

ISSN 0452-2982
UDC 681.31

航空宇宙技術研究所資料

TECHNICAL MEMORANDUM OF NATIONAL AEROSPACE LABORATORY

TM-431

航空宇宙技術研究所角田支所FACOM M-160Fシステム

石垣 博 ・ 中村 絹代 ・ 末松 和代

DP-55

1981年4月

航空宇宙技術研究所
NATIONAL AEROSPACE LABORATORY

航空宇宙技術研究所角田支所FACOM M-160Fシステム*

石 垣 博** 中 村 絹 代*** 末 松 和 代***

FACOM M-160F Computer System at Kakuda Branch National Aerospace Laboratory

Hiroshi ISHIGAKI, Kinuyo NAKAMURA
and Kazuyo SUEMATSU

ABSTRACT

As a new computer system, FACOM M-160F has been installed at the Kakuda Branch, National Aerospace Laboratory. This report gives an outline of the hardware and the operating system as well as a description of the operating procedure and a summary of the center routine. The characteristics of the FORTRAN language compiler are also described.

1. 緒 言

航空宇宙技術研究所角田支所には、電子計算機システムとして、昭和48年2月以来、TOSBAC-3400モデル41が設置され、研究業務遂行に貢献してきた。TOSBAC-3400は、浮動小数点演算に48ビットの高精度計算を行なう、科学技術用の計算機である。しかし、進歩のはげしい計算機技術のなかにあって導入以来8年、ハードウェア、ソフトウェアの老朽化、陳腐化は必然の帰結であった。その間、支所研究業務の進展とともに、計算機システムへの要求、期待も年々高いものとなり、昭和56年5月、FACOM M-160Fシステムに代替の運びとなった。

代替に伴う諸作業、とくにベンチマークテスト、センタールーチンの作成などは、本所計算センターと支所担当者との緊密な協力のもとに行なわれた。その協力成果の一端は、本報告からもうかがい知ることができよう。

本報告は、新システムとして角田支所に導入されるFACOM M-160Fシステムのハードウェア、ソフトウェアの概要を述べるとともに、システム運用方針とその実現手段であるセンタールーチンの内容、およびフォートラン言語コンパイラの特長などについて述べたものである。M-160Fのような汎用計算機は、事務計算からリアルタイム処理まで、きわめて多種類の要求にひととおりこたえられるように設計されている。一方、支所での使用は科学技術計算、実験データ処理などに限られる。したがって、汎用機のもつ膨大で複雑なソフトウェア体系の中から、必要な機能を迅速にひきだし、システムの効率的運用を可能にするセンタールーチンの役割は、きわめて大きいといえる。また、導入後の利用状況の詳細を把握、分析するのに必要なアカウント情報やシステム計測手段も種々用意されているから、運用パラメータの設定値改良などの努力により、いっそう効率的な運用を目ざすことも可能である。本報告ではさらに、ベンチマークテストの具体例にもとづき、3種のフォートランコンパイラ特性の長短が述べられている。したがって、目的に応じたコンパイラを

* 昭和56年2月23日受付

** 角田支所

*** 計算センタ

選択することにより、ユーザー自身が効率よく計算機を利用することができよう。また、導入時のプログラム変換では、ビット構成の相違に起因するエラーが発生すると予想されるが、第5章の浮動小数点演算についての項は、そのトラブル解決のてだすけともなろう。

2. ハードウェアとソフトウェア

本章においては、角田新システム (FACOM M-160F システムを指し、これに対し、TOSBAC-3400 システムを旧システムと言う) についての理解を深めるために、システムを構成するハードウェアとオペレーティングシステムについて簡単に説明する。

2.1 ハードウェア構成とハードウェア性能

図2-1にハードウェア構成図を示す。M-160F^{注1)}中央処理装置(以下CPUと言う)を中心に、4MB^{注1)}の容量を持つ主記憶装置(以下MSUと言う)、自動電源制御装置(以下APCUと言う)、メインコンソールが直結しており、各制御装置を通じて、磁気ディスク装置6台、磁気テープ装置3台、グラフィックディスプレイ装置1台、オープンコンソール1台、カード読取装置1台、ラインプリンタ装置1台、紙テープ読取装置1台、フロッピーディスク装置1台、XYプロッタ装置1台およびTSS端末装置8台が接続されている。他に、オフラインのカードパンチ機2台、フロッピーデータエントリ1台が置かれる。図2-2は、それらのレイアウトである。

表2-1は、図2-1のハードウェアの性能をまとめたものである。それに基づき、重要なCPU、MSUおよび新装置であるAPCUについて若干の説明を行なう。¹⁾

MSUは、CPUまたはチャネル装置(以下CHNLUと言う)から転送された情報の書き込みおよび記憶されている情報の読み出しを仕事とするが、この情報の読み出しおよび書き込みに関して、CPUまたはCHNLUが情報の読み出しをMSUに要求してから要求元に転送されるまでにかかる時間をアクセス

タイムと言い、読み出しまたは書き込みの起動がなされてからそれらの動作が完結するまでにかかる時間をサイクルタイムと言う。アクセスタイムおよびサイクルタイムは表2-1に載せてあるが、サイクルタイムが読み出しと書き込みの場合で異なるのは、CPUのマイクロプログラム制御方式のために、ハードウェアが指定されたアドレスに対する8バイトの情報の読み出しを行なった後、指定された情報の書き込みを行なうためである。

MSUは、その内容の破壊および不当な読み出し^{注2)}を未然に防ぐために、主記憶キーおよび保護キーを用いてMSUの記憶保護動作の規定を行なっている。しかし、すべての主記憶アクセス動作がこの記憶保護の対象となるわけではなく、後に説明するアドレス変換の場合にはその対象とならない。

CPUは、シーケンス制御部、記憶制御部、命令制御部、演算制御部から成り、コンピュータシステムの中枢をなす部分であり、これらの制御部で扱われるデータの形式は図2-3に示す通りである。数値形式には、固定小数点、浮動小数点、10進数および論理数の4種類があり、その形式は図2-4に示す通りである。旧システムと新システムの浮動小数点数の精度の比較を表2-2に示しておく。

CPUにおいて使用される命令数は189個であり、それらはその形式により6種類に分けられる。図2-5にその形式を示した。

CPUはいくつかの記憶領域を持っており、その中で、固定小数点演算、論理演算、アドレスの計算等に使用される汎用レジスタ、浮動小数点演算に使用される浮動小数点レジスタ、システムの動作の制御や特定の情報の指示に使用される制御レジスタは、プログラムで取り扱うことのできるレジスタである。これらのレジスタの個数は表2-1に示されている。

また、CPUはマイクロプログラム制御方式を採用しているため、そのマシクロックは、マイクロプログラムのサイクルタイムである42nsである。主な命令の演算速度は表2-3に示す通りであるが、ギブソンミックス比によると、新システムは旧システムの4.3倍となっている。

注1) 1MB = 1024KB, 1KB = 1024B, Bはバイトを示し、8ビットから成る。

注2) MSUの内容2KB毎に7ビットのキーを置いて、その内容の書き込みおよび読み出しの制限を行なう。

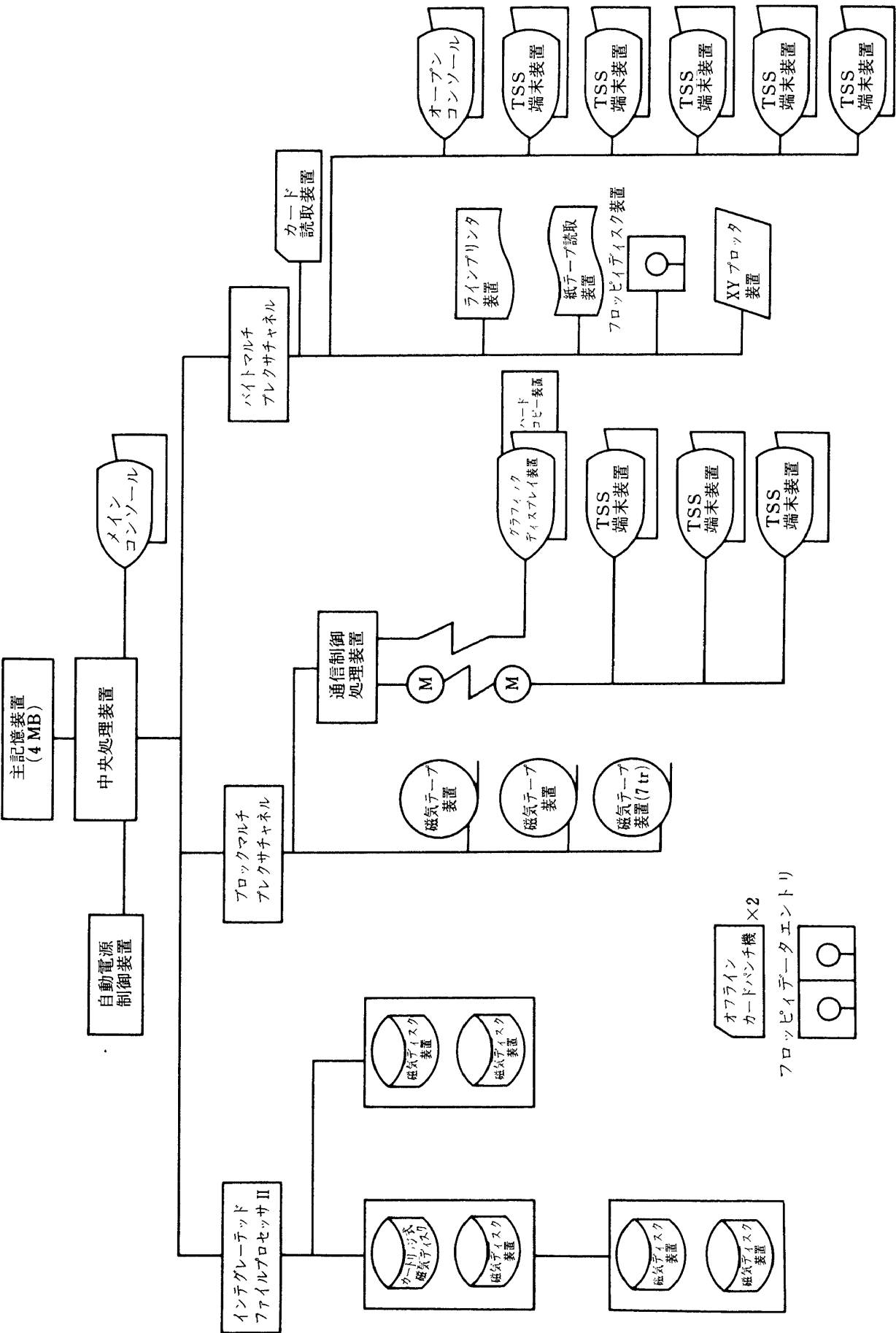


図 2-1 新システムハードウェア構成図

表 2-1 新システムハードウェア性能表

装 置 名	型 名	性 能		
中央処理装置	M5660 L	命令数	189	
		命令形式	6 種	
		制御レジスタ数	16 (各 32 ビット)	
		はん用レジスタ数	16 (各 32 ビット)	
		浮動小数点レジスタ数	4 (各 64 ビット)	
		データ形式	バイト 半 語 語 倍長語 4 倍長語	
主記憶装置		数値表現	2 進固定小数点数 浮動小数点数 10 進数数 論理数	
		主記憶容量	4MB	
		サイクル タイム	読 出 し	360 ns / 8B
			書 込 み	640 ns / 8B
自動電源制御装置	F9140 A	アクセス タイム		360 ns / 8B
メインコンソール	F5665 A	サイズ	14 インチ	
		表示容量	1920 字 (80 字 × 24 行)	
		カラー	三色 (赤 白 緑)	
インテグレートッド ファイルプロセッサⅡ	F5676 B	転送速度	885KB/秒	
磁気ディスク装置	F6411 A	記憶容量	135MB/台	
		データ転送速度	885KB/秒	
		平均回転待時間	10.1ms	
		平均ポジショニング時間	27ms	
カートリッジ式 磁気ディスク装置	F6417 A	記憶容量	67MB/台	
		データ転送速度	717KB/秒	
		平均回転待時間	12.5ms	
		平均ポジショニング時間	30ms	
ブロックマルチ プレクサチャネル	F5671 A	転送速度	1.6MB/秒	
磁気テープ装置	F610 A ₂	トラック数	9	
		記 録 密 度	1600/800 rpi	
		転 送 速 度	320/160KB/秒	
	F610 A ₃	トラック数	7	
		記 録 密 度	800/566 rpi	
		転 送 速 度	160/111KB/秒	

装 置 名	型 名	性 能	
通信制御装置	F 5656 A	通 信 方 式	全二重, 半二重通信
		回 線 速 度	50 ~ 56,000 ビット / 秒
グラフィックディスプレイ装置	F 9431 A	サ イ ズ	14 インチ
		表 示 容 量	1920 字 (80 字 × 24 行)
		表示文字種	96 字種
ハードコピー装置	F 9436 HC ₁	コピー時間	30 秒
バイトマルチプレクサチャネル	F 5672 A	転 送 速 度	{ マルチプレクスモード 40KB / 秒 バーストモード 100KB / 秒
カード読取装置	F 671 D	読 取 速 度	1250 枚 / 分
ラインプリンタ装置	F 654 C	印 字 速 度	660 行 / 分
紙テープ読取装置	F 749 G	読 取 速 度	300 字 / 秒
フロッピーディスク装置	F 442 A	回 転 速 度	360 RPM
		処 理 速 度	{ 読 取 1871 レコード / 分 書 込 1560 レコード / 分
XYプロッタ装置	F 6202 B	プロット速度	{ X 軸 1000 ステップ / 秒 Y 軸 1000 ステップ / 秒
オープンコンソール	F 6262 R	サ イ ズ	14 インチ
		表 示 容 量	1920 字
		表示文字種	128 字種
TSS 端末装置	F 6262 R	オープンコンソールと同様	
	F 9526 R	オープンコンソールと同様	
フロッピーデータエントリ	F 6852 A	フロッピーディスク部 レコード長 128 桁 / レコード 記 録 容 量 1898 レコード / 枚 回 転 速 度 360 rpm ディスプレイ部 サ イ ズ 9 インチ	
オフラインカードパンチ機	F 6822 A		

新システムでは仮想記憶方式を採用しており、それにより主記憶（実記憶）の有効利用がはかられている。この方式は実記憶とは別に仮想記憶を想定して行なう方式のために、ユーザはプログラム作成時に実記憶の大きさを意識することなくコーディングできるが、ジョブの実行においては、仮想記憶と実記憶を対応させるためのアドレス変換機構が必要となる。ユーザがプログラムを作成する場合、16MBの仮想空間を想定してコーディングすることができ、この16MBの仮想空間は256セグメント（64KB /

セグメント）に分けられ、さらに1セグメントは4KBを1ページとする16ページに分割されている。従って、仮想空間のアドレスは、セグメント、ページ、ページ内の番地から成り、これらはMSU内に、セグメントテーブル、ページテーブルとして空間毎に作成されMSUに格納されている。このように、セグメントテーブルおよびページテーブルを参照して、あるアルゴリズムに従い、実記憶アドレスに対応づけることをアドレス変換と言う。仮想記憶へのアクセスの度にセグメントテーブルおよびページテ

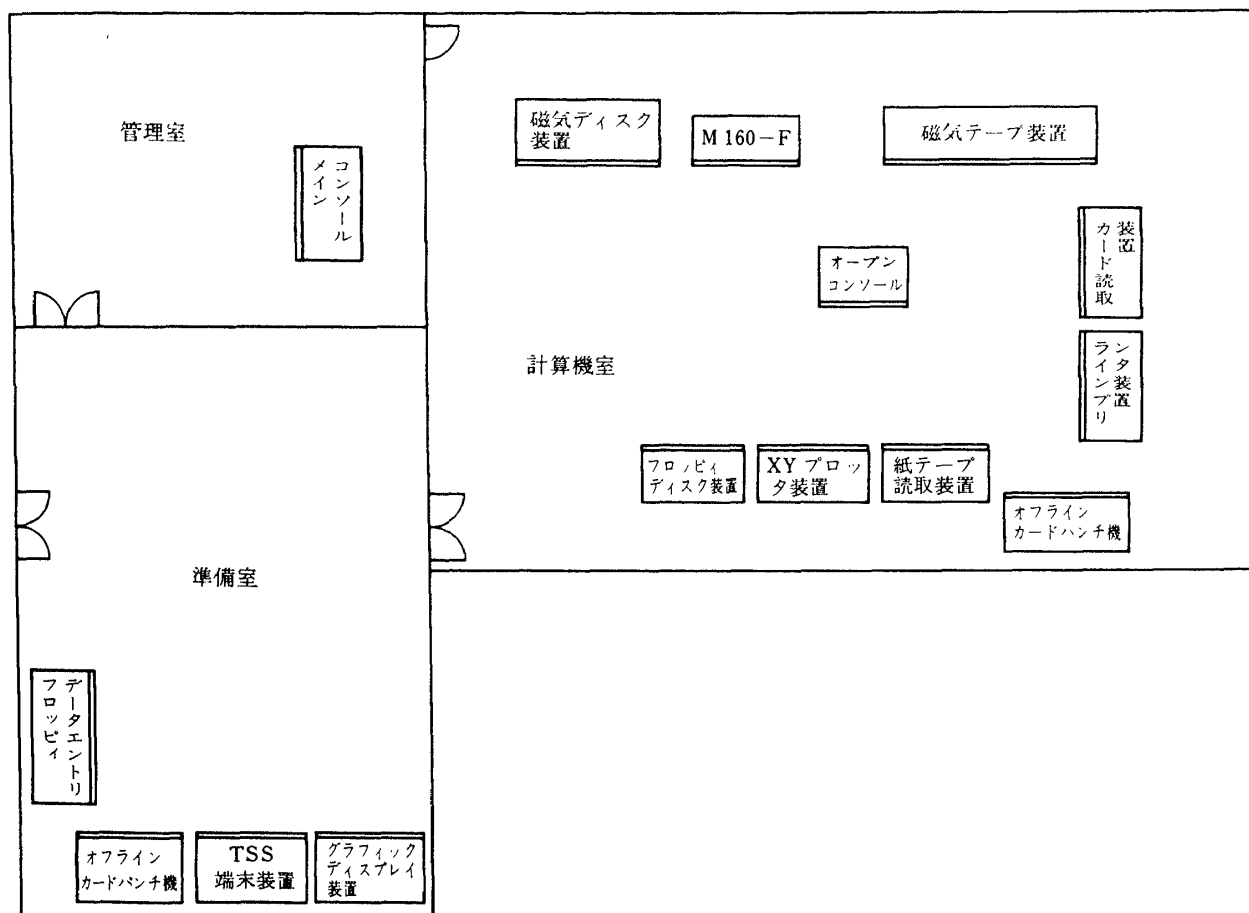


図 2-2 新システムレイアウト

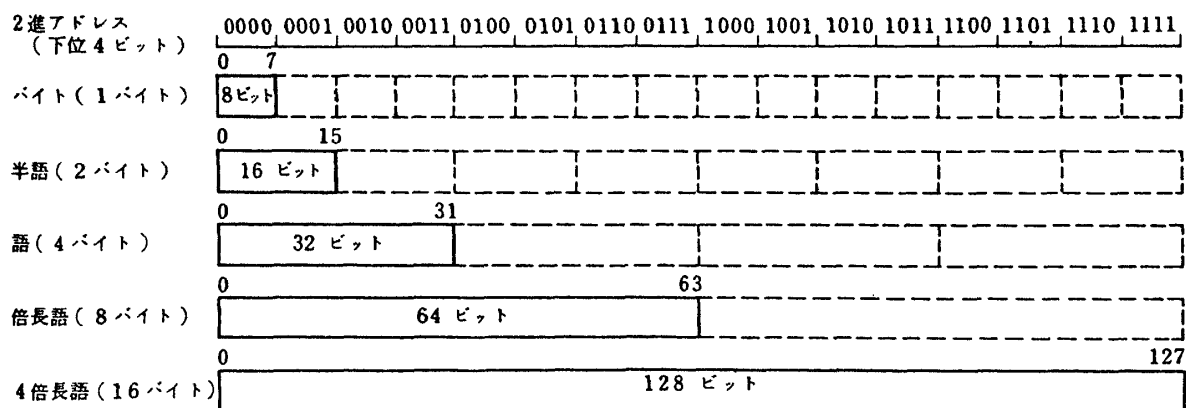


図 2-3 データ形式 (FACOM M-160 Fハードウェア解説書より転載)

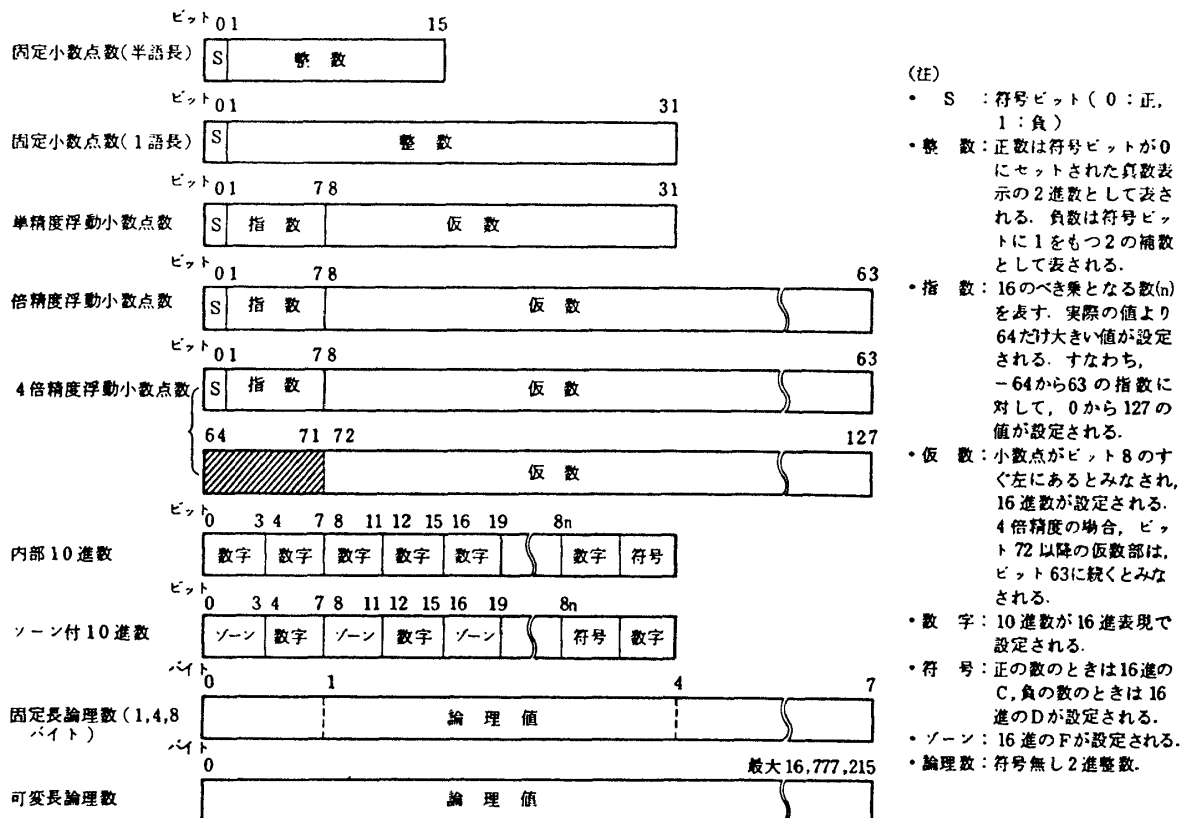


図 2-4 数値形式 (FACOM M-160F ハードウェア解説書より転載)

表 2-2 精度の比較

FACOM M-160F

項目 精度区分	データ形式					仮数部精度		指数部倍数	
	バイト	ビット	ビット構成			16 進	10 進	16 進	10 進
			仮数符号	指数	仮数				
単精度	4 (1 語)	32	ビット 1	ビット 7	ビット 24	桁 6	桁 7.2	16^{-64} ~ 16^{+63}	10^{-78} ~ 10^{+75}
倍精度	8 (2 語)	64	1	7	56	14	16.8		
4 倍精度	16 (4 語)	128	1	7	112	28	33.6		

TOSBAC 3400

項目 精度区分	データ形式					仮数部精度		指数部倍数	
	バイト	ビット	ビット構成			16 進	10 進	16 進	10 進
			仮数符号	指数	仮数				
単精度	6 (2 語)	48	ビット 1	ビット 10	ビット 37		桁 約 11		10^{-154} ~ 10^{154}
倍精度	12 (4 語)	96	1	21	74		約 22		

種類	形 式	長 さ (バイト)	アセンブラ記述形式
RR形式		2	OP R ₁ , R ₂
RX形式		4	OP R ₁ , D ₂ (X ₂ , B ₂)
RS形式		4	OP R ₁ , R ₃ , D ₂ (B ₂)
SI形式		4	OP D ₁ (B ₁), I ₂
S形式		4	OP D ₂ (B ₂)
SS形式		6	OP D ₁ (L, B ₁), D ₂ (B ₂)
		6	OP D ₁ (L ₁ , B ₁), D ₂ (L ₂ , B ₂)

OP: 命令コードを示す。
命令コードの先頭2ビットにより、命令の形式が決まる。
 { 00: RR形式
 01: RX形式
 10: RS/SI/S/RX形式
 11: SS形式

R: 演算用レジスタを指定する。
 { 一般命令: はん用レジスタ
 浮動小数点命令: 浮動小数点レジスタ
 ロードコントロール/ストアコントロール命令: 制御レジスタ

X: インデックスレジスタを指定する。
 B: ベースレジスタを指定する。
 D: 変位を表す。
 I: 直接数値を指定する。
 L: オペランド長を指定する。
 添字: オペランド番号を示す。

図 2-5 命令形式 (FACOM M-160Fハードウェア解説書より転載)

表 2-3 演算速度

単位 μ s

命 令		FACOM M-160F
固 定 小 数 点	ロ ー ド	0.84
	ス ト ア	1.1
	加 算	1.1
	乗 算	2.27
	除 算	7.35
浮 動 小 数 点	短 精 度 加 算	2.27
	倍 精 度 加 算	2.98
	短 精 度 乗 算	3.11
	倍 精 度 乗 算	6.89
	短 精 度 除 算	7.94
	倍 精 度 除 算	30.96

ーブルを参照することは、各テーブルに1回ずつアクセスすることであるが、このアクセス時間を節約するためにTLB^{注1)}という高速バッファを用意している。

TLBでは仮想アドレスと実ページアドレスの対応付けが行なわれており、TLBに登録されているものについては、セグメントテーブルおよびページテーブルにアクセスすることなしにその対応付けが行なわれる。TLBに登録されていないものについては先のアドレス変換が行なわれ、その時後の変換に備えてTLBに登録される。その方式はLRU方式^{注2)}で行なわれている。

従って、ユーザはプログラムの大きさを意識する必要はないが、メモリアクセスの方法については意識してプログラムコーディングする必要がある。詳細については、4章の「FORTRANについて」を参照されたい。

APCUを使用して自動電源切断を行なうことができる。また、異常発生の場合には警報信号をオペレータ控え室等に送ることができる。

上記に述べた仮想記憶方式およびAPCUの付加は、旧システムには見られないものであるが、その他に異なるものとしてTSS端末をあげることができる。TSS端末を利用することによりユーザは計算機設置場所から離れた場所でもファイルの保守、ジョブの実行等を行なうことができる。

注1) Translation Lookaside Buffer の略。

注2) Least Recently Used Method。最も近い過去に参照されたものは、再度参照される可能性が高いという性質を考慮した方式。

2.2 ソフトウェア

新システムにおけるジョブ処理形態は、バッチ処理および会話型処理に大別される。本節では、バッチ処理と会話型処理のあらましについてまず述べ、次にその中で用いられているアルゴリズムについて簡単に説明する。

2.2.1 バッチ処理のあらまし²⁾ (図2-6参照)

(1) ジョブの入力

ユーザがセンタの入出力室から投入したジョブや、TSS 端末から投入したジョブ（会話型リモートバッチジョブと呼ぶ）は、バッチジョブとして区別なく処理される。まずジョブは、ジョブ制御文とカード形式のデータに分けられ、データはそのまま磁気ディスク上に書き込まれ、ジョブ制御文は文法チェックを受ける。その結果、ジョブ制御文に誤りがなければ指定されたジョブクラスの入力待ち行列に、誤りがあれば出力待ち行列に接続される。

(2) ジョブの実行

入力待ち行列に並んでいるジョブの取り出しは、マスタスケジューラによって行われる（2.2.3項の(1)参照）。取り出されたジョブに対して、まずジョブ全体に関する開始処理（ジョブ開始処理と呼ぶ）が行われ、続いて最初のジョブステップを実行する。

のに必要な資源が用意される（ジョブステップ開始処理と呼ぶ）。（このうち、仮想空間の割付および実記憶の割付に関しては、2.2.3項の(2)参照。）その後、ユーザプログラムが実行される（各プログラムに対するCPUの割付制御方式に関しては2.2.3項の(3)参照）。

実行時に出力された結果は磁気ディスク上に書き出され、実行終了後には不要になった資源の解放等が行われる（ジョブステップ終了処理と呼ぶ）。もし、そのジョブに後続のジョブステップがあれば、再び次のジョブステップの開始処理がなされ、実行される。

全てのジョブステップが終了したジョブに対しては、そのジョブが使用した全ての資源の解放、システム出力待ち行列への接続等（ジョブ終了処理と呼ぶ）が行われる。

(3) ジョブの出力

システム出力待ち行列に接続されているジョブは、順次、指定された装置へ出力される。ジョブ制御文のホールド指定により出力を見合せていたジョブも、コンソールからのホールド解除の命令（リリース指令）を投入することにより同様に出力される。

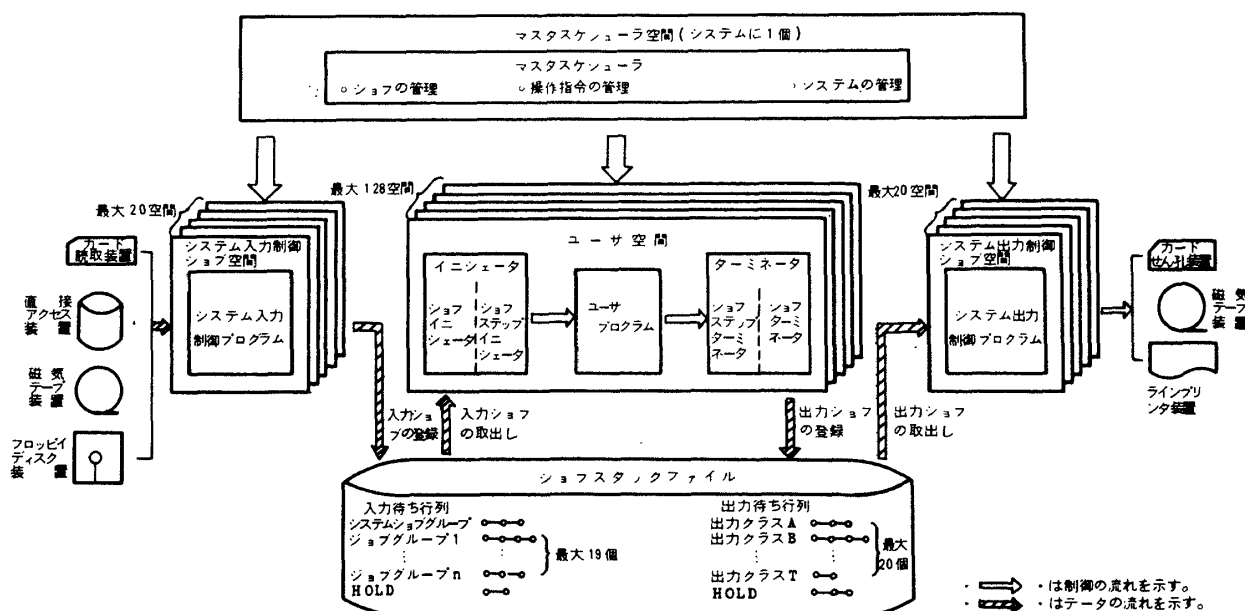


図2-6 バッチジョブのあらまし(FACOM OSIV/X8制御プログラムの諸機能より転載)

2.2.2 TSS処理³⁾

(1) TSSセッションの開設

TSS処理とは、TSS端末からの通信回線を介して計算機システムと「会話」しながら処理する処理方式であり、そのためにはまず、TSSセッションを所定の手順で開設しなくてはならない。ただし、ユーザは一時点で複数のセッションの開設は行えない。

(2) TSS処理の実行

ユーザは、TSSコマンドをシステムに投入することによって、ファイルの作成・編集、プログラムの翻訳・実行、会話型リモートバッチジョブの投入等の処理を行える。TSSコマンドは、バッチ処理の場合と同様、CPUの割付制御の対象となる。

(3) TSSセッションの閉設

TSS処理を終了させる場合には、所定の手順でセッションの閉設を行えば良い。この時点からその端末は、他のユーザも使用可能な状態となる。

2.2.3 システムの各種アルゴリズム

(1) ジョブのスケジューリング方式⁴⁾

ジョブのスケジューリングを制御するパラメータには、システム全体の起動多重度、実行多重度の外に、各ジョブクラス毎の優先順位、起動多重度、実行多重度がある。ただし、ジョブクラス毎の実行多重度は起動多重度より小さいか等しい値とし、各ジョブクラスの実行多重度の和をシステムの実行多重度とする。

マスタスケジューラは、ジョブの投入時およびジョブの終了時にシステム全体の起動多重度に空があるか否かを調べ、空があれば、下記の手順によりジョブの起動を試みる。

まず、ジョブ起動多重度に空があるジョブクラスの中で起動待ちジョブを有するものの有無を調べる。もし条件を満たすジョブクラスがあれば、その中で最も優先順位の高いジョブクラスのジョブを起動する。同一優先順位のジョブクラスが複数あった場合には、それぞれのジョブクラスから交互に取り出される。そしてシステムはそのジョブクラスに起動ジョブがある限り実行多重度分のジョブをロールイン状態（プログラムの一部または全部が主記憶上にある状態、すなわち実行可能状態）としておき、残りのジョブはロールアウト状態（プログラムを磁気ディ

スク上に退避させた状態、すなわち実行不可能状態）としておく。したがって、混雑時には、各ジョブクラスの実行中ジョブは実行多重度で制限されている。また、起動すべきジョブがないために実行多重度に満たないジョブクラスがあれば、優先順位に従ってかわりに実行させるジョブクラスを決定し、そのジョブクラスのロールアウト状態のジョブをロールイン状態にして実行させ、再び先のジョブクラスのジョブが起動された時点でロールアウト状態にもどす。

(2) 仮想記憶制御⁴⁾

システムで実行されているジョブは、その全体がMSU上に存在しなければならない理由は全くない。仮想記憶方式とは、多重処理されている各ユーザジョブのプログラムのうち、頻繁に使用されている部分のみを実記憶上に存在させ、その他の部分は磁気ディスク上に退避させておく方式であり、必要に応じて実記憶上に読み込むことにより処理効率をあまり変化させることなく実行させることを狙っている。

ここでは、プログラムの持つ論理的なアドレス空間を仮想空間と呼ぶが、これはMSUおよび外部ページファイルと呼ばれる磁気ディスク上の退避領域を実現媒体とする仮想的なアドレス空間である。したがって、ユーザは実記憶を一切意識することなく、仮想空間で実行されるプログラムを作成すれば良い。

新システムにおける仮想空間はジョブおよびTSSセッション毎に1個存在し（この方式を多重仮想記憶方式と呼ぶ）、その割付はジョブステップおよびTSSセッション開始処理で行われる。各ジョブおよびTSSセッションに対して許される仮想空間の大きさは最大16MBであり、その内のシステム共通領域（新システムの場合、約4MB程度である）を除く領域がユーザのプログラムに与えられるジョブ専用領域となる（ユーザプログラム以外の言語処理等サービス処理プログラムの大きさに関しては、2.2.4項参照）。

仮想記憶制御では、各ユーザの仮想空間はページと呼ばれる4KBの単位で管理され、各ページはLRU方式で自動的にMSUへの割付（ページイン処理と呼ぶ）や、外部ページファイルへの掃き出し（ページアウト処理と呼ぶ）が行われる。したがって、実行中の全仮想空間の和が使用可能な実記憶容

量に比べて相対的に大きい場合程、ページイン・ページアウト処理のための入出力回数が増大することになる。また、特に応答性を要求するプログラムに対しては、その実行中全てのページを MSU 上に割付けておくことも可能である。

なお、新システムでは、多重仮想記憶方式の採用

によりジョブ間の機密保護を行うとともに、実記憶の有効利用とシステムオーバーヘッドの減少のために、各種の技法を採用している。

(3) CPU の割付制御⁴⁾

ジョブに CPU 使用権を与えるものをディスパッチャといい、ディスパッチャはバッチジョブ、TSS

表 2-4 サービス処理プログラムの機能とプログラムサイズ

	機 能	仮 想 記 憶 使用量(KB)
FORTRAN	FORTRANプログラムの翻訳	256
FASP	FASPプログラムの翻訳	128
ソート・マージ	ファイル内レコードの並べかえ、及び複数ファイル内レコードの併合	180
リンケージエディタ	オブジェクトモジュール・ロードモジュールの結合編集	128
ローダ	オブジェクトモジュール・ロードモジュールの結合編集及び実行	128
LIBE	ファイルの登録・更新・複写・印刷・退避・復元等	140
ユーティリティ	直接アクセスボリュームの初期化(VINT)	44
	磁気テープボリュームの初期化(MINT)	40
	磁気テープボリューム上のデータを他の磁気テープに複写(MCPY)	256
	直接アクセスボリューム上のファイル名変更(RNAM)	64
	交代トラックの割当て(GALT)	64
	直接アクセスボリューム上のデータの印刷(DLST)	36
	磁気テープボリューム上のデータの印刷(MLST)	108
	磁気テープボリューム上のSYSOUTファイルの媒体変換(MSOUT)	128
	磁気テープボリューム上のデータの比較、照合(MCMP)	256
	ファイルの複写(FCPY)	128
	索引順編成ファイルの作成(ISFG)	64
	ファイル内容の印刷(FDMP)	64
	ファイルの消去(FDLT)	64
	ジョブ制御マクロライブラリ、操作指令マクロライブラリの編集(JCME)	128
	カタログ登録簿の編集(CTFE)	64
	直接アクセスボリューム上のファイルを磁気テープボリューム上へ退避(VSAV)	128
	VSAVで退避したファイルを直接アクセスボリューム上へ復元(VRCV)	128
	索引順編成ファイルの属性の印刷(IDIS)	64
	直接アクセスボリューム上のファイルの属性の印刷(FCHK)	64
	VTOCの修復(MVOL)	64
	オペレータメールボックスファイルの編集(OMBX)	64
	カタログ世代ファイルの編集(GDGM)	64

コマンドの区別なく実行可能なもののうち実行優先順位の最も高いジョブに CPU 使用権を与える。バッチジョブの場合、実行優先順位は、ジョブまたはジョブクラスであらかじめ定められているが、入出力頻度の高いものは優遇されるように、ジョブの実行優先順位を動的に制御すること（ダイナミックディスパッチング制御と呼ぶ）も可能である。また、TSS コマンドの場合は、実行優先順位は 1 種類しかなく固定であるが、TSS コマンド間では CPU 使用時間の短いコマンド程優遇されるようなスケジュールがなされている。

2.2.4 サービス処理プログラムについて^{5), 6)}

新システムでサポートされているサービス処理プログラムの機能および仮想記憶使用量を表 2-4 に示す。

これらのプログラムは、ユーザがジョブ制御マクロを使用したり、コンソールから指令を投入したりすることによって、適時呼ばれる。

3. 運用について

本章では、新システムにおける運用方針を明確にし、その実現手段であるセンタルーチンおよびシステム編集について述べる。

3.1 運用方式

システムの運用方式は、システム効率、保全性、操作性、融通性の観点から定める必要がある。

第一のシステム効率の観点からは、CPU、MSU、入出力機器、補助記憶装置、チャネル等のシステム資源の競合によるジョブ処理の遅れをできる限り防止するようなジョブ運用方式を定めることが要請される。ここで言うジョブ運用方式とは、ジョブをその特性に応じてクラス別けし、各クラス毎に仮想記憶使用量、CPU 使用時間等の適切な制限値を設け、各々のジョブクラスのジョブを適切な多重度で実行させることであり、その結果、システム資源に対する要求はより効果的に処理される。

第二の保全性の観点からは、ユーザのシステム使用権の確認やファイルの機密保護を行うとともに、定期的に磁気ディスク上のファイルを磁気テープ等にすい上げ、ファイルのバックアップ等を行う事が要請

される。

また、第三の操作性の観点からは、システムの開始・終了処理を含め、必要な機能が簡単な手順で、しかも誤操作なく使用できること、また、ファイルの媒体変換やシステムの状況表示など使用頻度の高い機能をできるだけ簡単に操作できることなどが要請される。

最後に、第四の融通性の観点からは、運用方針に基づいてシステム編集（3.3 節参照）を行った後でも、ハードウェア構成の変更、運用方針の変更などに応じて編集をやり直すことはもちろん、その一部のみを再編集したり、システム開始時にシステムパラメータ値の変更を行う機能等が必要となる。また、同様の理由により、センタルーチン作成後も、その修正や変更が随時可能でなければならない。

本節では、運用方式を、運用管理方式、ユーザ管理方式、ファイル管理方式、装置管理方式に大別し、各管理方式の内容とその実現方法について述べる。

3.1.1 運用管理方式

先に述べたように、システムの運用管理方式は、各センタの特殊性を十分に活かすように検討されなければならない。また、ジョブの運用方式は、システム資源を有効に使用するために不可欠なものであるが、反面、ユーザを拘束する条件にもなるため、慎重に決定しなければならない。

本節では、運用全般に関する通常運用時間帯およびユーザタイム（後述する）についての運用方式について述べるとともに、ジョブの運用方式についても明らかにする。

(1) 通常運用時間帯の運用方式

システムの開始・終了処理および運用のための日常の処理はできるだけ自動化し、オペレータの負荷の軽減とオペレーションミスの防止を図る。たとえば、システムの開始処理は、空調および電源の投入を行うだけで、システムの立ち上げに必要な一連の手続きが自動的に行われ、ジョブ処理が開始される。また、システムの終了処理は、その旨のコマンドを投入することによって立ち下げに必要な一連の手続きが終了するので、後は空調の電源を切断するだけで良い。この場合、実行中のジョブがあっても、主記憶上の情報は全てディスク上に退避され、システ

ムの開始処理によってその時点の情報は自動復元される（この機能を翌日再開機能と呼ぶ）。

また、ユーザジョブの操作やユーザファイルの管理は全てユーザ自身が行えるようにするので、そのために必要な操作指令はオープンコンソールから随時使用できるように、簡単なコマンドを用意する。

(2) ユーザタイムの運用方式

新システムでは、通常運用時間帯に実行できないジョブや緊急ジョブを処理できるように、通常運用時間帯以外にシステムを使用できる時間帯（ユーザタイムと呼ぶ）を設ける。

ユーザタイムは通常、1人のユーザがシステムを独占使用できる時間帯であるため、実行されるジョブに対する資源の規制はほとんど必要ない。そのため、通常のジョブクラスに比べて規制の少ないジョブクラスを設ける必要がある。

また、システムの終了処理には、そのクラスで実行中のジョブを次の通常運用時間帯へ持ち越すことは、通常の運用を円滑に行う上での妨げとなるため許さない。

(3) ジョブ運用方式

新システムのバッチジョブクラスは、仮想記憶使用量・CPU使用時間ともに小規模なクラス（SHRTと呼ぶ）、中規模なクラス（MIDLと呼ぶ）、

および大規模なクラス（LONGと呼ぶ）を設け、運用時間帯は常に各ジョブクラスの実行多重度を1～2多重にして運用する。ただし、LONGでも処理できないジョブに対しては特殊なジョブクラス（SPECと呼ぶ）を設けるが、このクラスのジョブは運用時間帯に常に実行が許されるというのではなく、システムの実行多重度に空きがある場合に実行される。

また、上記の他に、センタ業務用ジョブクラスとしてCENTを、ユーザタイム用ジョブクラスとしてUSERを設ける。

表3-1にジョブ運用方式について示す。ただし、表中の値は旧システムにおけるユーザジョブの属性をもとにして決めたものであるため、システム導入後、変更されることもある。

この表中のTSSというジョブクラスはTSS端末から投入されたコマンドを実行するためのものであり、UTLTというジョブクラスはコンソールから投入された媒体変換等のコマンドを実行するものであるため、直接ユーザがこれらのジョブクラスを指定する必要はない。

また、同表で指定された各種制限値は、ジョブ投入時に呼ばれるセンタルーチンにその機能を組み込むことにより実現する。

表3-1 ジョブ運用方式

ジョブクラス	実行優先 順位	ジョブクラス別 起動多重度	ジョブクラス別 実行多重度	ジョブ制限値			
				CPU時間 (分)	LP 出力 (注1) (枚)	仮想記憶 使用量(KW)	(注2) R/V
TSS	高 ↑ ↓ 低	10	3	1	—	70	V
CENT		0	0	∞	5,000	2,800	V
UTLT		1	1	—	5,000	70	V
SHRT		3	2	3	500	100	R
MIDL		2	1	15	500	250	V
LONG		1	1	60	500	400	V
SPEC		1	0	180	5,000	2,800	V
USER		0	0	∞	5,000	2,800	V
合 計		18	8				

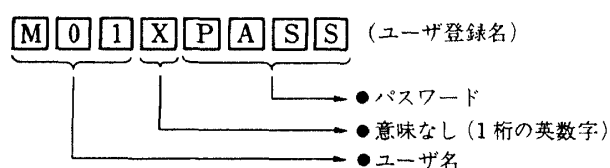
(注1) LP出力の場合は、出力量の制限値の他に、ユーザが出力量を省略した場合にセットされる省略値を別に設ける方針である。

(注2) Rのジョブは、ジョブ実行中そのプログラムが常時主記憶に割り付けられ、Vのジョブはシステムによりページインやページアウトが適時なされることを示す。

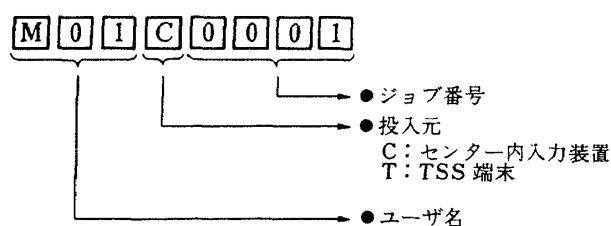
3.1.2 ユーザ管理方式

新システムでは、ユーザに対し可能な限りシステムの機能を解放する方針である。したがって、システムを円滑に運用し、かつ、システムの保全を考えると、ユーザの資格チェックを行わなければならない。そこで、ユーザのジョブ投入時および TSS セッション開設時に、そのユーザのシステム使用権を確認する。ジョブ投入時の場合、ユーザの指定したユーザ登録名（その形式を図 3-1 に示す）を、あらかじめシステムに登録してあるユーザ名とパスワードの対と比較照合することによって行う。この比較照合の結果、システム使用権の確認されたジョブは同図に示すジョブ識別番号が発行される。以後このジョブの操作は全てこの番号で行われる。また、TSS セッション開設時の場合には、ユーザ名、パスワードの順にシステムに登録してあるものと照合され、システム使用権が確認された場合にのみコマンド投入が許される。このことにより、システム使用権のないユーザは、システムを使用できないことになる。

また、ユーザが割当量を越えて大量のファイルを作成し、他のユーザに迷惑を及ぼすことを防ぐために、3.1.3 節で述べるファイル管理方針に反するユーザに対しては、一時的にシステム使用権を保留にする。ただし、システム使用権が保留中のユーザでも、ファイルの消去やシステム使用権回復の手続き



(i) ユーザ登録名の形式



(ii) ジョブ識別番号

図 3-1 ユーザ登録名とジョブ識別番号

は、TSS 端末およびオープンコンソールからは行える。

3.1.3 ファイル管理方式

磁気ディスク上のファイルは、それぞれの機能、目的、管理方針に応じて各種のチェックや処理がなされるべきである。ここでは、特にユーザに関係の深いファイルに対して行うべきファイル管理方式について述べる。

(1) 個人ファイルの割当管理および期限管理

個人ファイルは、システムから半永久的に貸出されるもの（永久貸出個人ファイルと呼ぶ）と、期限付で貸出されるもの（期限付貸出個人ファイルと呼ぶ）を設けて運用する。前者は、システム使用権のあるユーザが規定の割当容量の範囲内で常時使用可能な領域であり、後者は一時的に大容量の個人ファイルが必要とするユーザに対して期限付で割当られる領域である。

したがって、これらのファイルに対しては、常時各ユーザの割当量と使用量を把握し、ファイルの作成・更新時には各種のチェックを行わなければならない。しかし、新システムでは、TSS 処理に対して常時ユーザのファイル使用量を更新することがむずかしく、また、各種チェックを行うためのオーバーヘッドも大きいと考えられるため、一定の間隔で全ユーザのファイル使用量をまとめて更新し、作成・更新時のチェックは1切行わないものとする。また、ファイル使用量の更新時には、同時に各ユーザのファイル使用量と割当量の比較を行い、使用量が割当量を越えたユーザのシステム使用権を保留にし、使用権が割当量以内に納まったユーザのシステム使用権を回復させる処理を行う。

また、期限付貸出個人ファイルに対しては、期限の切れた時点でそのユーザの割当量を規定の容量にもどすため、使用量がその値を起えていると、前述の方式でそのユーザのシステム使用権は保留になる。

これらの処理は、システム開始処理に組込むセンタルーチンで行い、必要に応じてセンタ管理者がメインコンソールから随時使用できるようなコマンドを用意する。

(2) 個人ファイルの機密保護管理

個人ファイルの機密保護のために、新システムで

は①参照・更新共に自由，②参照は自由だが，更新はパスワードチェックを行う，③参照・更新共にパスワードチェックを行うという3つの機密保護レベルを設け，ファイル作成時にそのファイルのパスワードと機密保護レベルを指定できるようにする。また，機密保護に対する指定のないファイルは，①を指定したものとみなして処理を行う。

したがって，パスワードチェックの指定がなされているファイルを使用する場合には，必ずそのファイルのパスワードを指定しなければならない。

また，ファイルの作成および消去に関しては本人以外には行えないようにする。

以上に述べた個人ファイルの保護は，ジョブ投入時に呼ばれるセンタルーチンにその機能を組み込むことによって実現する。

(3) ディスクの有効利用について

個人ファイルのために割当てられている磁気ディスクは2本あり，その目的上，多数のユーザが頻繁に使用する。その結果，磁気ディスク上の空領域は分散化しやすく，それに対処するためにボリュームの圧縮を行い，空領域を1箇所にとめる必要性がでくる。そのため新システムでは定期的に個人ファイル用ボリュームの圧縮を行う。

ボリュームの圧縮は，その内容を一端磁気テープ等へ移し，再び元のボリュームへ戻すことにより行われる。この時使用した磁気テープ等のボリュームを一定期間保管することによって，個人ファイル

のバックアップ用としても使用できる。

3.1.4 装置管理方式⁷⁾

新システムに設置されている装置のうち，コンソール（特にオープンコンソール）およびTSS端末等に関してはその機能を明確にし，それ以外の装置に対しては使用方法を規制することによって装置の管理方式が定まる。ここで使用方法の規制とは，各装置毎にジョブ入力，プログラム入力（カードリーダーからジョブを入力する際，プログラムの全部または一部をカードリーダー以外の装置から入力することを示す），実行時使用，およびシステム出力の使用を規制することである。

表3-2に，新システムの各種装置の運用方式を示す。このうち，フロッピーディスクおよび7トラックの磁気テープの実行時使用は，現オペレーティングシステムではサポートされていないが，ユーザの便宜を図るために特に追加した機能である。

また，紙テープには，プログラムやジョブを保存しておく必要性がきわめて低いため，実行時使用のみを許した。

以下に各装置の使用方法について述べるが，媒体変換機能に関しては，オープンコンソールの項でまとめて述べることにする。

(1) カード読取装置（CR）

CRはバッチジョブ投入のための装置であり，ユーザはカードをセットしてスタートボタンを押すことによりジョブの投入ができ，CRはその完了と同

表3-2 各種装置の運用方式

装置名 \ 使用方法	ジョブ入力	プログラム入力	実行時使用	システム出力	備 考
カード読取装置	○	×	×	×	
フロッピーディスク装置	○	×	○	×	
磁気テープ装置(7トラック)	×	×	○	×	
磁気テープ装置(9トラック)	○	○	○	×	
磁気ディスク装置	○	○	○	×	
磁気ディスク装置 (カートリッジ式)	○	○	○	×	
紙テープ読取装置	×	×	○	×	
ラインプリンタ装置	×	×	×	○	
XYプロッタ装置	×	×	×	○	

時に自動停止する。また、ジョブ入力の際、磁気テープやディスクからプログラムの一部を同時に読み込ませることができる。その場合は、必要に応じて磁気テープやカートリッジ式磁気ディスクにマウント要求^{注)}があがるので、その指示に従って媒体をセットすれば良い。

(2) フロッピーディスク装置 (FPD)

FPDは、バッチジョブ投入用および実行時のデータの入出力に使用できる。

FPDからジョブを投入する場合には、まずオープンコンソールからジョブ投入用コマンドを入力し、システムからマウント要求があがった時点でその指示に従ってFPDをセットすれば良い。

また、実行時に使用する場合には、ジョブが実行状態になる直前にマウント要求があがるので、その指示に従ってセットする。

(3) 磁気テープ装置 (MT)

MTは、7トラックMTと9トラックMTとがあり、7トラックMTは実行時のデータ入出力用（ただし、使用に当っては通常のFORTRANのREAD/WRITE文ではなくCALL文形式でなければならない）として、9トラックMTはバッチジョブ投入用、プログラム入力用（(1)カードリーダー参照）および実行時のデータ入出力用として使用できる。使用方法是FPDの場合と同様である。

(4) 磁気ディスク装置 (DK, DC)

磁気ディスク装置には、媒体が装置固定のもの（DK）と、取りはずしが可能なカートリッジ式のもの（DC）があり、共にジョブ投入用（TSS端末から行える）、プログラム入力用（(1)カードリーダー参照）、実行時のデータ入出力用として使用できる。DKは媒体が固定であるため、使用時にユーザが行うべき操作は一切ないが、DCはMTと同様にマウント要求があがった時点でその指示に従いセットしなければならない。

(5) 紙テープ読取装置 (PTR)

PTRは、ジョブ実行時にデータの入力用として使用できる。その場合には、ジョブ実行直前にマウント要求があがるので、その指示に従いセットす

れば良い。

(6) ラインプリンタ装置 (LP)

LPは出力専用の装置であり、ジョブは終了順に自動出力される。自動出力させない場合には、ジョブ制御文でその旨の指定をすれば、ジョブ実行終了後結果の出力は保留され、ユーザがそれを解除するコマンドをオープンコンソールから投入した時点で出力される。ただし、保管期日には制限を設ける。

(7) XYプロッタ装置 (XY)

XYは出力専用の装置であり、出力結果はユーザの要求があるまでシステム内に保管される。出力の要求はオープンコンソールからその旨のコマンドを投入することによって随時可能である。ただし、LP出力結果と同様に保管期日に制限を設ける。

また、XYの出力結果をグラフィックディスプレイでモニタし、その出力をシステムに依頼した場合にも同様に処理される。

(8) グラフィックディスプレイ装置 (GD)

GDはTSS端末の1つであるが、図形表示を行えるところに大きな特徴がある。したがって、TSS端末と同様の手続きでセッションが開設でき、またTSS端末としても使用できる。XY出力結果をモニタする場合には、まずそのジョブの実行時にGDを使用する旨の制御文をつけ、個人ファイル上に出力結果を保存しておく（これをGDファイルと呼ぶ）。ユーザは任意の時点でGDのセッションを開設し、そのファイルを指定して出力結果をモニタすることができる。また、その結果をXYへ出力するためのコマンドも用意されている。

(9) オープンコンソール (OCSL)

新システムでは、ジョブの投入から取り出しまでに要するジョブの操作や、ユーザに割当られた個人ファイルの管理は、ユーザ自身がオープンコンソールを使用して行う事を原則とする。したがって、そのために必要な各種ディスプレイ機能やジョブ操作機能、ファイルの媒体変換機能等は、できるだけ簡単なコマンド操作で行えるようにしなければならない。また、運用方針によって許せない機能の使用をチェックしたり、誤操作をできるだけなくすための

注) システムがユーザに対して、媒体を装置に取り付け実行可能状態にすることを要求するためのメッセージである。

配慮も必要である。

以下に、オープンコンソールで行える機能についての説明を行う。

a) ジョブ操作のための機能

- 制御文で指定したジョブの起動保留・出力保留の解除
- ジョブのキャンセル
- FPD, MTからのジョブの投入
- XY結果の出力
- LP, XYのキャンセルおよび再出力

b) ディスプレイ機能

- システム資源の状態表示
- システム内ジョブの状況表示
- 自分の全ファイルのファイル名および使用量表示

c) ファイルの媒体変換機能^{注1)} およびファイル操作の機能

- ファイル単位の媒体変換（入力側としては、FPD, MT, DK, DC, PTR を、出力側としては、FPD, MT, DK, DC, LP を考えている。ただし、FPD間の媒体変換は装置が1台しかないため、オフラインの FPD で行えるようにする。）
- ボリューム単位の媒体変換（入力側、出力側共に MT）
- ファイルの圧縮
- ファイルの消去

d) その他の機能

- 自分のファイル使用量の更新およびそれに伴うシステム使用権の保留・保留解除
- MT, DCの初期化^{注2)}（ FPDの初期化はオフラインのフロッピーデータエントリで行える）

また、オープンコンソールは簡単に TSS 端末に切り換えることが可能である。

(10) TSS 端末

TSS 端末は、通常行えるファイルの作成・編集、

プログラムの翻訳・実行、および会話型リモートバッチジョブの投入等の機能に加えて、以下の機能をサポートする。

- TSS 端末から投入したジョブの状態表示およびキャンセル（ただし、起動前のジョブに対しては行えない）
- 出力量の多いコマンド処理結果の LP 出力
- 出力保留中ジョブのリスト検索

なお、本節の(1)～(7)項までに述べた装置の使用方法は、ジョブ投入時に呼ばれるセンタルーチンに組み込むことにより実現する。また、(8)～(10)項で述べた装置の使用方法に関しては、コマンド投入時に呼ばれるセンタルーチンで、そのコマンドが許されているか否かを判定することにより実現する。

3.2 センタルーチンについて

現在の汎用計算機システムは、不特定ユーザの種々多様な要求に対処すべく、多様化し、複雑化する傾向にある。また、各センタでは、運用方針、ユーザジョブの属性、システム構成が異なるために、センタ固有の管理方式の実現が必要になる。

このような条件のもとで、どんなユーザでも容易にシステムを利用できるようにするためには、ユーザとシステムの間に位置するセンタルーチンの役割は、非常に大きいと考えられる。

本節では、新システムで考えられている種々のユーザ管理案を実現するために必要となる航技研独自のセンタルーチンの内容について述べる。ただし、これらの機能は、実際に行う上で、必ずしも説明の順序で行われるものとは限らない。

(1) システム初期設定時の処理

毎日のシステム運用開始の前処理に、次の機能をもつセンタルーチンを用意する。

- 期限付貸出個人ファイルの期限が満了になったユーザの割当量変更処置
- 全ユーザの個人ファイル使用量の更新と、それ

注1) ファイルの媒体変換の機能は、バッチジョブからも使用可能なように、ジョブ制御マクロを用意する方針である。

注2) 初期化とは、ボリュームにラベルを書き込み、システムで使用可能な状態にすることである。当システムでは、MT, FPD, DKのラベルは全て通し番号をつけて管理する方針であるため、NL形式のMT以外は全てセンタ指定のボリューム通番を書き込まなければならない。

に伴うシステム使用権の保留および保留解除処理

- 保管期日が満了になったジョブの処理結果，およびXY出力結果の自動出力

(2) バッチジョブ入力確認時の処理

システムが，バッチジョブ中にユーザ登録名を確認した時点で次の処理を行う（図3-2参照）。

- ユーザ名確認
- システム使用権保留か否かの確認
- パスワードの確認
- ジョブの追番発行と追番更新
- 受付処理状況の表示
- エラージョブのアボート処理

(3) ジョブ文確認時の処理

システムが，バッチジョブ中にジョブ文を確認した時点で次の処理を行う（図3-3参照）。

- ジョブクラスの確認と必要に応じた標準値セッ

ト

- ジョブ打切値の確認と必要に応じた標準値セッ

ト

- 出力先の確認と必要に応じた標準値セット
- ユーザが指定できない全パラメータのセット
- ユーザが指定可能なパラメータの確認と必要に応じた標準値セット

- エラー内容の表示

- エラージョブのアボート処理

(4) ジョブステップ文確認時の処理

システムが，バッチジョブ中にジョブステップ文を確認した時点で次の処理を行う（図3-4参照）。

- 使用プログラムの実行権の確認
- ジョブステップ打切値の確認と必要に応じた標準値セット
- ユーザが指定できない全パラメータのセット

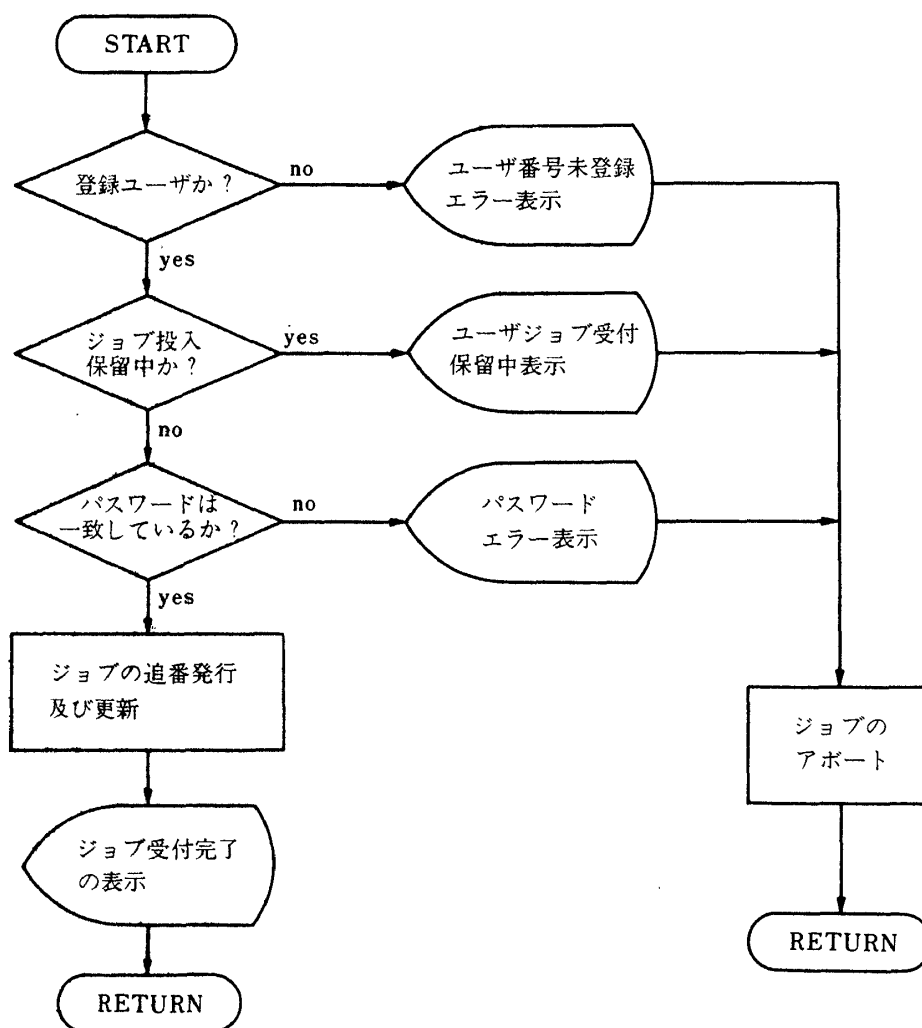


図3-2 バッチジョブ入力確認時のセンタルーチンの処理

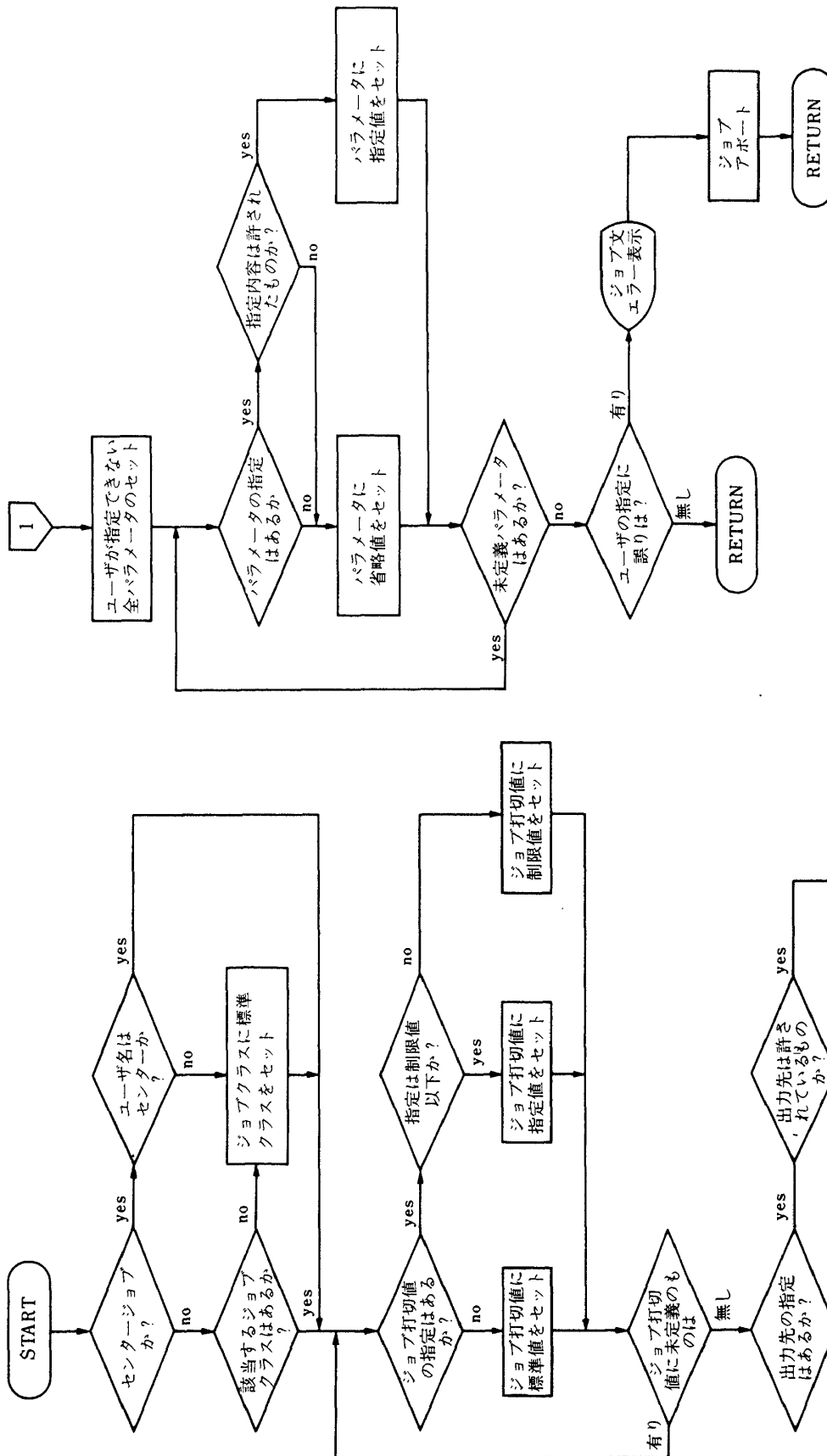


図3-3 ジョブ文確認時のセンタルーチンの処理

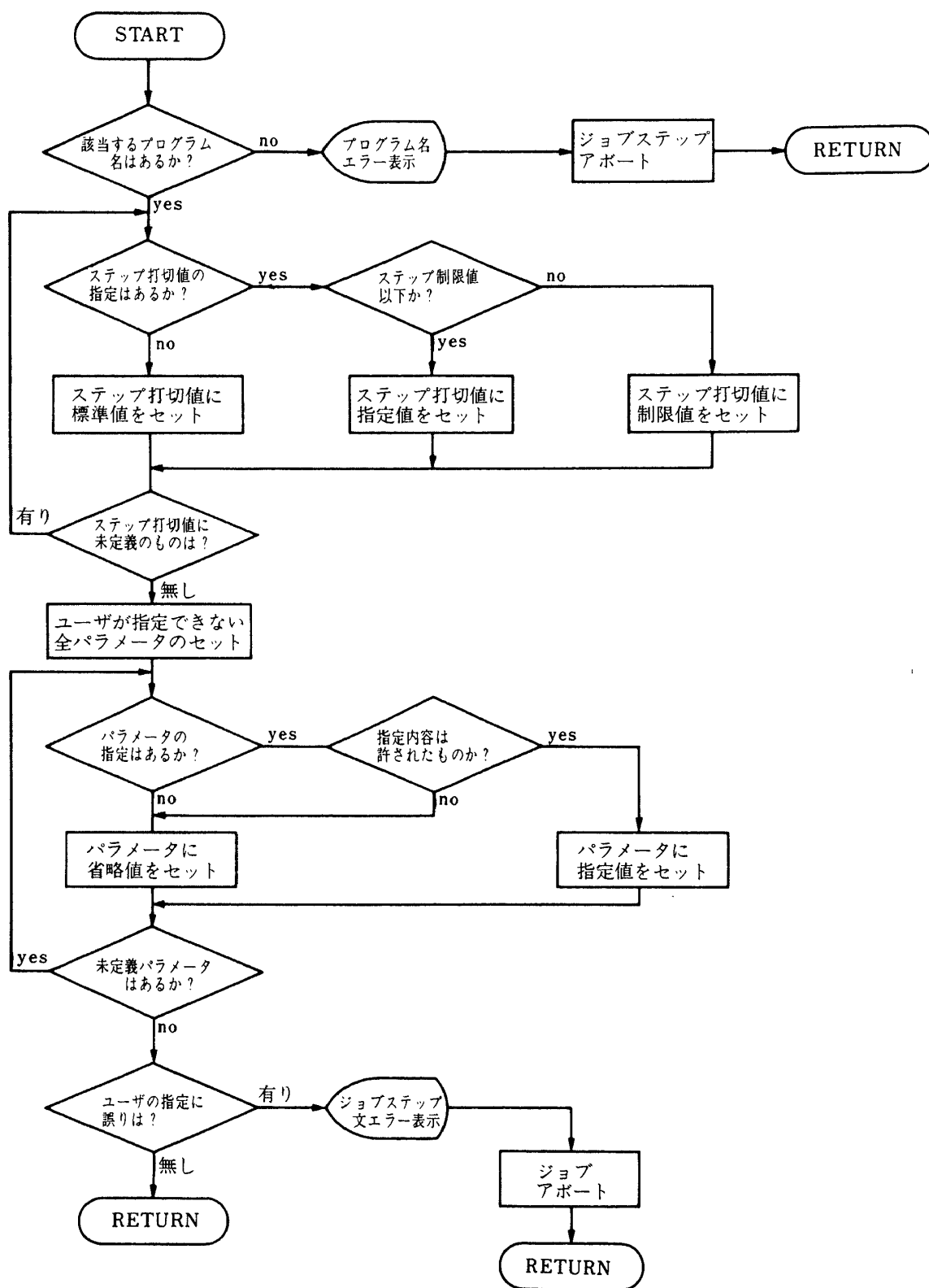


図 3-4 ジョブステップ文確認時のセンタルーチンの処理

- ユーザが指定可能なパラメータの確認と必要に応じた標準値セット

- エラー内容の表示

- エラージョブのアボート処理

(5) ファイル文確認時

システムがバッチジョブの中にファイル文を確認した時点で次の処理を行う(図3-5参照)。

- 使用装置の確認

- 使用装置の使用権の確認

- 使用ファイルの使用権の確認

- 個人ファイルの消去権の確認

- 個人ファイルの自動カタログ化処理

- エラー内容の表示

- エラージョブのアボート処理

(6) TSSセッション開設時の処理

TSSセッション開設時に、次の処理を行う(図

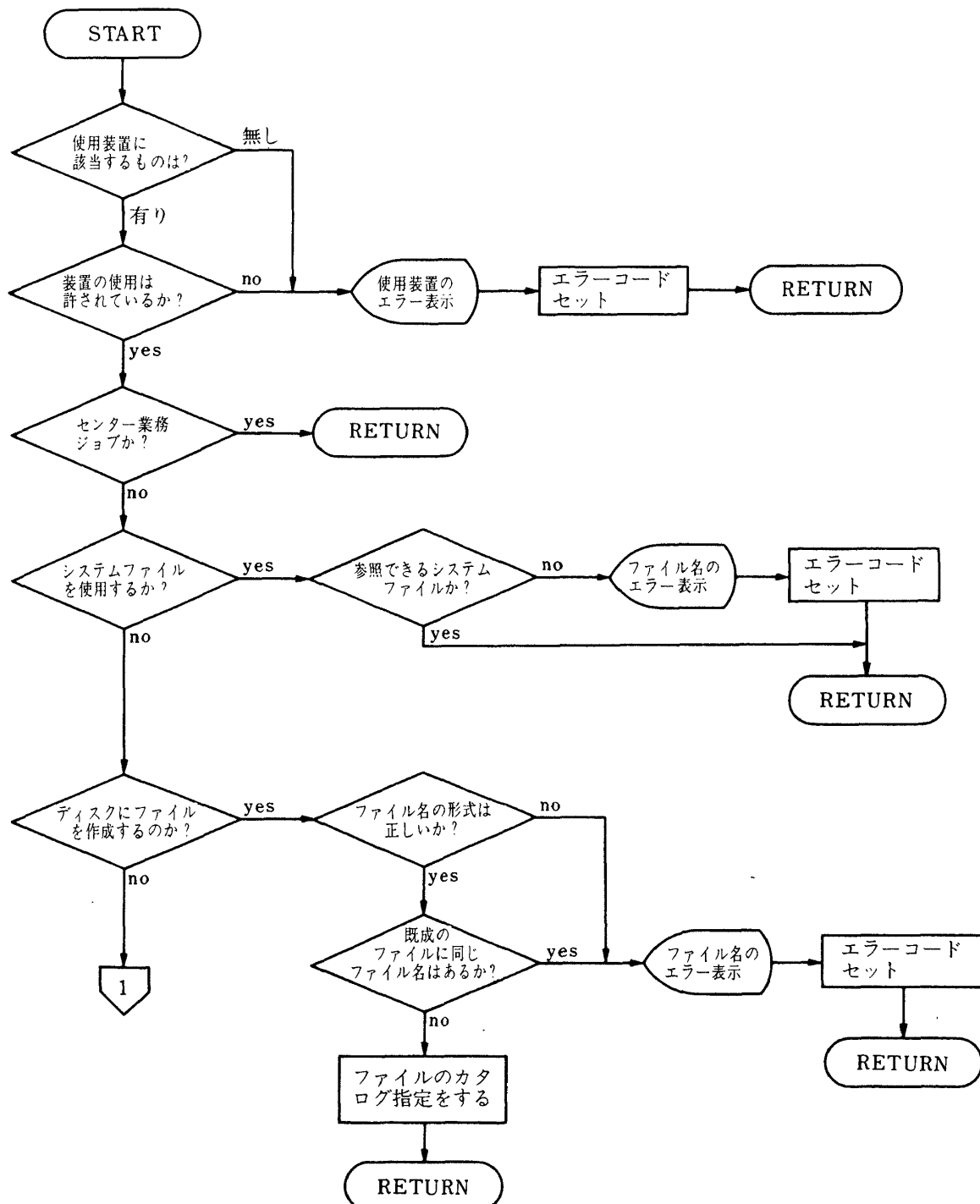


図3-5 ファイル文確認時のセンタルーチンの処理

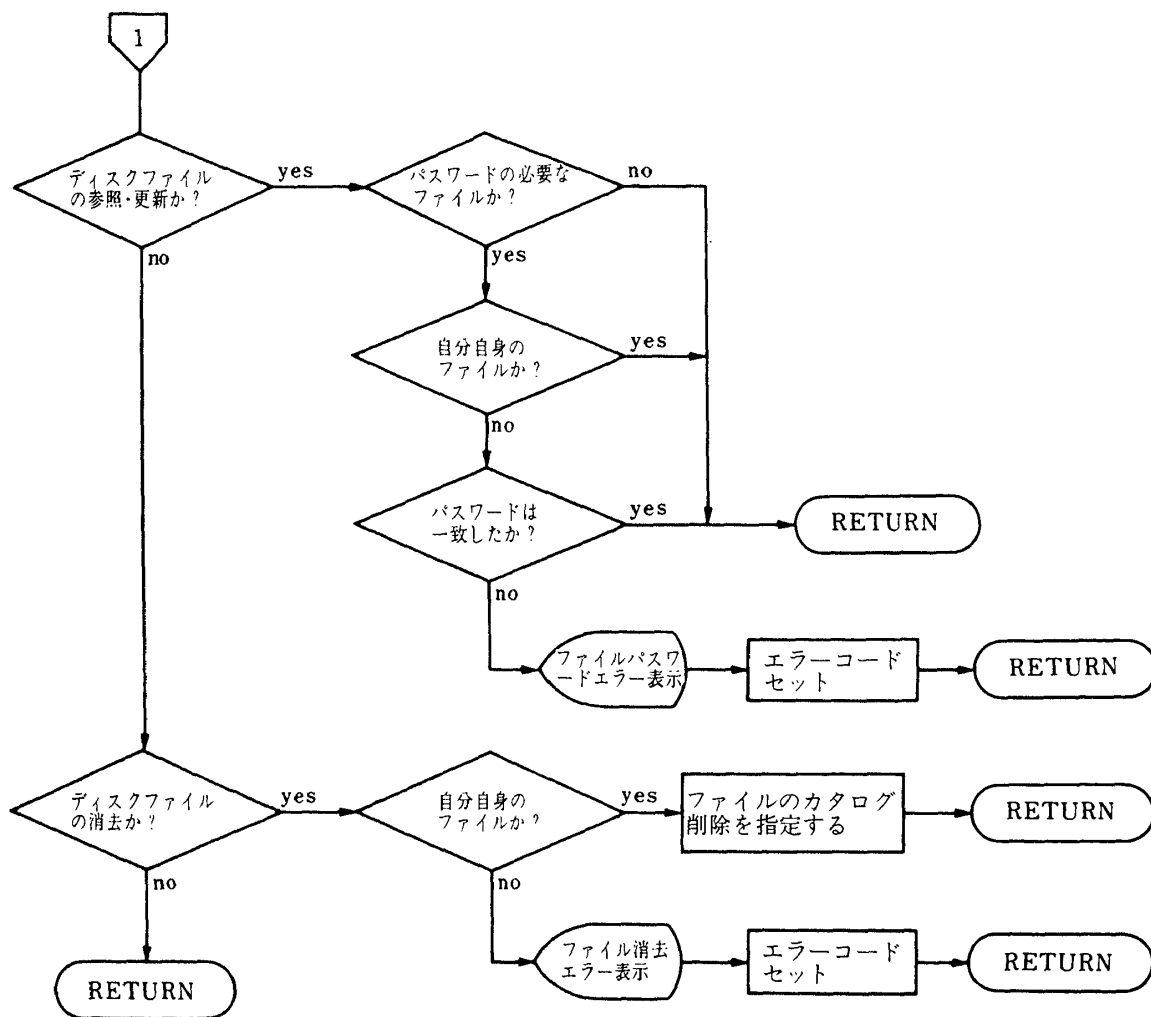


図 3-5 つづき

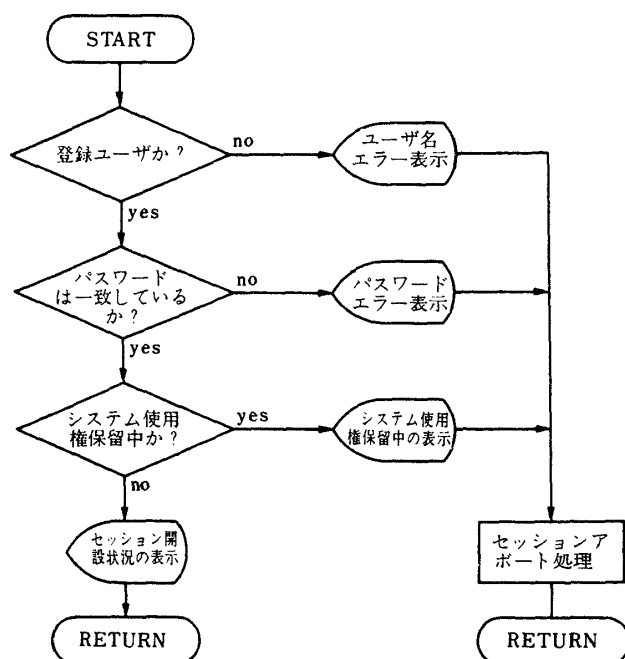
3-6 参照)。

- ユーザ名の確認
- パスワードの確認
- システム使用権保留ユーザに対して、その回復をうながす旨の表示
- セッション開設状況の表示
- エラー内容の表示
- エラーのあったセッションのアポート処理

(7) TSS コマンド投入時の処理

TSS コマンド投入直後に、次の処理を行う(図 3-7 参照)。

- コマンド名の確認
- バッチジョブと同一のファイル管理のための処置
- 会話型リモートバッチジョブ投入用コマンドに対する処置
- 会話型リモートバッチジョブ操作作用コマンドに



(注) システム使用権回復の手続きは、別の方法で行う

図 3-6 TSS セッション開設時のセンタルーチンの処理

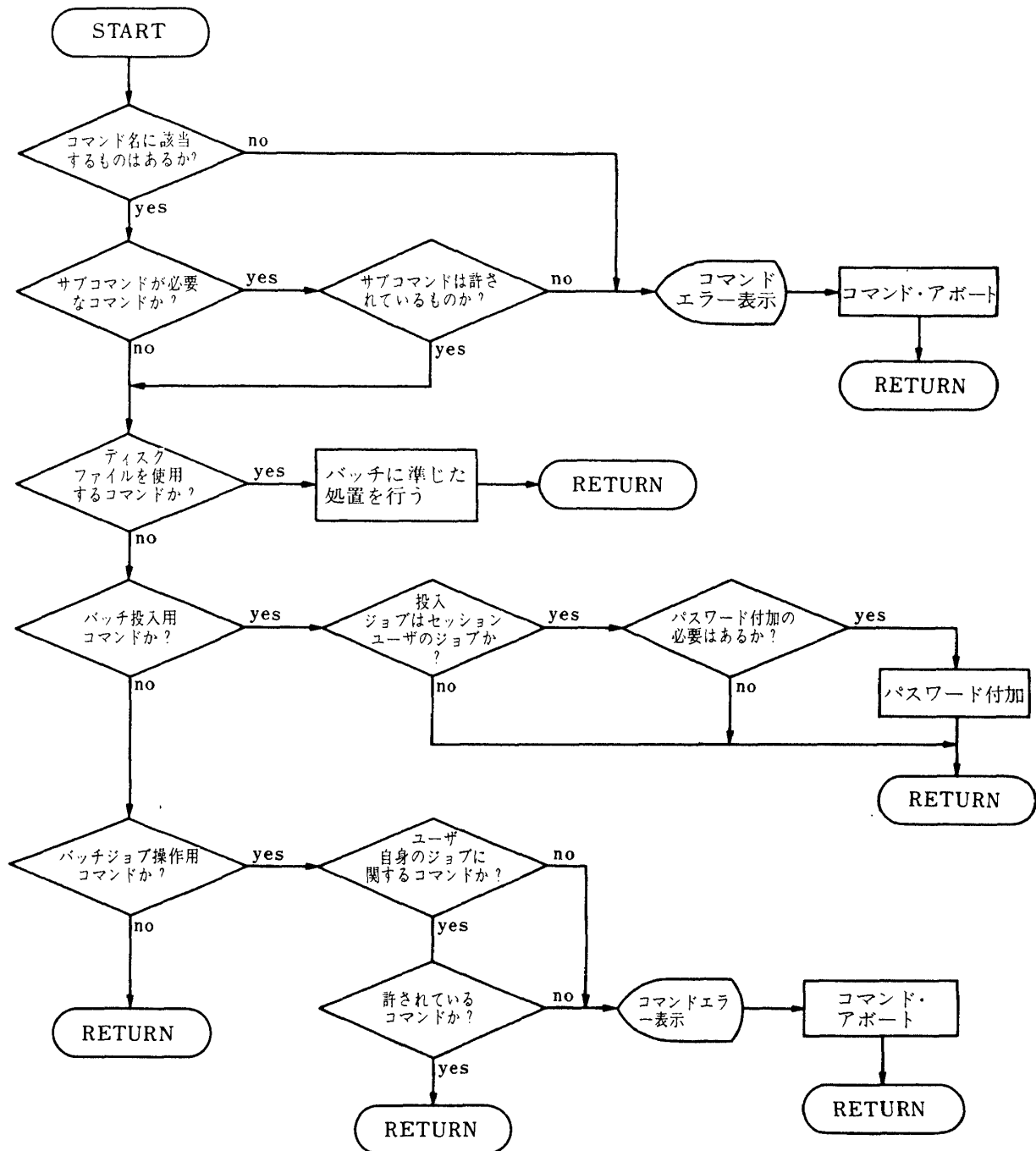


図 3-7 TSS コマンド投入時のセンタルーチンの処理

対する処置

- エラー内容の表示
- コマンドアボート処置

(8) TSSセッション閉設時の処理

TSSセッション閉設時に、次の処理を行う（図 3-8 参照）。

- 会計情報の表示

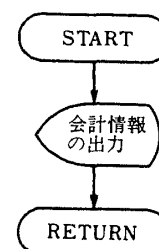


図 3-8 TSSセッション閉設時のセンタルーチンの処理

3.3 システム編集について⁷⁾

システムを実際に運用する場合には、システムの規模、運用形態およびハードウェア機器構成などに応じて、あらかじめシステム編集をしておかなければならない。しかし、システム変更に対する柔軟性をもたせるために、システム編集の後でも一部だけを再編集したり、システム開始時にシステムパラメータ値の変更を行える機能が用意されている。また、運用形態が複数あるシステムではその都度システムを変更しなくても良いように、複数のシステム編集が同時に登録できる。

システム編集の機能は、システムファイルの定義に関するもの、入出力機器構成に関するもの、運用に関するもの、その他に大別できる。本節では、これらの各項目について代表的なパラメータの説明を行うが、その設定値は、現時点での運用方針に基づいて定めたものであるため、ハードウェア機器構成の変更やユーザジョブの属性の変化、運用方針の変更等に応じて、変更される。

3.3.1 システムファイルの定義

システムファイルとは、システムを運用する上で制御プログラム自体が必要とするファイルをいう。本項では、システム編集時に定義すべきシステムファイルについて述べる。

まず、新システムで必要となるシステムファイルの種類および所要量を表3-3に示す。これらのファイルは、運用方針やユーザジョブの属性にあまり関係なく定まるものと、それなしには決定できないものがある。したがって、前者は業者提供値を参考にすることにより、過不足のきわめて少ない値を定めることができる。逆に、後者は、各センタ毎に条件が異なるため、それを考えずに適切な値を決めることはむずかしく、適切さを欠くと、システム運用の混乱を招くことにもなりかねない。そこで、それらのファイルについてはその根拠も合せて記した。

3.3.2 入出力機器構成に関する定義

入出力機器構成の定義には、そのシステムの入出力機器のタイプや接続方法を定義するものと、装置名や機器名を定義するものがあるが、本節では、ジョブ制御文やコンソール命令等で特にユーザと関連の深い機種名と装置名について述べる。

新システムでは、装置を効果的に使用するために表3-4に示す装置名と機種名を設けた。装置名は各装置と1対1に対応しており、同種の装置をグループ化したものに機種名をつけた。機種名は最大3段階までつけることができる。

ユーザは装置を使用する際、目的に応じて装置名または機種名が指定できる。たとえば、ある装置に対して強制終了や停止を行う場合には装置名を指定しなければならないが、9トラックのMTを使用する場合や、ユーザ用ボリュームにファイルを作成する場合などは、装置を指定しないで、“MT1”または“DK2”という様に機種名を指定すれば、システムはその時点での使用状況に応じて、その中の適当な装置を割当てするため、装置の効果的な使用が図れる。

3.3.3 運用に関する定義

運用に関するものの定義の中には、3.1.3項で述べたジョブ運用方式を実現するためのパラメータがある。本節では、それらのパラメータについて“パラメータ名(設定値)……内容および説明”の形式で述べる。

(1) システム運用パラメータの定義

- ユーザジョブ空間数(25)……システムで実行可能なユーザジョブの空間数。ここでの指定の範囲内ならばシステム運用時にジョブ多重度を変更できるため、大きめの値を設定した。
- ジョブ多重度(18)……同時に実行可能なユーザジョブの個数。ここでは表3-1のジョブクラス別起動多重度に基づいて定めた。
- システム実行多重度(8)……多重度制御の配下で実行されるユーザジョブのうち、同時にローラインされて実行可能な個数。
- TSS最大ユーザ数(10)……TSSサービスを同時に受けうる最大ユーザ数。当センタでTSS端末として使用できる装置の数とした。

(2) パフォーマンスグループ(PG)の定義

- PG識別記号(2~9)……ジョブクラス毎に定義した。
- PG最大多重度(0~10)……PG毎に、そのPGに属する空間のうちローライン状態で実行できる空間の最大数。表3-1のジョブクラ

表 3-3 システムファイルの種類および所要量

<システムボリューム>

システムファイル	ファイル名	所要量	備 考
NIP外部ページファイル	SYS.NIPEPS	*1.2 ^{MB}	(NIPの格納ファイル)
パラメータライブラリ	SYS.システム名 PARMLIB	*2.6	(システムパラメータの登録ファイル)
リンクパックライブラリ	SYS.システム名 LPALIB	*7.5	(制御プログラムの格納ファイル)
リンクライブラリ	SYS.システム名 LINKLIB	*4.0	(処理プログラムの格納ファイル)全部対象
外部仮パラメータライブラリ	SYS.システム名 EPARMLIB	*0.2	(外部仮パラメータの登録ファイル)
システムプロシジャライブラリ	SYS.システム名 PROCLIB	*0.3	(システムジョブの起動用ジョブ制御文の格納ファイル)
システムジョブ制御マクロライブラリ	SYS.システム名 JCMLIB	*1.0	(ジョブ制御マクロの登録ファイル)
イメージライブラリ	SYS.システム名 IMAGELIB	*0.2	(変換テーブル, 紙送り制御コード, キャラクタセットコードの登録ファイル)
システムダンプファイル	SYS.システム名 SYSDUMP	*5.0	(主記憶ダンプ等の格納ファイル)
システム外部ページファイル	SYS.システム名 LPAEPS	*3.5	(システム共通空間用の外部ページファイル)
ジョブ外部ページファイル1	SYS.システム名 JBEPS1	32.0	(ジョブ専用共通空間用の外部ページファイル)
” 2~15	SYS.システム名 JBEPS2~15	32.0×1	(同 上)
ハードエラーロギングファイル	SYS.システム名 LOGREC	0.6	0.3MB/週
システム管理ロギングファイル	SYS.システム名 SMF X, Y	15×2	5KB×3,000ジョブ/月(3ジョブステップ/ジョブであると約2.5MB程度)
ジョブスタックファイル	SYS.システム名 JSF	0.1	10枚×100件(ジョブ制御文の格納ファイル)
カタログ登録簿	SYS.システム名 CATALOG	*0.2	(カタログ情報の登録ファイル)
会計コードファイル	SYS.システム名 ACRFILE	*0.2	(ユーザ資格チェック用データの格納ファイル)
システムメッセージロギングファイル1~3	SYS.システム名 MLOG1~3	1.0	0.2MB/日(システムメッセージファイルはディスプレイメッセージを含むが, メインコンソールは通常使用しないため)
ディスプレイメッセージロギングファイル	SYS.システム名 装置名, MLOG1~2	0.2	0.2MB/日
コマンドマクロライブラリ	SYS.システム名 CMDLIB	*0.2	(操作指令マクロの格納ファイル)
オペレータメールボックスファイル	SYS.システム名 MBOX	*0.2	(オペレータ用のメッセージを格納するファイル)
システムマクロライブラリ	SYS.システム名 MACLIB	*3.6	(スーパーバイザサービスマクロ, データ管理マクロ等の格納ファイル)
HMPライブラリ	SYS.システム名 OLTFLIB	*0.9	(ハードウェア保守プログラムの格納ファイル)

所要量に*印のあるものはメーカー提供値を参考にした。

表 3-4 機種名および装置名

機 器 名	機 番	台数	機 種 名		装 置 名	備 考
カード読取装置	F 671 D	1	CR	—	C00	
ラインプリンタ装置	F 654 C	1	LP	—	L00	
フロッピー ディスク装置	F 442 A	1	FP	—	F00	
X Yプロッタ	F 6202 B	1	XY	—	X00	
紙テープ読取装置	F 749 G	1	PT	—	P00	
グラフィック ディスプレイ装置	F 9431 A	1	GD	—	G00	
磁気テープ装置	F 610 A ₃	1	MT	MT0	M00	7トラック
	F 610 A ₂	2	"	MT1	M11	9トラック
			"	"	M12	"
磁気ディスク装置	F 6411 A	5	DK	DK0	D00, K00	システムボリューム
			"	"	D01, K01	"
			"	DK1	D12, K12	ワークボリューム
			"	DK2	D23, K23	ユーザボリューム
			"	"	D24, K24	"
	F 6417 A	1	"	DK3	D35	オープン用ボリューム
C R T 端末	F 6262 R	6	CRT0	—	T00	オープンコンソール
			CRT1	—	T11	TSS 端末
			"	—	T12	"
			"	—	T13	"
			"	—	T14	"
			"	—	T15	"
	F 9526 R	3	CRT2	—	T26	"
			"	—	T27	"
			"	—	T28	"

注) •磁気テープ装置, 磁気ディスク装置の機種名は2種類あるが, 目的に応じて使い分けられる。
 •磁気ディスクボリュームは, F6411A用は135MB/台, F6417A用は67MB/台のため, 管理上, F6411Aは2台として取扱う。

ス別起動多重度に基づいて定めた。

- PG最小多重度 (0 ~ 3) ……システムの負荷状態が高まり、空間をロールアウトする必要性が高まった場合でもロールアウトされずに実行が保証される PG 毎の空間数。表 3-1 のジョブクラス別実行多重度に基づいて定めた。

(3) システム運用資源の定義

- WORK1 ボリューム群 (D00, D01, D12, K00, K01, K12) ……ディスクのうちこのグループに属するボリューム通番の指定。当システムでは、ワークボリュームを WORK1 ボリューム群とした。したがって、ワークボリューム使用の際はこのボリューム群を指定すれば、システムが使用状況に応じて、適当なボリュームを決定する。
- WORK2 ボリューム群 (D23, D24, K23, K24) ……当システムではユーザボリュームを WORK2 ボリューム群とした。したがってユーザボリュームを使用する場合には、このボリューム群を指定する。
- RVOL ボリューム群 (D00, D01, K00, K01) ……このボリューム群は常駐ボリュームとする必要のあるものに限る。そのため、当センタではシステムボリュームをこのボリューム群とした。

(4) ダイナミックアロケーションの定義 (ディスクに新規ファイルを作成する場合のパラメタの省略値の定義)

- ボリューム (WORK2) ……ファイルを作成すべきボリューム通番またはボリューム群を指定する。したがって、ファイル作成時にボリューム指定を省略すると、ユーザボリュームへの作成とみなされる。
- スペース単位 (トラック) ……ファイル作成時のスペース単位で、平均ブロック長単位 (この場合は同時にブロック長も指定する)、トラック単位、シリンダ単位のいずれかを指定する。
- スペース量 (10, 10) ……スペース割当量の初

期値と増分値を指定する。

- ディレクトリ領域トラック数(1) ……区分編成ファイルの場合のディレクトリ領域の大きさをトラック数で指定する。

4. FORTRAN について

2 章および 3 章において述べたように、ハードウェアシステム、ソフトウェアシステムおよびその計算機の運用に関しては、効率の良いシステムの作成および運用方式が考慮されているが、システムの有効利用のためにはこれらだけでは十分でなく、ユーザプログラムが効率的に作成されていることが望まれる。効率の良いプログラムを作成するためには、FORTRAN コンパイラの特性を知る必要があるので、それについてベンチマークジョブの処理結果にふれながら述べることにする。

4.1 FORTRAN77 について

科学技術計算におけるプログラム言語は FORTRAN 言語であると言っても過言ではない。新システムにおいても、主としてプログラム作成は FORTRAN を用いて行なわれると考えられるので、FORTRAN プログラムの使い易さ、オブジェクトプログラムの効率および他機種との互換性の高さが望まれる。新システムにおいて使用される FORTRAN 77 は ANS FORTRAN77^{注1)}に準じたものであり、その上に各種の拡張機能が備わっている。従って、使い易さおよび互換性は最高のものであると思われる。また、TOSBAC-FORTRAN と OSW-FORTRAN77 の機能の詳細な相違に関しては、後日それに関する資料が発行されることになっている。

4.2 FORTRAN77-OPT0 と OPT2 の比較

特にベンチマークジョブ処理の結果を通してユーザの作成したソースプログラムは FORTRAN ジョブステップにおいてコンパイルが実行されるがコンパイルされたオブジェクトプログラムの実行ステップ数はできるだけ少ないことが望ましい。実行ステップ数が少なく、効率化されたオブジェクトプ

注1) 米国規格協会 (ANSI) において制定された FORTRAN の新しい言語規格のことである。

ログラムを生成するコンパイラの機能を最適化という。コンパイラの最適化の機能には以下のものがある（詳細は、文献 9 を参照）。

(1) 局所的な最適化

- (イ) 組込み関数および文関数の内部展開－呼び出しオーバーヘッドの減少－
- (ロ) IF 文の最適化
- (ハ) 入出力文の最適化－入出力ライブラリ呼び出し回数の減少－
- (ニ) 単項減算，2 倍演算および 2 分演算等の高速化等。

(2) テキストの最適化

- (イ) 共通式の削除，ループ内の不変式のループ外への移動
- (ロ) 誘導変数の最適化
- (ハ) 単純代入の削除等。

(3) レジスタ割り当ておよび分岐の最適化

FORTRAN コンパイラの最適化機能には 3 つのレベルがあり，そのレベルに応じてそれぞれ OPT0，OPT1 および OPT2 と呼んでいる。OPT0 は局所的な最適化の一部のみを行ない，OPT1 は局所的な最適化，テキストの最適化およびレジスタ割り当ての最適化のそれぞれ一部のみを行ない，OPT2 はすべての最適化を行なうことになっている。これらの最適化機能が FORTRAN プログラムの実行においてどれ程有効であるかを知るために，計算機システムの選定に使用されたベンチマークジョブについて，これを分析することにする。

表 4-1 は，ベンチマークジョブの FORTRAN77-OPT0 および FORTRAN77-OPT2 におけるジョブ処理結果を一覧にしたものである。上段にジョブの各ジョブステップの実行時間を示し，下段に OPT0 処理の各ジョブステップの実行時間を 1 とした場合の OPT2 処理における計算速度の向上比率を示した。

FORTRAN ジョブステップにおけるコンパイル時間を OPT0 と OPT2 とで比較すると，どのジョブにおいても OPT2 の場合の方がコンパイル時間が大きく，OPT0 の場合の平均約 2 倍程かかっている。こ

れは，最適化のレベルが高いのでそれだけソースプログラムの翻訳に時間がかかるためである。このように FORTRAN ジョブステップにおいてはコンパイル時間がかかるが，ユーザプログラムが走行する RUN ジョブステップにおいては，その実行時間を小さくすることができる。OPT0 の場合と OPT2 の場合の RUN ジョブステップの実行時間を比較すると OPT2 の場合の実行時間は OPT0 の場合の実行時間の 1.3 倍～4.1 倍，平均 2.3 倍の処理速度の向上となっている。この結果から最適化はユーザプログラムの処理効率の向上に絶大な効果のあることがわかる。

以下では，最適化がどのようなプログラムに対して特に有効になるのかを FORTRAN のステートメントレベルで示すために，ベンチマークジョブのうち，OPT2 による処理が OPT0 による処理の約 3 倍以上速くなっている KKT2，KKT8，KKT12，KKT13 および KKT17 について考察することにする。

(a) KKT2

	OPT0 処理	OPT2 処理
FORTRAN ジョブステップにおけるコンパイル時間	0.839 秒	1.691 秒
RUN ジョブステップにおける実行時間	108.374 秒	31.740 秒

RUN ジョブステップにおける実行時間の差は約 76 秒であり，その比は OPT0 : OPT2 = 1 : 0.3 である。このように差の出る理由は以下の通りである。このプログラムはメインプログラムのみから成り，その実行ステートメントは 75 ステートメントである。各ステートメントの実行回数の和 N は 13,898,969 回

$$N = \sum_{i=1}^L n_i$$

L : ステートメント数

n_i : ステートメント i の実行回数

であったが，そのうち約 97 % を占める 13,527,360 回の実行回数は下記のステートメントで占められている。

```
DO 40 J = 1, 19
  IF(M(J).EQ.1) GO TO 43
```

表 4-1 FORTRAN77 OPT0 および OPT2 処理時のベンチマークジョブ処理結果

No.	ジョブ名	実行時間 (OPT0)				実行時間 (OPT2)			
		FORTTRAN	LIED	RUN	TOTAL	FORTTRAN	LIED	RUN	TOTAL
1	KKT 1	6.288 ^秒 1	0.788 ^秒 1	106.753 ^秒 1	113.829 ^秒 1	13.359 ^秒 0.5	0.680 ^秒 1.2	47.372 ^秒 2.3	61.411 ^秒 1.9
2	KKT 2	0.839	0.536	108.374	109.749	1.691 0.5	0.558 1.0	31.740 3.4	33.989 3.2
3	KKT 4	2.872	0.601	133.626	137.099	6.727 0.4	0.562 1.1	73.927 1.8	81.216 1.7
4	KKT 6	3.400	0.478	3.527	7.405	9.044 0.4	0.507 0.9	2.670 1.3	12.221 0.6
5	KKT 7	3.342	0.810	27.557	31.709	6.628 0.5	0.813 1.0	21.277 1.3	28.718 1.1
6	KKT 8	0.906	0.481	155.028	156.415	1.700 0.5	0.484 1.0	53.614 2.9	55.798 2.8
7	KKT 9	6.161	1.128	29.066	36.355	12.990 0.5	1.043 1.1	20.220 1.4	34.253 1.1
8	KKT 10	3.330	0.841	363.913	368.084	6.963 0.5	0.838 1.0	232.706 1.6	240.507 1.5
9	KKT 11	0.309	0.337	25.892	26.538	0.547 0.6	0.343 1.0	16.548 1.6	17.438 1.5
10	KKT 12	0.853	0.358	132.321	133.532	2.158 0.4	0.348 1.0	32.058 4.1	34.564 3.9
11	KKT 13	12.605	1.300	425.187	439.092	27.714 0.5	1.115 1.2	148.863 2.9	177.692 2.5
12	KKT 15	8.790	1.126	23.857	33.773	20.693 0.4	1.042 1.1	10.619 2.2	32.354 1.0
13	KKT 16	1.025	0.483	193.325	194.833	2.377 0.4	0.480 1.0	144.961 1.3	147.818 1.3
14	KKT 17	5.974	0.797	1165.522	1172.293	15.708 0.4	0.714 1.1	307.304 3.8	323.726 3.6
15	KKT 19	6.210	1.479	262.792	270.481	13.227 0.5	1.443 1.0	160.900 1.6	175.570 1.5
16	KKT 20	5.732	0.662	3093.549	3099.943	14.492 0.4	0.587 1.1	1424.781 2.2	1439.860 2.2
17	KKT 21	4.310	0.979	685.021	690.310	11.899 0.4	0.882 1.1	292.352 2.3	305.133 2.3
合 計		72.946	13.184	6935.310	7021.440	167.917 0.4	12.439 1.1	3021.912 2.3	3202.268 2.2
平均値		4.291	0.776	407.959	413.026	9.877 0.4	0.732 1.1	177.760 2.3	188.369 2.2

```

      IF(M(J). EQ. 2) GO TO 44
      IF(Y(J). LT. X(J)) GO TO 41
      IF(Y(J). GE. T. AND. X(J). LE.
1  T) GO TO 42
      Z(J) = 1H
      GO TO 45
41  IF(T. GE. X(J). OR. T. LE. Y(J))
1  GO TO 42
      Z(J) = 1H
      GO TO 45
42  Z(J) = 1H*
      GO TO 45
43  Z(J) = 1H*
      GO TO 45
44  Z(J) = 1H
45  IF(Z(J). EQ. 1H*) N = N + 1
40  CONTINUE

```

上記のステートメントを OPT2 の最適化を行なった後実行すると、先の(1)～(4)の IF 文の最適化を受けて、実行時間の短縮化をはかることができた。このプログラムコーディングは拙劣であるが、このような場合においても FORTRAN の最適化により実行時間を節約することができる。

(b) KKT8

	OPT0 処理	OPT2 処理
FORTRAN ジョブ ステップにおける コンパイル時間	0.906 秒	1.700 秒
RUN ジョブステップ における実行時間	155.028 秒	53.614 秒

RUN ジョブステップにおける実行時間の差は約 102 秒であり、その比は OPT0 : OPT2 = 1 : 0.34 である。このプログラムはメインプログラムおよびサブプログラム合わせて 81 ステートメントから成り、その実行回数の和 N は 19,838,043 回である。そのうち約 93 % を占める 18,456,504 回の実行回数は次のステートメントに費されている。

```

      DO 10 N = 1, 61
      I = I + 1
      IF(OF. EQ. X(N)) GO TO 30
      IF(OF. LE. X(N)) GO TO 20

```

10 CONTINUE

これも、KKT2 の場合と同様に IF 文の最適化を受けて、OPT2 の実行において実行時間の短縮化をはかることができる例である。

(c) KKT12

	OPT0 処理	OPT2 処理
FORTRAN ジョブ ステップにおける コンパイル時間	0.853 秒	2.158 秒
RUN ジョブステップ における実行時間	132.321 秒	32.058 秒

RUN ジョブステップにおける実行時間の差は約 100 秒であり、その比は OPT0 : OPT2 = 1 : 0.24 である。このプログラムはメインプログラムおよびサブプログラム合わせて 87 ステートメントから成り、各ステートメントの実行回数の和 N は 7,134,886 回である。そのうち約 94 % を占める 6,705,000 回の実行回数は下記のステートメントで占められている。

```
DO 90 J = 1, N
```

```
A(I, J) = A(I, J) + A(K, J) * T
```

90 CONTINUE

これらのステートメントにおける OPT0 と OPT2 の最適化の相違は、OPT0 の場合には、I, J および K はその都度インデックスレジスタにロードされるが、OPT2 の場合にはそのようなことはなく、先に示した(3)のレジスタ割り当ての最適化を受けてその実行時間が短縮されている。

参考のために、本所 FACOM230-75 システムにおいてこのジョブを処理した際の上記のステートメントのオブジェクトプログラムのステップの処理に要する時間は、OPT0 を用いた場合よりも OPT2 を用いた場合の方が速く、その倍率は 3.86 倍にもなった。

(d) KKT13

	OPT0 処理	OPT2 処理
FORTRAN ジョブ ステップにおける コンパイル時間	12.605 秒	27.714 秒
RUN ジョブステップ におけるコンパイル時間	425.187 秒	148.863 秒

RUN ジョブステップにおける実行時間の差は約 278 秒であり、その比は OPT0 : OPT2 = 1 : 0.34 である。このプログラムの実行ステートメントは、メインプログラムおよびサブプログラム合わせて 729 ステートメントであり、各ステートメントの実行回数の和は 17,539,550 回である。上記の(a)~(c)のジョブと異なり、実行ステートメントの総実行回数に占める割合が圧倒的に大きいステートメントはない。従って、OPT0 と OPT2 の実行時間の差が大きい理由は、以下の部分の最適化が全体的に効を奏し、実行時間の短縮が行なわれたものと思われる。

(i) $AN(I, J) = AMAX\ 1(ABS(\underline{0.5 * CN}),$
 $DN) - \underline{0.5 * CN}$

下線部分の計算は、OPT0 ではその都度計算されるので、このステートメントを実行する場合 2 度計算されるが、OPT2 実行時には、1 度計算されるだけで、2 度目の下線部分は、前回計算した値が用いられる。

(ii) DO 100 I = 2, NIM1
 :
 CAMN = 0.5 * (VIS(I, J)
 + VIS(I, J - 1)) / PRED ①
 :
 100 CONTINUE

↓
 $A = 1. / PRED$

DO 100 I = 2, NIM1
 CAMN = 0.5 * (VIS(I, J)
 : + VIS(I, J - 1)) * A
 :
 100 CONTINUE

OPT0 では PRED でその都度除するが、OPT2 で処理した場合には、その部分を DO ループの外に出し、しかも A という変数を新たに作り、除算を乗算にしている。乗算と除算とでは乗算の方が演算速度は速い(表 2-3 演算速度参照)。

ここで、上記プログラムの①の部分

$CAMN = 0.5 / PRED * (VIS(I, J)$
 $+ VIS(I, J - 1))$

とコーディングすることにより、0.5 / PRED が DO の外に出されることを注意しておく。①の場

合は 0.5 と PRED が離れているためにコンパイラが認識できず、PRED だけ DO の外に出される。

(iv) $C(\underline{J}) = AE(\underline{I, J}) * PHI(\underline{I+1, J})$
 $+ AW(\underline{I, J}) * PHI(\underline{I-1, J})$
 $+ SU(\underline{I, J})$

OPT0 では下線部分および 2 重下線部分の 6 個のアドレス計算を行なうが、OPT2 では 2 重下線部分の 4 個の計算で済む。他に、(I+1, J) および (I-1, J) のアドレス計算においては、OPT2 の場合には I に対する加算および減算を行なって I+1 および I-1 を表現するので、参照する必要がないことが原因となっている。

ここにあげた例(i), (ii), (iv)のうち、(i)と(ii)はユーザがコーディングに気をつければ防げるものであるが、もしこう書いたとしても OPT2 で処理すれば、FORTRAN の最適化機能によりその実行時間を短くすることができる。

(e) KKT17

	OPT0 処理	OPT2 処理
FORTRAN ジョブ ステップにおける コンパイル時間	5.974 秒	15.708 秒
RUN ジョブステッ プにおける実行時 間	1165.522 秒	307.304 秒

RUN ジョブステップにおける実行時間の差は約 858 秒であり、その比は OPT0 : OPT2 = 1 : 0.26 である。このように差の出る理由は、前記の KKT12 の場合と同様に 2 次元配列データのインデックスレジスタの使用法にその原因がある。

逆に、最適化があまり有効でないジョブとしては、KKT6, KKT7, KKT9 および KKT16 があげられる。これらのジョブの共通点は、全ステートメントの総実行回数に占める割合の大きいステートメントのある場合においても、その割合が平均している場合においても最適化のきかないステートメントが大部分を占めること、すなわちそのステートメントの部分をオブジェクトプログラムにコンパイルした場合、OPT0 と OPT2 とではそのオブジェクトプログラムのステップ数の長さに変化がみられないことである。

以上の例より最適化の有効なジョブのプログラム属性と有効性の程度がどのようなものであるかを理解できるであろう。このように既にデバッグ済みのプログラムに関しては最適化を行なうことは有効である。しかし、デバッグ中のジョブには必ずしも最適化が有効であるとは言えない。それは、デバッグ中のジョブには RUN ジョブステップまで実行されずに途中のジョブステップで終了しているジョブがかなりあり、この場合には最適化を行なわない方がジョブの実行時間が短いからである。

最後にジョブの実行時間について OPT0 の場合と OPT2 の場合とで比較を行なうと、全体的には速度が上がっているが、KKT6 は OPT0 よりもむしろ OPT2 の場合の方が実行時間が大きい。また、KKT 7, KKT9 および KKT15 は、OPT0 と OPT2 とではその実行時間にほとんど差がない。KKT6 および KKT15 のジョブについてのその理由はジョブの規模が小さいために RUN ジョブステップの実行時間は OPT2 の方が小さいが、OPT2 の FORTRAN ジョブステップのコンパイル時間が長いので、その合計時間であるジョブの実行時間にそれが影響していることに起因するものである。KKT7 および KKT9 のジョブについてのその理由は、先に最適化の効果のないジョブに関する共通点を述べたが、それと同様の理由によるものである。

表 4-2 は単精度と倍精度の比較表である。RUN

ジョブステップにおける実行時間を比較すると、倍精度のジョブは単精度のジョブの約 1.2 倍～1.8 倍かかっている。表 2-3 の演算速度比較表をみると、浮動小数点の加算においては単精度と倍精度とではあまり変わらないが、乗算の場合は約 2 倍、除算の場合は約 4 倍、倍精度の方が時間がかかる。ジョブ毎に演算の形やその実行回数が異なるので、倍精度の場合は単精度の場合の何倍余計に時間がかかるとは言えないが、表 4-2 の結果はおよその目安となるであろう。

4.3 SSOPTRAN の有効性について

前節において、FORTRAN コンパイラの最適化機能がユーザプログラムの実行時間の短縮化に非常に有効であることが明らかになった。FORTRAN コンパイラの行ないうるオブジェクトプログラムの最適化は現時点においてはほぼ限界に達していると考えられるが、ソースプログラムレベルで最適化を行なうという立場に立つならば、FORTRAN コンパイラの最適化機能とは別の最適化機能を考えることができる。FACOM M シリーズのシステムにおいて、これを実現しているのが SSOPTRAN である。SSOPTRAN はユーザプログラムの処理において、FORTRAN ジョブステップの前に置かれ、ソースプログラムからソースプログラムへの変換を行なうことによりプログラムの最適化をはかるトランスレー

表 4-2 単精度と倍精度の比較

No.	ジョブ名	RUNジョブステップの実行時間 (FORTRAN77・OPT2 処理)		備 考
		単 精 度	倍 精 度	
1	KKT 6	2.670 秒	3.514 秒 1.32 倍	
2	KKT 7	21.277	37.895 1.78	
3	KKT10	232.706	375.447 1.61	
4	KKT15	10.619	12.481 1.18	
合 計		267.272	429.337 1.61	

タである。

以下では、SSOPTRAN の最適化機能とその効果を例をもって示す。

(1) DO ループ制御変数の入れ換え

これは記憶空間の階層構造を考慮した最適化である。新システムにおいて仮想記憶空間は 4MB の高速の主記憶および低速の磁気ディスクによりサポートされており、2 章のハードウェア構成において示したように、主記憶は 4KB (= 単精度 1KW) を単位とするページに分割されて管理されている。従って、プログラム上連続した領域も主記憶上連続番地に存在するわけではないばかりか、主記憶上に存在せず、磁気ディスク上に分割されて存在することもしばしば起こる。データの存在する場所が異なれば、アドレス変換のための情報が TLB 上に存在しない可能性があり、その場合アドレス変換にかかるオーバーヘッドが生じることになる。データの存在するページが磁気ディスク上にしか存在しない場合にはその読み込みのオーバーヘッドは非常に大きい。従って、プログラムまたはデータの読み出し、書き込みはできる限り同一ページから連続して行なわれることが望ましく、DO ループ制御変数の入れ換えはこの点に着目した最適化機能である。以下、簡単な例により説明する。

```
DIMENSION A( 500, 500 ), B( 500,
      500 ), C( 500, 500 )
DO 1 I=1, 500
DO 1 J=1, 500
A( I, J )=B( I, J )+C( I, J )
1 CONTINUE
```

と書かれた FORTRAN ステートメント(A)は SSOPTRAN により、

```
DIMENSION A( 500, 500 ), B( 500,
      500 ), C( 500, 500 )
DO 1 J=1, 500
DO 1 I=1, 500
A( I, J )=B( I, J )+C( I, J )
1 CONTINUE
```

と変換される。その結果(A)では、A(I, J), B(I, J) および C(I, J) はそれぞれ 500 番地おきにアクセスされ、同一ページからの連続したアクセスは

それぞれ高々 3 回しか起こらなかったものが、(B)ではほとんど連続して同一ページからアクセスされることになる。

これは実記憶量よりも大きな仮想記憶量が要求されているような状況、たとえば多重ジョブからの総仮想記憶要求量が実記憶量を上回る場合、あるいは単独ジョブでも実記憶量を上回る仮想記憶を使用する大記憶ジョブが実行されるような場合において、磁気ディスクからのデータの出し入れの時間およびシステムのオーバーヘッドが増大し、それがジョブの経過時間の大巾な増大となってユーザにはねかえることを防止する最適化機能である。

実際に、要求仮想記憶量 / 実記憶量の比が大きくなると(A)を処理するための経過時間は(B)を処理するための経過時間の何十倍にも達することが報告されている。¹⁰⁾

(2) ユーザサブプログラムの内部展開

これはユーザの作成したサブプログラムをその呼ばれた場所に展開することによる最適化機能である。これにより、サブプログラム呼び出しのオーバーヘッドを削減することができる。また、内部展開することにより、FORTRAN コンパイラの最適化機能がメインプログラムとサブプログラムを合わせて同時に働くことになり、DO ループの内側でサブプログラムが呼ばれているような場合には、この効果は絶大である。KKT16 においては

```
メインプログラム
      :
DO 14 M=1, 49
IF( I-M ) 15, 16, 15
16 S=S
GO TO 14
15 U=UI( M )
RAM=0.4
S=S+( RAM*UXI( M )
      +( 1.-RAM ) *UXSNC ) *
      G( I, M, DX, M8, U ) *UI( M )
14 CONTINUE
      :
サブプログラム
      FUNCTION G( I, M, DX, M8, U )
```

```

REAL M8
P=3.141592
X=I * DX
XS=M * DX
A=X-XS
B=-2.4 * M8 ** 2 / SQRT
  ((1.-M8 * M8) ** 3) *
  (-.4) * .5 / U
D=-2. * A / (1.+(A * B) ** 2)
  / (1.+(A * B) ** 2)
G=D * (.5 * P / ABS(A) *
  (1.-(A * B) ** 2)
  +B * LOG((A * B) ** 2)
  +B * (1.+(A * B) ** 2))
RETURN
END

```

の部分が全実行回数の約 99 %を占める。FUNCTION サブプログラム G(I, M, DX, M8, U)をメインプログラムの G の位置に展開することにより、FUNCTION G 内の定数計算、共通式をメインプログラムの DO の外に出すという FORTRAN コンパイラの最適化機能が働き、KKT16 の実行時間は、SSOPTRAN を通した場合には、これを通さない場合の約 2 倍の処理効率をあげることができた。

(3) (1)および(2)の最適化機能の他に SSOPTRAN には、(1)および(2)ほど効果があるとは思えないが、IF 文、DO 文の実行回数に関する最適化機能および演算順序の変更による不変式のループ外への移行による最適化機能等がある（詳細は文献 11 参照）。

表 4-3 はベンチマークジョブに SSOPTRAN を適用した結果を示したものである。表 4-1 と同様に上段に各ジョブステップの実行時間、下段に SSOPTRAN を通さない場合の各ジョブステップの実行時間を 1 とした場合の処理効率の向上比率を示した。KKT16 を除く他のジョブは、SSOPTRAN の最適化機能を発揮するプログラムがなかったため、RUN ジョブステップの実行時間短縮の効果は殆んどないか、あったとしても 1 割程度であり、SSOPTRAN の実行時間が加わったことにより、むしろ逆効果になった例が多い結果になった。それは、SSOPTRAN の最適化機能が FORTRAN の最適化機能

に比べて普遍性がないためであり、SSOPTRAN の最適化機能の効果の有無がユーザのプログラムのコーディングの性格に大きく左右されることを意味するものである。

4.4 FORTRAN コンパイラ OPT0, OPT2

および SSOPTRAN の使用法について

本節では前節までに明らかとなった FORTRAN コンパイラ OPT0, OPT2 および SSOPTRAN の特性と効果を考慮した使用法について述べる。

OPT1 は OPT0 と OPT2 との中間的な特性をもつが、ユーザの立場からは余り魅力がないので考察の対象から除く。

表 4-4 はベンチマークジョブの FORTRAN77 OPT0 および OPT2 処理における 1 秒当りのコンパイル枚数を示したものである。表をみると、多少のばらつきはあるが OPT0 の 1 秒当りのコンパイル枚数は約 100 枚、OPT2 のそれは約 50 枚と考えてよい。今、ソースカード枚数を x 枚とすると、OPT0 および OPT2 で処理した場合にコンパイルに要する時間はそれぞれ $x/100$ 秒、 $x/50$ 秒と推定できる。一方表 4-1 に見られるように OPT2 を用いた場合の実行時間は OPT0 を用いた場合の約半分である。従って、OPT2 を用いて実行時間が s 秒かかった場合には OPT0 を用いた場合に比較して s 秒の利得があることになる。故に

$$x/50 < s$$

ならば FORTRAN のコンパイルには OPT2 を用いるべきであることがわかる。 x は平均的に 500 枚程度と考えてよいので、OPT2 を用いた方が得な s の下限は 10 秒となる。 x が 2,000 枚というのは相当大的いプログラムであるが、それでも s の値が 40 秒ならば OPT2 を用いた方が得である。

科学技術計算機システムの場合、ユーザは計算機の性能向上に対応してプログラムの規模を拡大する傾向がある。このことと、これまでの考察から、FORTRAN コンパイラの OPT0 と OPT2 のそれぞれの主記憶使用量の差が大きい場合には、FORTRAN ジョブステップのマクロの標準値を OPT2 にした方がよいという結論が得られる。但し、OPT2 では種々のデバッグツール（デバックバリュウ、

表 4-3 SSOPTRANを通した場合のベンチマークジョブ処理結果

No.	ジョブ名	実 行 時 間					備 考
		SSOPTRAN	FORTTRAN	LI ED	RUN	TOTAL	
1	KKT 1	13.675 秒	15.252 秒	0.374 秒	41.580秒	70.881秒	FORTRAN-HE (OPT2)
			0.9	1.0	1.1	0.8	
2	KKT 2	1.633	1.702	0.283	31.695	35.313	"
			1.0	1.0	1.1	1.0	
3	KKT 4	9.541	8.977	0.321	71.447	90.286	"
			0.8	0.9	1.1	0.9	
4	KKT 6	14.362	7.943	0.356	3.483	26.144	"
			1.2	1.0	1.0	0.5	
5	KKT 7	7.833	6.263	0.468	21.618	36.182	"
			1.0	1.0	1.1	0.8	
6	KKT 8	1.715	1.582	0.295	54.797	58.389	"
			1.0	1.0	1.0	1.0	
7	KKT 9	14.823	15.464	0.594	20.500	51.381	"
			0.8	1.0	1.0	0.7	
8	KKT10	7.175	6.590	0.459	242.286	256.510	"
			1.0	1.0	1.0	1.0	
9	KKT11	0.436	0.471	0.207	16.970	18.084	"
			1.0	1.0	1.0	0.9	
10	KKT12	1.695	3.512	0.213	29.721	35.141	"
			0.6	1.0	1.0	1.0	
11	KKT13	27.194	35.784	0.707	133.458	197.143	"
			0.8	1.0	1.1	0.9	
12	KKT15	22.020	19.996	0.626	12.454	55.096	"
			1.0	1.0	1.0	0.6	
13	KKT16	1.864	2.749	0.292	78.553	83.458	"
			0.8	1.0	1.9	1.8	
14	KKT17	18.074	20.137	0.398	258.741	297.350	"
			0.9	1.0	1.2	1.1	
15	KKT19	37.276	14.031	1.434	160.215	212.940	FORTRAN77 (OPT2)
			0.9	1.0	1.0	0.8	
16	KKT20	17.256	13.867	0.578	1356.609	1388.310	"
			1.0	1.0	1.1	1.0	
17	KKT21	10.133	11.865	1.012	292.165	315.175	"
			1.0	0.9	1.0	1.0	

表 4-4 FORTRAN77 OPT0 および OPT2 処理における 1 秒当りのコンパイル枚数

No.	ジョブ名	入力カード 枚数 (枚)	FORTRAN77 (OPT0)		FORTRAN77 (OPT2)	
			コンパイル時間 (秒)	1秒当りのコンパイル 枚数 (枚/秒)	コンパイル時間 (秒)	1秒当りのコンパイル 枚数 (枚/秒)
1	KKT 1	720	6.288	115	13.359	54
2	KKT 2	128	0.839	153	1.691	76
3	KKT 4	369	2.872	128	6.727	55
4	KKT 6	426	3.400	125	9.044	47
5	KKT 7	484	3.342	145	6.628	73
6	KKT 8	127	0.906	140	1.700	75
7	KKT 9	679	6.161	110	12.990	52
8	KKT 10	535	3.330	161	6.963	77
9	KKT 11	57	0.309	184	0.547	104
10	KKT 12	117	0.853	137	2.158	54
11	KKT 13	1110	12.605	88	27.714	40
12	KKT 15	959	8.790	109	20.693	46
13	KKT 16	119	1.025	116	2.377	50
14	KKT 17	501	5.974	84	15.708	32
15	KKT 19	629	6.210	101	13.227	48
16	KKT 20	580	5.732	101	14.492	40
17	KKT 21	315	4.310	73	11.899	26
合 計		7855	72.946		167.917	
平 均 値		462	4.291	108	9.877	47

サブチェック等)が使用できないこと、テストランでのエラーのトレースがしにくいことがあること、コンパイル速度が遅いこと等の理由から、ディバック中のジョブに関してはOPT0を使用した方が良いと考えられるので、その場合には、FORTRANジョブステップの制御文のパラメータを省略せずにOPT0とすべきである。

表4-5はベンチマークジョブのSSOPTRAN使用時の1秒当りのソースカードトランスレート枚数を示したものである。表からは、1秒当りのソースカードトランスレート枚数にはかなりのばらつきがあるようであるが、平均的には約40枚と考えてよいであろう。

前節に示したように許容実記憶に対して大きな仮想記憶を要求するようなジョブに関しては、安全のために是非ともこのSSOPTRANを通すべきである。

それは前節の(1)に示したような事態が発生すれば、システムのオーバヘッドが増大し、磁気ディスクへのチャネルに隘路が生じ、そのジョブばかりか、他のジョブにも影響を及ぼすからである。

また、DOループの内側でユーザの作成したサブプログラムを呼んでいるようなプログラムを実行させる場合には、SSOPTRANを用いることによるプログラム実行時間の短縮は大きいものがあるので、この場合にもSSOPTRANを用いるべきである。

上記の場合を除けばSSOPTRANによる利得は少ないと考えられるが、SSOPTRANを用いてランジョブステップの実行時間が長くなる例のないこと、1割程度実行時間が短縮する場合はしばしばあることから、長時間ジョブ(20分程度以上を目安として)に対してはSSOPTRANを通した方が良いという結論が得られる。

表 4-5 SSOPTRAN 処理における 1 秒当りのトランスレート枚数

No.	ジョブ名	入力カード 枚数 (枚)	SSOPTRAN		備 考
			実 行 時 間 (秒)	1 秒当りのトランス レート枚数 (枚/秒)	
1	KKT1	720	13.675	53	
2	KKT2	128	1.633	78	
3	KKT4	369	9.541	39	
4	KKT6	426	14.362	30	
5	KKT7	484	7.833	62	
6	KKT8	127	1.715	74	
7	KKT9	679	14.823	46	
8	KKT10	535	7.175	75	
9	KKT11	57	0.436	131	
10	KKT12	117	1.695	69	
11	KKT13	1110	27.194	41	
12	KKT15	959	22.020	44	
13	KKT16	119	1.864	64	
14	KKT17	501	18.074	28	
15	KKT19	629	37.276	17	
16	KKT20	580	17.256	34	
17	KKT21	315	10.133	31	
合 計		7855	206.705		
平 均 値		462	12.159	38	

5. 浮動小数点演算について

新システム選定のためのベンチマークテストの際に生じたトラブルは、新システムと旧システムの浮動小数点演算の精度の相違に起因するものが多かった。これはシステム移行に際してもユーザにとって大きな問題であり、本章ではこの相違を引き起こすいくつかの事実について若干述べることにする。

仮数部は、旧システムの方が単精度で約 4 桁、倍精度で約 5 桁精度が高い。但し、旧システムには 4 倍精度がない。また、指数部は、旧システムの方が単精度、倍精度共に上にも下にも約 2 倍の大きさである（表 2-2 参照）。以上の結果に起因する計算結果の相違の主なものは

- (1) 指数部の範囲が新システムの方が約半分と小さくなっているため、旧システムで生じてなかった

浮動小数点オーバーフローが生ずることがある。

- (2) 仮数部の精度の相違は演算結果に影響を与え、数値的不安定性の高い計算においてはこの影響は致命的となることがある。

- (3) 数値的不安定性が低い、あるいは全くない場合であっても、算術 IF 文等による判断結果に相違が生じ、その結果全く異なる結果を生ずることがある。また、違った判断分岐を行なうため、従来発見されなかった計算上のエラーや論理的なエラーが発見されることがある。これは意外に多く存在する。

- (4) 反復計算

$$u^{n+1} = f(u^n)$$

において、収束判定を

$$|u^{n+1} - u^n| / |u^n| < \epsilon$$

により行なう場合、 ϵ の大きさは仮数部の精度に

関係しているので適切に定める必要があるが、これに無神経なユーザが多く、そのため無駄な回復を行なうことや理論的には収束すべき計算が実行上収束しないことが生ずる。

等があるので、十分に注意する必要がある。

6. アカウント情報およびシステム計測に関する実現について

計算機システムに投入されたジョブの行動の把握の一手段としてアカウント情報の処理をあげることができる。しかし、これはジョブが実行され計算結果が出力された後に、そのジョブが中央処理装置、主記憶装置、チャネル装置等のシステムリソースをどの位使用したか、計算機システム内のジョブの経過、およびラインプリンタ出力枚数、カード

出力枚数のような消耗品をどれだけ使用したかというような情報についての静的なデータを与えるものである。これらの情報はユーザに対する課金情報として、また、ジョブの特性を判断する材料として利用されるものである。

現時点においては、ジョブ、ジョブステップ、ファイルおよび TSS に関する情報として、表 6-1 のようなアカウント項目の収集を考えている。これらのジョブ毎の情報を磁気ディスク上に格納し、一定の周期で磁気テープに移し、これを保存する方針である。

このようにして収集された情報は、以下のような処理プログラムを通して処理されることになる。

その処理プログラムは、

- (1) ユーザの計算機使用実績の報告を目的とした

表 6-1 アカウント情報収集項目について

(a) ジョブに関する収集項目

No.	項 目
1	ジョブ番号
2	ジョブクラス
3	ジョブ入力開始日付および時刻
4	ジョブ入力終了日付および時刻
5	入力装置、入力データ形式および入力量
6	ジョブ実行開始日付および時刻
7	ジョブ実行終了日付および時刻
8	ジョブ入力開始からジョブ実行開始までの経過時間
9	ジョブ入力開始からジョブ実行開始までの計算機稼働時の経過時間
10	ジョブ実行開始からジョブ実行終了までの経過時間
11	ジョブ実行開始からジョブ実行終了までの計算機稼働時の経過時間
12	ジョブの終了状態および完了コード 制御文エラージョブ。ジョブイロット時エラージョブ。 途中ジョブステップ実行中に異常終了。最終ジョブステップまで実行終了。 キャンセルジョブ、スキップされたジョブステップの有無等。
13	ジョブの処理装置使用時間（処理装置別）
14	ジョブのファイルアクセス回数（read/write 別）
15	ジョブのチャネル使用時間（read/write 別）
16	ジョブのラインプリンタ使用枚数
17	ジョブのプリンタプロッタ使用枚数
18	ジョブのXYプロッタ用紙使用量
19	ジョブのカードせん孔枚数
20	ジョブのモニタの処理装置使用時間（処理装置別）

(b) ジョブステップに関する収集項目

No.	項 目
1	ジョブ番号
2	ジョブステップ番号
3	プログラム名
4	ジョブステップ実行開始日付および時刻
5	ジョブステップ実行終了日付および時刻
6	ジョブステップ実行開始からジョブステップ実行終了までの経過時間
7	ジョブステップ実行開始からジョブステップ実行終了までの計算機稼動時の経過時間
8	ジョブステップで使用するための入力データの入力装置、入力形式および量
9	ジョブステップ終了状態および完了コード ジョブステップ実行の有無、異常終了等。
10	使用した記憶装置の大きさおよび占有時間 ○仮想記憶領域を使用したか否か ○主記憶装置のみ使用した場合 最大主記憶使用量および平均主記憶使用量 主記憶占有時間 ○主記憶および仮想記憶装置を使用した場合 最大主記憶使用量および平均主記憶使用量 主記憶占有時間 最大仮想記憶使用量 平均仮想記憶使用量
11	ジョブステップの処理装置使用時間（処理装置別）
12	ジョブステップのファイルアクセス回数（read/write 別）
13	ジョブステップのチャネル使用時間（read/write 別）
14	ジョブステップのページング回数（イン/アウト別）
15	ジョブステップのページ取戻しを受けたページ数
16	ジョブステップのスワッピング回数
17	ジョブステップのスワッピングページ数（イン/アウト別）
18	ジョブステップの仮想入出力ページング回数（イン/アウト別）
19	ジョブステップのモニタの処理装置使用時間（処理装置別）

(c) ファイルに関する収集項目

No.	項 目
1	ジョブ番号
2	ファイルが使用されたジョブステップの番号およびプログラム名
3	ファイル番号
4	装置名
5	ボリューム通番
6	ファイル定義名
7	ファイルの使用状態 ダミーファイル、ファイル使用の有無等

8	ファイルの種類 ○SYSINファイル, SYSOUTファイル(ラインプリンタ, カードパンチ, XYプロッタ等の別), 一時ファイル, 個人ファイル, プログラムファイル, エラーメッセージファイル等 ○仮想入出力ファイルか否か
9	ファイルアクセス回数(read/write 別)
10	チャネル使用時間(read/write 別)
11	ステージング回数
12	ディスステージング回数
13	ファイル確保量
14	ファイル使用量

(d) TSSに関する収集項目*

No.	項 目
1	ジョブ番号
2	セッション開設日付および時刻
3	セッション閉設日付および時刻
4	端末入力回数または量
5	端末出力回数または量
6	セッション経過時間
7	セッションの終了状態および完了コード
8	使用されたコマンドの種類別回数および応答時間

* TSS ジョブのシステム資源の使用に関する項目は、ジョブ、ジョブステップおよびファイルの収集項目と同様である。

もの

- (2) ジョブおよびジョブステップの特性を引き出すことを目的としたもの

の2つに分けて作成し、(1)については、ジョブ処理件数、中央処理装置使用時間、ラインプリンタ出力枚数等のような各ユーザの計算機使用実績を出力し、(2)については、ジョブクラス毎のジョブのシステムリソースの使用に関するヒストグラム、平均値、標準偏差値等の情報、同種ジョブステップのシステムリソースの使用に関するヒストグラム、平均値、標準偏差値等の情報を出力し、ファイルおよびTSSに関するアカウント情報についても上記のジョブおよびジョブステップと同様の出力を行なう。

アカウント情報収集が静的データ収集であるのに

対し、稼動状況に関する時系列データである動的データの収集方法としてPDL/PDA^{注1)}をあげることができる。これは、計算機システムのハードウェアおよびソフトウェア双方の効率を測定するためのソフトウェアモニタである。PDLの機能は予め決められた項目を一定時間収集することであり、PDAの機能はPDLにより収集された情報に統計処理をほどこし、システム動作状況をレポートするものである。これらにより、中央処理装置の利用状況、実記憶の利用状況等を把握することができる。

M-160F計算機運転開始時においては、種々の運用パラメータは特に根拠のある値がはめられているわけではないが、ある期間、計算機の運転を行ない、その期間の静的および動的な収集データを蓄積する

注1) PDL(Performance Data Logger)およびPDA(Performance Data Analyzer)。

ことにより、それらのデータを組み合わせてシステムの状況を詳細に把握することができ、それらは適切なパラメータ値を決定するための判断の材料として有効に利用することができると同時に、将来のシステム変更の基礎資料ともなるものである。

7. 結 言

本報告では、角田支所に導入を予定しているFACOM M-160 F システムのハードウェア、ソフトウェア、システム運用方式とその実現およびユーザにとって最も関心の深いと思われるFORTRANコンパイラの特性等について述べた。旧システムに比較して新システムのハードウェア速度は約4倍であるが、FORTRANを通しての使用実績によれば平均約14倍、高いものについては20倍以上の能力があると推定される。これは主としてFORTRANコンパイラの最適化技法等によるものであることは、ベンチマークテストの分析より明らかである。また、システムの機能もTSS機能、マルチジョブ機能をはじめ、旧システムに対して格段の向上をみている。これらの機能とシステム資源を十分に活用するためには種々の試行錯誤が必要であるが、そのための道具として、アカウント、稼動状況の計測手段も新システム

には取り揃えられており、これについても本報告ではその一端についてふれることができた。

従って、新システムの導入とその効果的利用は、角田支所で推進されている研究開発を大いに促進するのにあずかって力があると信ずる次第である。稿を閉じるに当り、新システムの準備においてFHL(株)の永野氏をはじめとするSE諸氏にお世話になったので、紙上をかりて感謝の意を表する。

参 考 文 献

- 1) FACOM M160 Fハードウェア解説書
- 2) FACOM OSW/X8 ジョブ管理機能説明書
- 3) FACOM OSW/X8 AIF解説
- 4) FACOM OSW/X8 解説
- 5) FACOM OSW/X8 LIBE使用手引書
- 6) FACOM OSW/X8 システムユーティリティ使用手引書
- 7) FACOM OSW/X8 操作手引書
- 8) FACOM OSW/X8 システム編集手引書
- 9) FACOM FORTRAN77 使用手引書
- 10) 情報処理 Vol.21, 1980, No.4
- 11) FACOM SSOPTRAN使用手引書

航空宇宙技術研究所資料 431号

昭和56年4月発行

発行所 航空宇宙技術研究所
東京都調布市深大寺町1880
電話武蔵野三鷹(0422)47-5911(大代表)〒182

印刷所 株式会社 三興印刷
東京都新宿区信濃町12 三河ビル
