

航空宇宙技術研究所報告

TECHNICAL REPORT OF NATIONAL AEROSPACE LABORATORY

TR-659

計算機システムにおけるジョブ処理用
新スケジューラの提案

土屋 雅子 ・ 末松 和代
吉田 正廣 ・ 畑山 茂樹

1981年3月

航空宇宙技術研究所
NATIONAL AEROSPACE LABORATORY

目 次

1. はじめに	1
2. ジョブ・ジョブステップスケジューラ の概念	3
3. APUシステム用ジョブ・ジョブステップスケジューラ の試作	4
3.1 現ジョブ処理方式の問題点	4
3.2 新ジョブ処理方式に対する運用系	7
3.3 試作ジョブ・ジョブステップスケジューラ	8
3.3.1 SCD1	8
3.3.2 SCD2	11
4. ジョブ・ジョブステップスケジューラ 効果評価用シミュレータ	13
4.1 シミュレータの概要	13
4.2 記号および制御表	13
4.3 各プログラムの処理概要	13
4.3.1 メインルーチン	13
4.3.2 ジョブ発生ルーチン	13
4.3.3 ジョブ到着処理ルーチン	13
4.3.4 ジョブイニシエート処理ルーチン	21
4.3.5 ジョブステップ実行保留処理ルーチン	22
4.3.6 ジョブステップ保留解除処理ルーチン	22
4.3.7 ジョブステップイニシエート処理ルーチン	23
4.3.8 主記憶管理ルーチン	23
4.3.9 タスクスケジューリングルーチン	25
4.3.10 ジョブステップターミネート処理ルーチン	25
4.3.11 ジョブターミネート処理ルーチン	25
4.3.12 出力ルーチン	26
4.3.13 ジョブ・ジョブステップスケジューリングルーチン	26
4.3.14 システム見直しルーチン, ジョブステップ実行中断処理ルーチン, ジョブステップ 実行再開処理ルーチン	26
5. シミュレーション実験	29
5.1 実験環境の設定	29
5.2 実験項目	29
(i) 実験Ⅰ	29
(ii) 実験Ⅱ	33
(iii) 実験Ⅲ	33
(iv) ラッシュアワー実験	33
5.3 実験結果	33
6. 考察	51
6.1 新旧スケジューラ の有効性比較	51
6.1.1 主記憶競合小なる環境での比較	51
6.1.2 主記憶競合大なる環境での比較	53

6.2	SCD1用の運用系	54
6.2.1	ラン多重度について	54
6.2.2	IO多重度について	54
6.2.3	優先権について	55
6.3	SCD2の運用系	55
6.4	APUシステム用ジョブ・ジョブステップスケジューラ	56
7.	おわりに	56

計算機システムにおけるジョブ処理用 新スケジューラの提案*

土屋 雅子** 末松 和代**
吉田 正廣** 畑山 茂樹**

New Scheduler for Job Processing in Computer Systems

Masako TUCHIYA, Kazuyo SUEMATSU,
Masahiro YOSHIDA and Shigeki HATAYAMA

ABSTRACT

This paper is concerned with the job processing method in computer systems. Firstly, we find fault with job processing by means of the conventional scheduler, and point out the fact that utilization of the central processing unit (CPU) is too sensitive to the input job-stream. This is the reason it is difficult to retain constantly high CPU utilization in keeping with the increasing throughput of the short job-class. Secondly, a new job processing method promising to solve this problem, which we shall call the job/job-step scheduler, is proposed. The main concept is to schedule a job or a job-step so that a number of so-called CPU-bound job-steps may always be in a state for immediate processing by CPU. In order to make the new scheduling possible, the operating systems must have the following job-processing capability: all job-steps setting up the job shall not necessarily be processed sequentially; it should be possible to halt the job processing temporarily at some job-step stage for preparation of the pool of CPU-bound job-steps. Thirdly, we describe the skeleton of the simulator which was programmed to evaluate the performance of the job/job-step scheduler in an application for job processing with the NAL computer system. Finally, the results of the simulation experiment are given to illustrate the effectiveness of our proposed scheduler.

1. はじめに

航技研計算センタは昭和53年4月以後、FACOM 230-75 APUシステム (APUシステムと略称) で運営している。このシステムは汎用プロセッサ 230-75 (CPUと略称) と専用プロセッサ Array Processor Unit (APUと略称) とを非対称マルチプロセッサ構成にしたものである。APUは膨大な量の

アレイ計算を中心とした、並列処理が可能な科学技術計算に対して極めて高速な演算処理能力を有したものであり、ユーザプログラム (即ち、ランジョブステップ) のみを実行する専用機である。^{1),2)} 一方、CPUはAPUを使用するジョブ (APジョブと略称) のコンパイル、リンケージエディット等のサービスプログラムの実行並びに入出力処理を受け持つほかに、APUを使用しないジョブ (CPジョブと略称) に関する全ての処理を行う。

APUシステムを効率的に運営していくためには、

* 昭和55年11月29日受付

** 計算センター

航技研のジョブ集団に対して適切な運用系を設定しなければならない。その際、限られた主記憶資源のもとで、いかにしてCPUとAPUの稼働率を高く維持するかが最重要課題となる（本システムの主記憶容量は768K語であり、この内、568K語がユーザ使用可能である）。現在はAPジョブを3本、CPジョブを3本の合計6多重でジョブ処理を行っており、CPジョブに関しては主記憶およびCPU資源要求量がともに少ないジョブを優先的に処理し、APジョブに関しては主記憶に余裕がない時には主記憶要求量の少ないジョブを優先的に処理し、余裕がある時には主記憶要求量の大きなジョブの実行を優先させるという方針で運用系を設定している³⁾。しかし、単にジョブクラスに対応した多重度、主記憶優先権、実行優先権等の外部設定パラメータをいかに最適に組合せた運用系をもってしても、APUシステムの稼働率を向上しえる程度には自ずから限界がある。

この限界を越えるため、筆者らはオペレーティングシステムに多少立ち入り、主記憶の利用率向上のための諸方策を考察し、それらをAPUシステムに適用した場合に得られる効果の予測を行った⁴⁾。方策の第一はロールイン・ロールアウト処理方式の改善であって、ロールアウト状態にあるジョブステップを可能な限りロールインするような方式に改めること、第二は主記憶上の断片化を出来る限り克服して、主記憶利用率を大巾に向上しうる主記憶アロケータの設計である。第三は物理的資源の増設によらず、主記憶再配置機能の活用によって主記憶の有効利用を最大限に計ること、第四は主記憶を増設して、物理的に資源の拡大を計ることである。我々はシミュレーション実験によって、第一から第三の方策を複合利用すれば、主記憶を増設することなく上記運用系のもとで、相当高く処理装置の稼働率を維持しえることを検証した。

しかし、このシミュレーション実験を通じて、次のような問題点の存在が明らかとなった。即ち、主記憶上の競合が全くなく、かつ処理待ちジョブも十

分に存在するという状況のもとにおいても、APUとCPUの稼働率が明らかにジョブストリームに依存して、時間的にかなり大巾な確率の変動を呈する^(注)。これは処理すべきジョブがあるにもかかわらず、実行中の6個のジョブによってAPUとCPUの処理能力を十分定常的に使い切れていないことを意味しており、ジョブの混雑時にはこの問題点が特に指摘される。

6多重でジョブ処理を行っているにもかかわらずAPU、CPUがアイドルになる原因は明らかに、全ジョブの処理状態が一時的に入出力処理待ち状態になってしまい、APUとCPUによる処理が全く中断してしまうためである。航技研ジョブに関していえば、サービスジョブステップは入出力頻度が平均して非常に高いものが多い（以後、この状態をチャンネルバウンドという）、ランジョブステップは平均して非常に少ないものが多い（以後、この状態を処理装置バウンドという）。従って、主記憶上の競合がない状況においても平均的にかなりの高率で処理装置がアイドルになるのは6多重共にサービスジョブステップの処理を行っている状況がかなりの頻度で出現することに起因している。

実は、上記の問題を惹起する根本的理由は計算機システム内でのジョブ処理方式そのものに内在している。現在のジョブ処理方式では、投入されたジョブはまず処理待ちキューに登録され、ジョブスケジューラが一定のアルゴリズムに従って、実行多重度の空きの件数だけのジョブを処理待ちキューから選定する。選定されたジョブは第一ジョブステップから最終ジョブステップまでの一連のジョブ処理を連続して行う。このため、処理装置バウンドな状態のジョブの存在を保証しえず、航技研ジョブストリームでは度々、実行中の各ジョブの処理状態が全てチャンネルバウンドになってしまい、APUもCPUも見掛け上一時的に負荷が軽くなる現象が起る。

上述の問題を解決し、処理待ちジョブがある限り

(注) 本稿ではAPU、CPU共に100%ユーザジョブの実行に利用しえるものとしており、OSの走行時間は考慮していない。従って、処理装置稼働率は、シミュレーション時間に対するユーザジョブが処理装置を使用した時間の割合を意味し、処理装置アイドル率はシミュレーション時間に対して処理装置が処理しえる状態のユーザジョブが存在しなかった時間の割合を意味する。

出来るだけ処理装置のアイドル率を小さくし、ジョブストリームの時間的变化に影響されずに処理装置を出来るだけ定常的に使い切るためには、実行多重度のいくつかが常にランジョブステップによって占有されるようなジョブ処理方式をとればよい。この一見素朴に思える考え方は我々が知る範囲ではまだ採用されていないようである。このようなジョブ処理方式を実現するために、我々は従来のジョブスケジューラとは全く異なるスケジューラ（ジョブ・ジョブステップスケジューラと命名）を考案した。オペレーティングシステムの観点からの詳細な検討は別に於いて、このようなジョブ処理方式のもつ非常に高い有効性は本稿において十分定量的に示し得たものと考えている。

また、同一ジョブ集団のもとでは処理装置の性能の向上と共に、どのジョブステップの平均処理装置サービス時間も低下し、^(注) それだけ入出力要求の頻度が増大するから、同一主記憶容量のもとで同一運用系を採用するならば、処理装置のアイドル率が増々高まらざるをえない。よって、主記憶コストをいたずらに増大することなく、処理装置の有効利用を計る観点からすれば、このようなジョブ処理方式は今後必須なものになってくると考えられる。

2. ジョブ・ジョブステップスケジューラ の概念

基本的な考え方は処理すべきジョブが存在する限り、単にランジョブステップの処理のためにいくつかの多重度（ラン多重度とよぶ）が占有されるようなジョブ処理方式を実現するところにある。

このためには、一つのランジョブステップの実行が終わったならば、直ちに別のランジョブステップの実行を開始しえるように、ランジョブステップをプールしておく必要がある。ジョブはいくつかのジョブステップで構成されているから、ランジョブステップをプールするためには、一つのランジョブステップに先行するサービスジョブステップの処理がすでに完了していなければならないし、また、ランジョブステップの終了をまって、それに後続するサー

ビスジョブステップの処理が開始されなければならない。もし一つのジョブでサービスジョブステップが複数個連続しているならば、それらは続けて処理してしまう。これらのサービスジョブステップの処理を実行するために、いくつかの多重度（IO多重度とよぶ）を用意しておかなければならない。一般的に於いて、プールすべきランジョブステップは処理装置使用時間の長いジョブクラスのものに限定してよい。この場合には、処理装置使用制限値の小さいジョブクラスのランジョブステップはサービスジョブステップと区別することなく、IO多重度の範囲内で処理することになる。即ち、IO多重度の空きを利用して、処理装置使用時間の短いクラスのジョブ処理と、プールされるランジョブステップに先行並びに後続しているサービスジョブステップの処理を行うことになる。ラン多重度の決定においては、ジョブクラスの数と処理装置の演算能力とを考慮し、IO多重度は資源要求量の小さいジョブのスループットを考慮して決定すべきである。

ランジョブステップはジョブクラス毎に適当な個数がプールできていれば十分であるが、この個数がある一定値以下になったら優先的にそのクラスのジョブの起動を行って、プール個数を一定値以上に保つようにする。このためのジョブ起動はIO多重度の空きを優先的に利用して行う。ジョブの起動に対して競合が発生した場合には、処理装置のアイドル率を出来るだけ逡減する目的から、処理装置使用時間の長いクラスのジョブを優先して起動する。このようなアルゴリズムにもとづいてジョブのスケジューリングを行っても、システム内でのジョブ処理が定常状態になれば、優先したクラスのジョブの起動は散発的にしか発生しないし、処理に要する時間も一般に十分短いから、IO多重度の空きは主に資源要求量の小さいクラスのジョブ処理に使用される。

また、プールされているランジョブステップの実行はラン多重度の空きを利用して行う。その際に必要となるジョブステップのスケジューリングは簡単に、終了したジョブステップと同じクラスのジョブステップをプールの中から先着順に（FIFO方式

(注) 平均処理装置サービス時間は入出力要求の平均間隔を意味する。

という) 起動する方式でよい。

さらに、ランジョブステップに後続するサービスジョブステップの実行は IO 多重度の空きを利用して行うが、スケジューリングの優先順位はランジョブステップに先行するサービスジョブステップよりは低くてよいが、IO 多重度の空きを利用して処理するクラスのジョブよりは高くしておく。

以上、概略的にジョブ・ジョブステップスケジューラによるジョブ処理方式を記述したが、これをより具体化するには更に、ジョブまたはジョブステップを起動する際の選定基準の設定が必要である。この基準に応じて、ジョブ・ジョブステップスケジューラは種々の形態をとることができる。

この新しいジョブ処理方式の実現によって次のような大きな効果が期待される。第一に、ジョブの混雑時には常にラン多重度分の処理装置バウンドなジョブが実行されているから、処理装置の稼働率を高くかつ、定常的に維持しえる。第二に、資源要求量の大きなジョブクラスのジョブでも、ランジョブステップに先行するサービスジョブステップ段階にエラーのあるジョブのターンアラウンド時間が格段と向上しえる。第三に、IO 多重度の空きを利用して処理するクラスのジョブは実質的に IO 多重度に近い個数だけ同時に処理されるから、そのクラスのターンアラウンド時間の短縮が保障しやすくなる。

3. APU システム用ジョブ・ジョブステップスケジューラの試作

3.1 現ジョブ処理方式の問題点

航技研では一般ユーザの使用できるジョブクラスを 5 つ設け、それぞれに表 3.1 に示す制限値を課し

ている。昭和 53 年度アカウント情報の統計解析結果によると、航技研ジョブ集団のジョブクラス構成比率、各ジョブクラスの制御文エラー率、および各クラスのジョブのジョブステップ列構成比率は表 3.2~3.3 に示す通りであり、各ジョブステップ毎の平均的システム資源使用量は表 3.4 に示す通りである。

APU システムでは概ね図 3.1 に示すような流れでジョブ処理が行われるが⁵⁾、システムの効率的運営の立場から運用系は次のように設定している。

主記憶優先権^{*} SY=AM>SH>DT>CM>AL

実行優先権^{*} SY=AL>SH=DT=AM>CM

多重度^{**} SH=DT=CM=AL=1, AM=2

• 記号 SY はサービスジョブステップを表わし、その他は各ジョブクラスのランジョブステップを表わす。

** 全記号はジョブクラスを表わす。

システムに投入されたジョブはまず、ユーザが指定したジョブクラスの処理待ちキューに登録される。ジョブスケジューラは、ジョブクラス毎に設定された上記多重度に空きがある限りそれぞれの処理待ちキューから空き多重度分のジョブを起動する。起動するジョブの選定は各ジョブクラス共に同一ユーザのジョブが連続していない限り FIFO 方式で行う。起動されたジョブは全て第一ジョブステップから最終ジョブステップまで連続的に処理される。その際、サービスジョブステップの処理は全て CPU でなされるが、ランジョブステップの処理は CPU のみでなされるもの (CP ジョブの場合) と APU と CPU の両方を使ってなされるもの (AP ジョブの場合) とがある。ただし、主記憶に対する競合が発生する

表 3.1 ジョブクラスと制限値

ジョブクラス	メモリ制限値 (K 語)		CPU 制限値 (秒)	APU 制限値 (秒)
	サービスジョブステップ	ランジョブステップ		
SH	400	60	120	0
DT	400	100	180	0
CM	400	160	1200	0
AM	400	160	120	1200
AL	400	300	360	3600

と、上記主記憶優先権に従っていくつかのジョブステップがロールアウトされるため、主記憶上にあるジョブステップのみが上記実行優先権に従って処理装置のサービスを受けることになる。ロールアウト状態のジョブステップは、他のジョブステップによる主記憶解放事象が発生する毎にロールイン処理を試みられる。

また、上記運用系を採用している根拠は次の通りである。第一に、AMとALクラスのジョブを共に1多重とし、主記憶優先権を最高にして運用した場

合、APUの稼働率は高くできるが、CPUの稼働率が激減する。第二に、AMとALクラスのジョブを共に1多重とし、ALクラスのジョブの主記憶優先権をCPクラスのジョブより低くして運用した場合、CPUの稼働率は高くできるが、APUの稼働率が激減する。以上の事情により、通常は2個のAMジョブを処理し、主記憶に余裕がある時期にはALジョブを優先的に処理するような運用系にして、APUとCPUが共にバランスした稼働率を維持できるように努めている。

以上からも判るように、このようなジョブ処理方式に対して、処理装置の稼働率を定常的に維持し、かつスループットをも保障しようとする観点からすれば、処理すべきジョブがあり、かつ主記憶上の競合もない場合においても次のような問題点が指摘できる。まず第一に、APUが実行可能なジョブステップ個数は最大3から最低0まで、確率的な意味で不定である。即ち、APジョブの多重度は3であるが、これらが全てサービスジョブステップの実行段階にあれば、APUが実行すべきジョブステップは

表3.2 ジョブクラス構成比率と制御文エラー率

ジョブクラス		項目	構成比率 (%)	制御文エラー率 (%)
CPクラス	SH		70.2	5.0
	DT		5.9	10.1
	CM		7.3	8.0
APクラス	AM		13.4	5.8
	AL		3.2	7.8

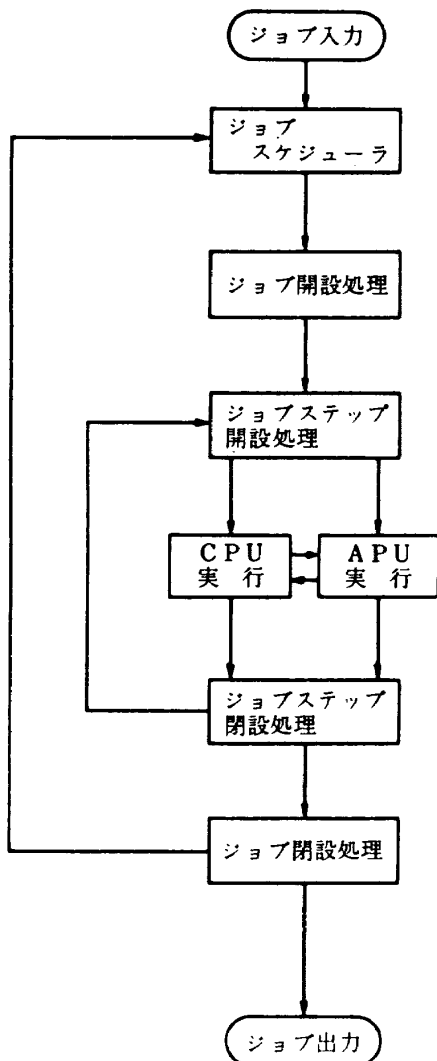
表3.3 ジョブステップ列構成比率

ジョブステップ列番号	AP又はCP FORTRAN	LIED	LIBE	UTILITY	AP又はCP RUN	ジョブステップ数	構成比率 (%)	
							SHクラス	DT, CH, AM ALクラス
1	1					1	11.2	12.9
2			1			1	4.4	0
3				3		3	5.7	0
4	1	1				2	1.8	2.1
5	1		1			2	1.3	0
6			2	2		4	1.4	0
7	2	1	2			5	0.6	0.7
8	1		2	3		6	0.2	0
9	2	1	2	2		7	0.2	0.2
10					1	1	10.7	12.3
11	2				1	3	0.2	0.2
12		1			1	2	0.8	0.9
13			2		1	3	0.4	0.5
14				3	1	4	0.2	0.2
15	1	1			1	3	54.4	62.5
16	2	1	2		1	6	4.5	5.2
17	1	1		2	1	5	1.6	1.8
18	2	1	2	2	1	8	0.4	0.5
合計							100.0	100.0

表 3.4 ジョブステップ毎システム資源使用量の平均値

項目 ジョブステップ名	AP 使用 時間 (秒)	CP 使用 時間 (秒)	主 記 憶 使 用 量 (K 語)	ディスクへの ファイルアクセス 回 数 (回/ジョブステップ)	ディスクへの ファイルアクセス 頻 度 (回/秒)	MT への ファイルアクセス 頻 度 (回/秒)	CP 通信の 頻 度 (回/秒)	CP サービス 時 間 (ミリ秒)	AP サービス 時 間 (ミリ秒)
サービス ジョブ ステップ	CP-FORTRAN	0.	3.1	90.0*	108.69	—	—	28.5	—
	AP-FORTRAN	0.	10.5	116.8	115.02	—	—	91.3	—
	LIED	0.	1.8	47.0*	538.59	—	—	3.3	—
	LIBE	0.	0.3	32.0*	146.45	—	—	2.0	—
	UTILITY	0.	1.2	31.4	828.98	—	—	1.4	—
ランジ ジョブ ステップ	SH-RUN	0.	17.0	66.0	—	4.01	—	249.4	—
	DT-RUN	0.	254	65.2	—	3.13	44.87	20.8	—
	CM-RUN	0.	331.2	81.6	—	0.65	—	1538.5	—
	AM-RUN	204.7	3.5	99.5	—	0.42	—	15.1	875.3
	AL-RUN	724.3	12.5	137.7	—	0.18	—	28.6	1654.9

(注) *印の値は固定



1件もなくなり、APUはアイドルになってしまう。表3.2～3.4の統計値で記述したジョブ集団に対するシミュレーション実験によると、このような状態が発生する確率は意外に大きく、そのためにAPUの稼働率をコンスタントに保つことが難しい⁴⁾。第二にCPUが実行可能なジョブステップの個数は最大6から最低3まで確率的に変化する。しかもシミュレーション実験からいえば、CPUが実行すべきジョブステップが全てチャンネルバウンドな状態になってしまう確率も予想外に大きく、そのためCPUの稼働率をコンスタントに維持することが非常に困難である⁴⁾。第三に、SHジョブクラスの多重度は1に固定されるため、このクラスのジョブのターンアラウンドタイムの短縮の手段は、主記憶および実行優先権を高位にする以外にない。第四に、起動されたジョブの全ジョブステップを連続処理するために、ジョブ処理に長時間要するジョブクラスでは、コンパイルエラー等で短時間で処理が終了するジョブがあったとしても早期発見が不可能である。

APUシステム用ジョブ・ジョブステップスケジューラは、これらの問題点を解決するという方針のもとに設計し、運用系を設定しなければならない。

図 3.1 APUシステムの現ジョブ処理方式

3.2 新ジョブ処理方式に対する運用系

まず始めに、ジョブ・ジョブステップスケジューラを用いてAPUシステムのジョブ処理を行う場合に必要な運用系に対する考え方を述べる。APUに対するラン多重度は表3.4から明らかのように、AMとALクラスのランジョブステップは共にAPUバウンドであるから、1多重でも相当高い稼働率が期待できる。しかし、1多重ではAPランジョブステップがCPUプログラムの実行に入るとAPUがアイドルになってしまうし、またAPジョブには2つのジョブクラスがあることを考え合せると、APUのラン多重度は2とするのがよい。次にCPUに対するラン多重度であるが、表3.4からCMのランジョブステップ用に1多重あれば十分なことが判る。

SHとDTクラスのランジョブステップは、表3.4に示されるように、CM, AM, ALクラスのそれと比べて処理装置使用時間が非常に短いから、これらはサービスジョブステップと区別せずにIO多重度の空きを利用して処理する方がスループットの向上につながる。この場合、SHとDTクラスのジョブは、その一連のジョブステップが連続して処理される。一方、CM, AM, ALクラスのサービスジョブステップは、IO多重度の空きを利用して処理され、ランジョブステップはラン多重度の空きを利用して処理される。従ってこのクラスのジョブ処理は一時的に中断する。

このIO多重度は、DTクラスのジョブ処理に1多重、SHクラスのジョブ処理にはターンアラウンドの保障を考慮して2~4多重と単純に割り振って推定してみると、3~5のIO多重度が必要と考えられる。システム立ち上げ時期にはランジョブステップのプールの補充を行うから、それらの先行サービスジョブステップの処理にIO多重度、ラン多重度共に占有される。このため、システム立ち上げの一時期、プールの対象にならないクラスのジョブ処理は行われぬこともあるが、プール補充が完了した定常時にはIO多重度の多くがSHとDTクラスのジョブ処理に利用されていると考えられる。

ランジョブステップのプール補充は表3.4の平均的資源使用量から判断すれば、CM, AM, ALのランジョブステップ共に1ラン多重度につき3個を目

途にして行えば十分であろう。しかし、非常に短時間に処理が終了するランジョブステップが続いた場合、一時的にせよこの最小個数を保持できなくなることも起きるし、またエラージョブの早期発見の立場からしてもプール個数設定値はもっと大きく設定するのがよいであろう。

勿論、ラン多重度で処理されているジョブステップがロールアウトされ易くしてはここでの目的は達成できない。このため、実行優先権は低くてもランジョブステップはできるだけAPUとCPUに対して実行可能な状態にしておく必要があるから、CM, AM, ALのランジョブステップの主記憶優先権は高位に設定しなければならない。サービスジョブステップの主記憶優先権も当然高位にすべきであるから、SHとDTクラスのランジョブステップがロールアウトされ易くなる。したがって、これらの実行優先権は高位に設定しておく必要がある。このように、主記憶要求量の大きいクラスのジョブの主記憶優先権を高くして運用するため、できるだけ主記憶の有効利用を計る必要性が高まる。⁴⁾

以上をまとめてみると、新ジョブ処理方式に対する運用系は次のようになる。

主記憶優先権

$SY \{ \supset \} AM = AL > CM > DT \{ \supseteq \} SH$

実行優先権

$SY \{ \supset \} AM = AL > SH \{ \supseteq \} DT > CM$

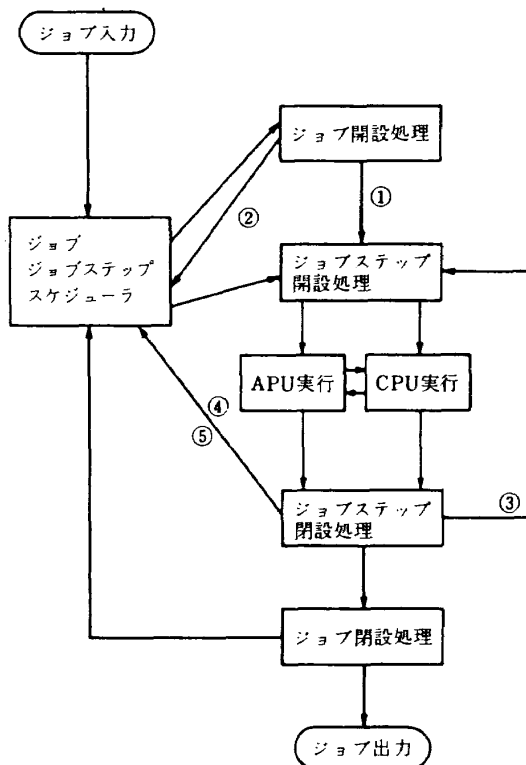
APラン多重度	2
CPラン多重度	1
IO多重度	3~5

APジョブの主記憶、実行優先権を共にCPジョブより高くしているのは、APUのアイドルをできるだけ減す配慮からである。こうしても表3.4から判るように、CPUに対する負荷はそれ程大きくはない。表3.4の平均値から推定すればAMとALクラスのジョブの実行優先権が等しい、いわゆるラウンドロビン方式を採用すると、AMジョブ3件につきALジョブ約1件の割合でAPジョブが処理されよう。なお、上記運用系において不確定な所は第5章のシミュレーション実験結果に基づいて定める必要がある。

3.3 試作ジョブ・ジョブステップスケジューラ

本節ではAPUシステム用に試作した2つのジョブ・ジョブステップスケジューラ SCD1 と SCD2 のアルゴリズムについて詳述する。起動の際のジョブまたはジョブステップ選定基準として、SCD1 はジョブクラス毎にFIFO方式を採用したものであり、SCD2 はジョブがシステムに投入されてから現在までの重み付き時間を採用したものである。これらのスケジューラは、

- (i) ジョブが入力された時点において多重度に空きがあった場合
 - (ii) 1つのジョブ処理が終了した時点
 - (iii) 起動したジョブの第1ジョブステップがランジョブステップであった場合
 - (iv) 処理中ジョブの後続ジョブステップがランジョブステップであった場合
 - (v) 1つのランジョブステップが終了した時点
- を契機として、どのジョブクラスのどのジョブまたは



- ① 第1ジョブステップがサービスジョブステップの時
- ② 第1ジョブステップがランジョブステップの時
- ③ SHとDTクラスのジョブおよびサービスジョブステップの連続処理
- ④ 後続ジョブステップがランジョブステップの時
- ⑤ ランジョブステップの処理が終了した時

図 3.2 APUシステムの新ジョブ処理方式

はジョブステップを起動するかを決定する。特に、(iii)～(v)の事象発生を境として、1つのジョブ処理が中断することに注意する。これらのスケジューラを用いた場合、APUシステムのジョブ処理は概略図3.2に示す如くなる。

3.3.1 SCD1

このジョブ・ジョブステップスケジューラは、APラン多重度2をAMとALクラスのランジョブステップに、IO多重度の1つをDTクラスのジョブにそれぞれ割当ててしまうところにSCD2との主な相違点がある。詳細は図3.3(記号については表3.5参照)の通りであるが、このスケジューラが起動を決定するジョブまたはジョブステップを優先順に記すと次のようである。

- ① 第1にAPラン多重度に空きがあればこれを優先的にふさぐ。その際、AMランジョブステップをALランジョブステップより優先して起動する。AMプールが零の時はAMクラスのジョブを、ALプールが零の時はALクラスのジョブを起動する。この場合、APラン多重度の空きでAPジョブの先行サービスジョブステップの処理が行われてしまうが、処理すべきジョブが多数ある限りこのような状態はシステム立ち上げ時以外めったに発生しない。
- ② APラン多重度に空きがなければ、次にCPラン多重度の空きを調べる。空きがあればCMランジョブステップを起動する。CMプールが零ならば、CMクラスのジョブを起動する。この場合にもCPラン多重度の空きでCMジョブの先行サービスジョブステップの処理が行われる。
- ③ ラン多重度に空きがなければ、AMプール個数を調べる。AMプール個数が設定値以下であれば、このIO多重度の空きを利用してAMジョブの先行サービスジョブステップの処理を行う。
- ④ AMプール個数が設定値以上であれば、ALプール個数を調べる。ALプール個数が設定値以下であればこのIO多重度の空きを利用してALジョブの先行サービスジョブステップの処理を行う。
- ⑤ ALプール個数が設定値以上であれば、CMプール個数を調べる。CMプール個数が設定値以下であればこのIO多重度の空きを利用してCMジ

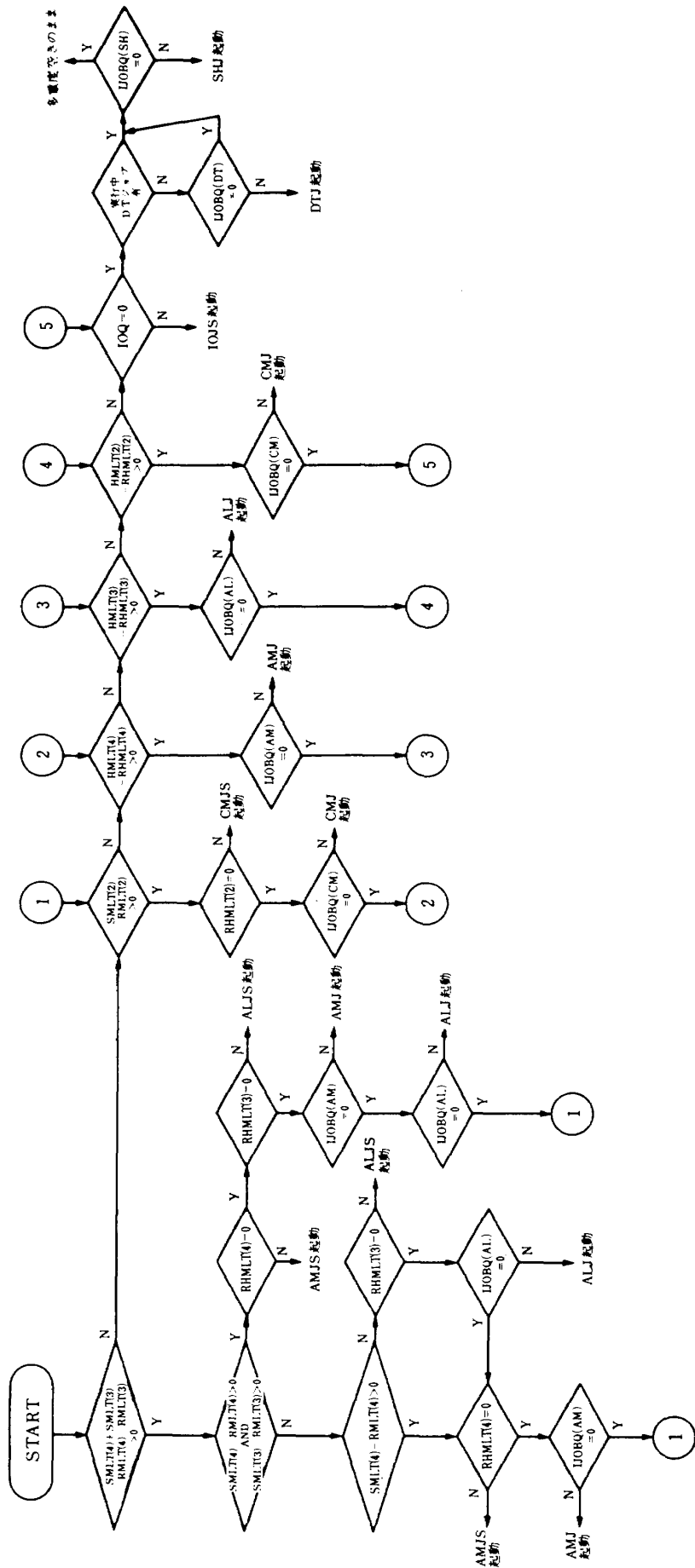


図 3.3 SCD 1 のジョブおよびジョブステップ起動のアルゴリズム

- ジョブの先行サービスジョブステップの処理を行う。
- ⑥ CMプール個数が設定値以上であればIOキューを調べる。これはランジョブステップに後続するサービスジョブステップの処理待ちキューであるが、このキューが零でない限り、このIO多重度の空きを利用してその処理を行う。
- ⑦ IOキューが零であれば、DTジョブが処理中であるか否かを調べる。処理していなければDTクラスのジョブを起動する。
- ⑧ DTジョブが処理中であれば、SHクラスのジョブを起動する。
- 上記において、起動が決定したクラスのジョブの

表3.5 記号の説明

記号	意味	記号	意味
SHJ	SHの処理待ちジョブ	SMLT(1)	システムIO多重度
DTJ	DTの処理待ちジョブ	SMLT(2)	システムCP多重度
CMJ	CMの処理待ちジョブ	SMLT(3)	システムAP多重度(図3.4)
AMJ	AMの処理待ちジョブ	SMLT(3)	システムAM多重度(図3.3)
ALJ	ALの処理待ちジョブ	SMLT(4)	システムAL多重度(図3.3)
SHJS	SHの実行保留ステップ	RMLT(1)	実行中IO多重度
DTJS	DTの実行保留ステップ	RMLT(2)	実行中CPラン多重度
CMJS	CMの実行保留ステップ	RMLT(3)	実行中APラン多重度(図3.4)
AMJS	AMの実行保留ステップ	RMLT(3)	実行中AMラン多重度(図3.3)
ALJS	ALの実行保留ステップ	RMLT(4)	実行中ALラン多重度(図3.3)
IJOBQ(SH)	SHの処理待ちキュー	HMLT(2)	CPプール個数設定値
IJOBQ(DT)	DTの処理待ちキュー	HMLT(3)	APプール個数設定値(図3.4)
IJOBQ(CM)	CMの処理待ちキュー	HMLT(3)	AMプール個数設定値(図3.3)
IJOBQ(AM)	AMの処理待ちキュー	HMLT(4)	ALプール個数設定値(図3.3)
IJOBQ(AL)	ALの処理待ちキュー		
IOQ	後続サービスジョブステップ処理待ちキュー		
RHMLT(2)	CPプール個数		
RHMLT(3)	APプール個数(図3.4)		
RHMLT(3)	AMプール個数(図3.3)		
RHMLT(4)	ALプール個数(図3.3)		

選定は、そのジョブクラスの処理待ちキューからFIFO方式で行う。また、起動が決定されたクラスのランジョブステップの選定は同一ユーザのジョブが連続しない限りFIFO方式で行う。処理待ちサービスジョブステップの選定はIOキューからFIFO方式で行う。

3.3.2 SCD 2

このジョブ・ジョブステップスケジューラはSCD 1と異って、APラン多重度をAMとALクラスのジョブに、IO多重度の1つをDTクラスのジョブに、特に割当していないところに特徴がある。そして、ジョブまたはジョブステップ選定基準としてSCD 1のようなFIFO方式をとらず、重み付き経過時間に基づく方式を採用している。^(注1) 詳細は図3.4（記号については表3.5参照）の通りであるが、このスケジューラが起動するジョブまたはジョブステップを優先順に記すと次のようである。

- ① まずAPラン多重度の空きを調べる。空きがあればAPプールの中で重み付き経過時間が最大のランジョブステップを1つ起動する。^(注2) APプールが零であれば、AMクラスとALクラスの処理待ちキューの先頭の2つのジョブの中で、重み付き経過時間の大きい方を起動する。
- ② APラン多重度に空きがなければ、CPラン多重度の空きを調べる。空きがあればCPプールからFIFO方式でCMランジョブステップを1つ起動する。CPプールが零であれば、処理待ちキューからFIFO方式でCMクラスのジョブを1つ起動する。

- ③ ラン多重度に空きがなければ、APプール個数を調べる。APプール個数が設定値以下であれば、AMクラスとALクラスの処理待ちキューの先頭の2つのジョブの内、重み付き経過時間の大きい方のジョブを起動する。
- ④ APプール個数が設定値以上であればCPプール個数を調べる。CPプール個数が設定値以下であれば、CMクラスの処理待ちキューからFIFO方式でジョブを1つ起動する。
- ⑤ CPプール個数が設定値以上であれば、IOキューを調べる。IOキューが零でなければFIFO方式で後続サービスジョブステップを1つ起動する。
- ⑥ IOキューが零であれば、SHクラスとDTクラスの処理待ちキューの先頭の2つのジョブの内、重み付き経過時間の大きい方のジョブを起動する。

以上から明らかなように、SCD 2ではAPランジョブステップが2件処理中であっても、AMとALクラスが1件ずつ処理されているとは限らず、2つ共AMまたはALジョブとなることがある。またCPジョブに関しても、CMクラスの1件は保障されているが、SHクラスとDTクラスが何件ずつ処理されるかは不確定である。このように、SCD2ではジョブクラス毎の多重度を重視してジョブ処理を行うのではなく、ジョブのシステム内滞在時間を重視して処理が行われる。

航技研の計算機システムに対して、どちらのスケジューラが望ましいかは第5章のシミュレーション実験結果を見て判断する。

(注1) 経過時間は処理待ちキューに登録されてから現在までの時間をいう。5.3節で出てくる経過時間（ジョブ開設から閉設までの時間）とは異なる点に注意

(注2) APプールにはAMとALランジョブステップが混在している。

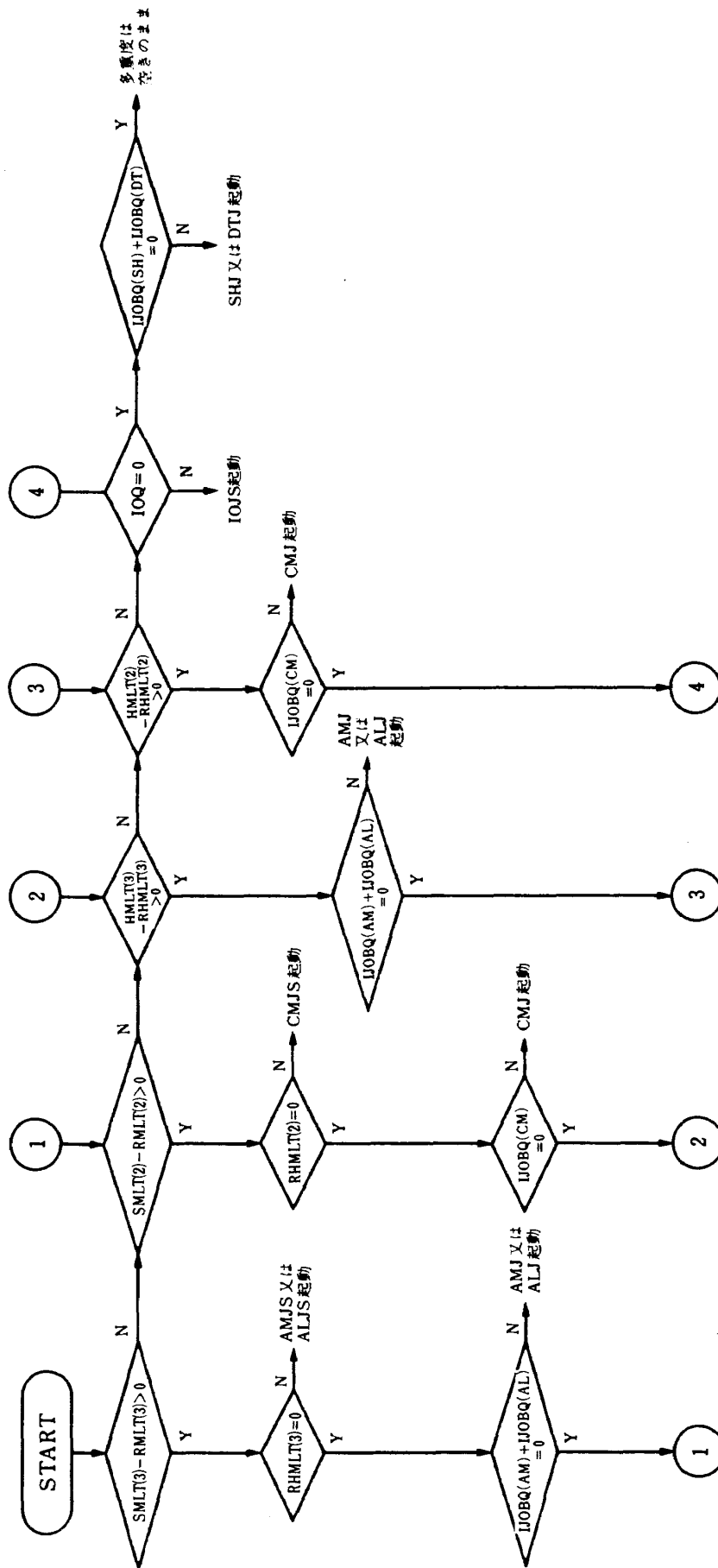


図 3.4 SCD 2 のジョブおよびジョブステップ起動のアルゴリズム

4. ジョブ・ジョブステップスケジューラ 効果評価用シミュレータ

本章では試作ジョブ・ジョブステップスケジューラの有効性を評価し、また、運用系の検討を行うための道具として開発したジョブプロセッシングシミュレータ（JPSと略記）について記述する。このシミュレータは先に開発したAPUシステム用JPSを母体とし、かつ主記憶有効利用の諸方策検討用に開発したプログラムをも利用しており、ジョブ・ジョブステップスケジューラによってAPUシステムのジョブ処理を行った場合のシミュレーション実験が可能な構造となっている。

4.1 シミュレータの概要

開発したJPSのプログラム構成は図4.1に示すとおりである。同図においてMAIN、TASK1、TASK2、GNRT、CALT および LPOUT の各ルーチンは、先に開発したものをを用いているので、詳細については文献を参照されたい。各プログラムの制御の受渡しは同図に示した事象の発生を契機として行われる。但し、SCDルーチンはMAINルーチンを介さず、INCONおよびJSTMルーチンの中で呼ばれる形式にしている。以下に、これら事象記号 E^i ($i = 1 \dots 26$)の意味について列記する。

- E^1 : シミュレーション結果出力
- E^2 : ジョブ発生
- E^3 : ジョブ投入
- E^4 : ジョブイニシエート（開設）処理終了
- E^5 : ジョブターミネート（閉設）処理終了
- E^6 : ジョブステップイニシエート（開設）処理終了
- E^7 : ジョブステップターミネート（閉設）処理終了
- E^8 : 最終ジョブステップターミネート（閉設）処理終了
- E^9 : タスクの主記憶への割付
- E^{10} : タスクの主記憶からの解放
- E^{11} : タスクの主記憶へのロールイン
- E^{12} : タスクの主記憶からのロールアウト
- E^{13} : APサービス開始

- E^{14} : CPサービス開始
- E^{15} : チャネルサービス開始
- E^{16} : APサービス終了
- E^{17} : CPサービス終了
- E^{18} : チャネルサービス終了
- E^{19} : ジョブステップ終了
- E^{20} : $ET_1 < ET_2$ のとき発生
- E^{21} : $ET_1 > ET_2$ のとき発生
- 但し、 $ET_1 = \text{Min} \{ E_1 \sim E_8, E_{10} \sim E_{19} \}$
- $ET_2 = \text{Min} \{ E_9, E_{11} \sim E_{18} \}$
- E^{22} : ジョブステップ実行保留処理終了
- E^{23} : ジョブステップ保留解除処理終了
- E^{24} : システム見直し処理終了
- E^{25} : ジョブステップ実行中断処理終了
- E^{26} : ジョブステップ実行再開処理終了

4.2 記号および制御表

次節において各ルーチンの概要を説明するが、その際に必要となる記号の意味および制御表の内容を表4.1～表4.2にまとめて示しておく。

4.3 各プログラムの処理概要

4.3.1 メインルーチン（MAIN）、タスクスケジューリング制御ルーチン（TASK1）

MAINルーチンにはジョブ単位の事象発生毎に制御が渡され、TASK1ルーチンにはタスク単位の事象発生毎に制御が渡る。これらのルーチンでは最も近い将来に発生する事象を決定し、シミュレーション時間をすすめ、各事象発生時間の更新を行ってから、決定された発生事象に対応するルーチンに制御を渡す。詳細については文献(5)を参照。

4.3.2 ジョブ発生ルーチン（GNRT）

GNRTルーチンは航技研ジョブ集団の統計的記述に従って計算機システムに投入されるジョブを発生する。詳細については文献(5)を参照。

4.3.3 ジョブ到着処理ルーチン（INCON）

このルーチンは、ジョブ発生事象が発生するとMAINルーチンからGNRTルーチンに引き続いて呼ばれる。まず発生したジョブがシステムに投入できるか否かを調べる。これは、システムに滞在しているジョブ数が滞在可能ジョブに達していれば（制

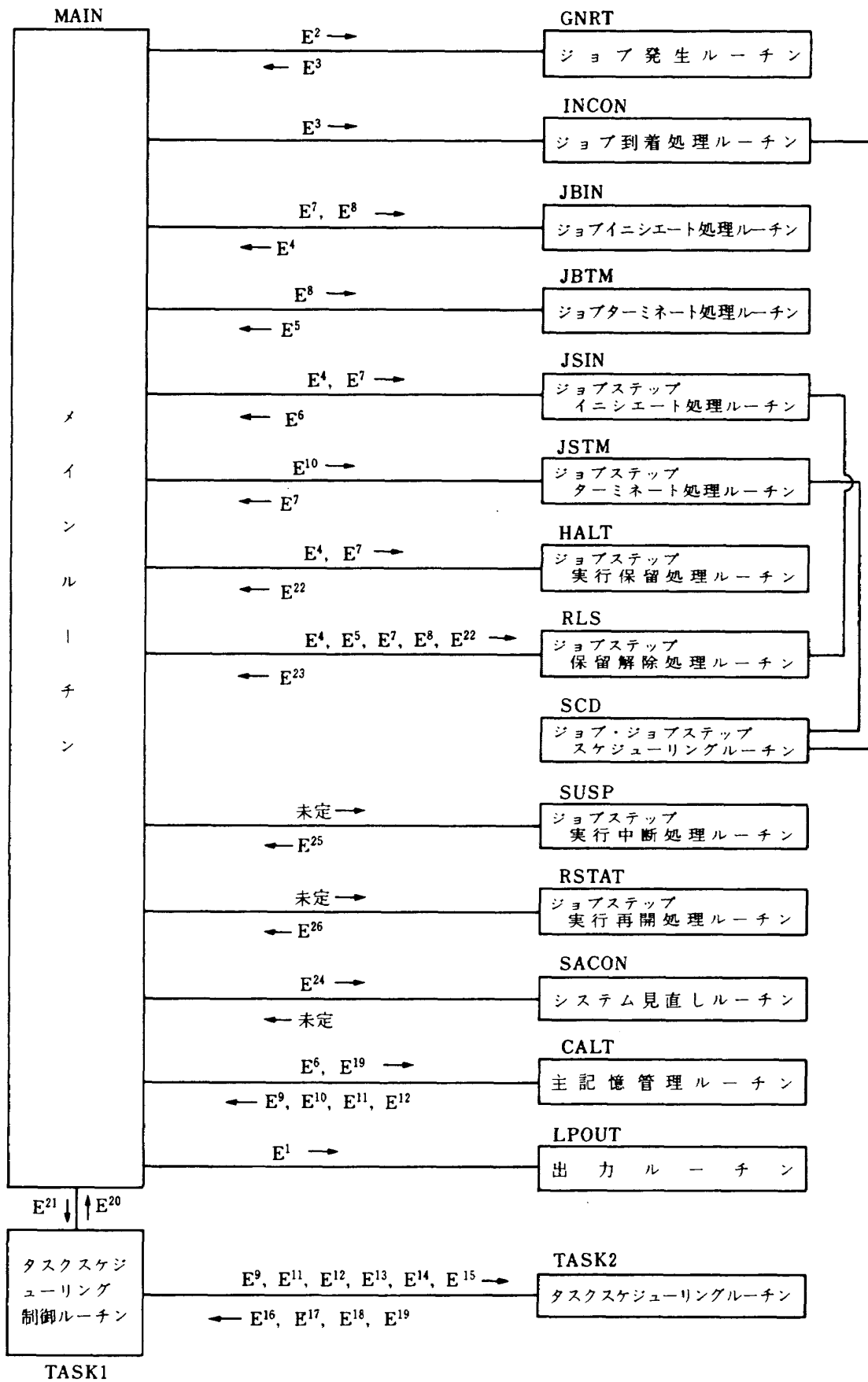


図 4.1 シミュレータのプログラム構成

表 4.1 引 数 と 定 数

(i) サブルーチンの引数と意味

サブルーチン名	引 数 と 意 味	持ち帰り引数と意味
INCON		IT ジョブ到着処理時間 IP=0 ジョブイニシエートなし IP=1 ジョブイニシエートあり
JBIN	IJ: ジョブイニシエートされるジョブ名	
JBTM	IJ: ジョブターミネートされるジョブ名	
JSIN	IJ: ジョブステップイニシエートされるジョブ名	
JSTM	IJ: ジョブステップターミネートされるジョブ名	IT ジョブステップターミネート処理時間
HALT	IJ: 実行保留されるジョブ名	
RLS	IJ: 保留解除されるジョブ名	
SCD	IJ: 事象発生の調整時間	IK スケジューリングするジョブ名 IK=0 スケジューリングするジョブなし

(ii) 定数値の意味

定義名	意 味	定義名	意 味
C(1)	ジョブイニシエート処理時間	IC(1)	ジョブステップ実行保留処理時間
C(2)	ジョブイニシエートオーバーヘッド時間	IC(2)	ジョブステップ実行保留オーバーヘッド
C(3)	ジョブターミネート処理時間	IC(3)	ジョブステップ実行中断処理時間
C(4)	ジョブターミネートオーバーヘッド	IC(4)	ジョブステップ実行中断オーバーヘッド
C(5)	ジョブステップイニシエート処理時間	IC(5)	ジョブステップ保留解除処理時間
C(6)	ジョブステップイニシエートオーバーヘッド	IC(6)	ジョブステップ保留解除オーバーヘッド
C(7)	ジョブステップターミネート処理時間	IC(7)	ジョブステップ実行再開処理時間
C(8)	ジョブステップターミネートオーバーヘッド	IC(8)	ジョブステップ実行再開オーバーヘッド

表 4.2 制 御 表

(i) ジョブ関係制御表

IJA(I) I; システムに滞在可能なジョブ数に対応

1	ジョブ発生番号	ジョブクラス	ジョブステップ数
---	---------	--------	----------

KJOB

1	ジョブ発生番号	ジョブクラス	ジョブステップ
---	---------	--------	---------

KJOBS(J) J; ジョブステップ数に対応

1	プログラム名
2	主記憶使用量
3	CP使用時間
4	AP使用時間
5	平均CPサービス時間
6	平均APサービス時間
7	平均チャネルサービス時間

IJOBQ(I) I; ジョブクラスの数に対応

1	ジョブ個数
2	ジョブ認識番号
3	ジョブ認識番号
4	ジョブ認識番号
5	ジョブ認識番号
6	ジョブ認識番号
7	ジョブ認識番号
...	...
200	ジョブ認識番号

I; システムに滞在可能なジョブ数に対応

IAJOB(I, J) J; ジョブステップ数に対応

1	プログラム名
2	主記憶使用量
3	CP使用時間
4	AP使用時間
5	平均CPサービス時間
6	平均APサービス時間
7	平均チャネルサービス時間

I; プール設定値に対応

HLT(I, J) J; ステップ種に対応

1	ジョブ名
2	実行保留又は実行中断開始時間
3	ジョブ認識番号
4	状態表示(実行保留又は実行中断)
5	状態表示(ロック中)

表 4.2 つづき

(ii) ジョブ統計情報用制御表

IABJ (K) K; ジョブクラスの数に対応

1	処理件数
2	経過時間(a)の和
3	経過時間(a)の2乗和
4	経過時間(b)の和
5	経過時間(b)の2乗和
6	主記憶使用時間の和
7	主記憶使用時間の2乗和
8	ロールアウト時間の和
9	ロールアウト時間の2乗和
10	ロールアウト回数の和
11	ロールアウト回数の2乗和
12	(主記憶量×主記憶時間)の和
13	(主記憶量×主記憶時間)の2乗和
14	ジョブ時間の和
15	ジョブ時間の2乗和
16	CP使用時間の和
17	CP使用時間の2乗和
18	AP使用時間の和
19	AP使用時間の2乗和
20	チャンネル使用時間の和
21	チャンネル使用時間の2乗和
22	(主記憶使用時間/ジョブ時間)の和
23	(主記憶使用時間/ジョブ時間)の2乗和
24	(経過時間(a)/ジョブ時間)の和
25	(経過時間(a)/ジョブ時間)の2乗和
26	(経過時間(b)/ジョブ時間)の和
27	(経過時間(b)/ジョブ時間)の2乗和

I; ジョブステップ個数に対応

MTAB(I, J) J; バッチ多重度に対応

1	実行保留回数
2	実行中断回数
3	実行保留時間
4	実行中断時間
5	プログラム名

TABP(L) L; プログラム名の数に対応

1	処理件数
2	経過時間の和
3	経過時間の2乗和
4	主記憶使用量の和
5	主記憶使用量の2乗和
6	ロールアウト時間の和
7	ロールアウト時間の2乗和
8	ロールアウト回数の和
9	ロールアウト回数の2乗和
10	(主記憶量×主記憶時間)の和
11	(主記憶量×主記憶時間)の2乗和
12	ジョブステップ時間の和
13	ジョブステップ時間の2乗和
14	(経過時間/ジョブステップ時間)の和
15	(経過時間/ジョブステップ時間)の2乗和
16	(主記憶使用時間/ジョブステップ時間)の和
17	(主記憶使用時間/ジョブステップ時間)の2乗和
18	CP使用時間の和
19	CP使用時間の2乗和
20	AP使用時間の和
21	AP使用時間の2乗和
22	チャンネル使用時間の和
23	チャンネル使用時間の2乗和

HOT(I) I; ジョブクラスの数に対応

1	実行保留時間の和
2	実行保留時間の2乗和
3	実行保留回数の和
4	実行保留回数の2乗和
5	実行中断時間の和
6	実行中断時間の2乗和
7	実行中断回数の和
8	実行中断回数の2乗和

表 4.2 つづき

I BOKE(I) I; ジョブクラスの数に対応

1	ボーキングジョブの数
---	------------

(ii) 実行中ジョブ・実行保留中ジョブ関係

I; システムに滞在可能なジョブ数
に対応

NJOB(I, J) J; ジョブステップ数に対応

1	ジョブ名	ジョブステップ 数	ジョブステップ 番号	プログラム名
2	主記憶使用量又は入力カード枚数			
3	CP 使用時間			
4	AP 使用時間			
5	平均 CP サービス時間			
6	平均 AP サービス時間			
7	平均チャンネルサービス時間			
8	ジョブ 優先権	主記憶 優先権	ロールアウト 優先権	実行優先権

I JOBS (M) M; バッチ多重度に対応

1	ジョブ名
2	CP 使用時間
3	AP 使用時間
4	平均 CP サービス時間
5	平均 AP サービス時間
6	平均チャンネルサービス時間
7	実行優先権
8	ジョブの種類(1; CPジョブ, 2; APジョブ)

I JOBC (M) M; バッチ多重度に対応

1	ジョブ名
2	プログラム名
3	主記憶使用量
4	主記憶優先権
5	ロールアウト優先権

I JOBSQ (M) M; バッチ多重度に対応

1	このジョブステップの開設時間
2	このジョブの総ジョブステップ数
3	実行中のジョブステップの順位

M; バッチ多重度に対応

ATAB (M, I) I; ジョブステップに対応

1	経過時間
2	主記憶使用時間
3	ロールアウト時間
4	ロールアウト回数
5	(主記憶量×主記憶時間)
6	ジョブステップ時間
7	プログラム名
8	CP 使用時間
9	AP 使用時間
10	チャンネル使用時間

ITJOB (M) M; バッチ多重度に対応

1	ジョブ到着時間
---	---------

IKJOB (M) M; バッチ多重度に対応

1	ジョブ実行開始時間
---	-----------

I ACT (M) M; バッチ多重度に対応

1	実行中はジョブ名
---	----------

I HALT (M) M; バッチ多重度に対応

1	実行保留中はジョブ名
---	------------

I STP (M) M; バッチ多重度に対応

1	実行中はプログラム名
---	------------

表 4.2 つづき

LOC1

1	ロック中ジョブの個数
2	ロック中ジョブ名
3	ロック中ジョブ名
4	ロック中ジョブ名
5	ロック中ジョブ名

ICNT(M) M; バッチ多重度に対応

1	ジョブ名
---	------

(iv) 多重度関係制御表

SMLT(I) I; ステップ種に対応

1	システム多重度
---	---------

RMLT(I) I; ステップ種に対応

1	実行中多重度
---	--------

(v) その他

LJOB(M) M; バッチ多重度に対応

1	事象名
2	ジョブ名
3	事象発生時間

LOC(I) I; 実行中, 実行保留中ステップ種に対応

1	ロック中ジョブ個数
2	ロック中ジョブ名
3	ロック中ジョブ名
4	ロック中ジョブ名
5	ロック中ジョブ名
6	ロック中ジョブ名

HMLT(I) I; ステップ種に対応

1	プール設定値
---	--------

RHMLT(I) I; ステップ種に対応

1	プール個数
---	-------

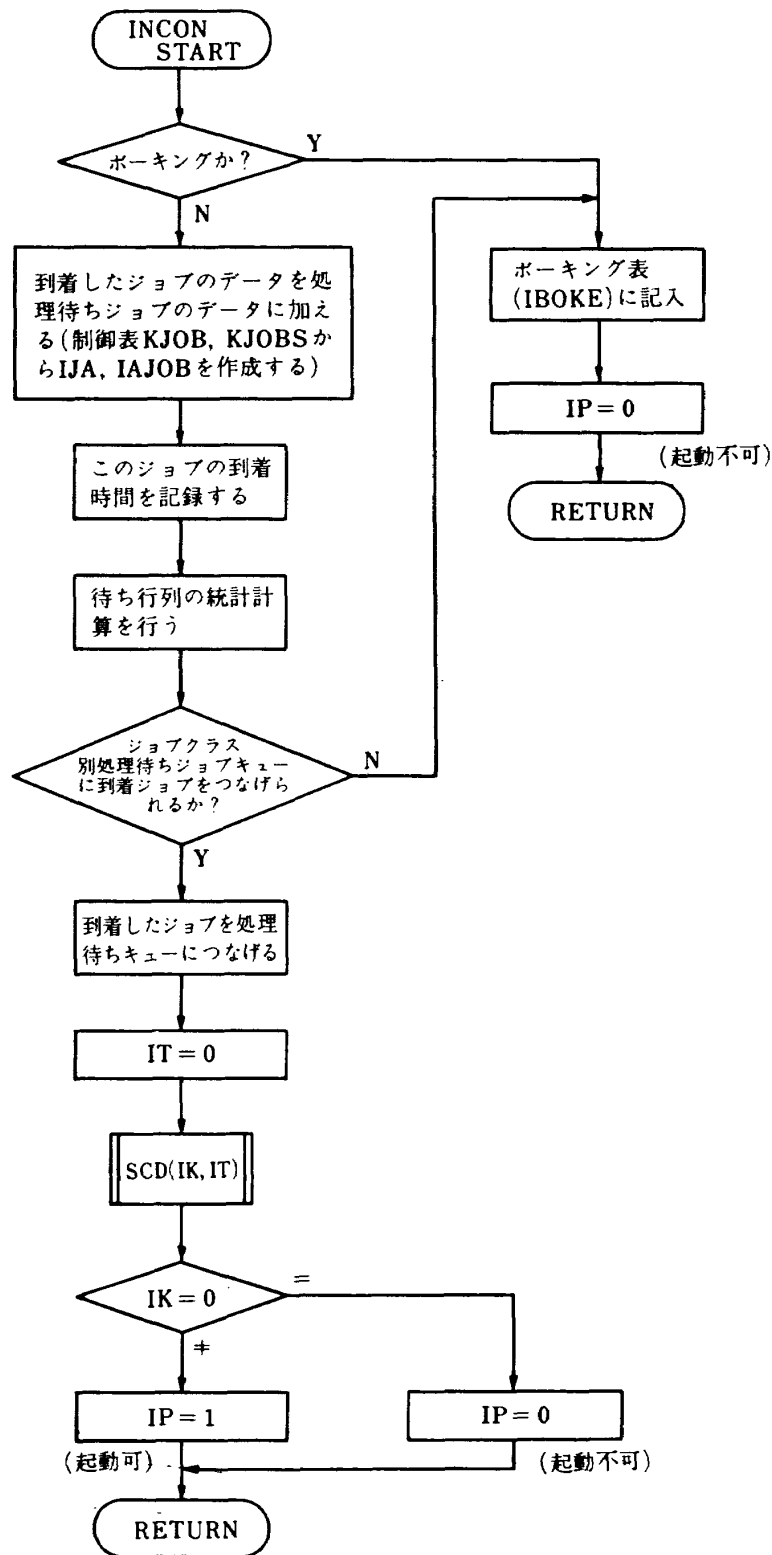


図 4.2 INCONルーチンのゼネラルフロー

御表 IJA に空きが無いとき)ポーキングとなり、INCONは制御表 IBOKE を更新して MAINルーチンに復帰する。又は、滞在ジョブ数が滞在可能ジョブ数に満たない場合は GNRTルーチンで作成された制御表 KJOB, KJOBS から制御表 IJA,

IAJOB を作成する。このとき、このジョブのシステム内におけるジョブ認識番号は制御表 IJA の 1 つの空き配列番号である。次に、制御表 KJOB よりこのジョブのジョブクラスを調べ、ジョブクラス別処理待ちキュー (制御表 IJOBQ) に空きが無い場

合は上記ポーキング処理をしてMAINルーチンに復帰する。空きがある場合は制御表 IJOBQ より、この時点までの待ち行列に関する統計計算を行って、到着したジョブを処理待ちジョブキューにつなげるために制御表 IJOBQ を更新する。次にSCDルーチンを呼び出し、ジョブのスケジューリングを行う。SCDルーチンでは次に述べるJBINルーチンで必要な制御表 ICNT の作成や、ロック処理も行う。(ロック処理については後述。) SCDルーチンのもち帰り引数が $IK=0$ のときはジョブ

起動なしで、INCONルーチンの引数を $IP=0$ としてMAINルーチンに復帰する。 $IK \neq 0$ のときはジョブ起動ありでINCONルーチンの引数を $IP=1$ としてMAINルーチンに復帰する。

4.3.4 ジョブイニシエート処理ルーチン (JBIN)

このルーチンは、ジョブイニシエート事象が発生するとMAINルーチンから呼ばれ、実行中ジョブに関する制御表NJOBを作成する。次に、イニシエートジョブの第一ジョブステップの実行がスケジ

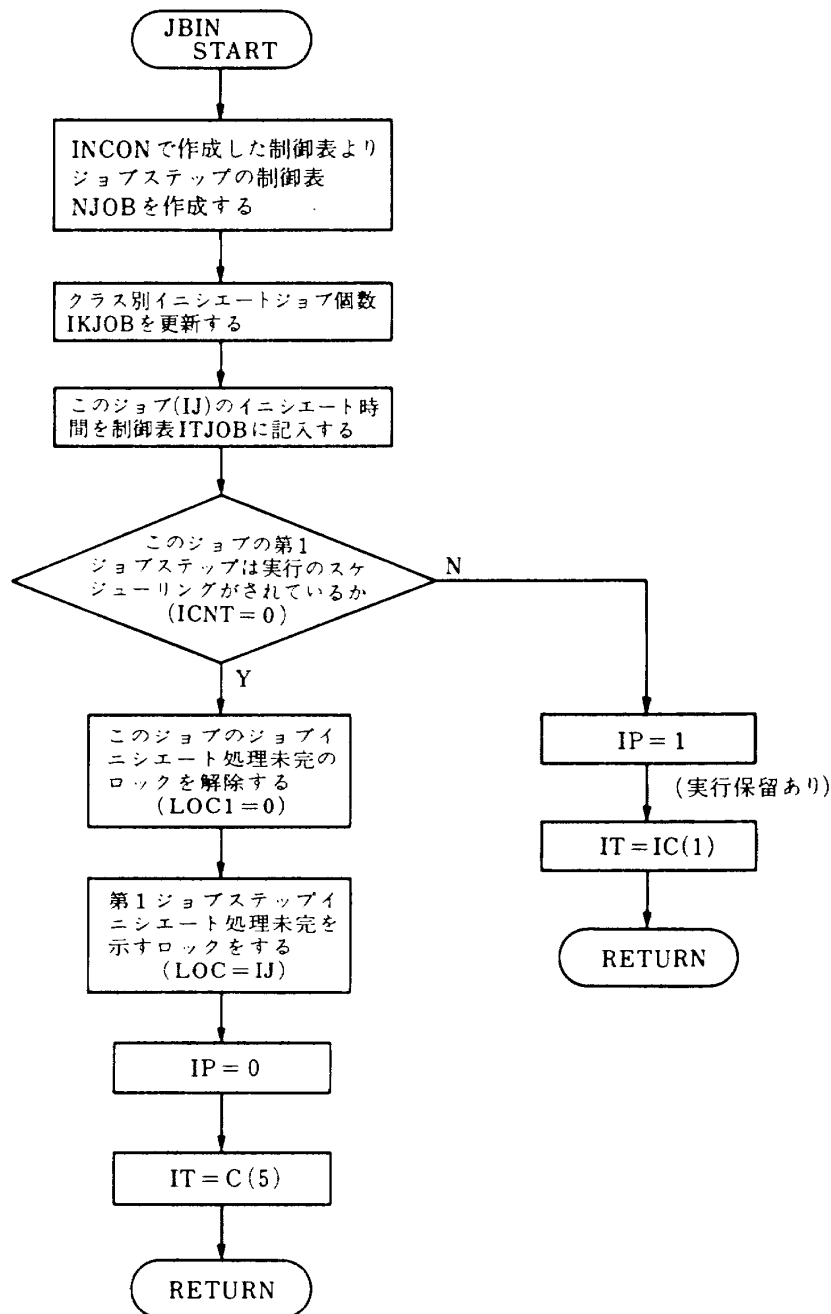


図 4.3 JBINルーチンのゼネラルフロー

ューリングされているか、実行保留がスケジューリングされているかを制御表 ICNT で調べる。実行がスケジューリングされている場合は、このジョブのイニシエート処理未完了を示す制御表 LOC 1 を書き替える。^(注)次に、ジョブステップイニシエート処理未完了を示す制御表 LOC にジョブ名等を書き込んで、引数を IP=0 (ジョブステップイニシエートあり)、IT=C(5) として MAIN ルーチンに復帰する。また、実行保留がスケジューリングされている場合は、それを示す制御表 ICNT を 0 にして、引数を IP=1 (ジョブステップ実行保留あり)、IT=IC(1) にして MAIN ルーチンに復帰する。ここで、C(5) はジョブステップイニシエート処理に要する時間、IC(1) はジョブステップ実行保留処理に要する時間である。なおジョブイニシエート処理に要する時間は、MAIN ルーチンでこのルーチンを呼ぶ前に進めている。

4.3.5 ジョブステップ実行保留処理ルーチン (HALT)

このルーチンは、ジョブステップ実行保留事象が発生すると MAIN ルーチンから呼ばれる。このルーチンにおいては、ジョブステップが実行保留になったことを示す制御表 IHALT にジョブ名を記入する。次に、実行保留プール個数が変わるので、この時点までのプール個数に関する統計データを収集し、その後に実行保留プール個数 (制御表 RHMLT) を 1 ふやす。つづいて、このジョブステップが実行保留処理未完了であることを示していた制御表 LOC を 0 に書き替え、このジョブステップを実行保留ジョブステップキュー (制御表 HLT) につなげ (これよりランジョブステップのプールを管理する)、実行保留開始時間等の情報をこのキューに記入する。最後に MAIN ルーチンに復帰する。

(注) ロック処置について

ジョブイニシエート決定からその処理完了まで、ジョブステップイニシエート決定からその処理完了まで、ジョブステップ実行保留決定からその処理完了まで、ジョブステップの保留解除決定からその処理完了まで、およびジョブステップの実行再開決定からその処理完了までには一定の時間がかかる。このため、ジョブやジョブステップがこれらの状態移行中にはまだ実行多重度、実行保留プール個数が加算されていない状況にあるが、これらの処理が完了すると多重度空きおよびプール個数空きはなくなるわけであり、通常の実行多重度空きとは異なる。この状態移行中に、システム多重度を越えてジョブやジョブステップのスケジューリングが行われないようにするためにロック処置を行う。

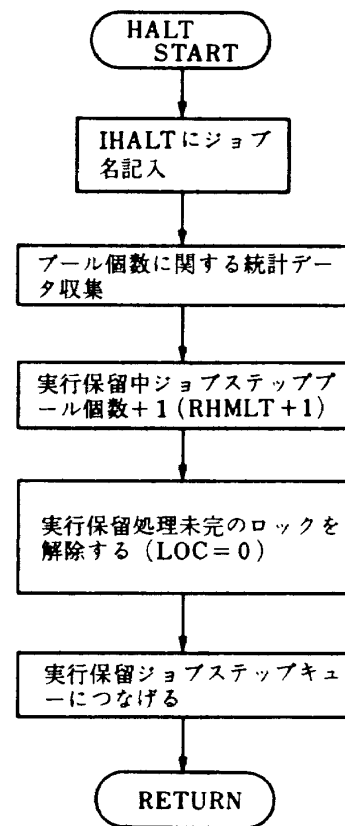


図 4.4 HALT ルーチンのゼネラルフロー

4.3.6 ジョブステップ保留解除処理ルーチン (RLS)

このルーチンは、ジョブステップ保留解除事象が発生すると、MAIN ルーチンから呼ばれる。このルーチンでは、このジョブステップのジョブが保留解除されたことを示すために HALT ルーチンでセットした制御表 IHALT を 0 にする。次に、実行保留プール個数が変わるので、この時点までのプール個数に関する統計データを収集する。その後に実行保留プール個数を 1 減ずる。つづいて、このジョブを実行保留ジョブステップキューよりはずす。また、このジョブステップの実行保留時間等を計算して制

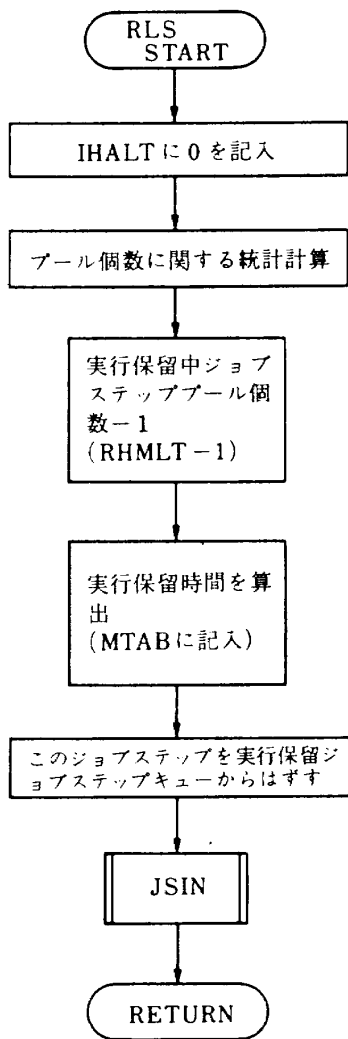


図 4.5 RLS ルーチンのゼネラルフロー

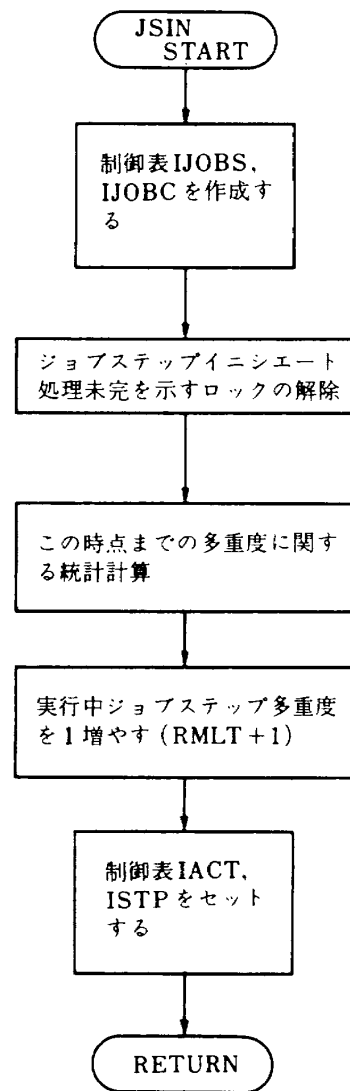


図 4.6 JSIN ルーチンのゼネラルフロー

御表MTABに記入する。次に、ジョブステップ起動のためにJSINルーチン呼び、復帰後このRLSルーチンもMAINルーチンに制御を渡す。

4.3.7 ジョブステップイニシエート処理ルーチン(JSIN)

このルーチンは、ジョブステップイニシエート事象が発生するとMAINルーチンから呼ばれる。このルーチンにおいては、ジョブステップが実行中にCALTルーチンやTASK1ルーチンで使用される制御表IJOBS, IJOBCを初期化し、JBINルーチンでセットしたジョブステップイニシエート処理未完了を示す制御表LOCを0に書き替える。次に実行中多重度が変わるので、この時点までの多重度に関する統計計算を行う。その後、イニシエートされたジョブステップのステップ種別を判定し、それぞれの多重度を1ふやす。つづいて、このジョブステ

ップが実行中であることを示すために、制御表IACTをセットする。また、このジョブステップのプログラム名が何であることを示す制御表ISTPにプログラム名を記入する。ジョブ・ジョブステップスケジューラでは、ジョブがいったんイニシエートされても第一ジョブステップが必ずしも実行に入らず、実行保留状態にもなるので、両者の状態を区別する必要があるためにこれらの制御表を用意した。以上のジョブステップイニシエート処理が終了するとMAINルーチンに復帰する。

4.3.8 主記憶管理ルーチン(CALT)

CALTルーチンは、JSINルーチンに引続いて呼ばれ、ジョブステップに対して主記憶の割付けを試みる。また、このルーチンはJSTMルーチンにも引続いて呼ばれ、ジョブステップが使用していた

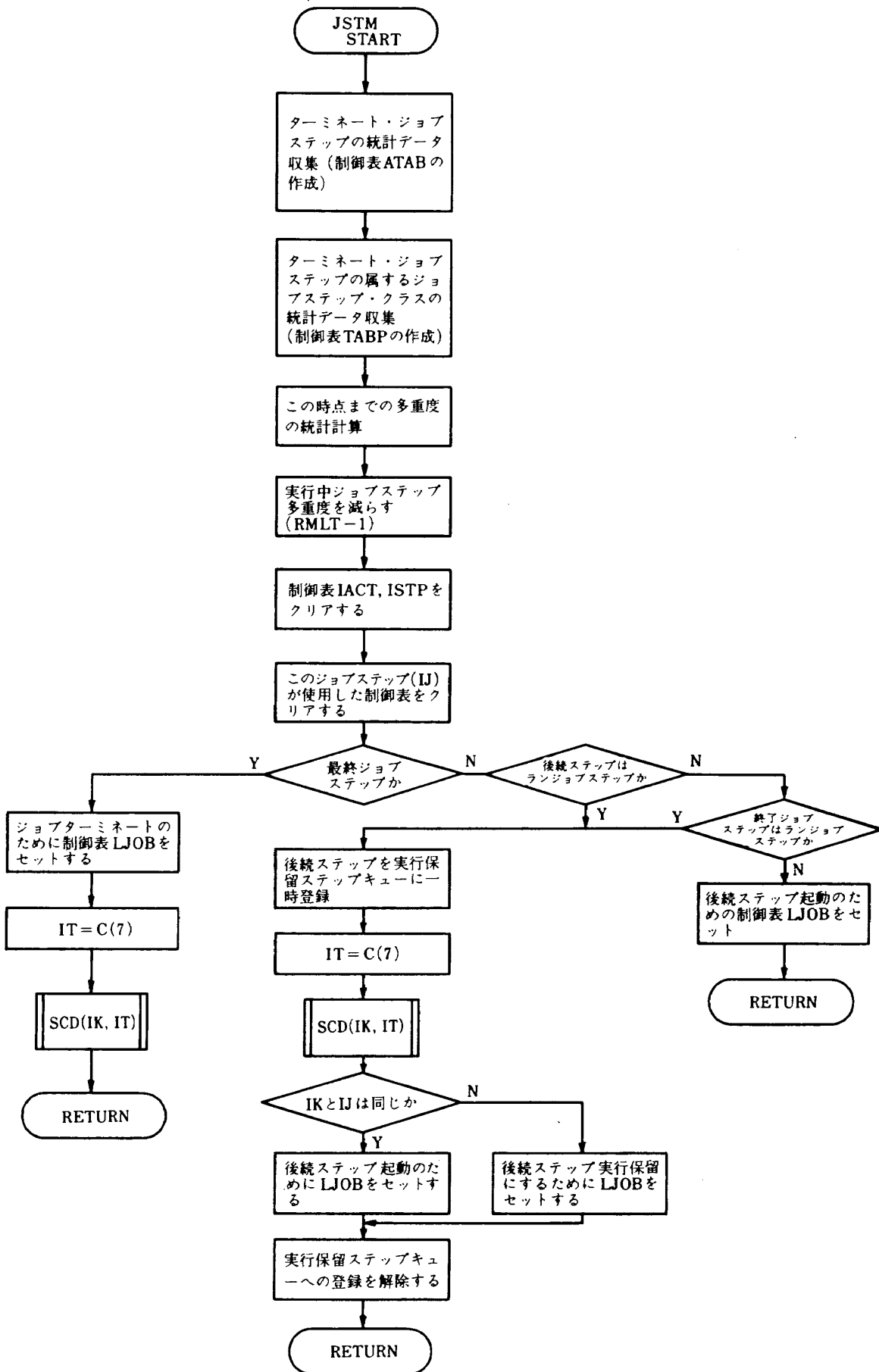


図 4.7 JSTMルーチンのゼネラルフロー

主記憶の解放を行う。主記憶の割付けはそのジョブステップの主記憶要求時と他のジョブステップの主記憶解放時に試みられ、ロールイン・ロールアウトを併用した一定のアルゴリズムに基づいて行われる。主記憶を割付けられたジョブステップはAPまたはCP実行待ちキューに登録される。詳細については文献(4)と(5)を参照。

4.3.9 タスクスケジューリングルーチン (TASK2)

TASK2 ルーチンはAPUまたはCPUの実行が中断される事象(即ち、入出力要求、タイムスライス切れ)発生の都度起動され、APまたはCP実行待ちキューから最も優先権の高いタスクにAPUまたはCPUの使用権を与える。詳細については文献(5)を参照。

4.3.10 ジョブステップターミネート処理ルーチン (JSTM)

このルーチンは、ジョブステップターミネート事象が発生すると、MAINルーチンから呼ばれる。このルーチンにおいては、CALTルーチンとTASK1ルーチンにより収集されたデータの集計を行い、このジョブステップが属するプログラムに関する統計計算を行って、制御表TABPに記入する。つづいてこの時点までの多重度に関する統計計算を行ってから、実行中多重度を減ずる。次に、制御表IACT, ISTEPを0に書き替え、また、このジョブステップが実行中に使用していた制御表もクリアする。その後、このジョブステップが最終ジョブステップであればジョブターミネートを示すために制御表LJOBをセットし、次に何をスケジューリングするかを決定するためにSCDルーチン呼び出して処理を行い、MAINルーチンに復帰する。また、後続のジョブステップがある場合には、このステップ種別を判定し、サービスジョブステップに後続するものがサービスジョブステップであれば、これを次に起動するために制御表LJOBにセットし、MAINルーチンに復帰する。また、サービスジョブステップに後続するものがランジョブステップの場合、または、ランジョブステップに後続するものがサービスジョブステップおよびランジョブステップである場合には、いったん実行保留ジョブステッ

プキュー(制御表HLT)に登録して、SCDルーチン呼びスケジュールリングを行う。復帰後、SCDルーチンで決定したジョブ(IK)と後続ジョブステップのジョブ名が一致する場合は、後続ジョブステップを起動するために制御表LJOBをセットし、先の登録を解除してMAINルーチンに復帰する。一致しない時には、後続ジョブステップは実行保留となるように制御表LJOBをセットし、先に行った登録を解除してMAINルーチンに復帰する。なお、SCDルーチン呼ぶときには、引数ITにジョブステップターミネート処理時間を入れておかなければならない。

4.3.11 ジョブターミネート処理ルーチン (JBTM)

このルーチンは、ジョブターミネート事象が発生するとMAINルーチンから呼ばれる。このルーチンにおいては、処理終了ジョブのジョブ統計を制御表ATAB, MTABより収集し、ジョブクラス毎の統計計算をして制御表IABJ, HOTに記入する。次に、実行中に使用していたこのジョブに関する制御表をゼロクリアし、MAINルーチンに復帰する。

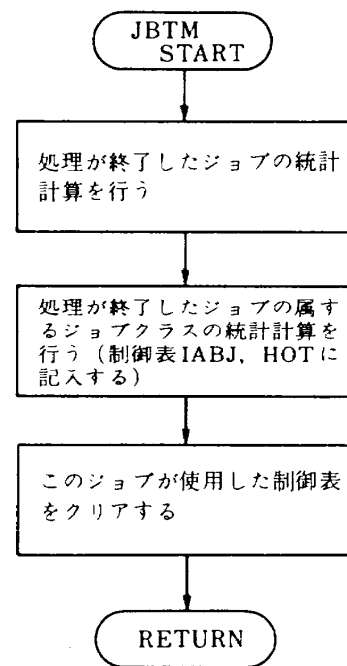


図 4.8 JBTMルーチンのゼネラルフロー

4.3.12 出力ルーチン(LPOUT)

LPOUT ルーチンはシミュレーション過程を記録するためのものであり、一定時間間隔で解析に必要な情報を収集する。詳細については文献(5)を参照。

4.3.13 ジョブ・ジョブステップスケジューリ ングルーチン(SCD)

このルーチンは、INCONおよびJSTMルーチンの中で呼ばれ、ジョブまたはジョブステップの起動をスケジューリングする。アルゴリズムの詳細については第3章で述べた。起動するジョブまたはジョブステップが決定すると、そのことを示すために制御表 LJOB をセットする。その内容は決定されたジョブ名、このジョブに関する発生事象およびそれが発生するまでの時間である。なお、この時間は SCD ルーチンの引数 IT にジョブイニシエートのときには C(1) を、ジョブステップ保留解除のときには IC(5) を、ジョブステップイニシエートのときには C(5) を加えたものとする。引数 IT はこのルーチンが JSTM ルーチンから呼ばれた場合にはジョブステップターミネート処理に要する時間を意味し、INCON から呼ばれた場合には 0 を意味する。また、起動ジョブの第一ジョブステップを実行するか実行保留にするかを制御表 ICNT にセットし、実行保留の場合には、多重度をうめるための準備としてふたたびスケジューリングを行う。さらに、ジョブまたはジョブステップの起動に際して必要となるロック処置を行ってから呼び出し元ルーチンに復帰する。^(注)

4.3.14 システム見直しルーチン(SACON)、 ジョブステップ実行中断処理ルーチン (SUSP)、ジョブステップ実行再開処 理ルーチン(RSTAT)

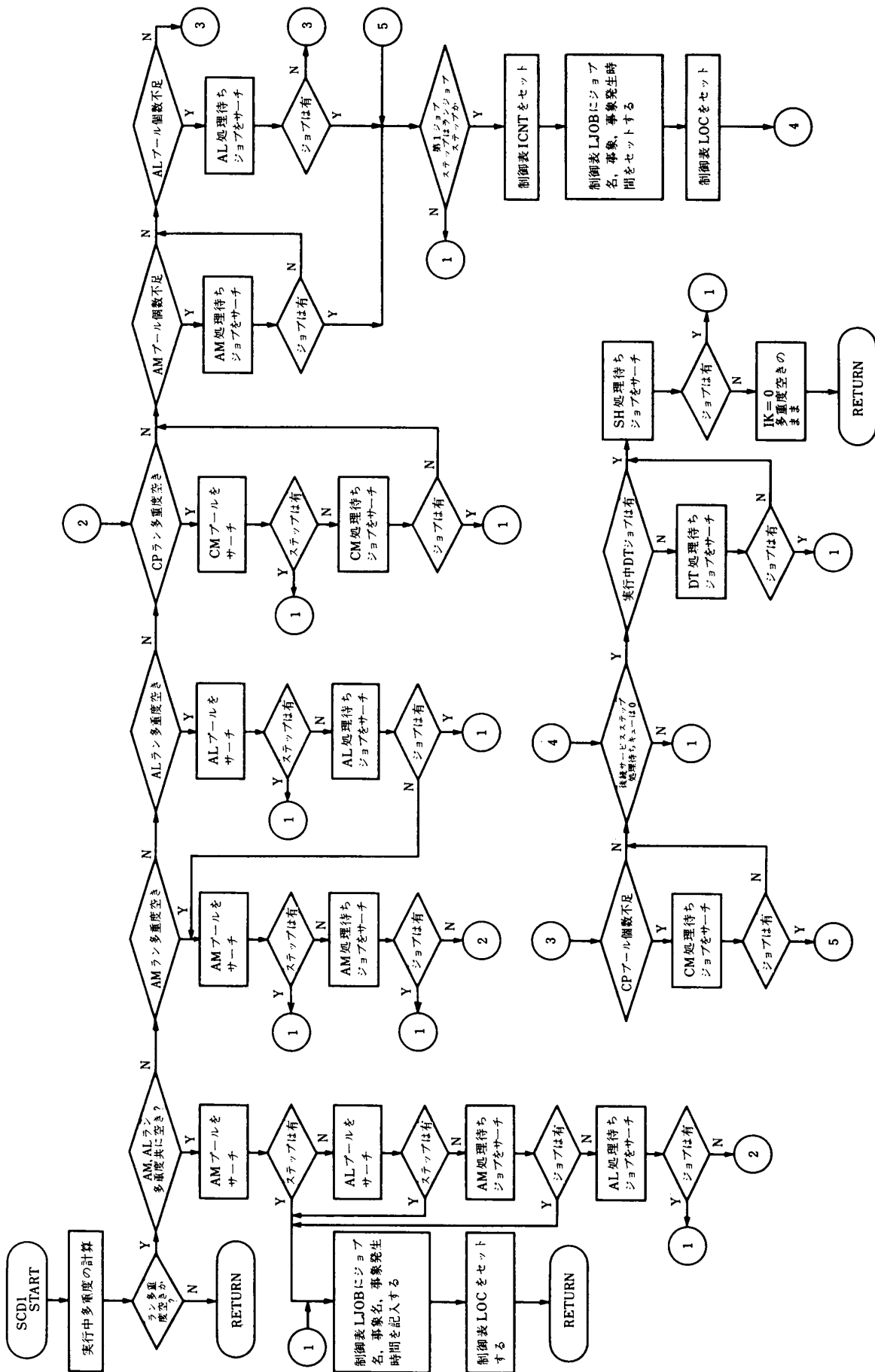
本稿ではこれら三つのルーチンを用いた実験を行っていない。従って、SACON ルーチンも用意していないが、これらを用いて次のような機能の実験が可能である。

SACON ルーチンは使用目的毎に異なるが、システム状況を計測し、そのデータを解析した結果に基づいて、ジョブステップスケジューラの設定値を適切値に変更したり、または全く別のスケジューリングアルゴリズムへの切換を行うことを可能とする機能を受けもつ。その際、実行中のジョブステップを中断したり、再開したりしえることが要求されよう。

SUSP ルーチンは、ジョブステップ実行中断事象が発生すると MAIN ルーチンから呼ばれる。このルーチンにおいては、まず、このジョブステップのジョブが実行中断中であることを示すために制御表 ISUSP をセットする。次に実行中ジョブステップ多重度が変わるので、この時点までの多重度に関する統計データを収集してから実行中ジョブステップ多重度を 1 減ずる。また、このジョブステップの属するクラスの実行中断ジョブステップキューにつなぎ、実行中断時間収集のために実行中断時間を制御表 HLT に記入した後、MAIN ルーチンに復帰する。

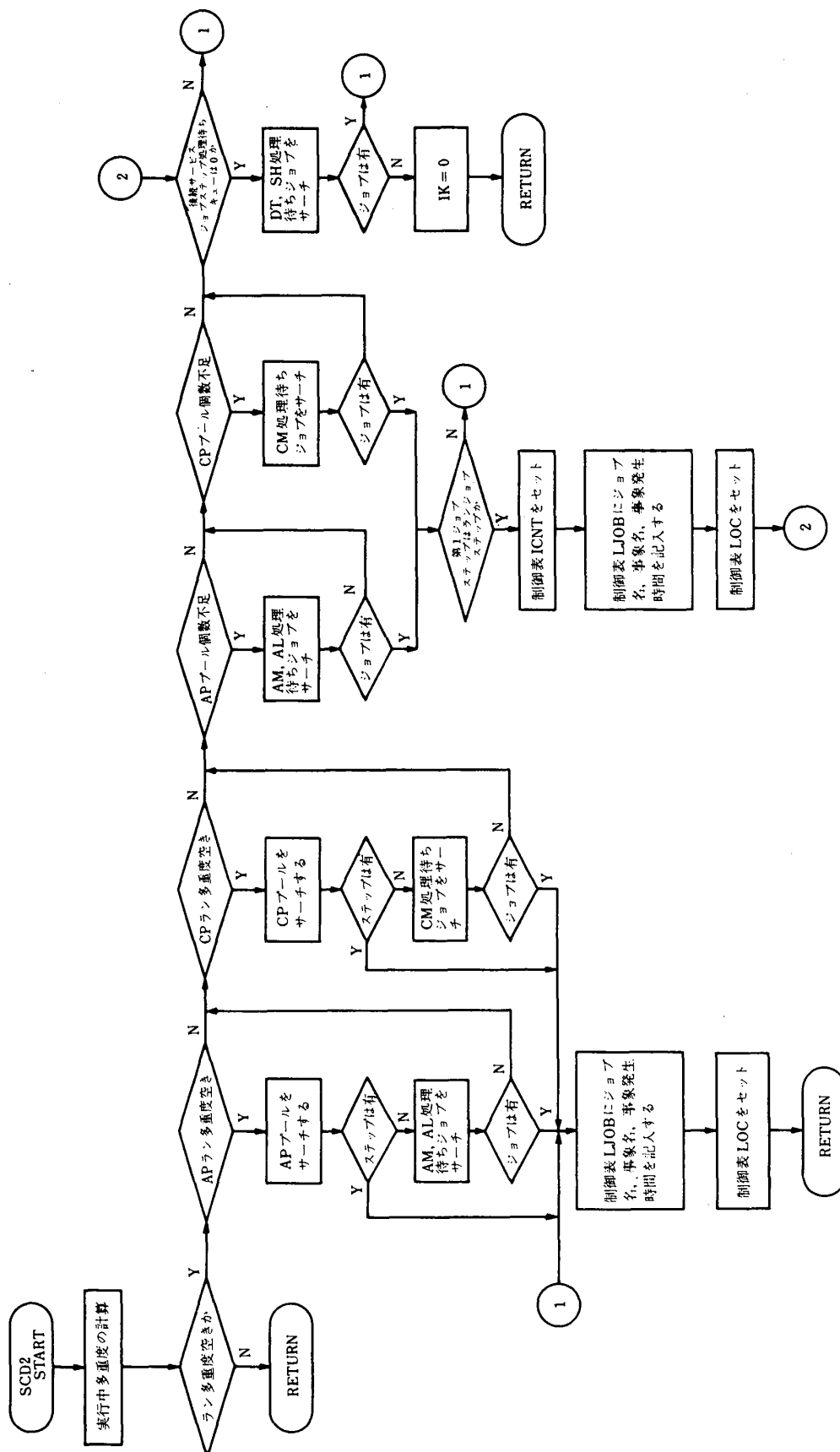
RSTAT ルーチンはジョブステップ実行再開事象が発生すると MAIN ルーチンから呼ばれ、このジョブステップの実行中断解除の処理を行う。まず、このジョブステップが実行中断解除されたことを示すために制御表 ISUSP をセットする。次に、実行中ジョブステップ多重度が変わるので、この時点までの多重度に関する統計データを収集し、実行中断ジョブステップキューよりこのジョブステップをはずした後、実行中ジョブステップ多重度を 1 ふやす。そしてこのジョブステップが実行を中断していた時間を算出して制御表 MTAB に記入し、MAIN ルーチンに復帰する。

(注) 図 4.9 において、AM, AL, CM, AP, CP 各プールをサーチする場合、制御表 RHMLT のみに基づき、実行保留処理未完(LOC中)のジョブステップ件数を無視してプール個数の補充を行っている。このため、各プール個数は設定値以上になりうる。



(i) SCD 1

図 4.9 SCDルーチンのゼネラルフロー



(ii) SCD 2

図 4.9 つづき

5. シミュレーション実験

試作ジョブ・ジョブステップスケジューラによる APU システムのジョブ処理効果を評価するために、下記の実験条件のもとでシミュレーション実験を行った。

5.1 実験環境の設定

航技研ジョブ集団を模擬するために必要な諸定数は昭和53年度アカウント情報の解析結果得られた統計値を用いた(表3.1~3.4)。ジョブ集団の記述手法については文献(4)を参照されたい。ジョブはポアソン過程に従ってこの集団からシステムに投入されるが、処理すべきジョブが常に存在している状況を実現する目的から、特に明記しない限り、ジョブの平均到着時間間隔は10秒に設定した。また、シミュレーション時間は実時間にして20時間で打ち切り、システムの平均的利用状況を表わす各種統計量の計算を行った。また、ラッシュアワー実験におけるジョブ個数は100件とし、上記ジョブ集団からランダムに取出した。^(注)

5.2 実験項目

本実験の目的は第一に新旧スケジューラによる APU システムのジョブ処理効果の比較、第二に試作した二つの新スケジューラの効果の比較、第三に新スケジューラに対して適切な運用系の決定とにある。これらの検討のために、表5.1~5.2に示した一連の実験を計画した。表5.1における主記憶管理ルーチン1と2はそれぞれ図5.1の(i)と(ii)の如き処理を行うプログラムであるが、アロケータ1,2およびリロケータ1については文献(4)を参照されたい。また、実験は主記憶容量を768K語(内、200K語はOSが使用)とし、CNPモードによる割付けによって主記憶上に256K語境界を生ずるものとして⁴⁾いる。

(i) 実験 I

この実験は主記憶競合がほとんど起らない状況のもとで、試作スケジューラ SCD1 による APU システムのジョブ処理状況を調べるのが目的である。そのような主記憶環境を模擬するため、文献(4)のアロケータ2とリロケータ1により主記憶の有効利用を最大限に計った主記憶管理ルーチン1を使用した。運用系としては表5.2(i)に示した種々の場合を検討する。同表における運用系0は旧スケジューラに関するもの

表 5.1 実験の種類^(注)

実験番号		使用ルーチン
新 ス ケ ジ ュー ラ	I	新スケジューラ SCD1 主記憶管理ルーチン1 アロケータ2 リロケータ1
	II	新スケジューラ SCD1 主記憶管理ルーチン2 アロケータ1
	III	新スケジューラ SCD2 主記憶管理ルーチン1 アロケータ2 リロケータ1
旧 ス ケ ジ ュー ラ	RI	主記憶管理ルーチン1 アロケータ2 リロケータ1
	RII	主記憶管理ルーチン2 アロケータ1

(注) 次の条件は全て共通
 ・ 主記憶容量は768Kw
 ・ 256境界有り

(注) ラッシュアワー実験とは各ジョブクラスの処理待ちキューに並んでいる多数のジョブの処理を完了するまでの処理状況の推移を調べる実験のことであって、その際新たなジョブ到着はないものとしている。

表 5.2 運用系の種類

(i) 実験 I および II (注)

運用系 番号	内 容	
	多 重 度	優 先 権
0	SH = 2, DT = CM = AM = AL = 1	主記憶; SY = AM = AL > CM > DT > SH 実 行; SY = AM = AL > SH = DT > CM
1	IO 多重度 = 3 (内; DT = 1) ラン多重度 = 3 (CM = AM = AL = 1)	主記憶; SY = AM = AL > CM > DT > SH 実 行; SY = AM = AL > SH = DT > CM
2		主記憶; SY > AM = AL > CM > DT > SH 実 行; SY = AM = AL > SH = DT > CM
3		主記憶; SY > AM = AL > CM > DT > SH 実 行; SY > AM = AL > SH = DT > CM
4		主記憶; SY = AM = AL > CM > SH > DT 実 行; SY = AM = AL > DT > SH > CM
5	IO 多重度 = 2 (内; DT = 1) ラン多重度 = 3 (CM = AM = AL = 1)	主記憶; SY = AM = AL > CM > DT > SH 実 行; SY = AM = AL > SH > DT > CM
6	IO 多重度 = 3 (内; DT = 1) ラン多重度 = 3 (CM = AM = AL = 1)	
7	IO 多重度 = 4 (内; DT = 1) ラン多重度 = 3 (CM = AM = AL = 1)	
8	IO 多重度 = 5 (内; DT = 1) ラン多重度 = 3 (CM = AM = AL = 1)	

(注) 次の条件は全て共通

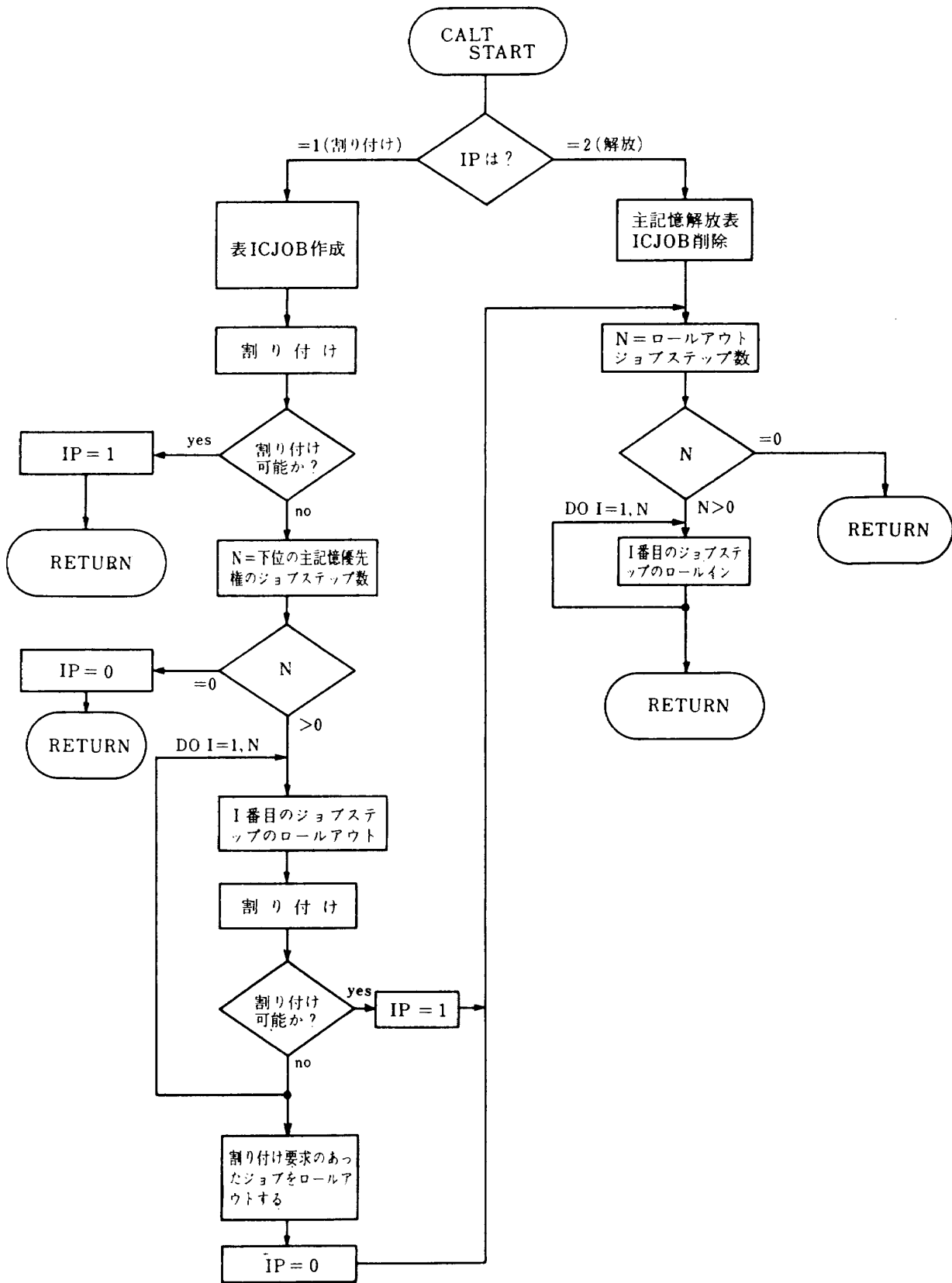
- CM, AM, AL 各プール個数設定値 = 3

(ii) 実験 III (注)

運用系番号	平均到着時間 間 隔	経過時間の重み 係 数
1	10 秒	$\begin{cases} SH = 1 \\ DT = 0.5 \\ AM = 1 \\ AL = 0.5 \end{cases}$
2	20 秒	
3	30 秒	
4	10 秒	$\begin{cases} SH = 1 \\ DT = 1 \\ AM = 1 \\ AL = 1 \end{cases}$
5	20 秒	
6	30 秒	
7	10 秒	$\begin{cases} SH = 1 \\ DT = 2 \\ AM = 1 \\ AL = 2 \end{cases}$
8	20 秒	
9	30 秒	

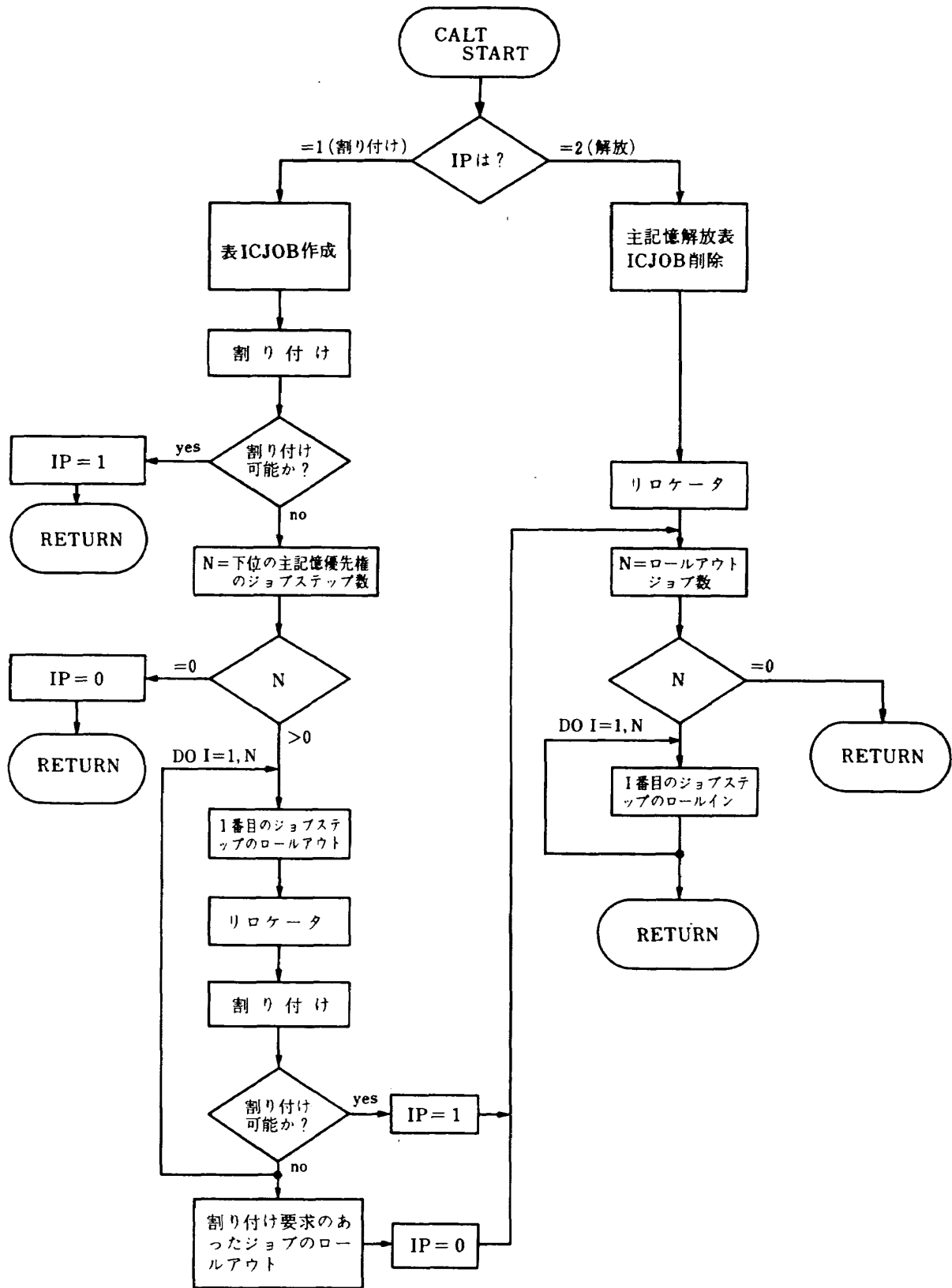
(注) 次の条件は全て共通

- 主記憶優先権
SY = AM = AL > CM > DT > SH
- 実行優先権
SY = AM = AL > SH = DT > CM
- IO 多重度 3
- ラン多重度 3 (CP = 1, AP = 2)
- CP, AP プール個数設定値 = 3



(1) ルーチン2

図 5.1 主記憶管理ルーチンのゼネラルフロー



(2) ルーチン 1

図 5.1 つづき

であり、同じ主記憶管理ルーチンのもとで最も適切と文献(4)で結論された運用系を表わしている。運用系1～8は新スケジューラに関するものである。運用系1は主記憶、実行優先権と多重度の合計を運用系0と同じに設定し、新旧スケジューラによるAPUシステムのジョブ処理状況の差異を明確にするためのものである。運用系2はサービスジョブステップの主記憶優先権を、運用系3はサービスジョブステップの主記憶、実行優先権共に、運用系1より上位に設定した場合の効果を調べるためのものである。運用系4と運用系6は運用系1に比して、SHとDTクラスの主記憶、実行優先権の組合せ方を変えた場合の効果を見るためのものである。また、運用系5～8は主記憶、実行優先権を同一にしているから、IO多重度の効果を調べることができる。従って、実験RI-0(運用系0による実験RIの意、以下同様)と実験I-1～8との結果に基づいて新・旧スケジューラによるAPUシステムのジョブ処理状況が比較できる。

(ii) 実験Ⅱ

この実験は主記憶競合がかなりの頻度で発生する状況のもとで、試作スケジューラSCD1によるAPUシステムのジョブ処理状況を調べるのが目的である。そのような主記憶環境を模擬するため、文献(4)のアロケータ1による主記憶管理ルーチン2を使用した。検討する運用系については実験Ⅰの場合と同じである。実験RII-0と実験II-1～8との結果に基づいて、新・旧スケジューラによるAPUシステムのジョブ処理状況が比較できる。

(ii) 実験Ⅲ

この実験は主記憶競合がほとんど起らない状況のもとで、試作スケジューラSCD2によるAPUシステムのジョブ処理状況を調べるのが目的である。そのような主記憶環境を模擬するため、実験Ⅰと同じ主記憶管理ルーチン1を使用した。運用系としては表5.2の(ii)に示した種々の場合を検討する。同表に示すように、主要な運用条件は表5.2(i)の運用系1と同一に設定

している。SCD2では起動に際してのジョブまたはジョブステップの選定基準を重み付き経過時間としているから、実験Ⅲの運用系の差異は重み係数においている。また、SCD2によるジョブ処理件数はジョブの混みぐあいに著るしく依存してくるから、非常に混在している状況を平均到着時間=10秒によって模擬し、それよりいくらか緩和した状況を20秒と30秒によって模擬する。表5.2(ii)の運用系1～3はSHとAMにDTとALよりそれぞれ2倍の重み付けをした場合を、運用系4～6は特に重み付けをしない場合を、運用系7～9はDTとALにSHとAMよりそれぞれ2倍の重み付けをした場合の効果を調べるためのものである。試作した二つの新スケジューラSCD1とSCD2の効果は実験I-1と実験Ⅲ-1,4,7との結果に基づいて比較できる。また、実験RI-0と実験Ⅲ-1～9の結果に基づき、新・旧スケジューラによるAPUシステムのジョブ処理状況が比較できる。

(iv) ラッシュアワー実験

この実験はジョブ集団からランダムに100件のジョブを取出し、取出した順番にそれぞれの属するジョブクラスの処理待ちキューに全て登録し終えてから、新たなジョブの到着はないものとして、処理待ちジョブ全ての処理が完了するまでの処理状況を実験ⅠとⅡのいくつかの運用系について比較検討することを目的とする。本稿では、このようにして取出したジョブの流れの一部分をジョブストリーム1,2と番号をつけて区別している。当然ながら、ラッシュアワー実験はスケジューラSCD2に対しては無意味である。

5.3 実験結果

シミュレーション実験結果は表5.3～5.5に示す通りである。同表の各項目の数値は全て20時間にわたって収集されたデータの統計量を表わしている。以下にいくつかの項目について多少の説明を加える。

(1) 本実験ではOSの走行時間を考慮していないから、APU,CPU共に100%ユーザジョブ処理に

表 5.3 シミュレーション実験結果 (実験 I)

実験番号	RI - 0	I - 1	I - 2	I - 3	I - 4	I - 5	I - 6	I - 7	I - 8
項目									
平均 AP 稼働率 (%)	94.47	99.30	98.80	98.23	98.71	99.26	99.31	98.28	99.31
平均 CP 稼働率 (%)	93.34	97.70	97.47	98.40	98.21	97.20	98.35	97.65	97.94
平均主記憶利用率 (%)	80.60	81.11	83.58	79.60	83.21	77.32	80.79	84.86	87.46
AP ジョブステップの 平均値	1.64	1.98	1.96	1.98	1.96	1.98	1.98	1.95	1.98
主記憶多重度 標準偏差	0.54	0.15	0.23	0.16	0.22	0.17	0.15	0.27	0.15
CP ジョブステップの 平均値	3.60	3.37	3.26	3.51	3.50	2.72	3.35	3.77	4.02
主記憶多重度 標準偏差	1.00	0.74	0.79	0.67	0.73	0.51	0.75	1.01	1.17
AP ジョブステップの 平均値	0.00	0.001	0.021	0.003	0.007	0.004	0.003	0.007	0.004
ロールアウト件数 標準偏差	0.00	0.022	0.143	0.059	0.08	0.06	0.05	0.084	0.062
CP ジョブステップの 平均値	0.46	0.47	0.59	0.33	0.34	0.18	0.49	1.02	1.73
ロールアウト件数 標準偏差	0.70	0.68	0.74	0.58	0.61	0.41	0.70	0.95	1.18
ジョブステップ総ロールアウト件数	917	586	1,069	712	860	134	560	1,793	3,126
空領域の平均サイズ (K語/個)	78.64	82.5	70.5	90.3	82.55	124.6	88.0	57.14	44.9
空領域の平均個数 (個)	1.48	1.4	1.4	1.41	1.24	1.07	1.34	1.57	1.59
リロケーション個数 (個)	5,410	3,752	4,198	4,068	4,389	2,246	3,824	5,739	7,222
リロケーションによる移動領域 (M語)	197.10	155.7	178.4	170.2	180.6	97.2	152.1	243.7	307.8
平均プール個数	AM	3.21	3.24	3.23	3.16	2.76	3.19	3.48	3.55
	AL	3.51	3.49	3.66	3.23	3.09	3.77	3.61	3.66
	CM	2.90	3.42	3.26	3.27	2.49	3.07	3.61	3.47
平均実行保留時間 (分/件)	AM	16.81	20.52	18.37	19.55	16.82	19.50	19.78	22.32
	AL	97.50	67.85	76.20	59.00	58.85	95.85	84.30	80.25
	CM	19.12	24.58	25.57	27.08	17.95	22.20	30.65	29.55
AP 寄与率	AM	40.04	50.46	48.90	49.80	49.56	48.78	47.91	49.48
	AL	54.43	48.34	49.33	48.91	47.70	50.53	50.37	49.83
合計		94.47	98.80	98.23	98.71	99.26	99.31	98.28	99.31
SH		32.52	18.62	19.08	21.31	7.56	20.05	27.72	35.98
DT		2.68	3.82	3.72	3.03	3.80	2.57	3.64	1.57

C P 寄与率	CM	49.51	67.40	67.64	66.06	64.43	77.08	67.99	58.59	51.98
	AM	6.31	6.89	5.65	7.74	6.47	6.71	5.96	5.91	6.20
	AL	2.32	1.36	1.74	1.80	2.97	2.05	1.78	1.79	2.21
平均ロールアウト時間 (秒/件)	合計	93.34	97.69	97.47	98.40	98.21	97.20	98.35	97.65	97.94
	SH	22.10	22.40	42.70	24.40	1.57	1.43	23.76	46.50	71.60
	DT	66.20	76.00	95.70	43.10	249.10	13.51	82.59	79.20	475.7
	CM	4.24	7.11	15.87	7.65	10.27	2.96	2.35	24.00	25.80
	AM	0.00	32.16	1.29	0.35	28.20	23.09	48.91	39.00	23.70
	AL	0.00	129.80	21.72	2.96	9.79	103.40	53.25	69.90	0.00
	SH	0.49	0.39	0.74	0.49	0.05	0.01	0.32	1.07	1.65
	DT	4.13	3.51	4.45	2.69	11.57	0.58	4.42	4.00	19.33
	CM	0.22	0.40	0.68	0.51	4.30	0.39	0.24	1.15	1.34
	AM	0.00	0.04	0.02	0.02	0.03	0.06	0.07	0.06	0.08
平均ロールアウト件数 (件/ジョブ)	AL	0.00	0.05	0.28	0.13	0.10	0.11	0.04	0.12	0.00
	SH	0.59	0.54	0.55	0.53	0.53	0.58	0.55	0.54	0.53
	DT	0.20	0.17	0.18	0.16	0.17	0.19	0.17	0.17	0.17
	CM	0.39	0.35	0.39	0.34	0.36	0.33	0.29	0.32	0.23
	AM	0.15	0.13	0.09	0.09	0.08	0.07	0.11	0.15	0.08
	SH	0.64	0.66	0.67	0.65	0.64	0.72	0.66	0.60	0.57
	DT	0.16	0.16	0.16	0.16	0.15	0.17	0.16	0.17	0.19
	CM	0.56	0.49	0.52	0.49	0.50	0.50	0.50	0.50	0.49
	AM	0.13	0.08	0.10	0.08	0.10	0.07	0.07	0.09	0.08
	AL	0.61	0.51	0.50	0.48	0.51	0.49	0.50	0.51	0.50
経過時間指数 (上段; 平均値) (下段; 標準偏差)	AL	0.12	0.03	0.04	0.06	0.05	0.05	0.05	0.05	0.05
	SH	4.30	3.75	5.31	3.97	2.94	2.62	3.94	5.00	6.22
	DT	14.20	5.52	10.73	6.83	0.22	1.39	8.79	8.68	11.15
	CM	3.29	3.57	3.47	3.67	3.67	3.60	3.95	4.60	5.67
	AM	11.00	1.17	0.99	1.24	1.60	1.41	0.94	2.74	1.67
	SH									
	DT									
	CM									
	AM									
	AL									

表5.4 シミュレーション実験結果(実験Ⅱ)

項目	実験番号	RⅡ-0	Ⅱ-1	Ⅱ-2	Ⅱ-3	Ⅱ-4	Ⅱ-5	Ⅱ-6	Ⅱ-7	Ⅱ-8
平均AP稼働率(%)		90.77	97.21	95.72	94.17	95.12	97.54	97.70	96.92	94.12
平均CP稼働率(%)		94.76	96.05	97.25	96.20	96.67	95.41	96.29	96.91	96.61
平均主記憶利用率(%)		74.87	73.53	76.01	71.85	72.98	71.45	74.91	77.69	75.03
APジョブステップの 主記憶多重度	平均値	1.51	1.76	1.68	1.60	1.66	1.85	1.79	1.69	1.60
	標準偏差	0.61	0.44	0.49	0.51	0.52	0.39	0.42	0.47	0.54
CPジョブステップの 主記憶多重度	平均値	3.72	3.04	3.41	3.36	3.28	3.53	3.24	3.60	3.68
	標準偏差	1.01	0.82	0.75	0.81	0.75	0.69	0.84	0.98	1.10
APジョブステップの ロールアウト件数	平均値	0.01	0.22	0.30	0.38	0.31	0.11	0.19	0.29	0.37
	標準偏差	0.12	0.42	0.46	0.49	0.47	0.31	0.39	0.46	0.51
CPジョブステップの ロールアウト件数	平均値	0.47	0.81	0.42	0.46	0.57	0.40	0.58	1.19	2.10
	標準偏差	0.66	0.79	0.66	0.47	0.64	0.60	0.79	0.95	1.12
ジョブステップ総ロールアウト件数		865	676	862	689	658	347	845	1,263	1,231
空領域の平均サイズ (K語/個)		30.06	41.80	33.90	39.60	39.20	45.70	33.00	29.49	36.18
空領域の平均個数 (個)		4.96	4.00	4.30	4.50	4.30	3.86	4.60	4.56	4.70
平均プール個数	AM	—	3.05	3.22	3.27	3.14	2.72	3.22	3.47	3.19
	AL	—	3.34	3.29	3.26	3.41	3.16	3.33	3.38	3.57
	CM	—	3.05	3.20	3.46	3.08	2.18	3.14	3.28	3.52
平均実行保留時間 (分/件)	AM	—	17.77	15.30	15.70	16.43	16.72	19.08	21.60	16.08
	AL	—	51.25	76.98	110.48	75.28	65.33	63.97	86.92	110.05
	CM	—	21.75	23.68	26.78	23.17	13.70	20.85	34.57	31.50
AP 寄与率	AM	33.69	47.53	55.63	57.38	50.99	46.77	46.37	44.63	55.59
	AL	57.08	49.68	40.09	36.79	44.13	50.77	51.33	52.29	38.53
合計		90.77	97.21	95.72	94.17	95.12	97.54	97.70	96.92	94.12
SH		34.82	15.42	21.46	20.57	19.25	5.97	21.52	28.45	29.65
DT		3.95	2.69	1.93	3.49	3.06	3.48	3.93	2.62	2.19

C P 寄与率	CM	48.81	68.79	64.53	63.98	64.55	67.46	62.35	58.97	56.82
	AM	5.30	7.18	7.49	6.98	6.91	6.50	6.79	4.33	6.86
	AL	1.88	1.97	1.84	1.18	1.90	3.00	1.70	2.54	1.09
	合 計	94.76	96.05	97.25	96.20	95.67	95.41	96.29	96.91	96.61
平均ロールアウト時間 (秒/件)	SH	15.06	28.50	16.60	21.40	4.17	20.00	19.70	21.80	37.86
	DT	64.82	121.00	227.70	92.20	196.10	67.30	93.20	139.70	258.50
	CM	10.80	15.50	17.20	23.40	38.09	12.10	24.20	36.50	53.19
平均ロールアウト件数 (件/ジョブ)	AM	53.50	137.20	15.50	39.40	109.10	94.10	80.30	283.90	319.70
	AL	63.95	211.20	325.50	749.80	244.00	60.60	175.40	317.80	948.90
	SH	0.41	0.35	0.38	0.36	0.10	0.16	0.40	0.64	0.97
有効主記憶時間比 (上段; 平均値) (下段; 標準偏差)	DT	2.77	0.36	6.29	3.25	5.72	1.64	2.89	5.29	10.00
	CM	0.60	0.86	0.75	0.74	0.85	0.66	1.04	1.14	1.67
	AM	0.06	0.13	0.07	0.08	0.11	0.11	0.11	0.11	0.21
経過時間指数 (上段; 平均値) (下段; 標準偏差)	AL	0.10	0.33	0.47	0.81	0.31	0.20	0.34	0.35	0.53
	SH	0.56	0.55	0.55	0.56	0.57	0.57	0.57	0.54	0.53
	DT	0.18	0.18	0.17	0.18	0.19	0.18	0.18	0.18	0.17
経過時間指数 (上段; 平均値) (下段; 標準偏差)	CM	0.38	0.35	0.38	0.38	0.40	0.35	0.34	0.34	0.33
	AM	0.12	0.09	0.14	0.12	0.13	0.09	0.12	0.19	0.17
	AL	0.62	0.68	0.64	0.65	0.67	0.75	0.65	0.64	0.62
経過時間指数 (上段; 平均値) (下段; 標準偏差)	SH	0.17	0.17	0.15	0.16	0.16	0.18	0.18	0.17	0.18
	AM	0.54	0.58	0.60	0.59	0.59	0.53	0.55	0.54	0.60
	AL	0.12	0.15	0.17	0.17	0.18	0.11	0.14	0.14	0.17
経過時間指数 (上段; 平均値) (下段; 標準偏差)	SH	0.63	0.54	0.53	0.54	0.57	0.55	0.56	0.57	0.55
	AM	0.13	0.12	0.12	0.10	0.11	0.12	0.11	0.12	0.10
	AL	3.55	3.82	3.57	4.02	2.89	3.57	3.81	3.88	4.61
経過時間指数 (上段; 平均値) (下段; 標準偏差)	SH	4.06	6.63	6.44	9.36	2.58	0.44	8.69	5.60	0.68
	AM	3.48	3.63	3.68	3.51	3.58	3.41	3.99	4.13	5.02
	AL	1.50	1.26	1.86	1.58	1.62	1.01	1.72	1.60	2.68

表 5.5 シミュレーション実験結果(実験Ⅲ)

項目	実験番号	RI-0	Ⅲ-1	Ⅲ-2	Ⅲ-3	Ⅲ-4	Ⅲ-5	Ⅲ-6	Ⅲ-7	Ⅲ-8	Ⅲ-9
平均AP稼働率(%)		94.47	98.91	96.57	98.76	98.22	95.98	97.31	99.31	96.99	99.14
平均CP稼働率(%)		93.34	97.94	98.29	97.19	97.57	98.08	95.63	95.50	96.93	92.16
平均主記憶利用率(%)		80.60	79.34	78.87	79.74	81.66	80.45	80.41	87.12	84.75	80.72
A Pジョブステップの 主記憶多重度	平均値	1.64	1.97	1.90	1.95	1.94	1.88	1.91	1.97	1.92	1.97
	標準偏差	0.54	0.19	0.37	0.23	0.27	0.40	0.34	0.18	0.34	0.20
C Pジョブステップの 主記憶多重度	平均値	3.60	3.39	3.68	3.33	3.43	3.62	3.52	3.19	3.33	3.32
	標準偏差	1.00	0.79	0.70	0.80	0.78	0.79	0.80	0.89	0.87	0.79
A Pジョブステップの ロールアウト件数	平均値	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00
	標準偏差	0.00	0.03	0.03	0.04	0.01	0.04	0.02	0.09	0.0	0.005
C Pジョブステップの ロールアウト件数	平均値	0.46	0.44	0.18	0.45	0.45	0.30	0.28	0.72	0.57	0.41
	標準偏差	0.70	0.72	0.45	0.68	0.67	0.58	0.54	0.86	0.75	0.65
ジョブステップ総ロールアウト件数		917	650	396	685	584	491	695	770	675	834
空領域の平均サイズ (K語/個)		78.64	91.15	95.13	92.08	80.70	94.64	93.27	63.02	76.21	89.57
空領域の平均個数 (個)		1.48	1.41	1.37	1.36	1.38	1.27	1.31	1.15	1.18	1.32
リロケーション個数 (個)		5,410	4,274	4,247	4,238	3,649	3,671	4,594	2,580	3,133	4,920
リロケーションによる移動領域(M語)		197.10	192.85	198.74	187.78	168.00	182.50	197.51	154.79	177.61	178.23
平均プール個数	AP	—	3.10	2.66	2.85	2.93	2.67	2.75	2.72	2.86	1.32
	CP	—	3.15	3.03	2.47	2.99	2.94	2.34	2.35	2.74	3.15
平均実行保留時間 (分/件)	AM	—	12.31	9.14	10.80	12.15	11.73	12.67	17.87	18.28	24.07
	AL	—	16.46	10.17	8.90	10.92	10.50	11.52	13.88	11.67	13.37
	CM	—	20.02	28.03	18.92	22.12	21.02	18.52	14.82	17.53	10.48
A P 寄与率	AM	40.04	70.45	68.48	58.08	55.42	54.81	48.09	30.52	32.73	28.57
	AL	54.43	28.46	28.09	40.68	42.80	41.17	49.22	68.79	64.26	70.57
	合計	94.47	98.91	96.57	98.76	98.22	95.98	97.31	99.31	96.99	99.14
	SH	32.52	19.60	23.59	20.74	15.88	15.88	25.39	5.17	10.37	28.99
	DT	2.68	2.11	2.66	2.09	3.97	4.67	3.74	5.27	6.12	5.20

C P 寄与率	C M	49.51	67.68	61.21	63.08	65.54	68.51	57.68	77.66	73.54	52.71
	A M	6.31	7.73	9.72	9.85	8.58	7.41	6.72	3.95	2.76	2.90
	A L	2.32	0.82	1.11	1.43	2.60	1.61	2.10	3.45	4.14	2.36
	合 計	93.34	97.94	98.29	97.19	97.57	98.08	95.63	95.50	96.93	92.16
	S H	22.10	24.38	5.84	21.62	27.20	15.30	12.01	47.48	21.56	15.17
	D T	66.20	225.3	95.12	134.5	64.72	94.31	52.34	302.7	216.1	56.09
	C M	4.24	4.11	6.19	8.02	9.36	3.25	11.72	12.18	3.93	9.32
	A M	0.00	5.16	6.08	26.82	20.74	3.50	9.14	0.27	29.23	9.91
	A L	0.00	0.34	1.66	6.14	32.71	14.73	1.74	84.52	1.42	70.47
	S H	0.49	0.40	0.15	0.40	0.43	0.29	0.39	0.66	0.33	0.43
	D T	4.13	5.21	2.82	4.24	1.82	2.53	2.17	5.03	3.57	1.84
	C M	0.22	0.40	0.58	0.78	0.61	0.26	0.68	0.27	0.39	0.81
	A M	0.00	0.03	0.03	0.05	0.06	0.02	0.04	0.08	0.04	0.04
	A L	0.00	0.03	0.07	0.09	0.06	0.04	0.04	0.06	0.01	0.04
	有効主記憶時間比 (上段; 平均値) (下段; 標準偏差)	S H	0.59	0.54	0.52	0.54	0.55	0.52	0.56	0.57	0.53
		0.20	0.17	0.16	0.18	0.17	0.17	0.18	0.18	0.16	0.19
D T		0.39	0.33	0.33	0.36	0.37	0.33	0.40	0.38	0.36	0.45
		0.15	0.05	0.08	0.10	0.12	0.06	0.17	0.12	0.08	0.17
C M		0.64	0.66	0.65	0.63	0.68	0.66	0.64	0.78	0.72	0.67
		0.16	0.17	0.17	0.18	0.16	0.17	0.02	0.16	0.17	0.18
A M		0.56	0.51	0.51	0.51	0.51	0.50	0.52	0.51	0.51	0.52
		0.13	0.07	0.09	0.10	0.10	0.09	0.10	0.08	0.11	0.09
A L		0.61	0.50	0.51	0.50	0.50	0.52	0.52	0.51	0.51	0.51
		0.12	0.02	0.08	0.06	0.05	0.09	0.05	0.04	0.07	0.06
S H		4.30	4.11	3.07	4.06	4.34	3.54	3.33	5.98	3.74	3.30
		14.20	6.29	2.16	8.44	9.54	4.25	4.01	18.21	6.47	5.97
D T		3.29	3.82	3.75	3.46	3.65	3.73	3.22	3.93	3.71	2.99
		11.00	1.92	1.77	1.20	1.79	1.41	0.91	1.90	1.60	1.27

利用できる。従って、 $(100\% - \text{平均稼働率})$ は処理装置が処理しえる状態のユーザジョブの存在しなかった平均時間割合を表わし、本稿ではその値をアイドル率と呼んでいる。

(2) 主記憶利用率は $(\text{使用主記憶容量} / 568) \times 100\%$ を意味する。

(3) 空領域の平均サイズと平均個数は主記憶上の断片化の平均的状況を表わしている。

(4) リロケーション回数はリロケータの起動回数を表わし、移動領域はリロケータが主記憶上で移動した領域の総量を表わす。

(5) 寄与率は各ジョブクラス毎の処理装置使用比率であって、平均AP, CP稼働率の内訳を表わす。

(6) 有効主記憶時間比は各ジョブクラス毎のジョブに主記憶が割付けられていた総時間に対する、各ジョブがシステム資源 (APU, CPU およびチャネ

ル)を使用した総時間の比率を表わす。

(7) 経過時間指数はSHとDTクラスのジョブがシステム資源を使用した総時間に対する、各ジョブが開設から閉設までにかかった時間の比率を表わす。CM, AM, AL各クラスのジョブについては実行保留中の時間があるのでこの値は著しく大きくなる。表5.3~5.5には新・旧スケジューラで比較可能なジョブクラスについてのみこの指標の値を示した。

また、図5.2~5.4は20時間平均値の細部を呈示するため、各一時間毎のAP, CP平均稼働率の時間的推移を表わしている。図5.5はジョブの平均到着時間間隔をパラメータとした、実験Ⅲの各運用系によるジョブ処理件数の時間的推移を示している。最後に、図5.6~5.9および表5.6はラッシュアワー実験結果である。

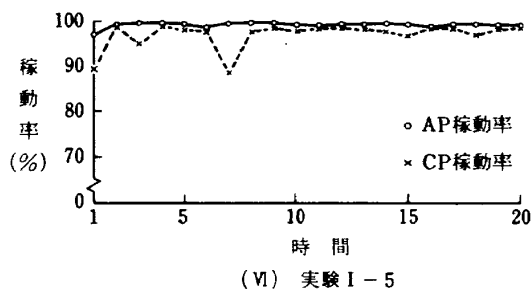
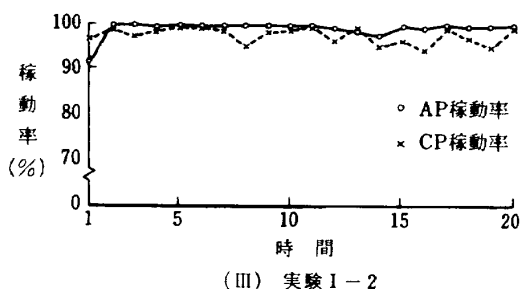
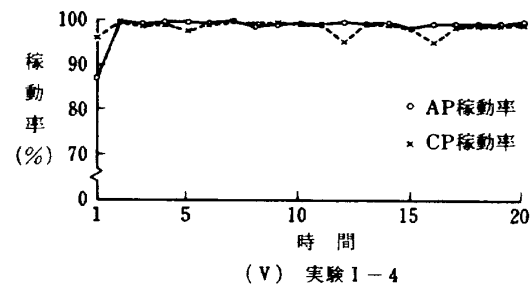
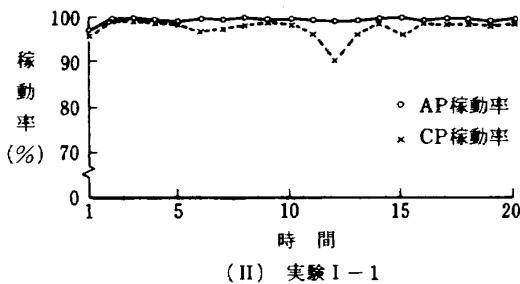
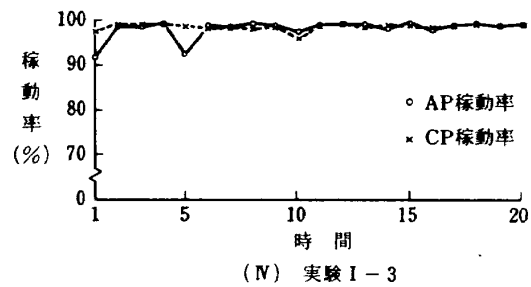
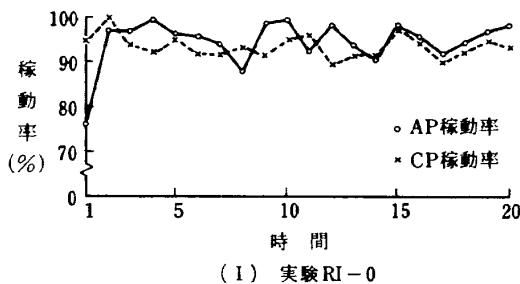
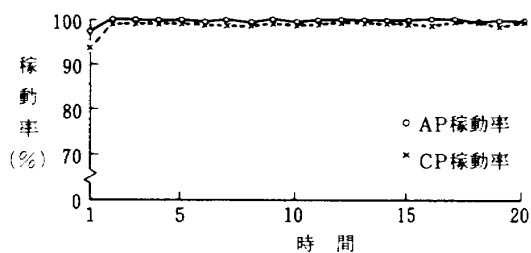
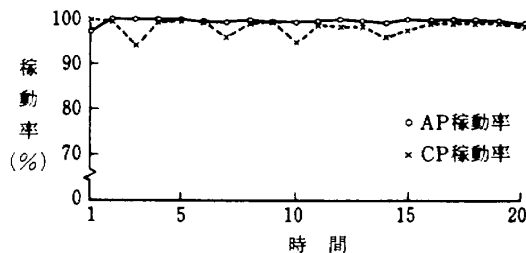


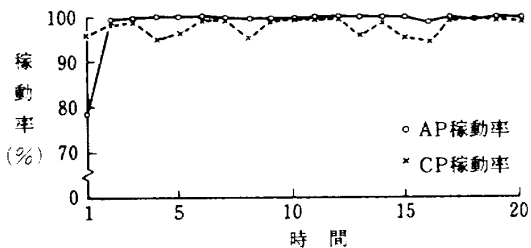
図 5.2 平均処理装置稼働率の推移 (実験 I)
(注) 平均は 1 時間毎の平均を意味する



(VII) 実験 I-6

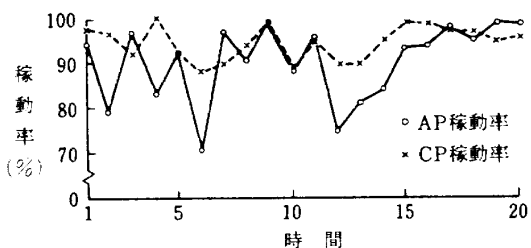


(IX) 実験 I-8

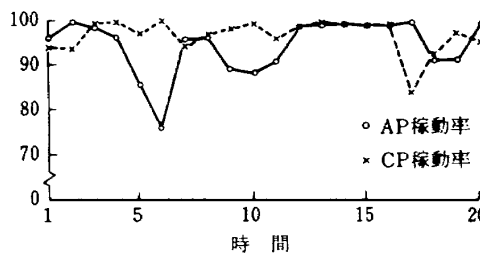


(VIII) 実験 I-7

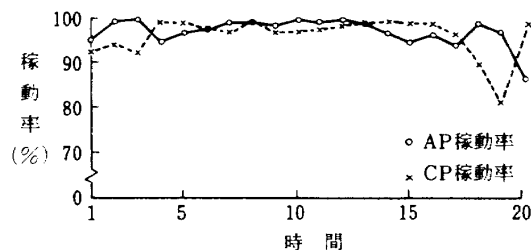
図 5.2 つづき



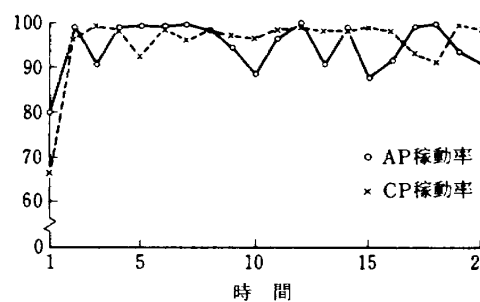
(I) 実験 II-R0



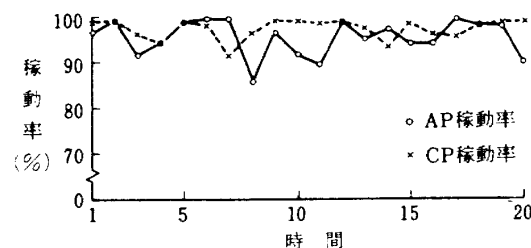
(IV) 実験 II-3



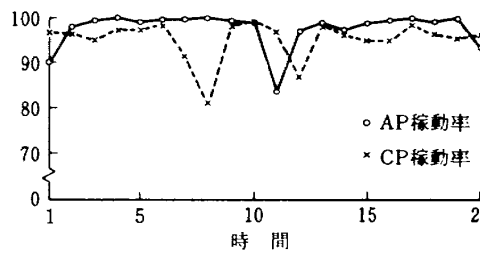
(II) 実験 II-1



(V) 実験 II-4



(III) 実験 II-2



(VI) 実験 II-5

図 5.3 平均処理装置稼働率の推移 (実験 II)

(注) 平均は 1 時間毎の平均を意味する

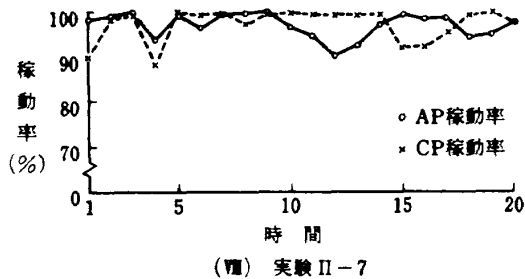
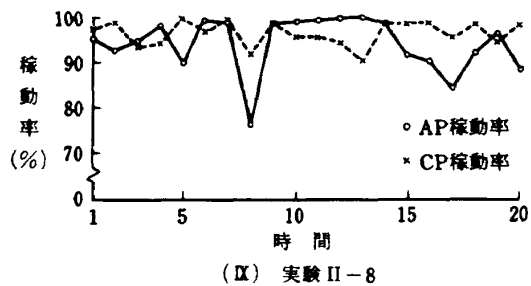
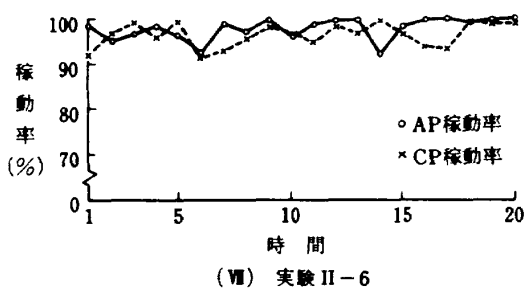


図 5.3 つづき

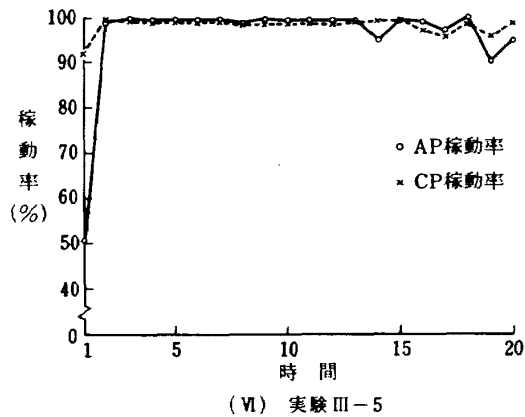
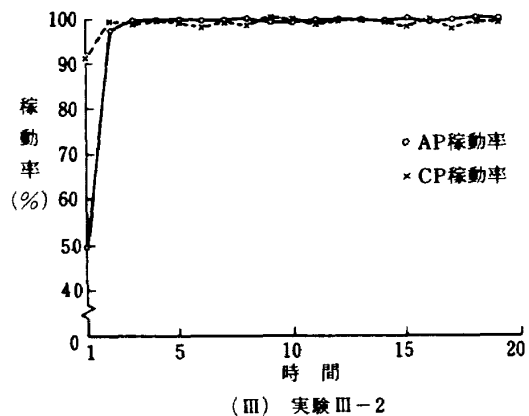
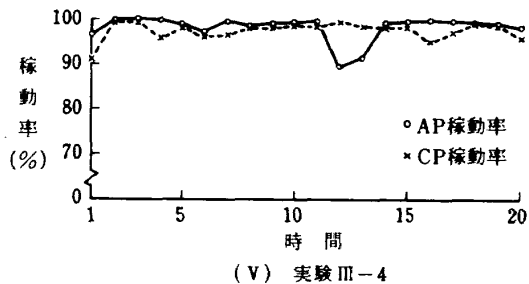
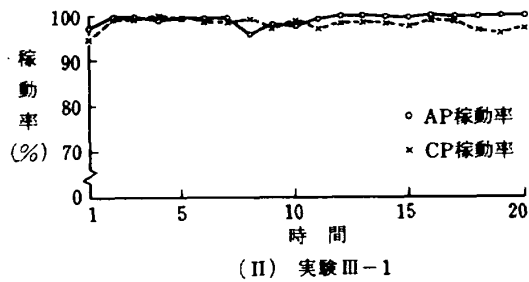
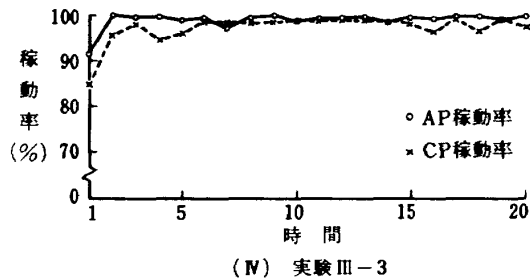
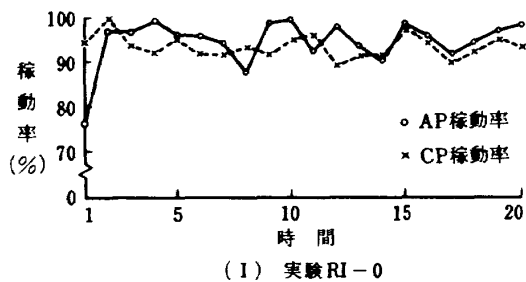


図 5.4 平均処理装置稼働率の推移 (実験 III)
 (注) 平均は 1 時間毎の平均を意味する

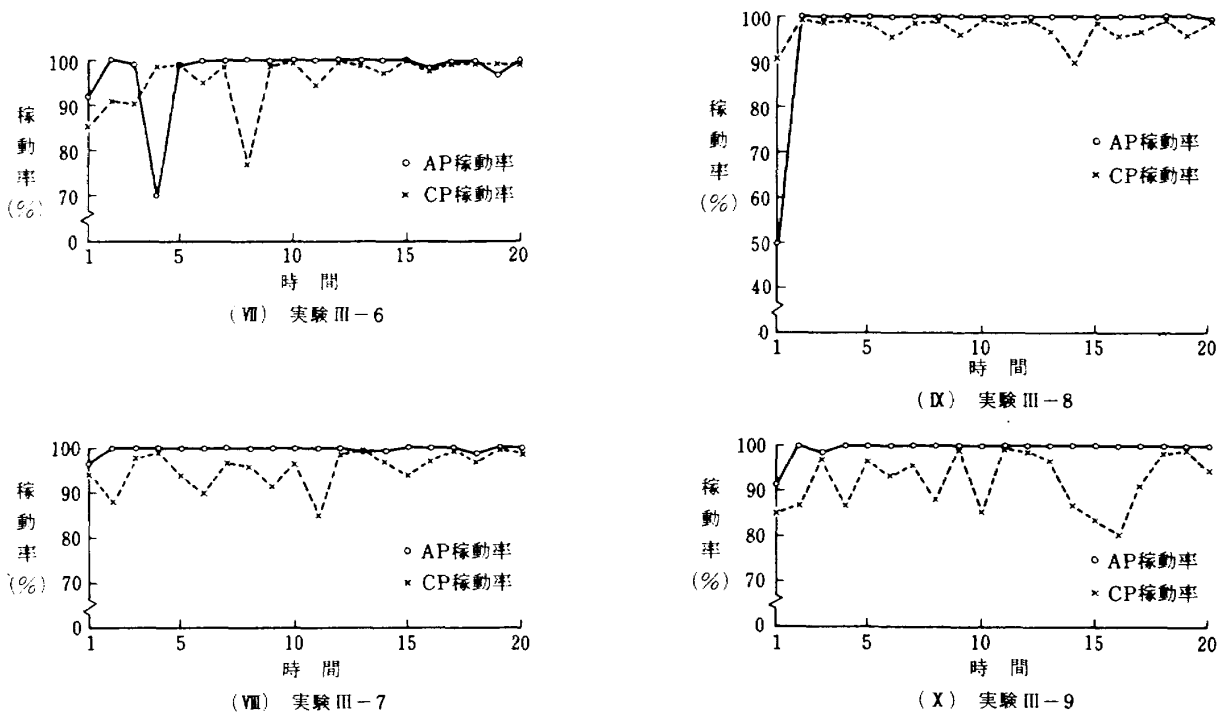


図 5.4 つづき

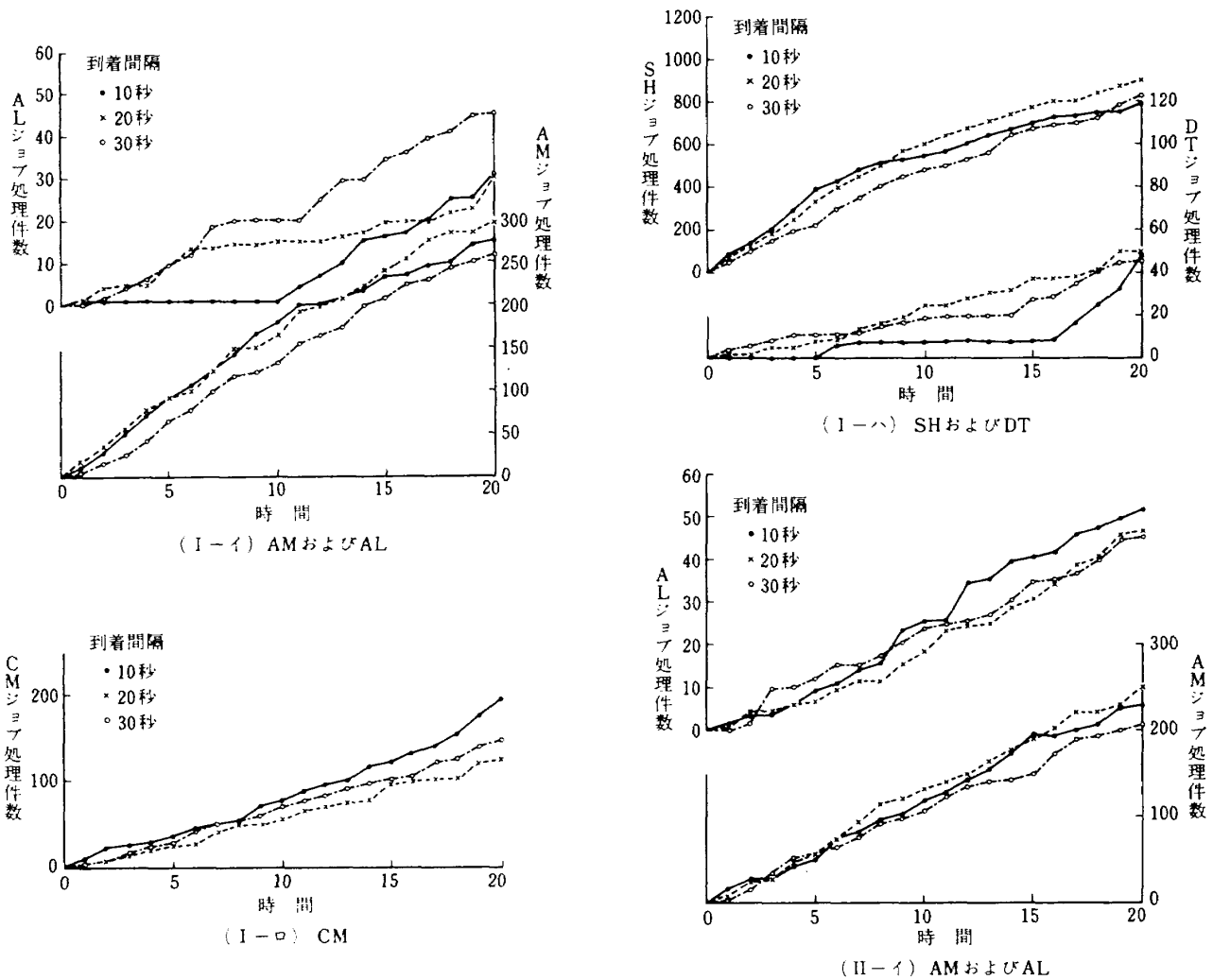


図 5.5 実験 III におけるジョブ処理件数の推移

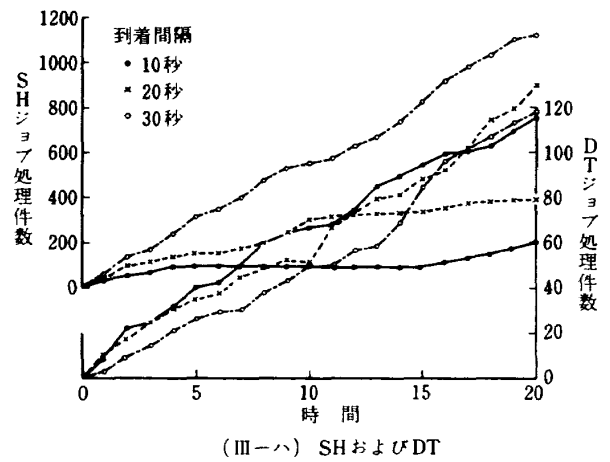
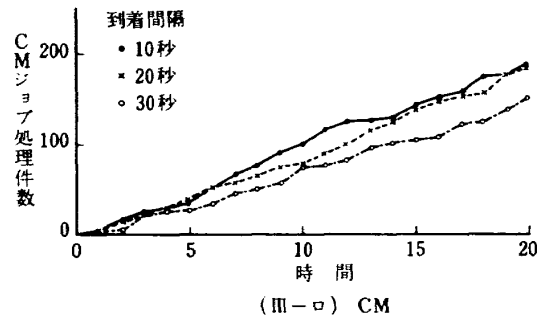
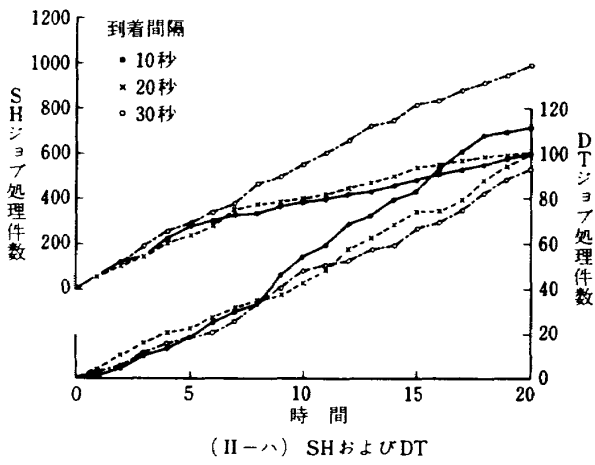
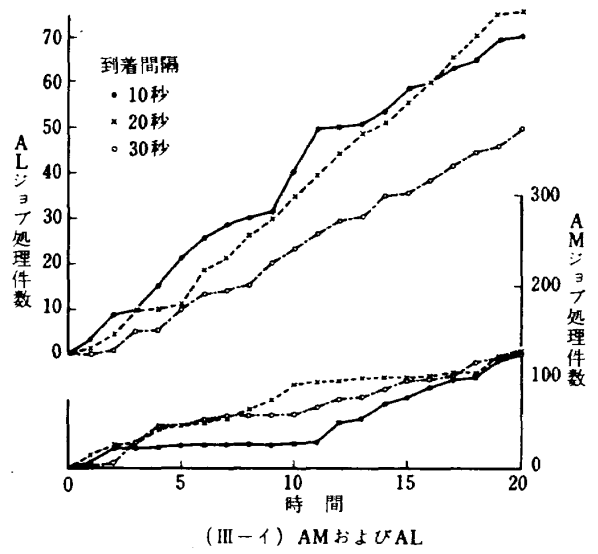
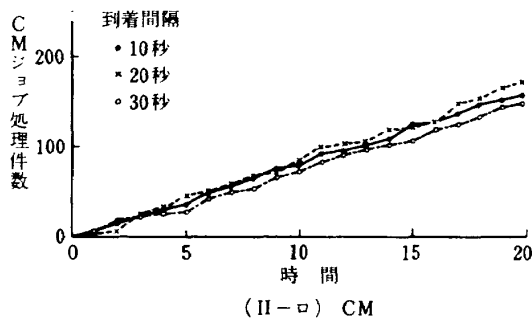


図 5.5 つづき

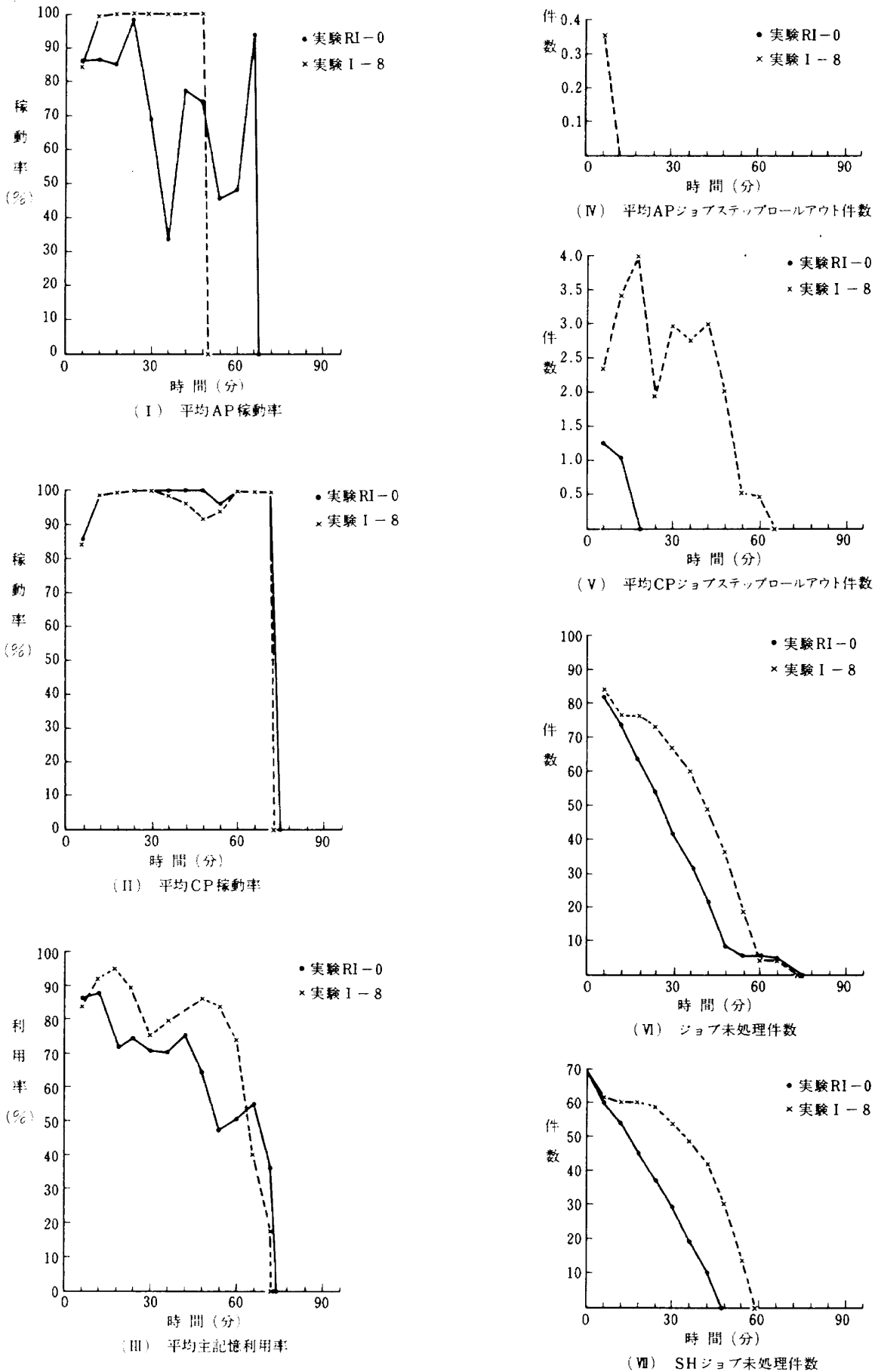


図 5.6 ラッシュアワー実験 (ジョブストリーム 1)
 (注) 平均は全て 6 分毎の平均を意味する

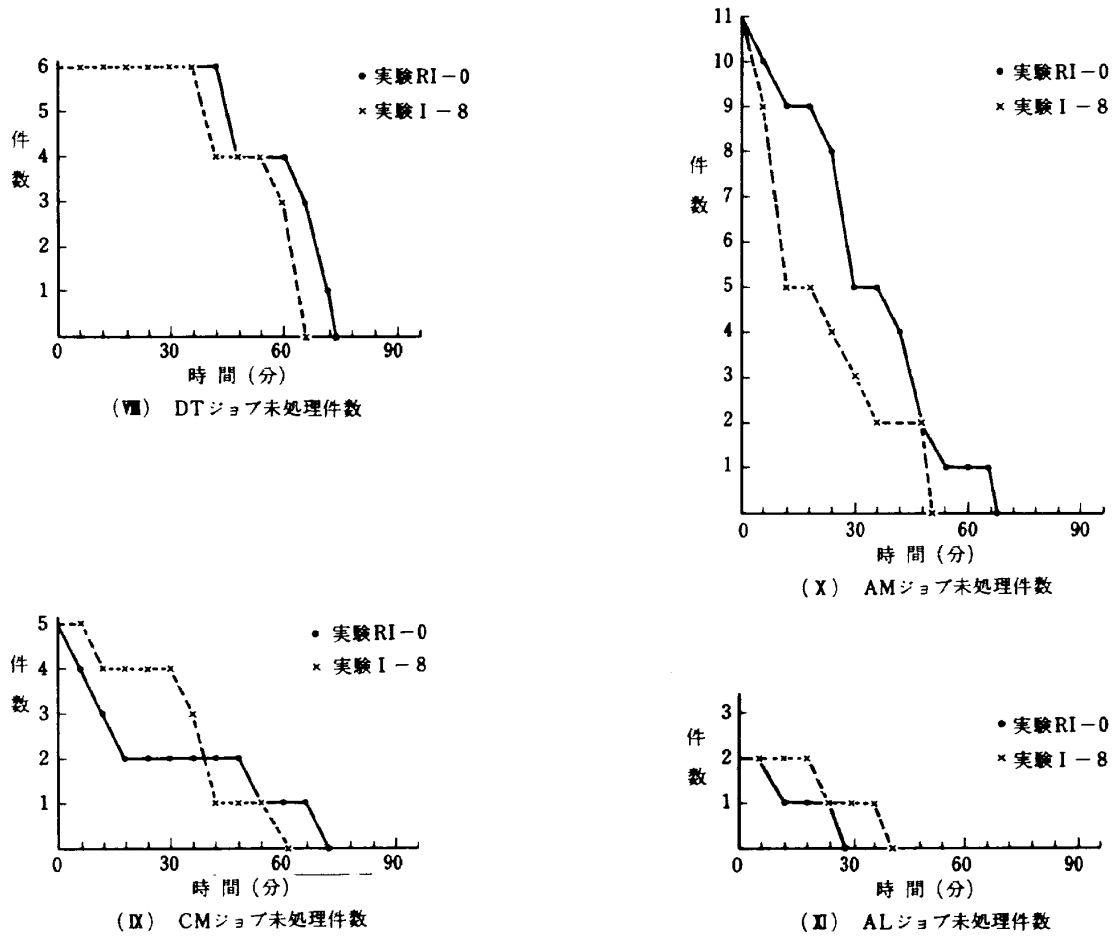


図 5.6 つづき

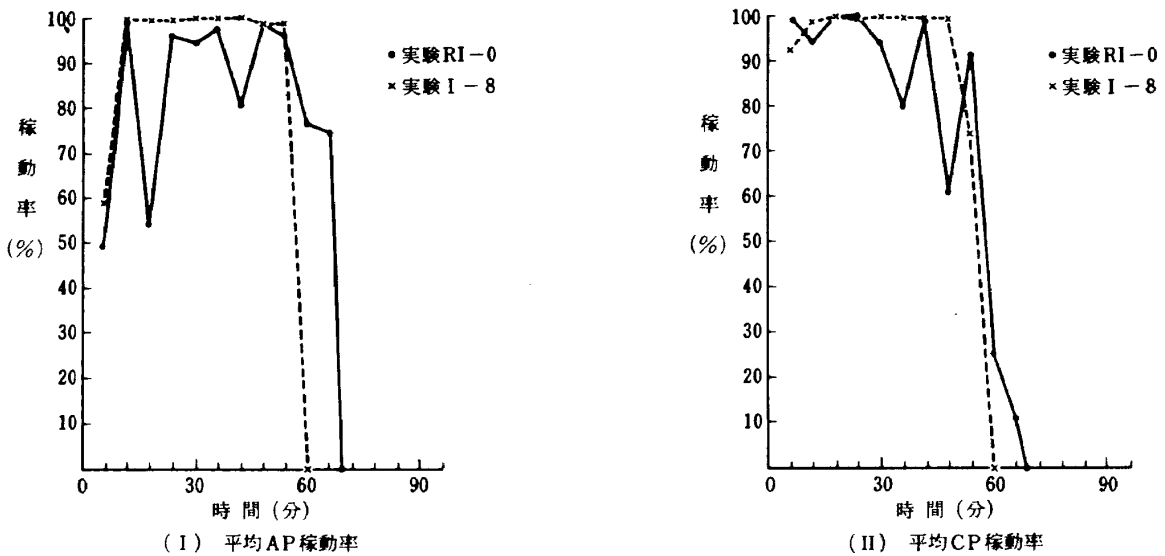


図 5.7 ラッシュアワー実験 (ジョブストリーム 2)
 (注) 平均は6分毎の平均を意味する

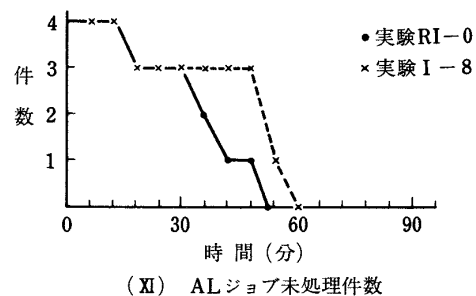
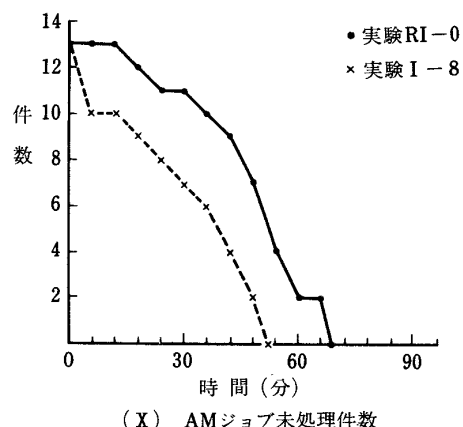
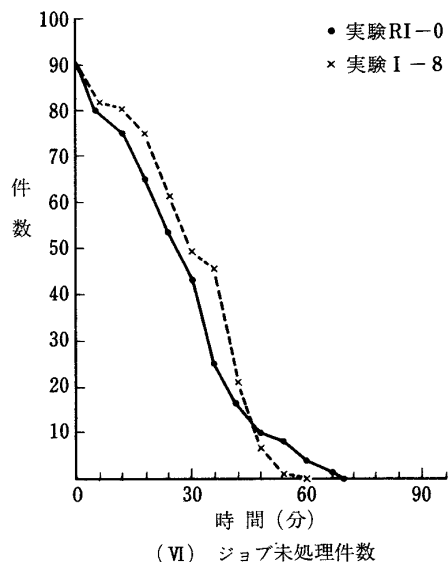
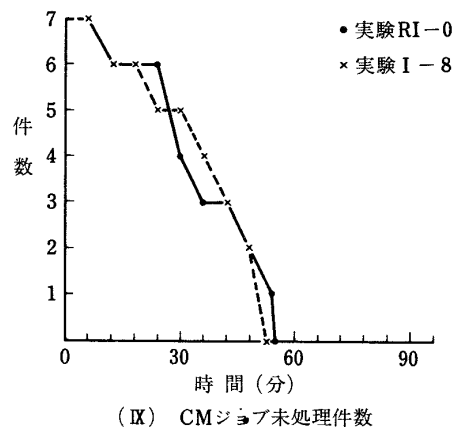
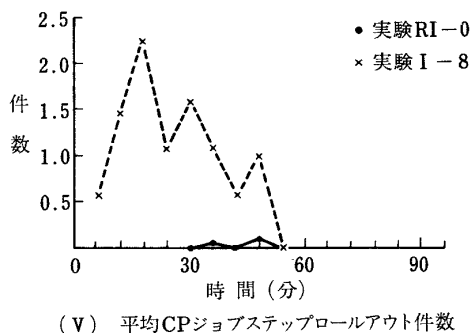
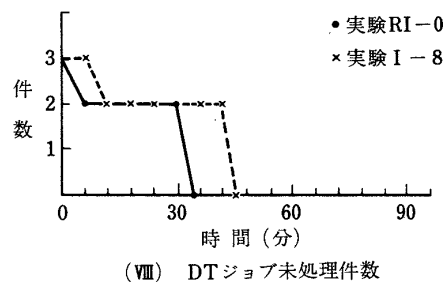
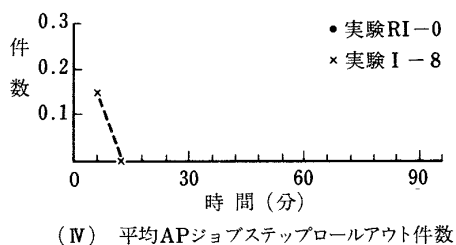
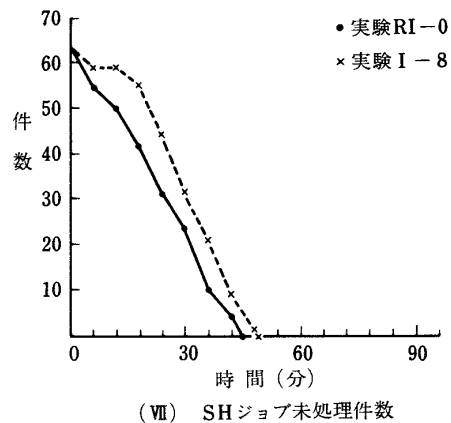
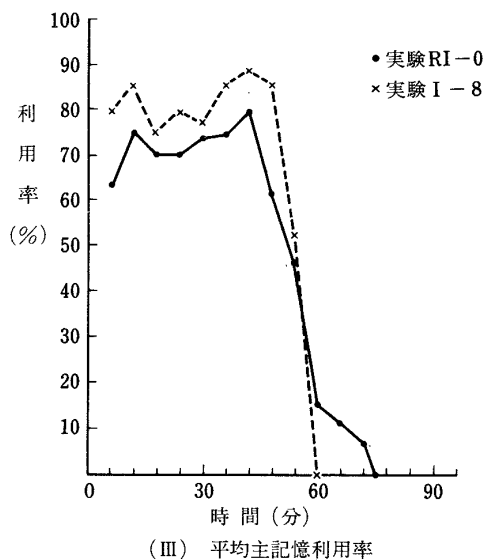


図 5.7 つづき

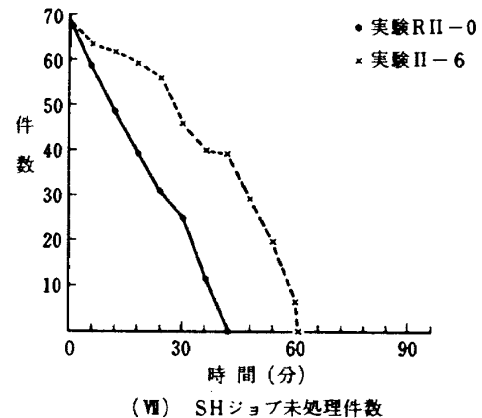
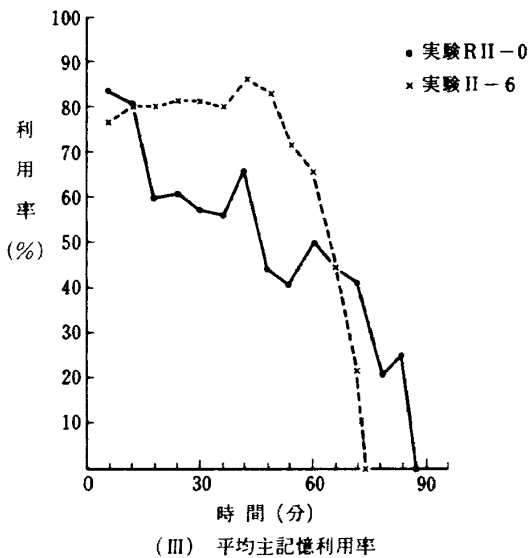
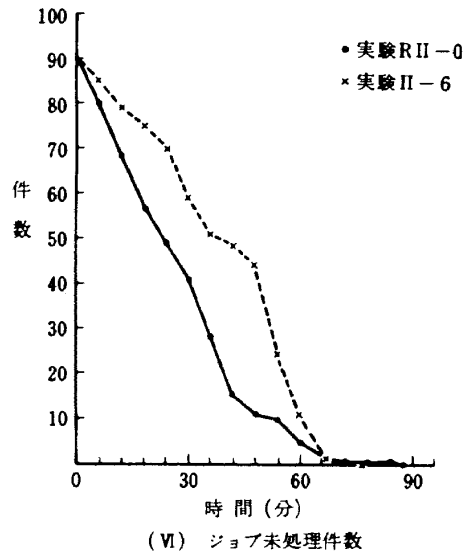
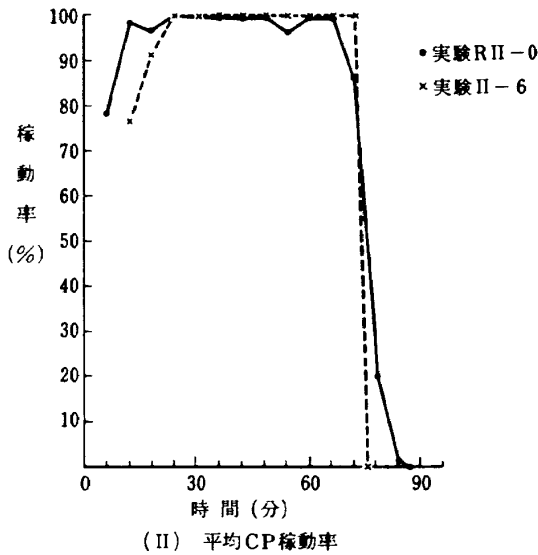
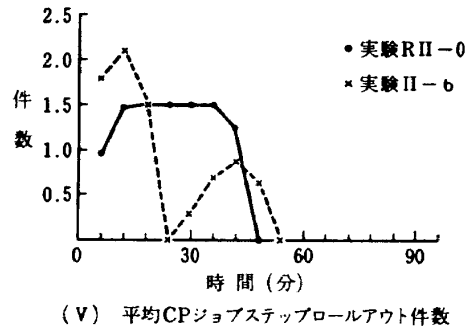
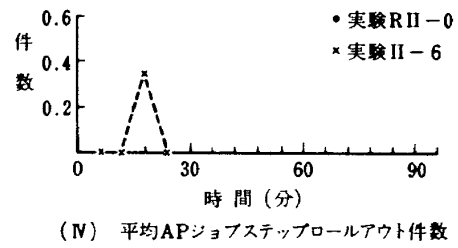
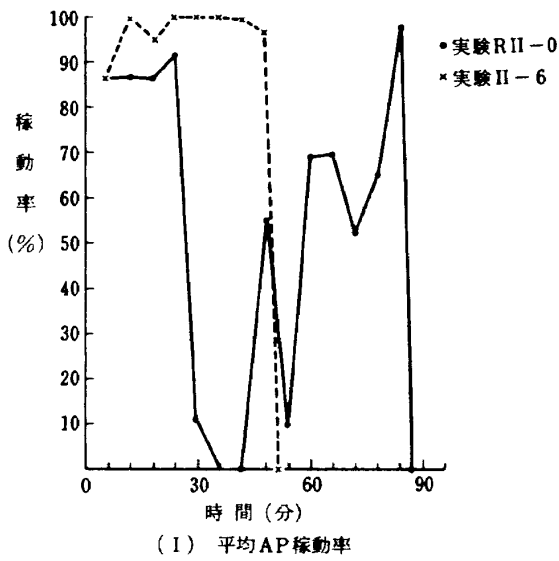


図 5.8 ラッシュアワー実験(ジョブストリーム1)
(注) 平均は6分毎の平均を意味する

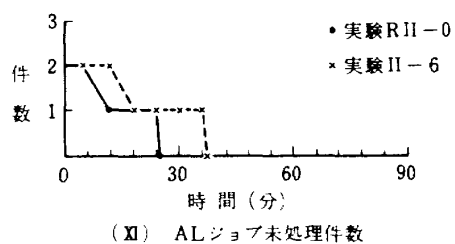
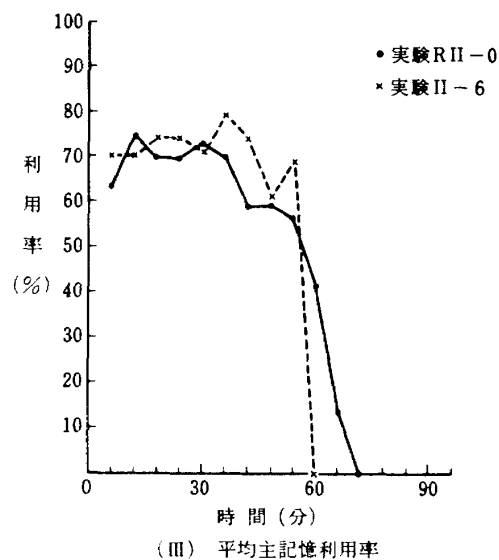
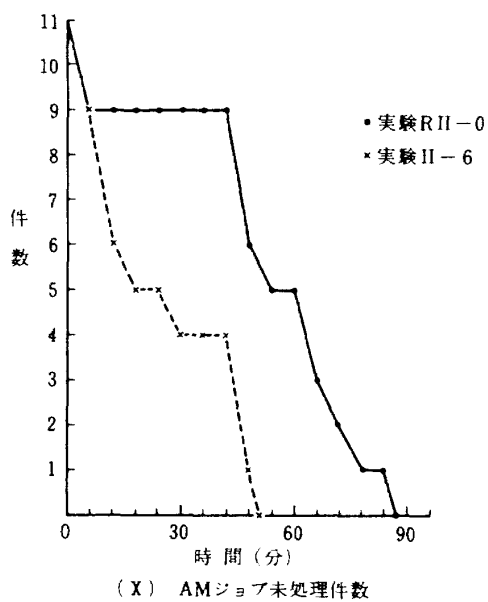
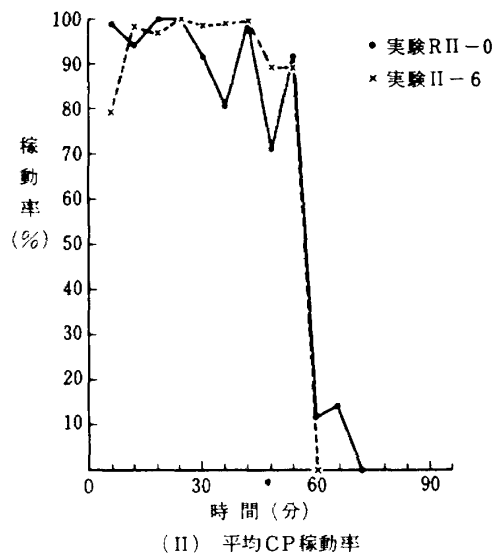
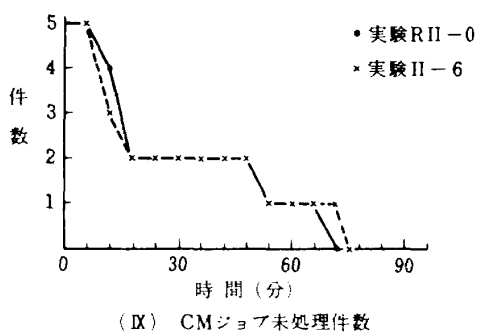
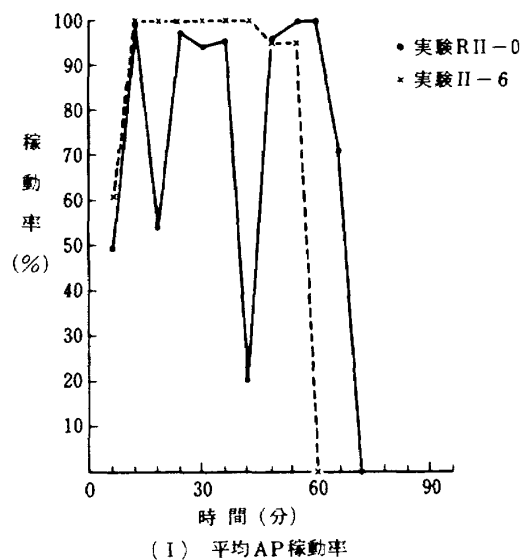
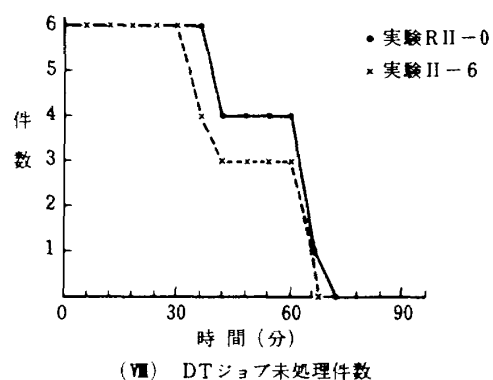
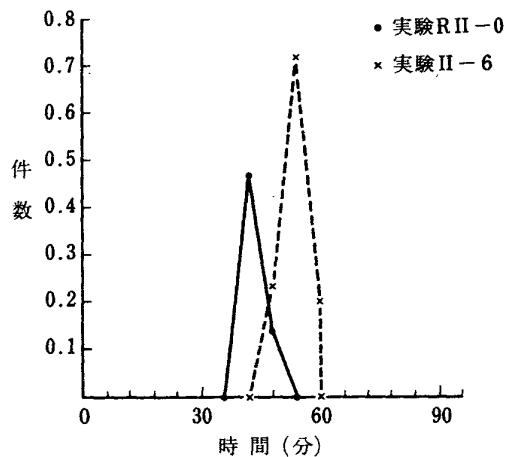
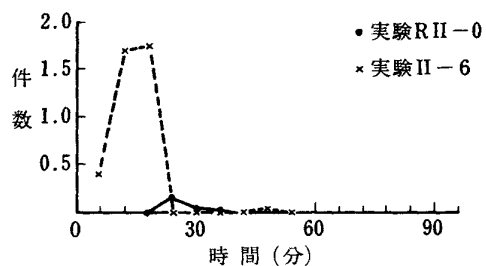


図 5.8 つづき

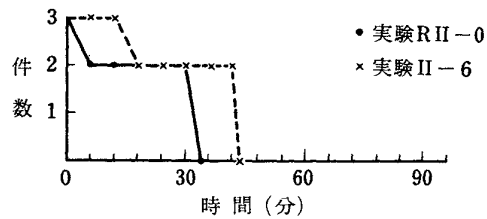
図 5.9 ラッシュアワー実験 (ジョブストリーム 2)
(注) 平均は全て 6 分毎の平均を意味する



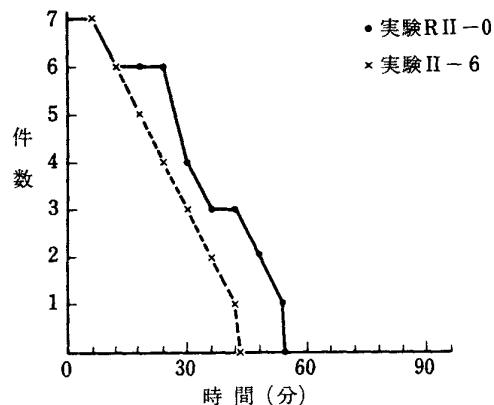
(IV) 平均APジョブステップロールアウト件数



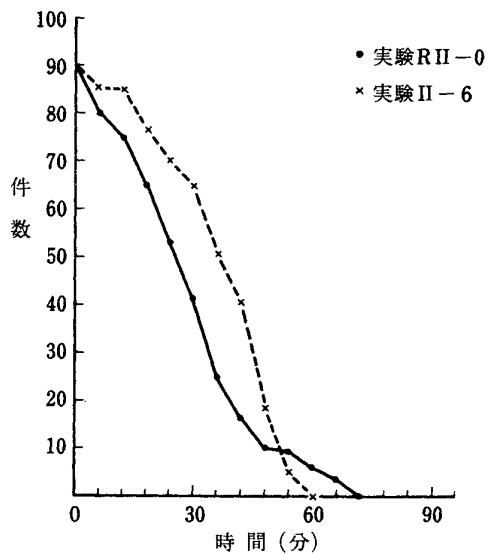
(V) 平均CPジョブステップロールアウト件数



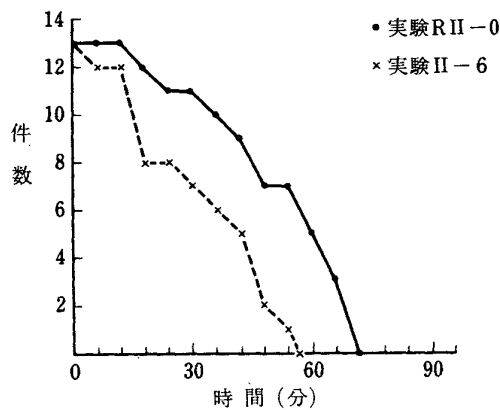
(VIII) DTジョブ未処理件数



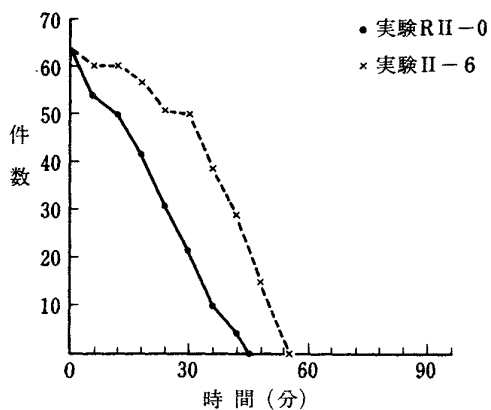
(IX) CMジョブ未処理件数



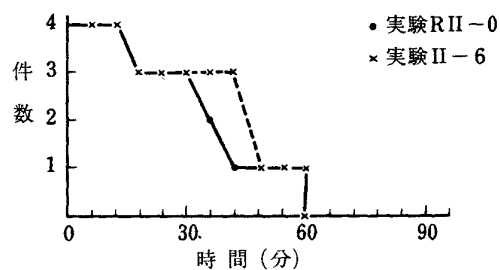
(VI) ジョブ未処理件数



(X) AMジョブ未処理件数



(VII) SHジョブ未処理件数



(XI) ALジョブ未処理件数

図 5.9 つづき

表 5.6 ラッシュアワー実験におけるジョブ処理終了時間

実験 クラス	図 5.6		図 5.7		図 5.8		図 5.9	
	旧スケジューラ	新スケジューラ	旧スケジューラ	新スケジューラ	旧スケジューラ	新スケジューラ	旧スケジューラ	新スケジューラ
SH	2,794	3,510	2,686	3,101	2,526	3,731	2,691	3,314
DT	4,415	3,948	2,049	2,714	4,304	4,112	2,014	2,737
CM	4,319	4,381	3,316	3,151	4,273	4,558	3,270	2,618
AM	4,066	3,040	4,127	3,100	5,205	3,070	4,282	3,446
AL	1,705	2,450	3,116	3,591	1,486	2,258	3,602	3,602

(単位秒)

6. 考 察

前節のシミュレーション実験結果に基づき、本節ではまず新・旧スケジューラによるジョブ処理効果の有効性を評価し、新スケジューラの優位性を検証してから、新スケジューラに対する適切な運用系の検討並びに試作スケジューラ SCD1 と SCD2 によるジョブ処理効果の比較とを行う。

6.1 新旧スケジューラの有効性比較

6.1.1 主記憶競合小なる環境での比較

アロケータ 2 とリロケータ 1 を使用した主記憶管理ルーチン 1 は主記憶を最大限に有効利用した場合の状況を模擬する⁴⁾。このような主記憶管理環境のもとでは実験 RI、実験 I、実験 III およびラッシュアワー実験結果から、以下のことが結論される。

(i) 表 5.3 は実験 RI と実験 I の 20 時間シミュレーション結果に基づいて、APU システムのジョブ処理状況を表わす各種指標の平均値を一覧にしたものである。まず、処理装置の平均稼働率を調べると、旧スケジューラの実験 RI-0 も非常に高い平均稼働率を呈しているが、新スケジューラ SCD1 の実験 I-1~8 では全てその値を凌駕しており、かつ我々が SCD1 設計の目標とした通り、APU、CPU 共に平均稼働率はほとんど限界的な値を示している。これは旧スケジューラに対する新スケジューラの優位性を結論する第一の根拠となる。

もう少し詳細に新・旧スケジューラによるジョブ処理効果の差異を調べると、第一に旧スケジューラよりも新スケジューラの方が平均的に主記憶競合の度合いが増大している点が挙げられる。但し、これは CP ジョブステップに関する

主記憶競合であって、AP ジョブステップに関してはほとんど主記憶競合が発生していない。従って、AP ジョブに関する平均ロールアウト時間と回数の値はそのサービスジョブステップ処理時におけるものと考えてよい。CP ジョブステップに関して発生するこの主記憶競合は次のように説明できる。運用系 1~8 は全て CM、AM および AL クラスのランジョブステップが各 1 件ずつ定期的に主記憶割付けされるように運用しているから、主記憶はこれらによって平均的に 320K 語占有されている(表 3.4 参照)。SCD1 は全クラスのサービスジョブステップが IO 多重度で処理されるように設計しているから、主記憶の平均的残り 250K 語を IO 多重度 (= 2~5) のジョブステップが競合して使用することになる。よって、IO 多重度が大きい程ロールアウト頻度が増す。その際、主記憶上にある SH および DT クラスのランジョブステップは最もロールアウトされ易い。また、FORTRAN の実行には平均的に 100K 語の主記憶が必要であるから(表 3.4 参照)、FORTRAN の同時処理件数が大きい程、主記憶競合が顕著に現われる。しかし、このような主記憶競合が CP 稼働率にほとんど影響を及ぼしていない点が重要である。

第二に、CM、AM および AL ジョブの平均実行保留時間を注目すべきである。この値は先行サービスジョブステップの処理が終わってから、ランジョブステップの実行に入るまでの一ジョブ当りの平均待ち時間である。従って、それぞれランジョブステップに先行するサービスジョブステップ段階にエラーのあるジョブのターンアラウンドの平均短縮時間をも意味している。

表 5.3 はプール個数設定値を 3 とした場合の値を示しているが、設定値を大きくするほど大巾にこのターンアラウンド時間を短縮できる。これは旧スケジューラにはない新たな特長であり、新スケジューラの優位性を結論する第二の根拠である。

第三の相違点は AP 寄与率に見られる。新スケジューラでは主記憶競合がほとんど起らない状況の場合、ラウンドロビン方式による APU 運用が正確に実現でき、AM と AL ジョブ処理のために APU が公平に使われている。一方、旧スケジューラでは主記憶競合がなくても、AL ジョブ処理のために APU が明らかに多く使用され、ラウンドロビン方式の公平さが実現しにくい。第四の相違点は AP ジョブの有効主記憶時間比に見られ、この値は旧スケジューラよりも新スケジューラの方が多少低下している。このことは主記憶有効利用の立場からは好ましくないが、CP ジョブの有効主記憶時間比への影響は見られないので特別問題とはならない。

- (ii) 表 5.5 は実験 RI - 0 と実験 III - 1 ~ 9 に関する 20 時間シミュレーション結果を表わしているが、同表から次の二点を除いて(i)と同じ結論がなし得る。一点は AP および CP 寄与率に関してである。新スケジューラ SCD2 は AP ラン多重度と IO 多重度の空きに際して、共にジョブクラスではなく重み付き経過時間に基づいてジョブ・ジョブステップ起動のスケジューリングを行うように設計している。従って、SCD1 のように寄与率は一定せず、SCD2 ではジョブクラス毎の AP および CP 寄与率は重みの付け方に応じて変化している。詳しくは 6.3 節で検討する。もう一点は AM と AL ジョブの平均実行保留時間である。実験 III では AP プール個数設定値を 3 としているの、AM と AL プール個数設定値をそれぞれ 3 にしている実験 I に比べて AM, AL ジョブ共に平均実行保留時間が短くなっている。
- (iii) 図 5.2 は実験 RI - 0 を実験 I - 1 ~ 8 の 20 時間シミュレーション結果から、一時間毎の平均処理装置稼働率の値を求めてその時間的推移

を図示したものである。同図の(i)は旧スケジューラを用いた場合の推移を示しているが、AP ジョブは全くロールアウトされずに処理されているにもかかわらず(表 5.3 参照)、一時間毎の平均 AP 稼働率にかなりの変動が見られる。一方、新スケジューラの場合には同図の(ii) ~ (ix) が示すように、AP 稼働率はコンスタントにほとんど限界的な値を呈している。また、一時間毎の平均 CP 稼働率は CP ジョブステップ相互の主記憶競合のため時間的変動を呈しているが、旧スケジューラに比べて新スケジューラの方が平均稼働率が高く、変動巾も小さくなっているのが明確に読みとれる。これらが旧よりも新スケジューラの優位性を結論する第三の根拠である。なお、同図の(ii) ~ (ix) において、AP 稼働率の大きな落込みは AP ジョブステップ(AM または AL ランジョブステップ)のロールアウトに起因し、CP 稼働率の大きな落込みは CM ランジョブステップのロールアウトに起因している。

- (iv) 図 5.4 は実験 RI - 0 と実験 III - 1 ~ 9 について、一時間毎の平均処理装置稼働率の値の時間的推移を図示したものであるが、同図から(iii)と同じ結論がなし得る。なお、同図の(ii) ~ (ix) では処理装置稼働率の落込みが図 5.2 の(ii) ~ (ix) に比べて多い。このように、SCD2 は SCD1 よりも主記憶競合を発生させ易いが、その理由については 6.3 節で詳述する。
- (v) 図 5.6 と図 5.7 は実験 I - 8 によって、ラッシュアワー時における新旧スケジューラのジョブ処理状況を比較したものである。図 5.6 から明らかなように、ジョブストリーム 1 に対しては次の 2 点に著しい差異が認められる。第一点は 6 分毎の平均 AP 稼働率の推移であって、新スケジューラの場合には我々の目標通り 100% 近い AP 稼働率を示している。一方、旧スケジューラの場合にははげしく確率的な変動を呈しているが、AP ジョブステップは全くロールアウトされていないから、このジョブストリーム 1 は APU に対して一時的に非常に軽微な負荷状況(即ち、AP ジョブがチャンネルバウンドな

状況)になることを意味している。このため、処理すべきAPジョブが存在するにもかかわらず、APUの処理能力を十分活かしえず、新スケジューラに較べてAPジョブ処理終了までに長時間を要している。図5.6の(i)はジョブ・ジョブステップスケジューラがこのような問題を解決する様子を明確にとらえている。第二点はSHジョブ未処理件数の推移にある。一般に、ラッシュアワー時には処理待ち件数の多いSHジョブのスループットを向上させて、出来るだけ早く混雑を解除した方がよい。しかし、CM, AMおよびALプール補充のためのCPジョブステップによる主記憶競合がはげしく、同図の(vii)は一時期SHジョブのスループットが著しく低下することを示している。これは、ジョブ・ジョブステップスケジューラによる計算機システムのジョブ処理開始時に現われる特徴であるが、一見欠点の如く見える。しかし、通常は常に各クラスのプールが補充された状態で運用されているのであるから、このような現象は起らない。図5.7はジョブストリーム2に対する実験結果を示すが、上記と同じことがいえる。特に同図の(vii)からわかるように、システム立ち上り時期を過ぎれば、むしろ旧よりも新スケジューラの方がSHジョブのスループットは向上している。以上から、新スケジューラの優位性を結論する第四の根拠が得られる。なお、本実験では各クラスのプール補充のために10～20分を要している。

6.1.2 主記憶競合大なる環境での比較

アロケータ1を使用した主記憶管理ルーチンは主記憶上の断片化を出来得る限り克服した場合の状況を模擬するが、再配置機能を利用していないから、6.1.1節の場合よりも主記憶競合が発生しやすい。⁴⁾このような主記憶管理環境のもとでは実験RII, 実験IIおよびラッシュアワー実験結果から、以下のことが結論される。

(i) 表5.4は実験RIIと実験IIの20時間シミュレーション結果から、APUシステムのジョブ処理状況を表わす各種指標の平均値をまとめた

ものである。同表から明らかなように、処理装置の平均稼働率は新スケジューラの実験II-1～8の値が全て旧スケジューラの実験RII-0の値を凌駕している。主記憶競合は旧よりも新スケジューラの方が発生し易い。これは6.1.1節の(i)で述べたIO多重度に起因するのみでなく、CM, AMおよびALクラスのランジョブステップによる主記憶競合も著しくなっている。しかし、それにもかかわらず平均AP, CP稼働率が高水準を示している点が重要である。CM, AMおよびALジョブの平均実行保留時間については6.1.1節の(i)に同じである。また、主記憶競合のため、AMとALランジョブステップのロールアウト頻度と時間とが確率的に増大している。このため、AMおよびALジョブのAP寄与率は一定せず、ラウンドロビン方式の公平さは失われている。

(ii) 図5.3は実験RII-0と実験II-1～8の20時間シミュレーション結果から、一時間毎の平均処理装置稼働率の時間的変動を図示したものである。同図の(i)は旧スケジューラを用いた場合を表わしているが、APUとCPU共に一時間毎の平均稼働率は全くランダムな変動を呈している。これは投入ジョブの確率的变化に起因して起る。一方、同図の(ii)～(ix)は新スケジューラを用いた場合を表わしているが、APUとCPU共に100%近い稼働率を出来るだけ維持しようとしている形跡がはっきりと認められる。実際、一時間毎の平均稼働率の落込みは全て主記憶競合によるランジョブステップのロールアウトに起因しており、ロールアウトの少い時間帯では必ず高い稼働率を示す。旧スケジューラではジョブストリームの負荷の軽重により稼働率がはげしく変動するのと比して、このことは重要なジョブ・ジョブステップスケジューラの特長であり、新スケジューラの優位性を結論する第五の根拠となる。

(iii) 図5.8と図5.9は実験RII-0と実験II-6によって、ラッシュアワー時における新旧スケジューラのジョブ処理状況を比較したものである。図5.8はジョブストリーム1による比較で

あるが、これには多少説明を要する。航技研システムの主記憶は図 6.1 の如く構成されており、ランジョブステップには特に制限がないが、サービスジョブステップに対しては 256K 語境界をまたいで主記憶を割付けることができない⁴⁾。ジョブストリーム 1 では AM ジョブクラスの処理待ちキューの 3 番目に第一ジョブステップとして主記憶を連続 187K 語要求するサービスジョブステップを有するジョブが並んでいる。主記憶に関する上記理由により、この 187K 語の割付けは図 6.1 の B 空間のみに可能であって、A および C 空間には不可能である。このため、B 空間が分割使用されている間、AM クラスの 3 番目のジョブは処理が開始されても、第一ジョブステップが直ちにロールアウトされ、なかなか主記憶割付けがなされない。図 5.8 の (X) が示すように、旧スケジューラではこのような事情のため 3 番目のジョブ終了まで AM クラスの 4 番目以後のジョブ処理が長時間待たされている。新スケジューラの場合でも同様な事情が発生しているにもかかわらず、3 番目のジョブ終了を待つことなく 4 番目以降のジョブ処理が続行されている。この旧スケジューラにはない柔軟性も新スケジューラの優位性を結論する重要な根拠である。図 5.9 はジョブストリーム 2 による比較を示すが、この場合にはジョブストリーム 1 のような特殊事情が起っていない。これらラッシュアワー実験から一般に、新スケジューラを使用すれば APU が圧倒的に有効利用されること、そのため AP ジョブのスループットが著しく向上すること、並びにシステム立ち上り時期を過ぎれば CP ジョブ（特に SH ジョブ）のスループットも旧スケジューラと同程度保障しえることが結論できる。なお、本実験では、CM, AM および AL ランジョブステップのプール補充のために、6.1.1 節の (V) と同じく 10 ～ 20 分を要している。

6.2 SCD 1 用の運用系

ジョブ・ジョブステップスケジューラに対する適切な運用系に関しては 3.2 節で考察したが、ここではシミュレーション実験結果に基づきその確認を行うと共に、そこで確定しえなかった事項について検討する。

6.2.1 ラン多重度について

SCD1 ではラン多重度 = 3 を CM, AM および AL クラスにそれぞれ 1 つずつ振り分けている。表 5.3 から判断すると、CP ラン多重度 = 1 は適切であるが、AP ラン多重度 = 2 は AP ランジョブステップによる主記憶占有率を多少高くする。しかし、AP ラン多重度 = 1 とした場合、平均 AP 稼働率は 70 ～ 80 % 程度となり、APU 処理能力を十分に活かすことができない。このため、主記憶有効利用の観点から多少問題はあっても、AP ラン多重度 = 2 は航技研システムの特異性からして必要である。

6.2.2 IO 多重度について

SCD1 は IO 多重度 (= 2 ～ 5) を DT クラスに 1 つ振り分け、残りを SH ジョブと CM, AM, AL クラスのサービスジョブステップとが競合して使用するよう設計している。よって、SCD1 ではシステム利用効率の低下をきたさない範囲内で IO 多重度を大きくできれば、柔軟に SH ジョブのスループット向上のために対処しえる。再配置機能を利用して主記憶の有効利用が計れる場合には表 5.3 から明らかなように、主記憶の断片化をほとんど克服しえる。従って、IO 多重度 = 5 としても CP ジョブステップによる主記憶競合のためにシステム利用効率の低下を招くことはない。しかし、再配置機能が利用できない場合には表 5.4 から判るように、主

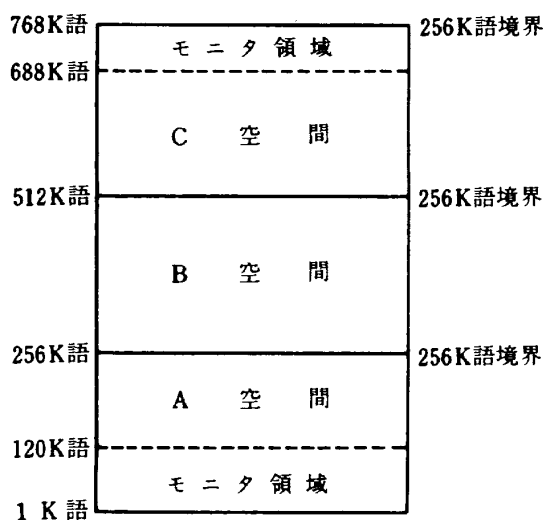


図 6.1 主記憶空間

記憶の断片化によって、IO多重度=5はシステム利用効率の低下を招いている。6.1.2節の(iii)などの実験経験からいえば、リロケータが利用できない場合にはIO多重度=3が最も適切である。

6.2.3 優先権について

表5.3と表5.4および図5.2と図5.3から、運用系3よりは1の方がシステム利用効率が高くかつ安定していると判断しえる。従って、サービスジョブステップとAPランジョブステップの主記憶および実行優先権は共にSY=AM=ALが適切である。また、3.2節で定めたように、APランジョブステップの主記憶および実行優先権をCPランジョブステップより高位に設定すると、APジョブ処理が優先され過ぎ、CPジョブ処理を冷遇する事態の発生が危惧される。しかし、表5.3と表5.4から判るように、平均AP稼働率の向上と共に平均CP稼働率も高い水準まで向上しており、APジョブによるCP寄与率も10%以内であり、またCPジョブの有効主記憶時間比もむしろ低下している等の諸事実から、APジョブによるシステム資源占有の影響は特別問題にならないと結論できる。但し、主記憶利用率が実験Ⅱよりよほど低下した場合にはこの影響が顕著になり、APUとCPUの稼働バランスを回復するために運用系の見直しが必要となろう。

次に、CP(即ち、SH,DTおよびCM)ランジョブステップ間の優先権について検討する。SCD1は常時CMランジョブステップの処理が実行可能なように設計しているから、CPクラスの中ではCMクラスの主記憶優先権を最高に設定する必要があるが、実行優先権はCMジョブのCP寄与率調整のため最低に設定しなければならない。このため、CMジョブによる主記憶占有の影響が懸念されるが、表5.3と表5.4から判るようにCMジョブの有効主記憶時間比は十分に大きく、かつ、SHおよびDTジョブの有効主記憶時間比への影響は見られない。残るSHおよびDTクラス間の優先権はこれらクラスのCP寄与率を目安として設定すればよい。表5.3と表5.4には主記憶優先権をDT>SHとし、実行優先権をSH>DTとした場合について、IO多重度を2から5まで増加していったときのCP寄与率の実験値を示してある。

なお、各ジョブクラスの処理装置寄与率は優先権の組合せのみに基づいて目標値を達成しえるものではない。寄与率は主記憶の利用状況に依存してかなり大巾に変動する。但し、主記憶競合が少なければ寄与率はかなり安定した値を示すので、表5.3は十分参考としえる。

6.3 SCD2の運用系

SCD2に対して適切な運用系も、主要項目については6.2節に同じである。しかし、SCD2ではラン多重度=2をAMとALクラスに振り分けない。同様にIO多重度(=2~5)も特にSHまたはDTクラスに振り分けない。APクラスジョブまたはジョブステップの起動に際しては、AMおよびALクラスの中から重み付き経過時間が最大のものをスケジューリングする。従って、SCD2によるシステムの運用では経過時間に対する各ジョブクラスの重み係数を設定しなければならない。SCD2設計の意図はAPおよびCPジョブ起動のスケジューリングにおいて、起動を優先するジョブクラスとその優先程度とをこの重み係数によって制御・調整する試みにある。

実験Ⅲはこの制御・調整可能な程度をさぐる目的で行ったが、図5.5は実験Ⅲ-1~9の20時間シミュレーション結果に基づいて、一時間刻み累積ジョブ処理件数の推移を図示したものである。同図の(i)と(iii)はCPおよびAPクラス共に、重み係数を大きく設定したクラスのジョブがまず優先的に処理され、小さくしたクラスのジョブ処理が後廻しにされる様子をよく表わしている。しかも、その様相はジョブの到着時間間隔が小さい程、きわだって顕著になる。また、重み係数を小さく設定したクラスのジョブ処理はある程度の個数が溜るのを待って間歇的に行われる傾向が強く表われている。このため、APラン多重度、IO多重度共に、同一クラスのジョブによって占有されている機会が多くなる。以上のことから説明がつくように、実Ⅲ-1~3ではALジョブ処理の機会が少く、多くはAMジョブが2個処理されている。従って、APランジョブステップの占有する主記憶容量は平均200K語と少いから(表3.4)、CPランジョブステップのロールア

ウト頻度も減少する。図 5.4 の (ii) ~ (iv) における処理装置稼働率のきわだった安定性はこのような事情に基づいている。一方、実験Ⅲ-7~9 では逆に AL ジョブ処理の機会が増し、かつ AL ジョブが 2 多重占有していることが多くなる。このため、AP ランジョブステップの占有する主記憶容量は平均 280K 語になる機会が多いから (表 3.4), CP ランジョブステップのロールアウト頻度は増大する。図 5.4 の (vii) ~ (x) はこのような事情を反映して、CP 稼働率が不安定になる様子をよくとらえている。

経過時間に重みを付けない場合は少しジョブ処理の様相が異ってくる。図 5.5 の (ii) から判るように、重み付けをしない場合にはジョブ処理件数がジョブの到着時間間隔に依存せず、SCD1 による実験 I-1 とよく似たジョブ処理状況を呈するが、AP および CP ジョブの各寄与率は当然ながら安定した値を示さない。

以上を総括すると、重み付き経過時間に基づいたジョブ・ジョブステップのスケジューリングはジョブの混雑状況に応じてある程度、ジョブ処理を適応制御し得る。しかし、システム資源の平均的使用状況をかなり変動させることがあり、それが処理装置稼働率を不安定化する要因となり、また、各クラスに対応したジョブ処理件数の予測を難かしくする欠点がある。

6.4 APU システム用ジョブ・ジョブステップスケジューラ

航技研 APU システムに適用することを考えた場合、

- (i) 各ジョブクラス毎にある程度スループットを一定にしえること
- (ii) システム資源の平均的使用状況をかなり安定化しえること
- (iii) 処理装置 (即ち、APU と CPU) 稼働率のバランスをとり易いこと

から、SCD2 より SCD1 の方が適切であると判断する。

しかし、SCD2 はある程度ジョブの混雑状況に適応しえる柔軟性を有している。この特長を活かすために、通常のジョブ混雑状況時には SCD1 によ

ってジョブ処理を行い、異常なジョブ混雑状況に入った時にその異状に適応しえる重み係数を設定して、SCD2 によるジョブ処理に切替えるという方法が考えられる。このようなシステム運用効果は SCD2 の機能を図 4.1 のシステム見直しルーチンに持たせれば、シミュレーション実験によって検討することが出来る。

7. おわりに

従来、計算機システムのジョブ処理にはジョブを単位としてスケジューリングを行い、起動するジョブが決定したならば、その一連のジョブステップを連続して実行するという方式がとられている。本稿では、このジョブ処理方式に対して批判的検討を加え、処理装置の能力をコンスタントに最大限引き出そうとした場合に重大な問題点のあることを指摘し、この問題を解決し得る新しいスケジューラの提案を行った。このスケジューラはジョブのみならずジョブステップをもスケジューリングの対象とすることから、我々はこれをジョブ・ジョブステップスケジューラと命名した。

ジョブ・ジョブステップスケジューラの基本はジョブ処理における実行多重度を、ジョブを単位として数えるのではなく、ジョブステップの処理状況を単位として数えるところにある。即ち、実行多重度をラン多重度と IO 多重度とに分割して、ラン多重度は処理装置バウンドなジョブステップの実行可能個数、IO 多重度はチャネルバウンドなジョブステップの実行可能個数とする。そして、ラン多重度の空きが常にランジョブステップの実行に使用し得るように、ランジョブステップのプールを用意しておく。一方、IO 多重度はランジョブステップがプールされるクラスのジョブのサービスジョブステップの処理およびプールの対象としないクラスのジョブの連続処理のために使用する。

ラン多重度は出来るだけ余分なシステム資源を保有せず処理装置のアイドルを防ぐことを目標として、処理装置の演算能力に見合った必要最小限の値に設定する。また、IO 多重度はシステム稼働率を劣化させない範囲内で、一連のジョブステップを連続処理するクラスのジョブのスループットを増大し、タ

ーンアラウンド時間の短縮を計る上で必要な値に設定する。さらに、プール個数を十分大きく設定するならば、ランジョブステップに先行するサービスジョブステップ段階にエラーのあるジョブのターンアラウンド時間を飛躍的に短縮し得る点も、新スケジューラの特長である。このように、ジョブ・ジョブステップスケジューラはスループットの増大並びに処理装置稼働率の向上とを同時に計るという、よく知られている計算機システムの効率的運用に関するジレンマに対して、一つの解決策を提示し得たものと考えられる。

新スケジューラの有効性を評価するために、航技研APUシステムに応用した場合を例として、適切なアルゴリズムを有するジョブ・ジョブステップスケジューラを試作し、そのスケジューラによってAPUシステムのジョブ処理を行った場合を模擬するシミュレータを作製した。シミュレーション実験により、新旧スケジューラによるジョブ処理効果の有効性比較および新スケジューラに対する運用系の検討を行い、定量的に新スケジューラの優位性と有用性とを十分検証し得たと思う。

ただ、オペレーティングシステムの観点からの詳細な検討は行っていない。従来のもとは違って、新スケジューラによるジョブ処理では一連のジョブステップを連続実行する機能の他に、一時的にジョブステップの実行を保留する機能が必要となる。問題はオペレーティングシステムが後者の機能をサポートする上で惹起してこようが、この点に関してはOS専門家からの御教示を待ちたい。

引用文献

- 1) 富士通；FACOM230-75 APU概要紹介, FACOMジャーナル, Vol. 5, No. 3, pp. 52-54 (1979)
- 2) 三好；航空宇宙技術研究所におけるアレイプロセッサシステム, FACOMジャーナル, Vol. 5, No. 3, pp. 37-43 (1979)
- 3) 中村, 石塚, 吉田；航技研FACOM 230-75 アレイプロセッサシステムセンタルーチンの作成, 航技研資料TM-344 (1978)
- 4) 末松, 吉田, 土屋, 畑山；主記憶有効利用のための諸方策および航技研システムへの適用検討, 航技研資料TM-419 (1980)
- 5) 末松, 中村, 吉田, 原田, 三好；FACOM230-75 アレイプロセッサシステムに対するジョブプロセッシングシミュレータ, 航技研資料TM-384 (1979)
- 6) 末松, 吉田, 中村, 三好；FACOM230-APシステムのシステムシミュレーション, 航技研報告TR-590 (1979)

航空宇宙技術研究所報告 659号

昭和56年3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
