 0' 8.559	0: 5'13.989	4	0	2	27	1	0 0	0:35		
	MPS YUSEI	6 18	0: 0'44.096	1:30' 7.987	24	0	13	167	6	0 0	1:38		
	NTT DENDEN	45 93	0: 8'32.793	44:36'54.312	180	0	76	959	26	5 14	39: 2		
	TTI SEN I	0 0	0: 0' 0.000	0: 0' 0.000	0	0	0	0	0	0 0	0: 0		
*		1240 1747	6:22'20.621	253:54'56.493	25128	24990	6497	194455	766	311 163	285:39		



ジョブコード別会計情報集計表

*** JIPDEC FACOM 230-60 SYSTEM-D ACCOUNTING LIST *** ジョブコード ヘッツ - 1973. 1 - PAGE 1

JOB CODE	JOB NAME	CPU TIME (HH:MM:SS.MS)	CORE TIME (HH:MM:SS.MS)	CARD IN	PRINTER PAGE	ERROR NORM JCER ABNORM	JOB NUMBER	JOB STEP NUMBER	USE TIME	CORE/USE (%)
0101	TSS ジ-ヨ-シツケン	0: 0' 0.000	0: 0' 0.000	0	0	0 0 0	0	0	0' 0	0.0
0102	ON-LINE-SIMULATION	0:33'22.905	6:32' 9.390	46599	4571	108 16	53	177	267 842' 6	46.6
0103	ILS	7:22'52.711	64:24'44.628	71910	6282	381 63	180	624	967 5565'35	69.4
0104	GRAPHIC	0: 0'11.191	0: 2'22.103	1463	40	4 2	0	6	6 44'54	5.3
0105	COMPILER-GENERATER	1:26'34.823	10:16'48.219	161366	7026	154 8	85	247	679 1793'32	34.4
0106	SYSTEM ヒヨウカホ-シキ	0: 0' 0.000	0: 0' 0.000	0	0	0 0 0	0	0	0' 0	0.0
0201	セイソンカンリ	0: 0' 0.000	0: 0' 0.000	0	0	0 0 0	0	0	0' 0	0.0
0204	シソンシ タイク	0: 0' 0.000	0: 0' 0.000	0	0	0 0 0	0	0	0' 0	0.0
0206	フツビン ケイク	0: 0' 0.000	0: 0' 0.000	0	0	0 0 0	0	0	0' 0	0.0
0208	TSS-SARVICE シソツイチヨウラ	0:55'33.184	21:30'23.777	244	112	46 9	5	60	109 1527' 2	84.5
0223	フツビン カンリ	0: 0' 0.000	0: 0' 0.000	0	0	0 0 0	0	0	0' 0	0.0
0301	ケンヨーベンカン	2: 7'47.851	20:15'40.832	45702	2364	52 7	23	82	382 1113'23	109.2
0302	FACOM ACCOUNT ROUTINE	0: 0' 0.249	0: 6'38.171	4	2	1 0	0	1	1 6'49	97.4
0303	CPL	0:20' 9.325	1:28'13.583	1392	117	16 1	1	18	23 110' 2	80.2
0304	JOLDER-2	2: 6'25.649	33:49'52.021	74107	10425	273 43	126	442	836 3524' 6	57.6
0305	TSS-FORTRAN	0: 1' 6.236	0:28'23.952	1008	111	8 0	0	8	17 41'25	68.6
0306	TSS-SIMULATION	0: 0' 0.000	0: 0' 0.000	0	0	0 0 0	0	0	0' 0	0.0
0307	COST	0: 0' 0.000	0: 0' 0.000	0	0	0 0 0	0	0	0' 0	0.0
0409	コウタクカ1	0: 2'16.114	0:22' 3.919	1996	202	7 2	4	13	23 38'18	57.6
0419	カツラン MODEL	3:53'25.979	7:48'33.595	19966	1979	32 15	52	99	124 887'15	52.8
0425	トウシリエクリツ ノ ケイソン	0: 0' 0.000	0: 0' 0.000	0	0	0 0 0	0	0	0' 0	0.0
0512	キカイカンリ (カンリカ)	3:23' 9.832	31:16'59.589	20953	3278	300 13	50	363	534 3007'45	62.4
3005	カイソク SYSTEM (ケンチイカ)	6:15'51.082	20:36'58.399	189548	12623	141 10	281	432	977 3790'30	32.6
3006	キヨウイカ	0: 1'46.236	0: 5'57.106	2254	150	0 0	6	6	14 20' 2	29.7
*		28:30'33.367	219: 5'49.284	638512	49282	1523 189	866	2578	4959 22312'44	58.9

プログラム別利用件数集計表

05

PROGRAM	KENSUU	CPU TIME
FSUMOVEZ	607	2.55.26.552
JMACCAT	207	4.54.079
ELOG	91	1.01.291
FORTRAN	4577	11.41.27.103
LIED	8380	31.09.24.937
EXEPRO	2934	68.44.31.459
FASP	5281	11.28.18.731
CDLP	66	24.34.231
FSUMTDMP	192	16.17.493
GPSS	39	4.58.07.630
TOMO	561	12.10.26.449
BACCUS	368	43.54.849
TCDPT	45	2.47.665
JOL2	78	45.498
COBOL	2702	3.26.33.451
SORT	549	5.17.25.600
DAILYACT	407	1.14.01.609
LIBE	5152	6.27.41.094
FSULIST	461	6.03.406
DEL:FILE	979	8.13.02.700
PLD	393	3.41.50.380
FSUPROGM	861	1.15.276
CPL	1239	10.10.14.450
SIMBOL	104	4.473
P1	40	18.416
P2	37	20.558
FSUMTCPY	14	4.35.787
NONAME	203	17.03.691
FSUEPOC	8	1.646
FSUDALTG	9	3.21.973

— 禁無断転載 —

昭和48年3月発行

発行所 財團法人 日本情報処理開発センター

東京都港区芝公園3丁目5番8号

機械振興会館内

TEL (434) 8211 (代表)

印刷所 三協印刷株式会社

東京都渋谷区渋谷3-11-11

TEL (407) 7316

