

航空宇宙技術研究所資料

TECHNICAL MEMORANDUM OF NATIONAL AEROSPACE LABORATORY

TM-419

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

末松和代・吉田正廣
土屋雅子・畑山茂樹

1980年8月

航空宇宙技術研究所
NATIONAL AEROSPACE LABORATORY

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*

末松和代** 吉田正廣**
土屋雅子** 畑山茂樹**

1. はじめに

航技研計算センタは国産初のアレイプロセッサ FACOM-APU (以下 APU と略記する) を導入し, FACOM 230-75 (以下 CPU と略記する) と共に非対称マルチプロセッサシステム (以下 APU システムと略記する) を構成して, 昭和53年4月より本運用を行っている。APU システム導入の目的は, 航空宇宙技術の研究に特徴的に現われる各種の離散的計算法に基づいた大規模問題を昭和50年2月から運用してきた FACOM 230-75 マルチプロセッサシステムに比べてより高速に処理するためである。

この非対称マルチプロセッサシステムは今日迄に例を見ないシステムであるが, APU システムの効率的運用とユーザの使い易さを主眼目として作成した APU システム・センタルーチン¹⁾ はほぼ所期の成果を収めている。しかし, センタルーチンのレベルで APU システムの稼働効率を改善しうる範囲には自ずから限界があり, APU システムのより効率的な運用を計るためにはオペレーティングシステム (以下 OS と略記する) そのものを見直し, 改良しうる点や改善すべき点を見出す事が必要となる。

そのような目的のために, 先に開発したジョブプロセッシング・シミュレータ (以下 JPS と略記する) が利用できる。この JPS を用いて種々の運用形態をシミュレーション実験した結果によれば³⁾, どのような運用形態を採用してみても今一つ主記憶利用率の低さが目につき, もっと主記憶の高利用を計る必要があるということが結論された。

主記憶利用率の低さの原因は, 航技研における各種ジ

ョブの主記憶要求量のばらつきに起因する主記憶の断片化を, OSの主記憶管理方式が防止できないところにある。このような観点から現有の主記憶管理方式を見直してみると, 本論でも詳述するように, 第一にロールイン・ロールアウト方式を改善することによって, 第二にアロケータを改良することによって, かなり主記憶の断片化を克服できるという見通しを持った。

本報告はその予測を実証すべく, ロールイン・ロールアウト方式を改善し, 各種の基本的なアロケータによる主記憶断片化克服効果を JPS を用いたシミュレーション実験により比較検討したものである。また, より以上の主記憶利用率の向上を目指すには, 割り付けられた領域に対して再配置機能を利用してできる限り主記憶の断片化を克服しなければならない。本稿では主記憶再配置を実現する方法を提案し, この再配置機能を APU システムに適用した場合の効果を示す実験も行っている。

2. 主記憶管理の方式と改善策

本節では, 現有の主記憶管理方式を説明する際に必要と思われる事項と, 主記憶管理に際して APU システムから要請される事項をまず述べ, 現有の処理方式を見直しながら改善すべき点を明らかにする。

2.1 APU システムの主記憶構成⁴⁾

APU システムの主記憶は図 2.1 のように構成されている。同図において, 主記憶装置 I から IV はそれぞれ1つのメモリアクセスコントローラを持ち, 2モジュールを実装している。モジュール 0 ~ 3 は 128 KW から成り, モジュール 4 ~ 7 は 64 KW から成っている。各モジュールは 4バンクから成っており, 各バンクは 2W の読出し巾を持ち, それぞれ他のバンクとは独立に動作できる。各モジュールは 4 ウェイ・インターリーブを行っており, さらにモジュール間に渡ってもインターリーブを行っている。したがって, 0 ~ 512 KW 目までは 32 ウェイ・イン

* 昭和55年7月2日 受付
** 計算センター

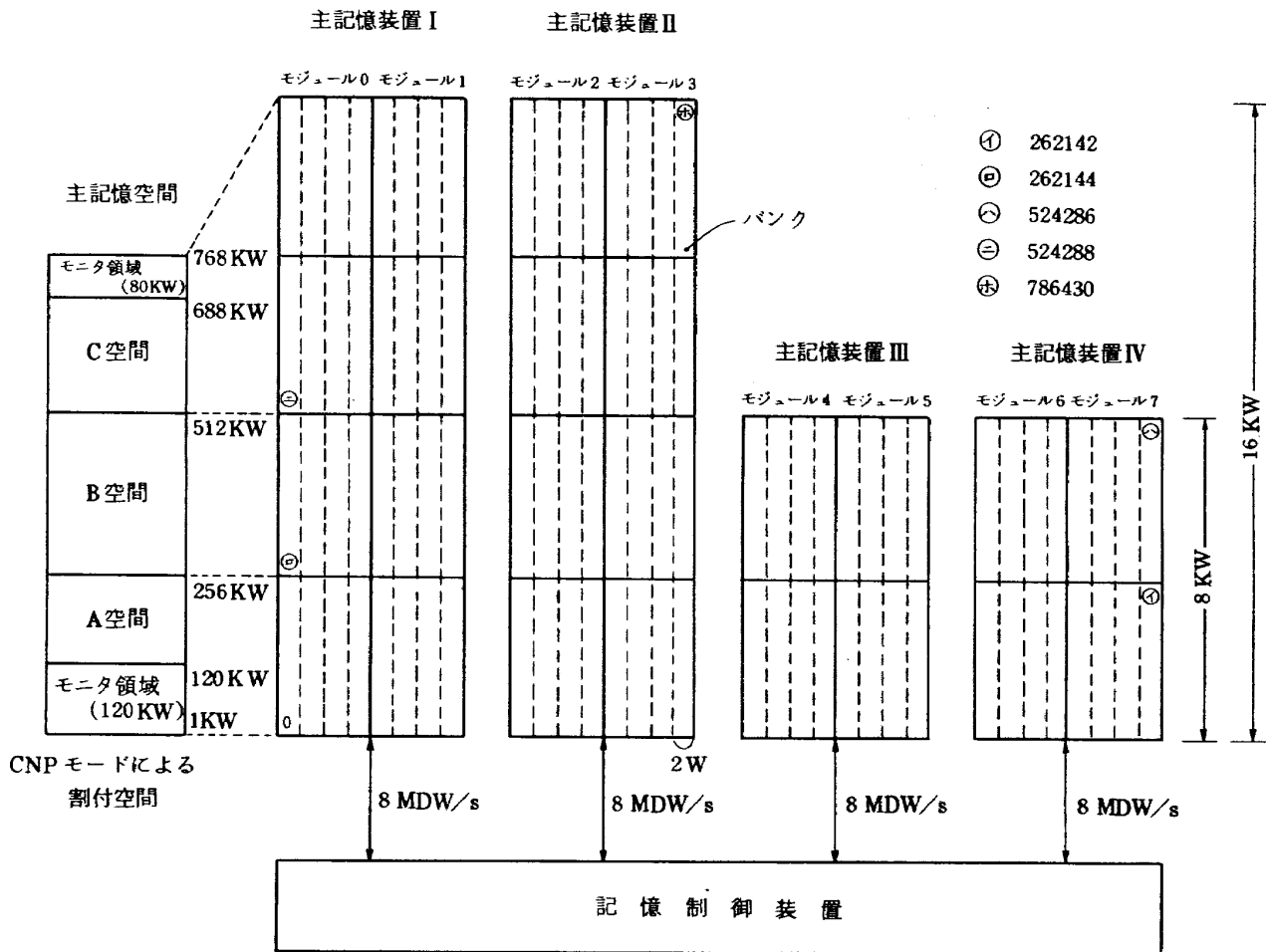


図 2.1 APUシステムの主記憶構成図

ターリーブであるが、512~768KW目までは16ウェイ・インターリーブとなる。よってメモリスイクルタイムが990nsであるから、番地ずけを図2.1のようにすることによって、0~524287番地までは32 MDW/s^{注1)}のデータ供給能力をもつが、524288~786431番地までは16 MDW/sのデータ供給能力しかないことになる。このデータ供給能力の相違はCPジョブステップに対しては全く影響を及ぼさないが、APジョブステップに対しては影響が大きい^{注2)}ため主記憶を割り付ける際には低データ供給能力領域への割り付けはできるだけ避ける必要がある。^{注2)}航技研におけるこの特殊事情は、アロケータの設計においては十分考慮されなければならない。⁵⁾

2.2 CNPモードとCPモード

FACOM 230-75にはFACOM 230-60と上向きの互換性をもたせるために、主記憶の利用法としてCNP・

CPの2つのモードが用意され使い分けられている。ここでCNPモードとは主記憶空間を256KW単位に区切って使用するモードであり、CPモードとは1MWまでの主記憶空間を完全に連続して使えるモードである。CNPモードではデータあるいは命令が256KW境界にさしかかると境界を越える事が許されないため、そのデータあるいは命令はその256KW単位の主記憶空間の先頭番地へ戻される。したがって、APUシステムではCNPモードによる割り付け空間は図2.1の様に3つ存在し、当然の事ながらそれぞれの境界では主記憶割り付けの断片化が起り易い。また、CPモードによる割り付け空間は明らかに1つである。

なお、APUシステムでは総主記憶容量768KWのうち、1~120KW目および689~768KW目までの合計200KWがメモリー領域にとられているため、アロケータが割

注1) MDW/sはMega-Double-Word-per-secondの略で1秒当りの転送能力を示す。なお、1kwは1024語を意味する。

注2) CPジョブステップとは主としてCPUで処理されるジョブステップをいい、APジョブステップとは主としてAPUで処理されるジョブステップをいう。

り付けできる空間は 121 ~ 688KW 目までに限られる。後の説明の都合上、主記憶空間を図 2.1 の様に 3 つに分割し、121 ~ 256KW 目までを A 空間、257 ~ 512 KW 目までを B 空間、513 ~ 688 KW 目までを C 空間と名付ける。明らかに A、B 空間は 32 MDW/s のデータ供給能力をもつが、C 空間は 16 MDW/s の供給能力しかない。

2.3 主記憶管理方式

各ジョブは翻訳から結合編集を経て実行段階に至るいくつかの順序付けられたジョブステップ列に分割されて処理される。主記憶使用の要求は、第一にそれらの各ジョブステップの開設時に発生する。各ジョブステップの要求する主記憶量は一つの連続した主記憶空間に確保される必要はなく、表 2.1 に示される様な 2 個から 4 個の領域に分割して確保できれば良い。^{注)}各ジョブステップに対する主記憶の割り付けは、同表の分割番号順に必要な領域が全て確保できたときに完了する。例えば、APRUN ジョブステップに対しては、6 KW の領域、ユーザ実行形式プログラム領域、および 4KW の領域が、この順番に確保できれば割り付け完了である。しかし、1 個でも分割番号の領域が確保できなければ、その時点で 2.4 節で詳述するロールアウト処理が行われる。そこで再度割り付けが試みられる。

表 2.1 各ジョブステップの主記憶要求量
単位 (KW)

領域分割 番号 プログラム名	1	2	3	4
APFORTRAN	6	45	15	α
CPFORTRAN	6	28	10	46
LIED	6	11	30	
LIBE	6	13	13	
UTILITY	6	α		
APRUN	6	α *	4	
CPRUN	6	α *	4	

注意 表中 * 印のものは CP モード、他は CNP モードで割り付けられる
 α については付録 B 参照のこと。

さて、分割された個々の領域を実際に主記憶空間上に割り付けるためには、2.5 節で詳述するアロケータが必要となる。表 2.1 に示す様に、APU システムではユーザの実行形式プログラムの領域のみが CP モードで割り付けられ、その他全ての領域は CNP モードで割り付けられなければならない。このため、アロケータには両方のモードによる割り付け機能が必要となる。アロケータは各ジョブステップ毎にその分割個数の数だけ呼ばれ、個々の領域毎に決められているモードでの割り付け処理を行う。例えば、APRUN ジョブステップの主記憶割り付けを完了するまでに、アロケータは CNP モードの割り付けのために 2 回、CP モードの割り付けのために 1 回呼ばれる。

ロールアウト処理によっても主記憶への割り付けが完了できなかったジョブステップは、補助記憶装置へロールアウトされる。また 2.4 節で述べる様に、主記憶上にあるジョブステップでも主記憶優先権の高いジョブステップの割り付けの際にロールアウトされる事がある。主記憶使用の要求の第二は、これらロールアウトされているジョブステップのどれかを、主記憶へロードする事が要請された時点に発生する。このロード手続きは 2.4 節で詳述するロールイン方式に従って行われる。ロールイン要請のあったジョブステップを主記憶へロードするには、先に述べた開設時と同じ方法がとられるが、分割された個々の領域の割り付け順序は次に述べる様に異なる。すなわち、割り付け要求に対する処理の場合は分割番号順に割り付けられるが、ロールイン処理の場合は分割番号 1 の 6 KW の領域が割り付けられた後、大きい領域から順に割り付けられる。

主記憶上のジョブステップがロールアウトされる場合、それまで占有していたいくつかの領域はその解放処理後は空きとなって、割り付けが可能となる。

1 つのジョブステップは、開設されてから主記憶上に何回かロードされるうちに処理装置から必要な AP および CP サービスを受け終ると、最終的に主記憶解放がなされて、そのジョブステップは閉設される。

以上が主記憶管理の概要であるが、次に現有のロールイン・ロールアウト方式とアロケータを詳述し、問題点をさぐり改善策を考える。

2.4 ロールイン・ロールアウト方式⁵⁾

開設時に主記憶の割り付けができなかったジョブステ

注) 表 2.1 において、APFORTRAN から UTILITY までと CPRUN が CP ジョブステップであり、APRUN のみが AP ジョブステップである。また APFORTRAN から UTILITY までをサービスジョブステップといい、CPRUN と APRUN を RUN ジョブステップという。

ップは図 2.2 (i) に示すところのロールアウト方式による処理を受ける。各ジョブステップには APU システム運用上の定数として主記憶優先権が付けられているが、まず主記憶上にあり、かつ割り付け要求のあったジョブステップの主記憶優先権より下位の優先権をもつジョブステップをさがす。その様なジョブステップが複数個ある場合にはその中で最下位のものから順に 1 個ずつロールアウトし、その都度割り付け要求のあったジョブステップの割り付けを、前節の方法で試みる。これにより割り付けが完了したならば引き続き下で述べるところのロールイン処理を受ける。一方、下位の主記憶優先権をもつジョブステップが全然ないか、あってもそれら全部をロールアウトしても主記憶への割り付けができない場合には、割り付け要求のあったジョブステップ自体をロールアウトして処理は終了する。

以上が現有のロールアウト処理の内容であるが、これには大きな欠点がある。すなわち、下位の主記憶優先権をもつジョブステップを全てロールアウトしても割り付け要求のあったジョブステップを主記憶上に割り付けできない場合には、当然この処理のためにロールアウトされたジョブステップを主記憶上へ復旧しなくてはならない。それにもかかわらず、現有の方式ではそれらはロールアウトされたままである。したがって、ロールアウト処理の際、図 2.2 (ii) に示すように、必ずロールイン処理を行うならば、元通りには復旧できない事があるにしても主記憶利用率は多少とも向上できるはずである。

ロールイン処理は各ジョブステップの閉設時と、上で述べたロールアウト処理の一部からの要請で呼ばれ、図 2.2 (iii) に示す処理を行う。すなわち、ロールアウトされているジョブステップのうち、主記憶優先権の最上位のものから順に割り付けを行い、割り付けができないジョブステップがあった時点でロールイン処理は終了する。

このロールイン処理にも次のような欠点がある。それは主記憶優先権の高いジョブステップがロールインできない場合でも、それより下位の優先権をもつジョブステップでロールインできるものが十分に存在しうるからである。しかし、現有の処理方式ではその様なジョブステップはロールインする事ができない。したがって図 2.2 (iv) のように、全てのロールアウトジョブステップに対してロールイン処理を試みるように改善し、図 2.2 (ii) と組合せるならば、多少とも主記憶利用率の向上が望めよう。

2.5 アロケータ

各ジョブステップの主記憶上の領域はアロケータによって割り付けられる。その場合、割り付け処理時には表 2.1 に示された分割番号順に、ロールイン処理時には分

割番号 1 を除いて大きい領域から順に、決められたモードで割り付けられる。一般に複数の領域を主記憶上に割り付ける場合、大きい方の領域から順に割り付ける方がより効果的である。その意味から、割り付け処理の場合もロールイン処理の場合と同じ様に、できるだけ大きい領域から順に割り付けるべきである。また、アロケータは主記憶の断片化が発生しにくいように設計されなければならないが、APU システムでは 2.1 節および 2.3 節に述べた理由により、更に CP モードによる割り付けは極力 C 空間を避けたいという要請がある。一方、CNP モードによる割り付けに関してはそのような制約は全くない。したがって、これら両者の事情により、アロケータは少なくとも CNP モードの領域に対しては C 空間から割り付けを始めるべきであり、CP モードの領域に対しては C 空間の割り付けは最後にすべきである。この事は、APU システムの主記憶構成と関連して要求される事であり、アロケータの満たすべき基本条件である。容易にわかるように CNP モードによる主記憶割り付け方法は、上記の条件により自ずから一義的に定まってしまうが、CP モードにより主記憶を割り付けるには図 2.3 に示す 4 つの基本的な方法があり、それを以下で詳しく述べる。同図に示した基本的なアロケータは、更に種々の機能を追加したり、主記憶を増設することによって、その改良による効果が検討できる。

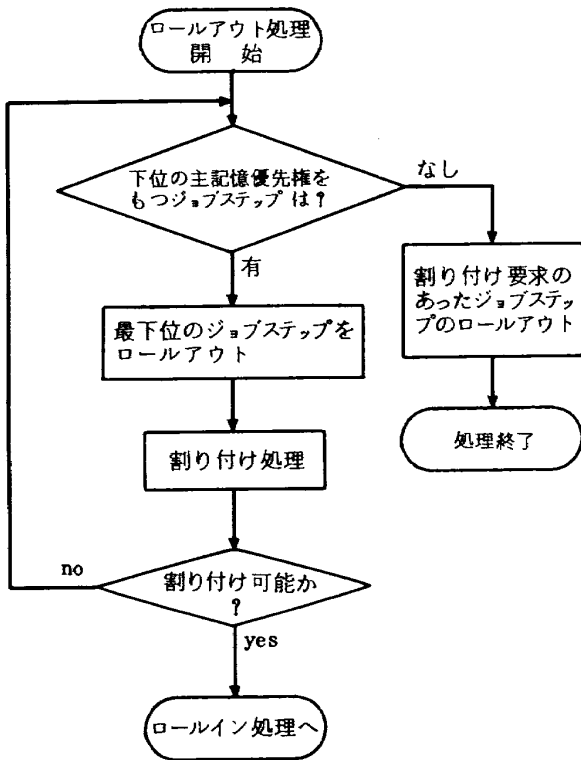
本稿の 1 つの主題は、JPS を用いたシミュレーション実験でこれらのアロケータによる主記憶割り付け状況を比較検討し、どのアロケータが APU システムにとって最も有効であるかを特定する試みにある。

2.5.1 アロケータ 0

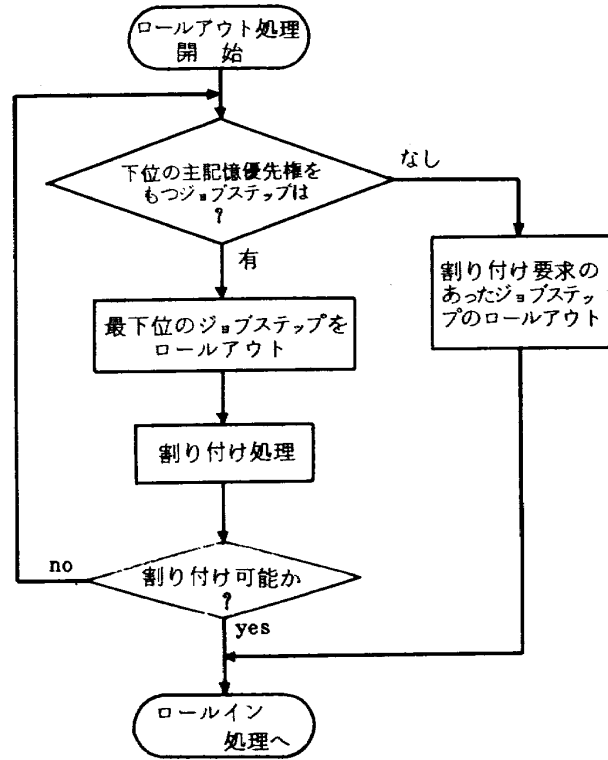
図 2.3 の (i) は現有アロケータ (アロケータ 0 と呼称) の割り付け手順と方向を示したものであり、全て高位番地から低位番地の方向に空領域をさがして割り付けを行うところに特徴がある。CNP モードによる割り付けはまず C 空間へ試み (手順①)、割り付けできなければ次に B 空間へ試み (手順②)、それでも割り付けできなければ最後に A 空間へ試みる (手順③)。

CP モードによる割り付けは、まず B 空間のみへ試み (手順①)、割り付けできなければ次に A 空間のみへ試み (手順②)、それでも完了しなければ A と B 空間にまたがって試みる (手順③)。割り付け要求量が 256 KW 未満のものはこの手順③までで割り付け処理は打ち切られるが、256 KW 以上のものは次に B と C 空間にまたがって試みられ (手順④)、それでも完了しなければ最後に A、B および C 空間にまたがって試みられる (手順⑤)。

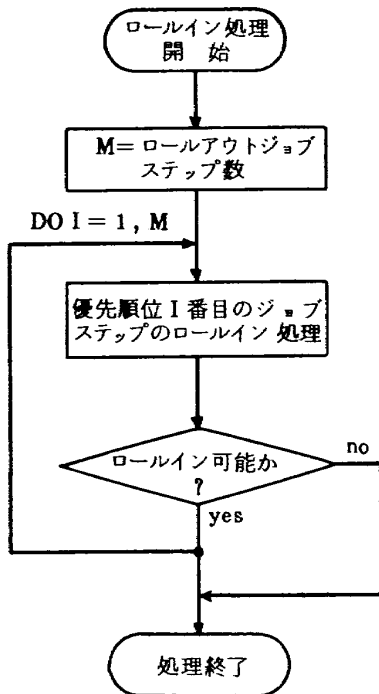
この現有アロケータには APU システムで使用する時



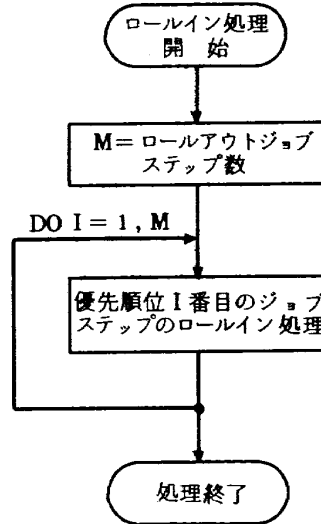
(i) 現在のロールアウト方式



(ii) 改善後のロールアウト方式



(iii) 現在のロールイン方式



(iv) 改善後のロールイン方式

図 2.2 ロールイン・ロールアウト方式

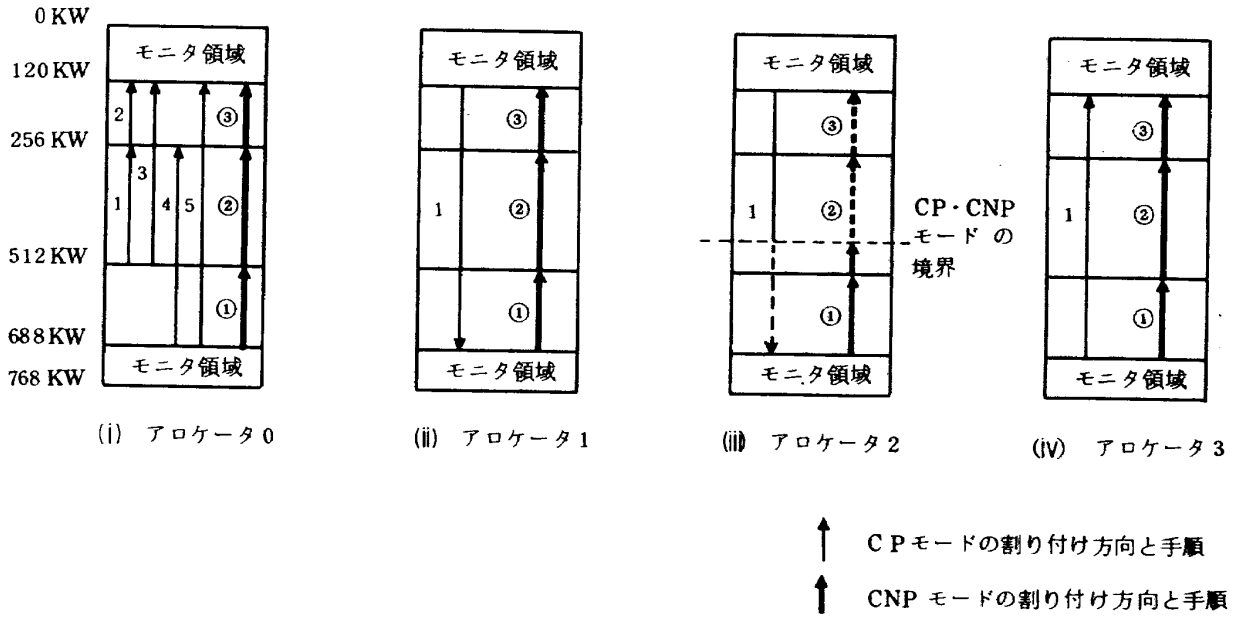


図 2.3 アロケータ

に次の2点に関して問題がある。第一に、このCPモードによる割り付け方法では、主記憶の断片化が発生し易くなるという点である。なぜならば、手順①と②はもととも空領域を作り易いからである。例として図2.4の場合を取り上げよう。まず手順①によってB空間に空領域Ⅰができたとする。つぎに手順②によってA空間に空領域Ⅱができたとする。このような主記憶利用状況の時に要求量Ⅲの割り付け要求が発生したとする。これが $I < III$ 、 $II < III$ で、 $III < I + II$ なる大きさのものであったならば、(i)で明らかなように手順③によってもⅢを主記憶上に割り付ける事ができない。しかし、もし手順①と②がなく、割り付けが手順③から始まっているならば、

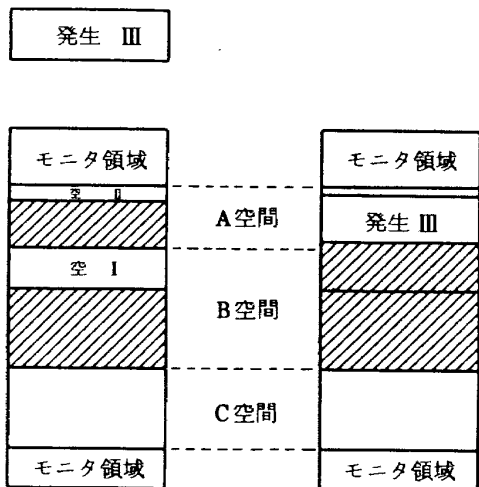
(ii)のようにⅢを割り付ける事ができる。また手順④も同じ理由から省いた方が良いであろう。このアロケータの第二の欠点は、手順④(または手順⑤)にある。これは“CPモードによるC空間への割り付けは最後に行うべきである”というAPUシステムにおけるアロケータ設計の基本的条件に明らかに反している。

2.5.2 アロケータ1

アロケータ0で述べた2つの欠点のうち、特に主記憶の断片化の克服を主眼とするならば、図2.3の(ii)に示すような主記憶割り付け方式(アロケータ1)が考えられる。同図の(i)と(ii)から明らかなように、CNPモードによる割り付け方法はアロケータ0と同じである。CPモードによる割り付けはアロケータ0とは逆に、低位番地から高位番地の方向へ行い、しかもA、B、およびC空間にまたがって試みる(手順①)。アロケータ1は確かにCPモードによるC空間の割り付けを最後に行っており、2.5.1節で述べたような断片化の発生に対する防止策も考慮されている。更に、CNPモードによる割り付けのために生じた空領域があれば、そこへのCPモードによる割り付けを可能にして、主記憶の利用率の向上を計っているが、反面、CPモードによる割り付けがC空間に行われる可能性を高める点が欠点にもなっている。

2.5.3 アロケータ2

アロケータ0の2つの欠点のうち、特にC空間へのCPモードによる割り付けを回避することに主眼をおけば、図2.3の(iii)に示す割り付け方式(アロケータ2)が考えられる。アロケータ2のCNPとCPモードによる割り付け方向と手順はアロケータ1と同じであるが、CNP



(i) 手順①, ②, ③の場合 (ii) 手順③の場合

図 2.4 アロケータ0による主記憶割り付け状況

(またはCP)モードによる割り付けはその時点で割り付けられているCP(またはCNP)モードの領域の最高(または最低)番地まで打ち切る。よって、アロケータ0と1とでは、CNP・CP両モードで割り付けられた領域が混在しているのとは対照的に、アロケータ2ではCPモードで割り付けられた領域が低位番地側に、CNPモードで割り付けられた領域が高位番地側にあって両者は混在していない。このことはC空間へのCPモードによる割り付けの可能性をできるだけ少なくする一方、主記憶の利用率を低下させる点が欠点にもなっている。

2.5.4 アロケータ3

もし、図2.1で示したAPUシステムの主記憶装置ⅢとⅣを増設して、1024KWまで一様に32MDW/sのデータ供給能力をもつ主記憶構成にできた場合には、アロケータは主記憶の断片化の克服のみに注目して設計すれば良い。この場合、比較の対象とすべきものはアロケータ1と図2.3の(i)に示した割り付け方式(アロケータ3)の2つである。アロケータ3はこのような目的のためにのみ取り上げたものである。

2.6 リロケータ

アロケータをAPUシステムに最も適したもので置きかえて、出来るだけ主記憶の断片化を防いだとしても、主記憶の要求および解放の、時刻も容量も共に事前に知る事ができないために、空領域が主記憶全体に分散してしまう可能性は十分にある。このような主記憶の断片化を克服し、更に主記憶利用率の向上を求めようとするならば、割り付けられている領域を強制的に再配置(リロケーション)して空領域をできるだけ1箇所に集めるための作業が必要になる。このような主記憶再配置機能を有するものをリロケータと呼ぶことにする。本節では、APUシステムの主記憶管理ルーチンにリロケータを組込む方法、それによる主記憶利用率の向上の程度、およびオーバーヘッドの大きさを調べる目的で、2.5節で示したアロケータに対して適当と思われるリロケータを設計してみる。そのために考慮しなければならないいくつかの側面について、まず述べておく。

主記憶再配置を実行する手段としてはロールイン・ロールアウト命令を用いる方法と、MVB(Move Byte)命令を用いる方法とがある。前者による場合には、移動したい領域がCP・CNPのどちらのモードで割り付けられていても、それを主記憶上のどこへでも再配置が可能である。しかしながら、ロールイン・ロールアウト命令で再配置するのに要するオーバーヘッドはMVB命令により再配置するのに要するオーバーヘッドに比べて大きすぎる。このため後者の方法を採用せざるを得ない。ただし、M

VB命令で主記憶の再配置を実行する際には、APUシステムの側から次のような制約を受ける。第一に、CPモードで割り付けられている領域は主記憶上どこへでも再配置可能であるが、A(またはB、C)空間へCNPモードで割り付けられている領域はA(またはB、C)空間内だけにしか再配置できない。第二に、C空間へ(またはC空間へまたがって)CPモードで割り付けられている領域はできるだけA、B空間内へ再配置しなければならない。これらの制約のもとで考えると、主記憶再配置の方式は複雑さの点において、CP・CNPモードで割り付けられている領域が主記憶上で混在している場合としていない場合とでは以下に示すように対照的になる。なお、“低(または高)位番地の方へ移動”とある箇所は全て“低(または高)位番地の方へ低(または高)位にある領域の順に移動”という意味に使っている。

2.6.1 リロケータ1

アロケータ2により割り付けられた領域を効果的に再配置し、主記憶の断片化を動的に克服する機能を有するものを、リロケータ1と名付ける。アロケータ2を用いた場合にはCP・CNPモードで割り付けられた領域が、主記憶上で混在していないのが特徴であり、そのためにリロケータ1は非常に簡単に実現できる。すなわち、図2.5の(i)に示した状況例のように、CPモードで割り付けられている領域は全て低位番地の方へ可能なだけ移動し、CNPモードで割り付けられている領域は全て高位番地の方へ可能なだけ移動する。このように最大限の再配置を行えば、CP・CNPモードの領域の境(同図に一点鎖線で示した箇所)がC(またはB、A)空間にあれば、空領域の数は1(または2、3)個にしよう。

2.6.2 リロケータ2

アロケータ0、1または3を用いた場合は、CP・CNPモードで割り付けられた領域が互に主記憶上で混在している。このために、上述した制約を満たしつつ空領域の個数を減じようとするならば、再配置の方法は多種考えられ、また複雑にならざるを得ないが、主記憶の断片化の克服のために簡単でかつ効果的と思われるリロケータは次の手順で再配置をする方式のものである。

step 1. A空間に割り付けられている領域を、A、B空間にまたがっているものも含め、全て低位番地の方へ可能なだけ移動し、更に、B、C空間に割り付けられている領域を低位番地のものから順次調べて、CNPモードで割り付けられている領域が見つかるまで全て低位番地方向へ可能なだけ移動する。(状況例図2.5の(ii)~(iv))

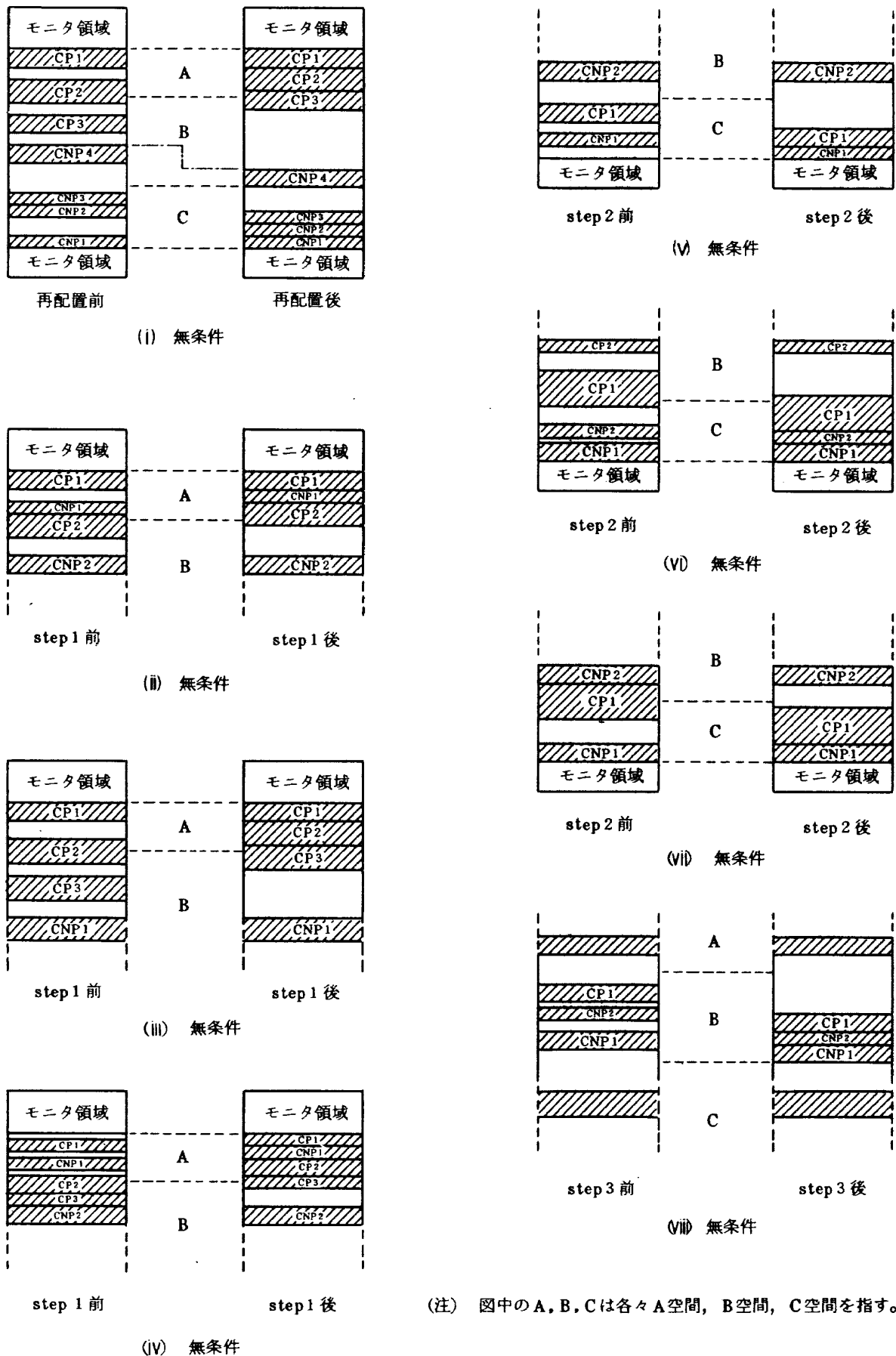


図 2.5 リロケータ 1 による状況例

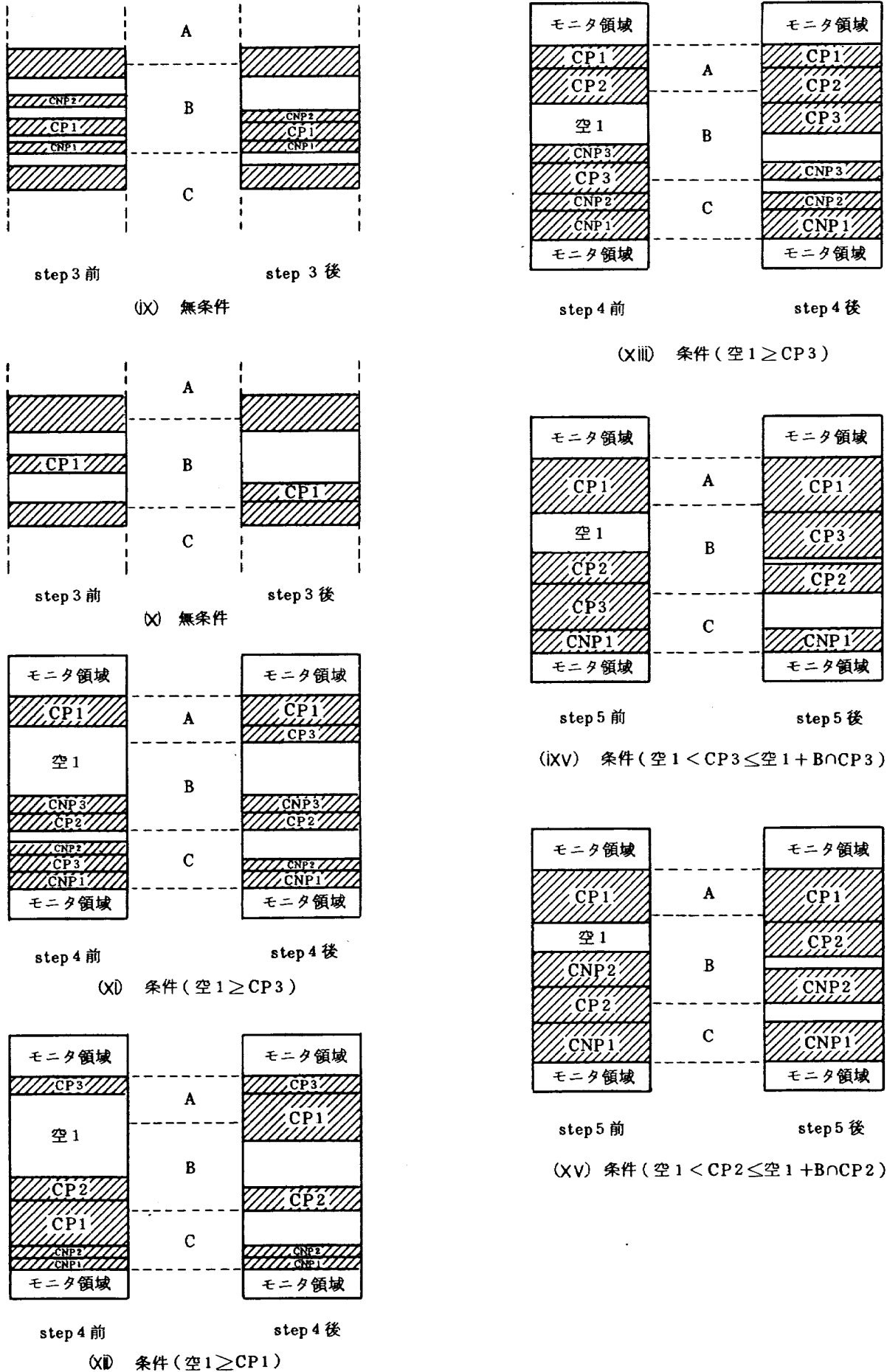


図 2.5 リロケータ 1 による状況例 (つづき)

step 2. C空間に割り付けられている領域をB空間にまたがっているものも含めて、全て高位番地の方向へ可能だけ移動する。(状況例図 2.5 の(V)~(VII))

step 3. B空間に割り付けられている領域を、全てB空間の高位番地の方へ可能だけ移動する。(状況例図 2.5 の(VIII)~(X))

step 4. こうして拡大された空領域へ、C空間に(またはC空間にまたがって)CPモードで割り付けられている領域を、移動可能ならば空領域の低位番地の方へ移動し、その結果B空間内に割り付けられている領域で移動可能なものができたならば、B空間の高位番地の方へ可能だけ移動する。(状況例図 2.5 の(XI)~(XIII))

step 1 ~ 3 で再配置が終了する場合には、どの領域の移動回数も高々1回である。しかし step 4 まで進むと、いくつかの領域に対しては2度目の移動が行われる。また、上述した制約の第2点が重要視されるならば、更に次の step を用意しておかなければならない。

step 5. C空間にまたがってCPモードで割り付けられている領域 D_1 と、その他にB空間内へ割り付けられている領域が少なくとも1個あるとする。領域 D_1 の大きさを \bar{D}_1 、step 1 ~ 3 で拡大された空領域 D_2 の大きさを \bar{D}_2 、B空間上にある D_1 の大きさを $\bar{D}_3 (= B \cap \bar{D}_1)$ とし、 \bar{D}_3 がある大きさ以上なら再配置を実行する。もし、

$$\bar{D}_2 < \bar{D}_1 \leq \bar{D}_2 + \bar{D}_3$$

が成立するならば、 D_1 を空領域の低位番地の方へ移動し、かつB空間内の領域を全てB

空間の高位番地の方へ可能だけ移動する。
(状況例図 2.5 の (IX)~(XV))

step 4 が実行されず、step 5 が実行される場合には、いくつかの領域に対しては2度目の移動が行われる。

以上から明らかなように、リロケータ2による再配置後、主記憶上の空領域の個数は高々2個にしうる。

2.6.3 再配置実行の条件

さて、領域の再配置の実行条件について考えてみる。まず他の事項を無視し、主記憶の断片化のみを最大限に防ぐ目的の場合から考える。そのためには主記憶の解放という事象が発生した時、常に再配置を実行するのが最善であろう。その場合、リロケータはジョブステップ閉設時と図 2.2 の(ii)における“最下位のジョブステップをロールアウト”した直後に呼ばれる。以上の手続きは確かに断片化の防止に対して最善であろうが、再配置が頻発するとオーバーヘッドが増大する。さらに、図 2.6 に示す状況例のような、あまり効果的とは思えない再配置も実行してしまう。従って多少の断片化を許すことによって、主記憶利用率はどの程度低下し、どの程度オーバーヘッドを軽減できるかを調べておく必要が生じてくる。なお図 2.6 はリロケータ2を用いた場合を示しているが、解放領域2と3は解放後空領域となってもなら再配置を起動する要因にならない。このように、リロケータ起動を伴わない解放領域を有効空領域と呼ぶことにする。一方、解放領域1と4は解放後リロケータによってそのすきまがうめられる。このようにリロケータ起動の要因となる解放領域を非有効空領域と呼ぶことにする。

この目的のためには、リロケータを主記憶解放の時点毎ではなく、さらに次の条件を満たしているときにのみ呼ぶようにしておけば良いであろう。 t を主記憶解放の時刻、 $C_r(t)$ を時刻 t における非有効空領域となる主記

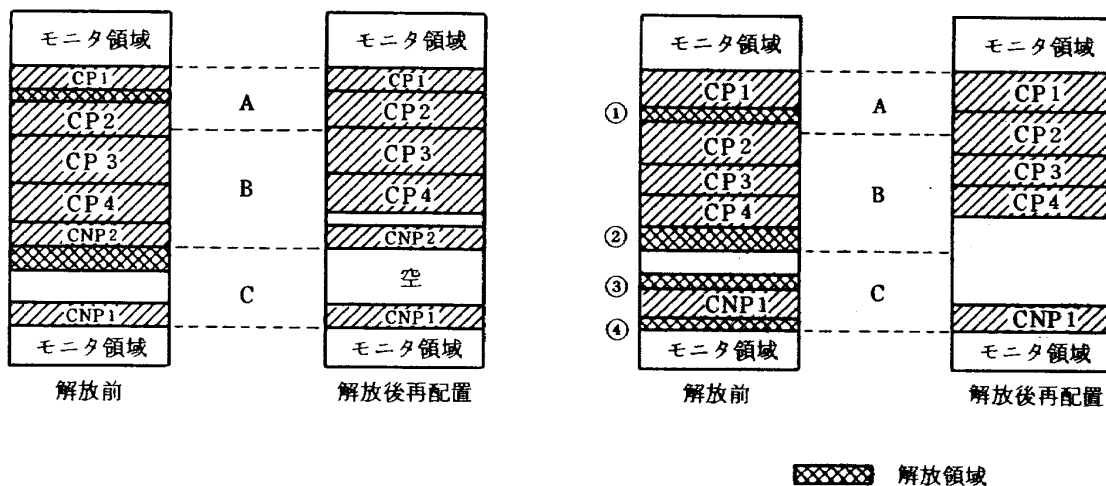


図 2.6 効果的でない再配置状況例

憶解放量, $n(t)$ を時刻 t における非有効空領域の個数, $C_e(t)$, $e = 0, 1, \dots, n(t)$ を時刻 t における e 番目の空領域量とし, $C_0(t) = 0$ とする。いま, C_s を非有効空領域の総和の許容量として規定し, 時刻 t で主記憶解放の事象が発生したとする。このとき

$$C_r(t) + \sum_{e=0}^{n(t)} C_e(t) \geq C_s$$

が成立するならば再配置を実行し, そうでなければ時刻 t では再配置を行わない。このように C_s をパラメータとして実験するならば, 主記憶の断片化と利用率の関係, および利用率とオーバーヘッドの関係をとらえることができて, これらのトレード・オフ関係から最も効率の良い C_s の値が推定できるであろう。

2.6.4 補足

確かにリロケータ2はリロケータ1より主記憶利用率は向上させ得る。しかし, リロケータ1作成の簡単さに比べて, リロケータ2では非常に複雑なアルゴリズムを用意しなければならない。このため我々はリロケータ1による主記憶利用率向上の度合いをまず実験し, それでは満足できない場合にリロケータ2を作成するという方針をとった。また, 再配置実行のタイミングも上記のような一般的アルゴリズムとせず, 主記憶解放事象発生時点と, ジョブステップ開設時に割り付けできない時, およびロールイン処理を行う前とに限定した。これら3つのタイミングが再配置効果を調べる上で最も重要であると考えたからである。

3. シミュレーション実験

前章で設計したアロケータやリロケータの有効性, 主記憶の増設効果, 256KW境界存在の影響等の評価は, 先に開発したジョブプロセッシングシミュレータ(JPS)を用いて行う。

3.1 JPSの概要

図3.1に基づいて, シミュレータの概要を以下で説明する(詳しくは文献2を参照)。

ジョブ発生ルーチンは計算機システムに投入されるジョブとその属性を全て模擬して, ジョブが発生する都度ジョブ発生情報をジョブスケジューリングルーチンへ渡す。ジョブスケジューリングルーチンは, 発生したジョブをそれが属するジョブクラスの待ち行列へ登録する。また, ジョブの発生時とジョブ終了時には, ジョブクラス毎の指定多重度に余裕があれば, ジョブクラス毎の待ち行列からそのジョブクラスのジョブを起動する。さらにジョブ開設時には開設処理を, 閉設時には閉設処理を行う。

ジョブステップスケジューリングルーチンは, ジョブ開設処理に引続いて, そのジョブの第一ジョブステップの開設処理を行い, ジョブステップ終了時にはジョブステップの閉設処理と後続のジョブステップの開設処理を行う。もし, 後続のジョブステップがない場合には, ジョブスケジューリングルーチンに対し, ジョブ閉設処理, つづいてそのジョブクラスのジョブの起動を促す。

主記憶管理ルーチンはジョブステップ開設処理に引続

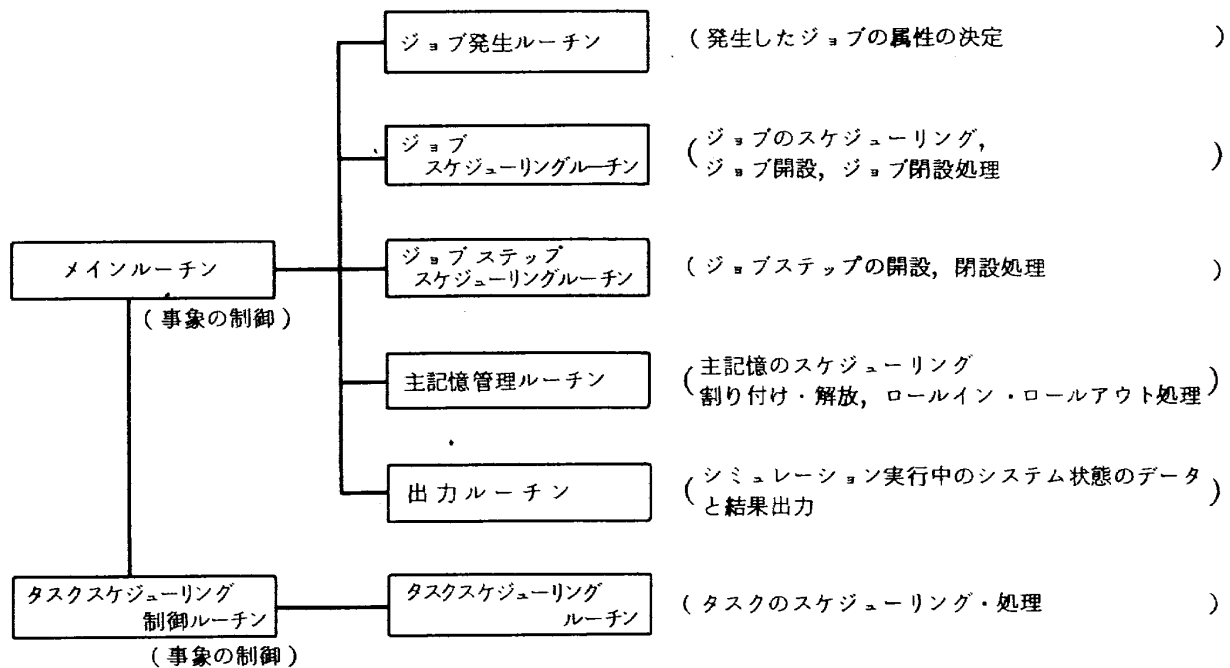


図3.1 JPSのプログラム構成

いて、そのジョブステップに対して主記憶の割り付けを試み、またジョブステップ終了時にその解放を行う。主記憶の割り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。主記憶を割り付けられたジョブステップは、AP または CP 実行待ち待ち行列に登録される。

タスクスケジューリングルーチンは、APU 又は CPU の実行が中断される事象（入出力要求、タイムスライス切れ）発生都度起動され、AP または CP 実行待ち待ち行列から最も優先権の高いタスクに APU または CPU の使用权を与える。図 3.2 は AP タスクと CP タスクの処理過程の詳細を示す。同図の(ii)がいわゆる非対称マルチプロセッサによるタスクの処理の流れを模擬している。

メインルーチンはジョブ単位の事象を制御し、タスクスケジューリング制御ルーチンはタスク単位の事象を制御する。これらのルーチンは最も近い将来に発生する事象を決定することによって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へ制御を渡す。

出力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、一定時間間隔毎に解析に必要な情報を収集する。

3.2 実験環境の設定

本稿におけるシミュレーション実験では、上記 JPS の実験環境を規定する諸定数を以下に示すように設定した。

3.2.1 ジョブの発生

ジョブ集団から計算機システムへジョブが到着すると、模擬ジョブが発生したことになる。JPS ではこのジョ

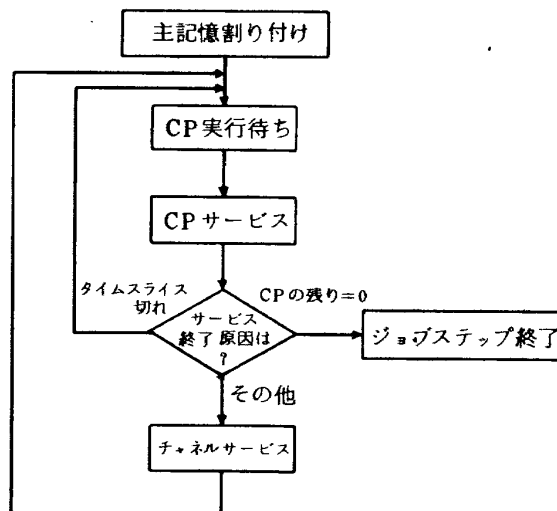
ブ到着に対する確率モデルをポアソン過程としているから、ジョブの到着時間間隔は指数分布に従って決定される。この指数分布は平均値のみによって完全に規定される。

昭和53年度のアカウント情報統計解析結果によると、ジョブの到着時間間隔の平均値は 90s と推定される。しかし、このジョブ発生率でシミュレーション実験を行うと、いくつかのジョブクラスの待ち行列が度々空になる。また、平均値を 45s としても DT クラスの待ち行列が空になる時がある。ジョブクラス毎の待ち行列で空になるものがあると、その期間実行多重度は低下し、システム資源使用要求の負荷が軽減され、見掛上各システム資源の利用率は低下する。

本実験の目的はシステム資源の実質的利用率を向上させ得る方途をさぐることにあるから、見掛上の利用率低下があっては精度の良い判断ができない。以上の理由から、本実験では平均値を 30s として、航技研ジョブストリームとしてはかなりの混雑時に相当する状態で実験を行った。この場合には、実験立ち上り時期を除いて、どのジョブクラスの待ち行列も空になることはない。

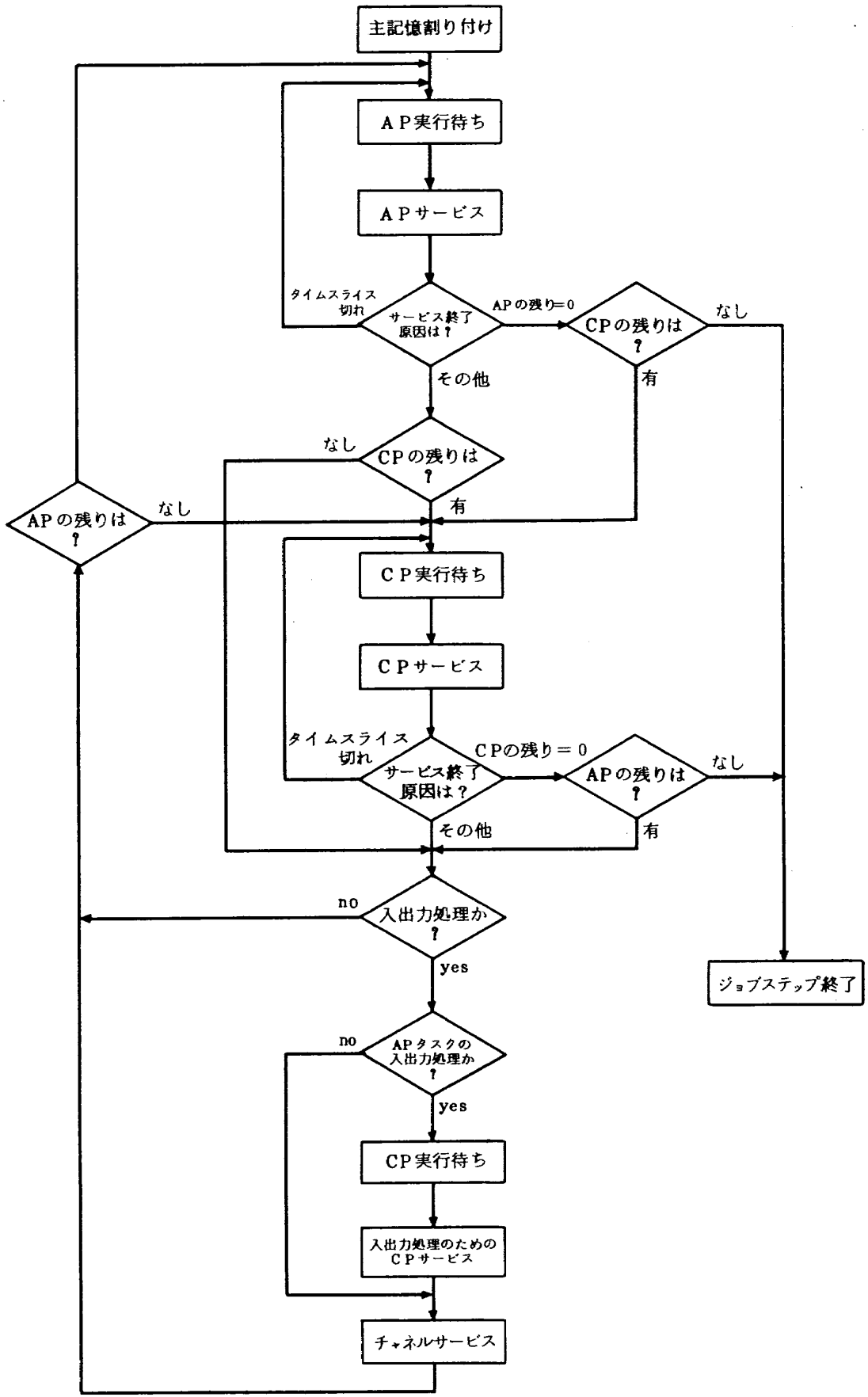
3.2.2 ジョブの属性

JPS へ投入されたジョブは、ジョブ集団のジョブクラス構成比率、ジョブクラス内ジョブステップ列構成比率、および各ジョブステップ毎のシステム資源使用量の確率分布に基づいて、まずジョブクラスが決まり、次にそのジョブステップ列が決まり、この構成の各ジョブステップに対して各々処理装置使用時間（APU および CPU 使用時間）、主記憶使用量、ファイルアクセス 1 回当たりのチャネルサービス時間、およびファイルアクセス頻度と



(ii) サービスジョブステップおよび CP-RUN ジョブステップのタスクの動き

図 3.2 AP・CP タスクの処理過程



(ii) APRUNジョブステップのタスクの動き

図 3.2 AP・CPタスクの処理過程 (つづき)

表 3.1 ジョブステップのシステム資源使用量期待値

項目	AP使用時間(秒)	CP使用時間(秒)	主記憶使用量(KW)	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)	CPサービス時間:チャネルサービス時間	APサービス時間:チャネルサービス時間
ジョブステップ名											
サービス・ジョブステップ											
CP FORTRAN	0.	3.1	90.0*	108.69	—	—	—	28.5	—	0.95:1	—
AP FORTRAN	0.	10.5	116.8	115.02	—	—	—	91.3	—	3.04:1	—
LIED	0.	1.8	47.0*	538.59	—	—	—	3.3	—	0.11:1	—
LIBE	0.	0.3	32.0*	146.45	—	—	—	2.0	—	0.07:1	—
UTILITY	0.	1.2	31.4	828.98	—	—	—	1.4	—	0.05:1	—
SH-RUN	0.	17.0	66.0	—	4.01	—	—	249.4	—	8.31:1	—
DT-RUN	0.	25.4	65.2	—	3.13	44.87	—	20.8	—	0.69:1	—
CM-RUN	0.	331.2	81.6	—	0.65	—	—	1538.5	—	51.28:1	—
AM-RUN	204.7	3.5	99.5	—	0.42	—	1.123	15.1	875.3	40.37:1	2340.39:1
AL-RUN	724.3	12.5	137.7	—	0.18	—	0.594	28.6	1654.9	94.38:1	5461.17:1

(注) *印の値は固定値

が決定される。なお、これらの値が運用上の制限値を越えている場合には、その制限値に置き換えられる。

このジョブ属性の決定過程と、昭和53年度のアカウ
ント情報統計解析結果に基づいたジョブ集団を規定する各
種パラメータの推定値については付録Bに詳述している。
本実験ではこれらの推定値に基づいてジョブの属性を決
定している。ここでは各種ジョブステップを付録Bに従
って多数発生した場合、それらが要求するシステム資源
使用量の期待値のみを表3.1に示すにとどめる。同表か
ら、航技研における各ジョブクラス毎の平均的システム
資源要求量がわかる。

また、本実験で使用するジョブ集団では、先の実験³⁾
に使用したものに比べて、SHクラスの発生比率が7%
程大きくなっている点を注意しておく。

3.2.3 シミュレーション時間

計算機システムの各種統計量が安定するまでシミュレ
ーションを続行する必要があるが、今迄の経験から判断
して、シミュレーションは実時間にして20時間で打ち切
った。

3.2.4 運用系

我々は、先に報告したFACOM 230-75 APUシステム
に対する運用系の検討結果をもとにして、以下の様な運
用系1を設定した。

主記憶優先権 $SJ = AM_R > SH_R > DT_R > CM_R$
 $> AL_R$

実行優先権 $SJ = AL_R > SH_R = DT_R = AM_R$
 $> CM_R$

多重度 $SH = DT = CM = AL = 1, AM = 2$

SJ : サービスジョブステップ

SH_R : SHクラスのRUNジョブス
テップ

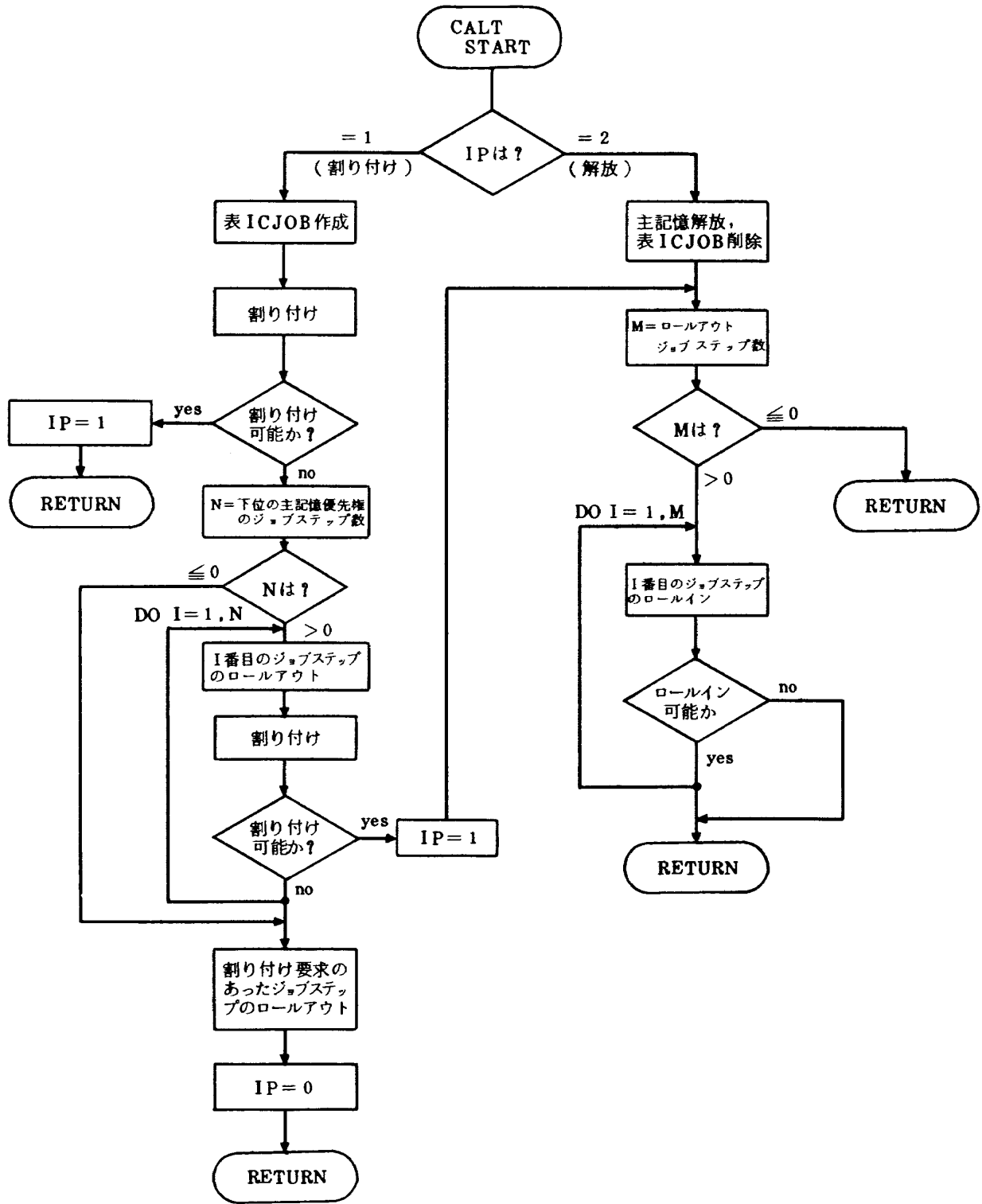
運用系の設定に際して、APUの稼働率は高めたいが、
APRUNジョブステップの主記憶優先権を高くしすぎると
CPUの稼働率をかなり低下させるという前回の実験
結果が強く反映されている。このことを、ALクラスの
主記憶優先権は低くするけれども実行優先権は最高にす
る事によって表現している。しかし、我々がこれから調
べようとしている諸方策によって主記憶利用率の向上が
十分に得られるならば、この運用系はALクラスを最優
先することになり、SHクラスのジョブのスループット
をかなり低下させる可能性がある。即ち、主記憶ネック
の状態が緩和すればこの運用系の見直しも当然必要とな
るが、これについては4章で考察する。

3.3 実験項目

本実験の目的は、ロールイン・ロールアウト方式の改
良効果、各種アロケータの効率比較、リロケーション、

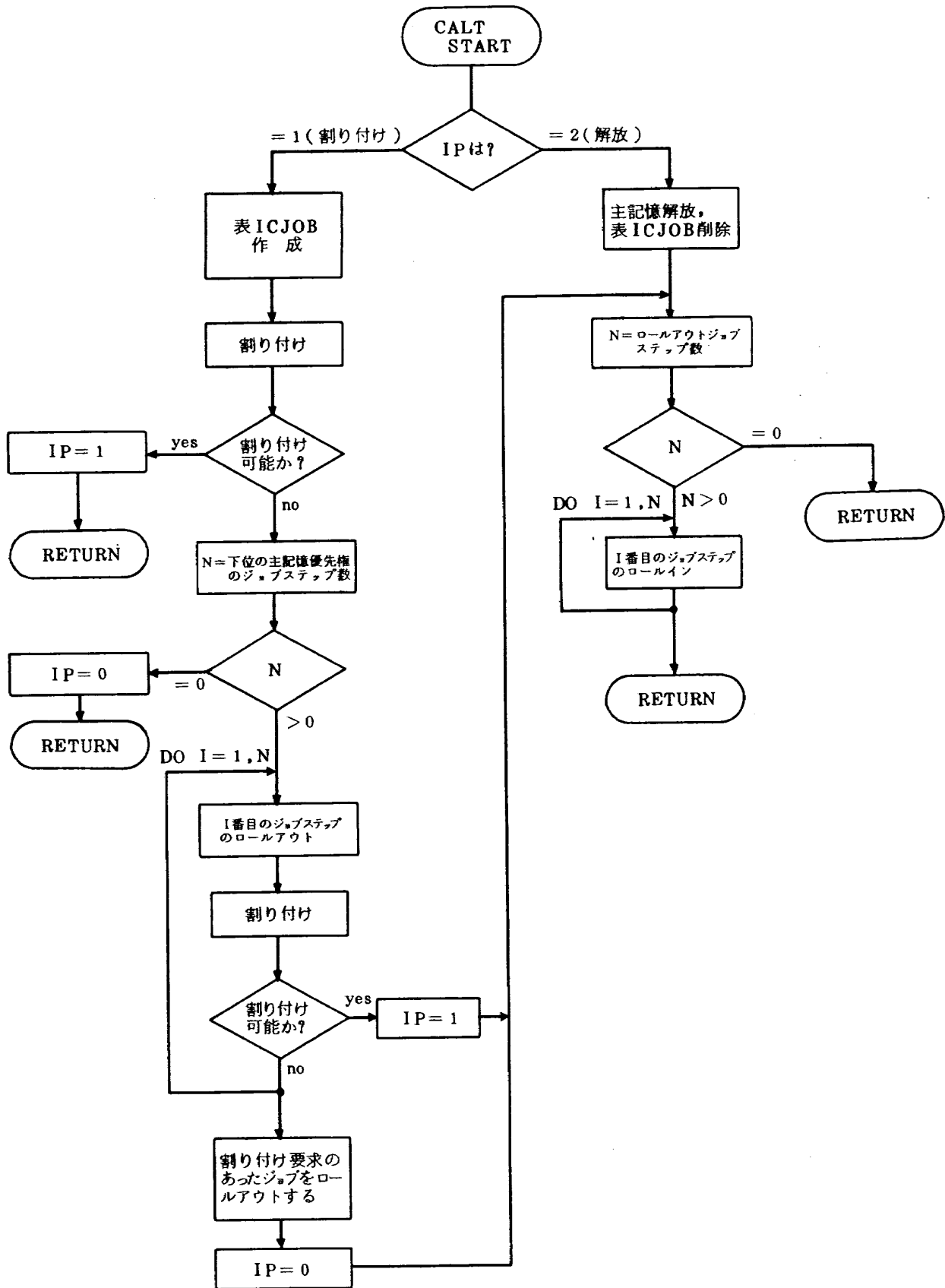
表3.2 実験内容

使用機能 実験番号	主記憶管理 ルーチン	ロールイン・ ロールアウト 方式	アロケータ	リロケータ	主記憶量 (KW)	256KW境界 の有無
0	0	旧	0	—	768	有
1	1	新	0	—	768	有
2	1	新	1	—	768	有
3	1	新	1	—	768	無
4	1	新	1	—	1024	有
5	1	新	3	—	768	無
6	1	新	3	—	1024	有
7	1	新	2	—	768	有
8	1	新	2	—	768	無
9	1	新	2	—	1024	有
10	2	新	2	1	768	有
11	2	新	2	1	768	無
12	2	新	2	1	1024	有
13	3	新	2	1	768	有
14	3	新	2	1	768	無
15	3	新	2	1	1024	無



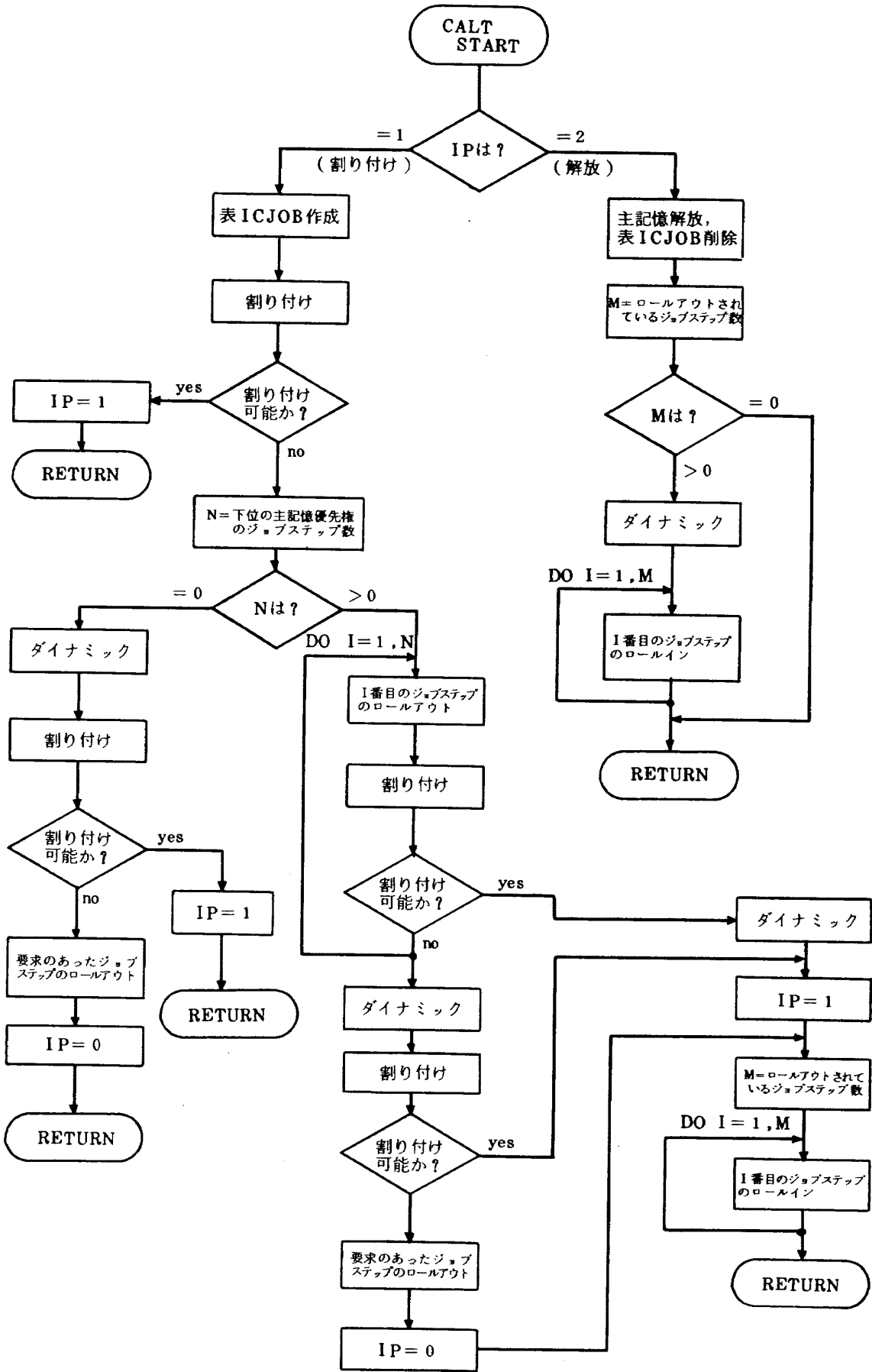
(1) ルーチン 0

図 3.3 主記憶管理ルーチン



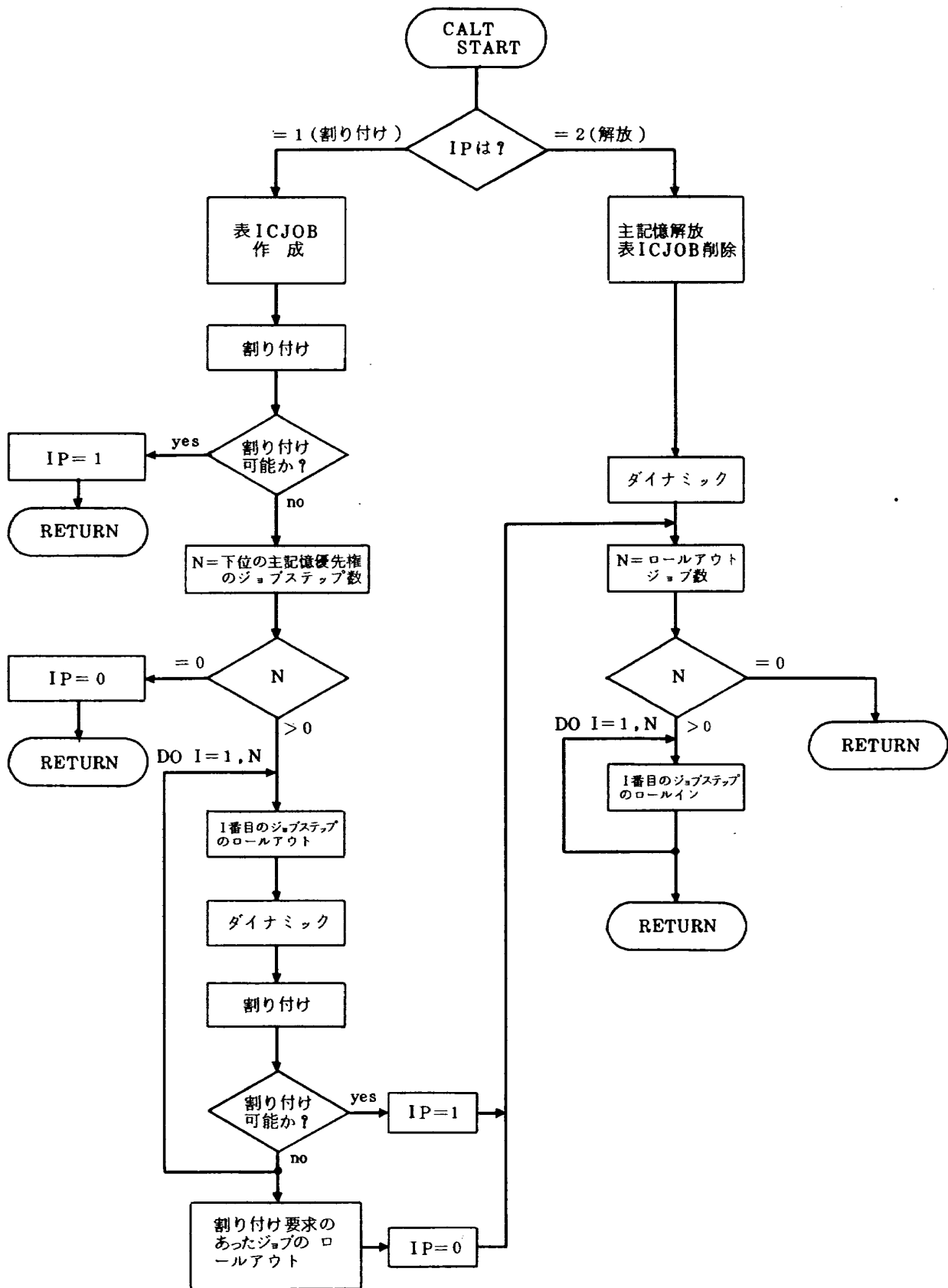
(2) ルーチン 1

図 3.3 主記憶管理ルーチン (つづき)



(3) ルーチン2

図 3.3 主記憶管理ルーチン (つづき)



(4) ルーチン 3

図 3.3 主記憶管理ルーチン (つづき)

表 3. 3 シ ミ ュ レ ー シ ョ ン

項目		実験番号	0	1	2	3	4	5	6
20時間の平均AP稼働率 (%)			79.48	79.89	88.80	90.27	89.86	85.07	89.37
平均CP稼働率 (%)			84.09	87.78	92.84	91.65	94.62	91.13	95.15
平均主記憶利用率 (%)			61.57	62.52	72.74	72.11	56.67	69.68	57.70
APジョブステップの主記憶多重度	平均値		1.34	1.41	1.79	1.89	2.14	1.65	2.05
	標準偏差		0.76	0.82	0.82	0.85	0.92	0.84	0.94
CPジョブステップの主記憶多重度	平均値		3.24	3.24	3.30	3.21	3.52	3.29	3.62
	標準偏差		1.19	1.07	0.84	0.88	0.95	0.87	0.97
APジョブステップのロールアウト件数	平均値		0.84	0.81	0.43	0.50	0.003	0.68	0.004
	標準偏差		0.36	0.40	0.50	0.50	0.05	0.47	0.06
CPジョブステップのロールアウト件数	平均値		0.25	0.19	0.16	0.06	0.01	0.05	0.01
	標準偏差		0.59	0.44	0.38	0.25	0.09	0.23	0.08
ジョブステップの総ロールアウト件数			576	509	509	497	14	568	30
空領域の平均サイズ (KW/個)			47.07	47.40	38.92	45.64	71.36	42.54	63.35
空領域の平均個数 (個)			4.91	4.92	4.44	3.95	5.58	4.41	6.05
リロケーション回数 (回)			—	—	—	—	—	—	—
リロケーションによる移動領域(MW)			—	—	—	—	—	—	—
AP寄与率	AM		70.6	67.7	57.7	54.7	44.6	62.6	51.7
	AL		8.9	12.2	31.1	35.6	45.3	22.5	37.7
	合計		79.5	79.9	88.8	90.3	89.9	85.1	89.4
CP寄与率	SH		28.7	30.3	24.7	22.4	26.2	24.2	28.3
	DT		3.7	3.4	3.4	3.4	4.8	3.9	3.1
	CM		45.4	45.9	57.2	59.3	55.9	54.0	57.2
	AM		5.9	7.9	6.4	5.5	5.1	7.1	5.0
	AL		0.4	0.3	1.1	1.2	2.6	1.9	1.6
	合計		84.1	87.8	92.8	91.7	94.6	91.1	95.2
平均ロールアウト時間 (秒/件)	SH		0.439	0.154	0.021	0.239	0.004	0.027	0.001
	DT		41.310	14.150	3.449	3.660	0.164	3.716	0.491
	CM		102.200	96.460	13.360	28.550	0.195	21.390	0.377
	AM		0.551	6.564	39.740	0.864	0.	3.069	0.971
	AL		6128.000	5225.000	889.500	1052.000	15.190	1974.000	9.823
平均ロールアウト回数 (回/件)	SH		0.006	0.007	0.006	0.011	0.001	0.008	0.001
	DT		1.000	0.532	0.539	0.188	0.040	0.354	0.123
	CM		1.524	1.628	0.841	0.641	0.034	0.823	0.084
	AM		0.018	0.032	0.040	0.005	0.	0.015	0.005
	AL		27.000	23.000	9.853	11.060	0.100	16.790	0.278
有効主記憶時間比 (上段:平均値 下段:標準偏差)	SH		0.64	0.63	0.60	0.61	0.57	0.60	0.56
			0.23	0.22	0.20	0.21	0.20	0.20	0.19
	DT		0.43	0.44	0.41	0.39	0.39	0.41	0.35
			0.16	0.16	0.14	0.12	0.13	0.13	0.09
	CM		0.70	0.67	0.71	0.73	0.68	0.70	0.67
			0.16	0.15	0.15	0.15	0.15	0.16	0.15
	AM		0.57	0.58	0.50	0.47	0.47	0.53	0.50
			0.15	0.16	0.17	0.19	0.21	0.16	0.20
	AL		0.83	0.84	0.80	0.84	0.83	0.85	0.83
			0.16	0.12	0.14	0.12	0.09	0.09	0.13
経過時間指数 (上段:平均値 下段:標準偏差)	SH		2.77	2.79	2.73	2.79	2.92	2.84	2.89
			1.90	1.72	1.65	2.03	1.98	1.87	1.81
	DT		3.04	2.95	3.16	2.96	3.04	3.00	3.18
			1.71	1.49	1.21	0.71	0.69	0.99	0.64
	CM		2.17	2.06	1.63	1.64	1.72	1.74	1.69
			1.83	1.03	0.55	0.67	0.67	0.65	0.46
	AM		2.19	2.17	3.15	3.05	3.37	2.46	3.12
			1.61	1.17	6.86	2.38	2.52	1.65	2.70
	AL		9.45	8.77	4.15	3.78	1.30	5.34	1.33
			16.84	11.19	8.63	5.99	0.25	6.58	0.39
<参考>立ち上り除外18時間の平均AP稼働率 (%)			78.87	82.23	90.53	94.12	92.18	87.69	91.64
平均CP稼働率 (%)			84.63	87.44	93.34	91.68	94.70	91.18	95.29
平均主記憶利用率 (%)			61.95	63.25	74.11	73.51	57.74	70.86	58.89

実験結果

7	8	9	10	11	12	13	14	15	運用系の 見直し実験
86.62	89.17	93.98	88.43	93.18	92.21	91.68	93.21	94.52	91.81
89.39	86.41	94.76	90.46	89.42	95.17	92.86	94.01	94.34	94.85
69.42	69.74	57.97	76.64	77.69	57.28	78.92	79.22	61.68	76.70
1.68	1.87	2.43	2.06	2.28	2.35	2.22	2.34	2.52	1.57
0.84	0.83	0.82	0.90	0.82	0.89	0.86	0.84	0.79	0.59
3.18	3.04	3.24	3.36	3.14	3.35	3.20	3.15	3.20	3.86
0.92	1.16	0.84	0.94	0.86	0.90	0.90	0.81	0.80	0.96
0.72	0.48	0.02	0.22	0.24	0.001	0.26	0.22	0.	0.
0.46	0.50	0.12	0.42	0.43	0.03	0.44	0.41	0.	0.
0.12	0.37	0.002	0.01	0.01	0.001	0.01	0.003	0.	0.24
0.44	0.83	0.05	0.14	0.09	0.04	0.06	0.05	0.	0.53
711	1106	32	1418	1443	36	720	627	0.	490
46.51	48.33	70.92	74.95	78.67	82.43	99.68	117.00	313.90	96.67
4.10	3.84	5.53	2.33	2.06	4.88	1.28	1.00	1.00	1.51
—	—	—	2088	2448	29	4414	4067	4295	4982
—	—	—	126.9	122.7	6.8	159.7	135.9	144.8	181.7
67.0	54.8	36.1	44.5	41.4	35.9	37.9	35.8	26.8	37.0
19.6	34.3	57.9	43.9	51.8	56.3	53.8	57.4	67.6	54.8
86.6	89.2	94.0	88.4	93.2	92.2	91.7	93.2	94.5	91.8
25.0	22.2	27.5	25.5	27.1	25.2	22.4	23.4	23.1	32.9
3.5	4.7	5.0	4.1	4.1	4.3	3.9	4.2	4.2	4.1
51.4	50.1	56.8	48.4	50.3	59.0	56.7	59.4	61.5	49.8
8.8	8.1	3.7	10.7	6.1	4.3	8.4	4.9	2.6	5.5
0.7	1.3	1.8	1.8	1.8	2.4	1.5	2.1	2.9	2.6
89.4	86.4	94.8	90.5	89.4	95.2	92.9	94.0	943	94.9
1.417	9.224	0.007	0.084	0.038	0.005	0.028	0.	0.	11.880
23.390	52.650	0.266	2.098	0.900	0.226	0.835	0.064	0.	19.170
21.080	67.940	0.646	4.949	1.999	0.283	1.871	1.292	0.	5.380
11.240	24.250	0.116	0.	0.	0.	0.025	0.	0.	0.
2113.000	847.000	21.380	329.300	318.500	0.822	423.500	280.000	0.	0.
0.024	0.098	0.001	0.040	0.020	0.001	0.002	0.	0.	0.270
0.581	2.126	0.056	0.691	0.374	0.083	0.023	0.011	0.	1.147
0.754	2.491	0.075	1.317	0.709	0.087	0.183	0.076	0.	0.274
0.062	0.126	0.021	0.	0.	0.	0.006	0.	0.	0.
21.320	12.930	0.250	23.540	23.710	0.263	15.480	11.000	0.	0.
0.63	0.61	0.57	0.62	0.64	0.58	0.59	0.60	0.60	0.55
0.21	0.21	0.19	0.22	0.22	0.19	0.21	0.20	0.20	0.18
0.43	0.43	0.41	0.43	0.44	0.37	0.40	0.38	0.38	0.37
0.15	0.15	0.14	0.18	0.18	0.11	0.14	0.09	0.12	0.14
0.70	0.69	0.69	0.70	0.70	0.70	0.70	0.73	0.74	0.62
0.16	0.15	0.17	0.16	0.16	0.15	0.16	0.15	0.13	0.16
0.53	0.49	0.42	0.43	0.40	0.40	0.41	0.38	0.35	0.54
0.16	0.17	0.21	0.20	0.19	0.22	0.20	0.20	0.22	0.13
0.83	0.80	0.82	0.81	0.82	0.81	0.82	0.84	0.84	0.62
0.14	0.17	0.13	0.11	0.12	0.12	0.13	0.12	0.08	0.12
2.76	3.59	2.84	2.89	2.82	2.88	2.88	2.83	2.68	3.53
2.29	12.09	1.82	1.92	2.20	0.19	1.82	1.78	1.28	4.64
2.94	3.86	3.10	2.98	2.97	3.13	3.15	3.27	3.04	3.36
1.52	6.43	1.31	1.33	1.29	0.73	1.06	1.32	0.61	1.18
1.76	2.09	1.70	1.73	1.67	1.66	1.71	1.57	1.54	1.88
0.67	1.89	0.64	1.15	0.56	0.80	0.79	0.48	0.46	0.62
2.41	3.07	4.14	3.43	3.77	4.54	3.93	4.21	5.31	2.13
1.51	3.14	3.92	2.26	2.82	4.35	3.44	3.27	4.20	0.64
4.40	4.00	1.38	1.79	1.86	1.35	2.05	1.73	1.25	1.75
5.12	6.27	0.47	0.98	1.00	0.40	2.10	1.17	0.19	0.39
85.34	91.56	96.76	91.50	96.82	95.00	93.90	95.60	97.36	92.13
89.72	88.14	94.85	90.15	89.00	95.32	94.02	94.21	94.38	95.40
69.23	70.69	59.18	78.08	79.23	58.46	80.82	80.21	63.31	77.50

および主記憶増設の有効性評価など、全て主記憶管理ルーチンにかかわっている。したがって、図 3.1 におけるこれ以外のルーチンは全て既設のものを使っている。現有の主記憶管理ルーチン 0 および、新たに用意した主記憶管理ルーチン 1～3 のフローチャートを図 3.3 に示す。これらを用いて、表 3.2 に示す様な一連の主記憶管理の諸方策に関する実験 0～15 を計画した。同表において、旧ロールイン・ロールアウト方式は図 2.2 の(i), (iii) を指し、新ロールイン・ロールアウト方式は同図の(ii), (iv) を指す。また、アロケータ 0～3 はそれぞれ図 2.3 の(i)～(iv) を指す。更に、リロケータは 2.6 節の処理方式のものを指す。

(i) 実験 0, 1

この 2 つの実験結果から、ロールイン・ロールアウト処理方式の改良効果を調べる。

(ii) 実験 1, 2, 5, 7

この 4 つの実験結果から、どの主記憶割り付け方式が主記憶利用率の向上のために最も適しているかを判定する。

(iii) 実験 4, 6, 9, 12

この 4 つの実験結果から、256KW の主記憶増設が主記憶ネックの状態をどの程度まで緩和し、その結果として処理装置の稼働率がどの程度向上しうるかを評価する。

(iv) 実験 10, 13

この 2 つの実験結果から、リロケータ 1 の有効性およびリロケーションを行うタイミング、ならびに、2.6.3 節で設計したリロケータ 2 を検討する必要性を調べる。

(v) 実験 3, 8, 11, 14

この 4 つの実験結果から、256KW 境界を有する主記憶割り付け領域の存在が、主記憶利用率に及ぼしている悪影響の程度を探る。

(vi) 実験 15

この実験は本研究において考えた有効機能の全てを兼備した場合の、計算機システム稼働状況を示すためのものである。

3.4 実験結果

以上の準備のもとで、20時間のシミュレーション実験 0～15 を行った。その結果は表 3.3 に示す通りである。同表の列の各項目は、JPS の出力ルーチンが一定時間間隔毎に収集する計算機システム利用情報記録項目の一部であり、同表の数値は各々20時間にわたる収集データの統計量を表わしている。以下に表 3.3 を読む上で注意すべき事項を列挙しておく。

(1) 表 3.3 における AP, CP 稼働率および主記憶利用率の立ち上り除外は、シミュレーションの立ち上り時

間を 2 時間として、この時間帯に収集されたデータを除外した場合の統計量を、20時間の統計量と比較するために載せた。

(2) JPS では、OS の走行時間としてジョブ開設・閉設処理時間、ジョブステップの開設・閉設処理時間、ロールイン・ロールアウトに要する時間、主記憶の割り付け・解放に要する時間、および入出力処理のために要する時間を考慮している。これら OS 処理には全て CPU が使用されるが、その統計は CPU モニタオーバーヘッド時間として記録される。しかし、経済的見地から、これら OS 走行事象をシミュレーション時間の更新事象から省き、実験最後に総 OS 走行時間を算出している。

(3) (2) で述べた理由により、シミュレーション実験における APU 時間も CPU 時間も 100% ユーザジョブステップの実行のために使用できるから、{100% - 稼働率(%)} は APU, CPU がアイドルとなる平均比率を表わす。ただし、20時間シミュレーション実験で必要とした OS の総走行時間を T_{os} とすれば、CPU が実際に必要とする処理時間は $(20 + T_{os})$ 時間である。本実験では、3.2.1 節で述べたように実験環境を設定しているから、処理待ちジョブは常に存在している。したがって、このアイドル比率はできるだけ小さくすることが望まれる。

(4) 主記憶利用率は(総主記憶量 - モニタ使用量)に対する使用量の比率で表わしているから、これだけでは主記憶が競合しているのか余裕があるのかが判定できない。しかし、上記(3)と同様の理由により、本実験ではかなりの競合状態にあると考えてよい。

(5) AP (または CP) クラスの待ち行列が十分に長く、かつ主記憶競合がなければ、AP (または CP) 多重度は最大 3 (または 6) 多重から最低 0 (または 3) 多重まで可変ではあるが、APU, CPU 共に 3 多重である比率が非常に大きい。しかし、主記憶に対する競合のためにロールアウトされるものがあると、主記憶上にあるジョブの多重度は低下する。その平均値は AP, CP ジョブステップのロールアウト件数の平均値によって示されるが、この値が大きい程主記憶競合の度合いが強いことを意味する。

(6) 空領域の平均サイズとその個数は、主記憶上の断片化の平均的状态を示している。空領域個数は 1 に近いことが望ましい。

(7) 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によって移動された領域の総量を表わす。

(8) 平均ロールアウト回数および平均ロールアウト時間は、値が小さい程主記憶競合の度合いが弱いことを示す。

(9) 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稼働率および平均CP稼働率の内訳を示している。これにより各実験において各ジョブクラスに分配された処理装置資源の割り合いを知ることができる。

(10) 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り付けられていた総時間に対して、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。この比率が1に近い程主記憶を有効に使用していることを意味するが、それは主記憶優先権が低く、かつ実行優先権が高いジョブステップ程実現されやすい。

(11) 経過時間指数は、各ジョブクラス毎のジョブがシステム資源を使用した総時間に対して、そのジョブの開設から閉設までにかかった時間の比率を表わす。この値が1に近い程、APU, CPU, または主記憶に対する待ち合せ時間が短いことを示している。

以上のデータから、各実験毎の平均的ジョブ処理能力を十分に比較評価することができる。さて、各実験におけるジョブストリームはジョブ発生件数としては十分であるが、時間的推移の中でシステム資源に対する競合の程度 (以下、負荷と言う) は非常に軽微な状態から非常に過大な状態まで発生している。各実験のジョブ処理能

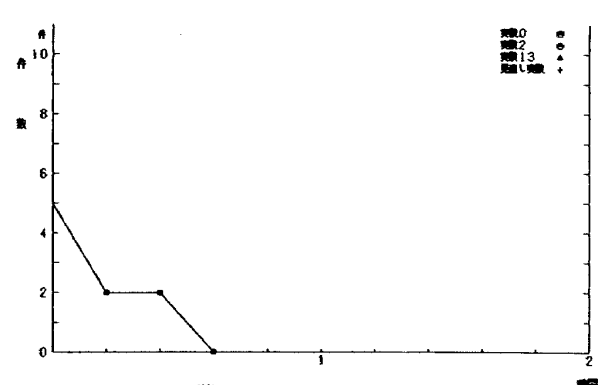
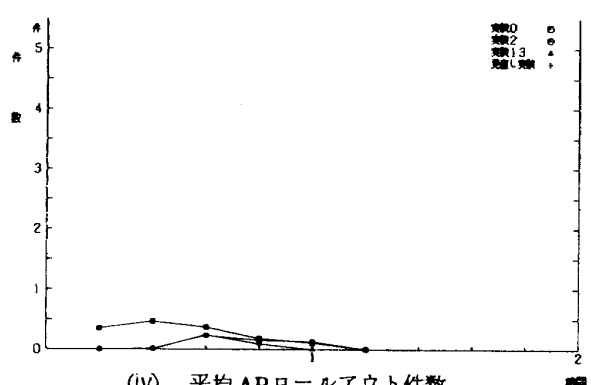
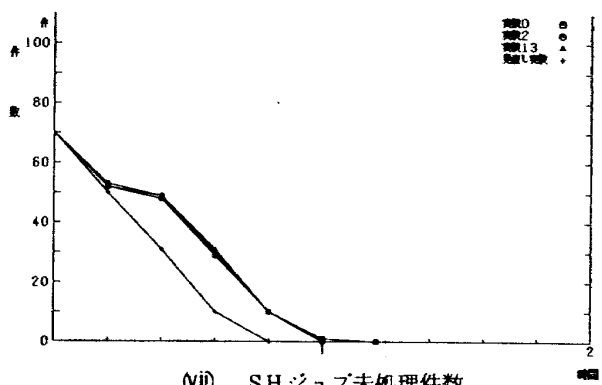
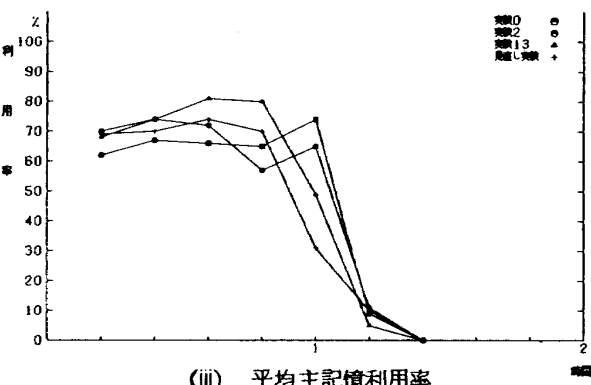
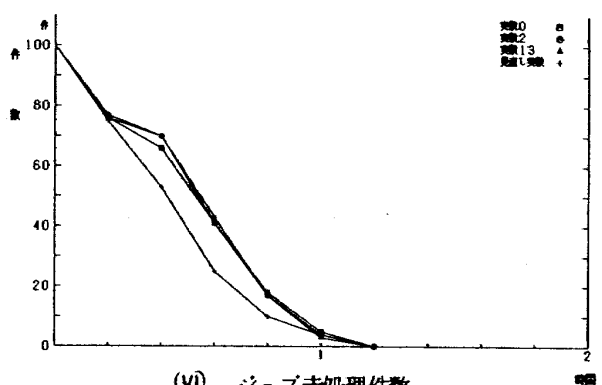
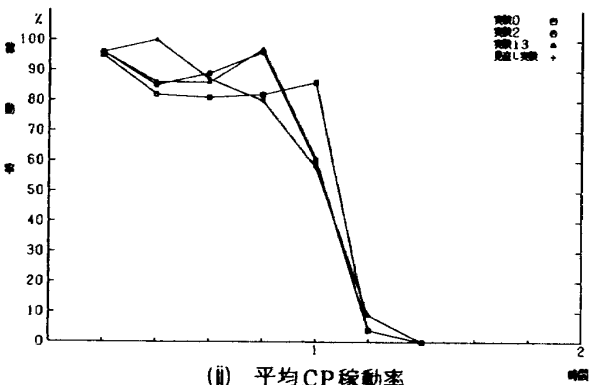
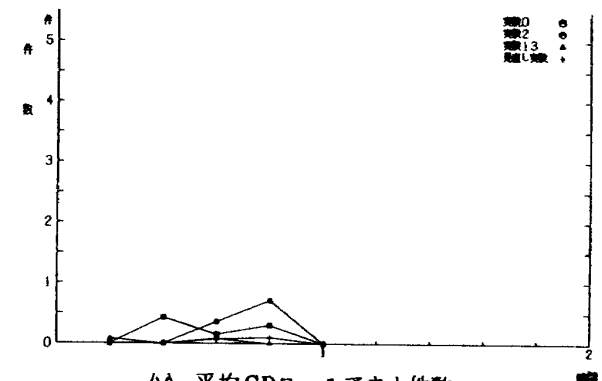
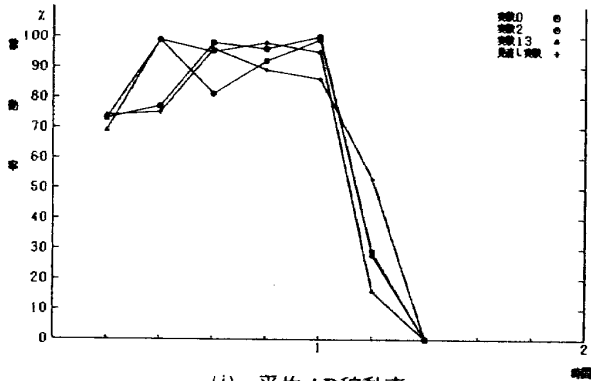
力の差は、軽微な負荷に対してはほとんど現われないが、過大な負荷に対しては顕著に現われる。そこで、平均的データよりもっと処理能力の差異を明示しうる実験を20時間シミュレーションを行った実験項目の中の代表的なもの4つにより行った。まず、付録Bで詳述したジョブ集団からランダムに10回、適当な長さのジョブストリームを取り出した。それらジョブを全てジョブクラス毎の待ち行列に登録してから、これら以外にはジョブを発生することなく、各実験項目毎のジョブ処理を開始し、その時間的過程を記録した (以後、ラッシュアワー実験と呼ぶ)。その結果、負荷が軽微と認められる事例が7件 (この代表例を図3.4に示す)、負荷が中程度と認められる事例が2件 (この代表例を図3.5に示す)、および負荷が過大であると認められる事例が1件 (これを図3.6に示す) の割合であった。図3.4~図3.6において、負荷の軽重の程度は(i)~(v)の平均的ロールアウト件数の推移図から推測される。また、(iii)の平均主記憶利用率推移図と(i)~(ii)の平均処理装置稼働率推移図の相関関係を調べると、主記憶の有効利用の効果が明らかとなる。特に注目されたい点は、(vi)~(ix)のジョブ未処理件数の推移図の内容 (その一部を表3.4, 図3.7に示す) である。実験0より実験2, 実験2より実験13の順に全ジョブを処理するのに要した時間はかなり短縮されているが、逆に

表3.4 ラッシュアワー実験における詳細データ

	多重度が5になった時刻 (秒)			ジョブ処理が全て終了した時刻 (秒)		
	負荷小	負荷中	負荷大	負荷小	負荷中	負荷大
実験0	3511	3250	3076	3820	5377	4701
2	3328 (-183)	3711 (+461)	3444 (+368)	3842 (+22)	4899 (-478)	4551 (-150)
13	3239 (-272)	3704 (+454)	3387 (+311)	3766 (-54)	4718 (-659)	4421 (-280)
運用系の見直し実験	2617 (-894)	2781 (-469)	2707 (-369)	4128 (+308)	5077 (-300)	4415 (-286)

()内は実験0の結果との差

注) この表の数値および図3.7のグラフは、多重度が実際に変化した時刻を示している。一方、図3.4~3.6は、時間帯毎の平均処理装置稼働率、平均ロールアウト件数およびその時点での未処理件数を示しているため、そのまま比較することはできない。例えば、平均AP稼働率は0.2時間毎のデータであるため、1.6時間目で5%を示し、1.8時間目で0.0%を示しているとすればAPUジョブの処理は1.4時間目まででは完了し、その後0.6分 (0.2時間×0.05=0.6分) 程で全て完了したと考えられる。



注) 平均AP・CP稼働率, および平均主記憶利用率は12分毎の平均値を示す。

図 3.4 ラッシュアワー実験結果 (負荷軽微な場合)

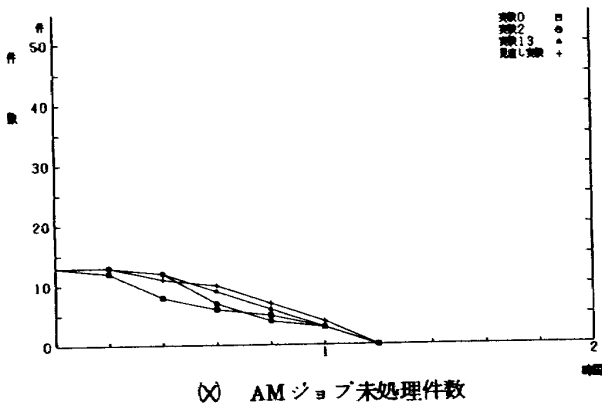
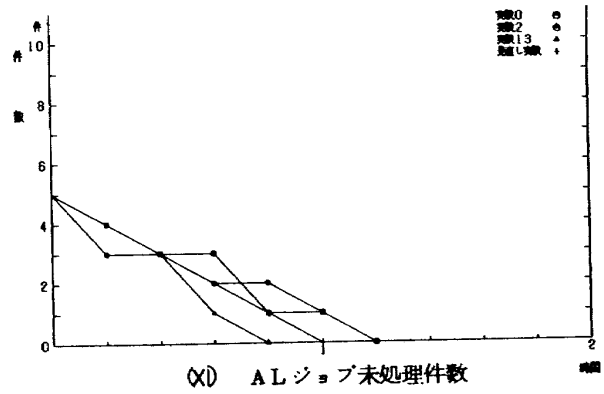
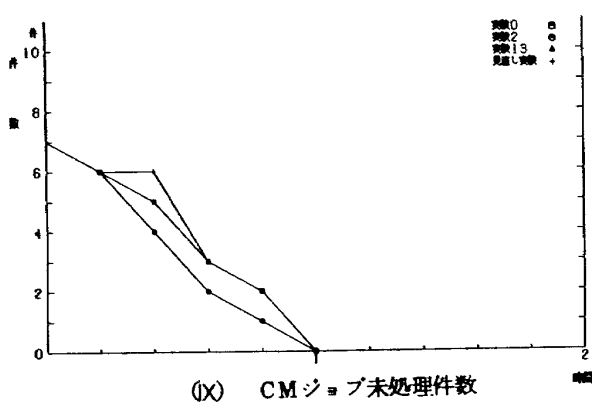
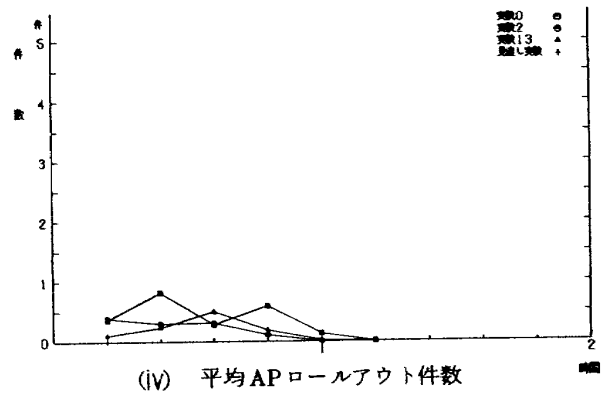
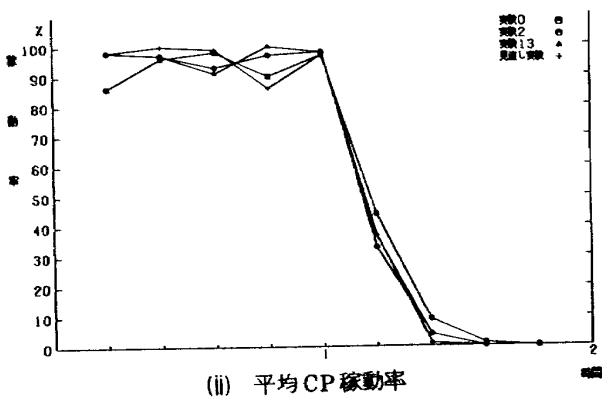
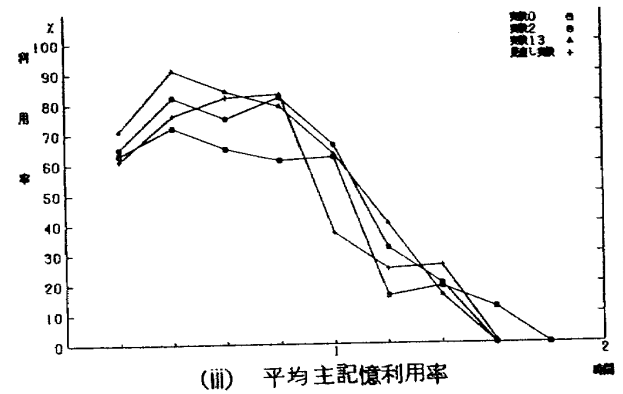
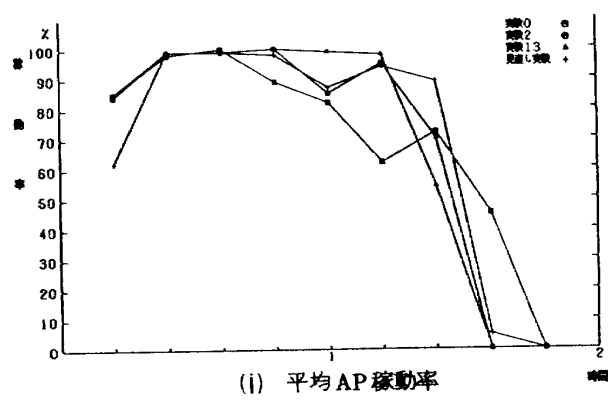


図 3.4 ラッシュアワー実験結果(負荷軽微な場合)つづき



注) 平均A・P・C稼働率, および平均主記憶利用率は12分毎の平均値を示す。

図 3.5 ラッシュアワー実験結果(中程度負荷の場合)

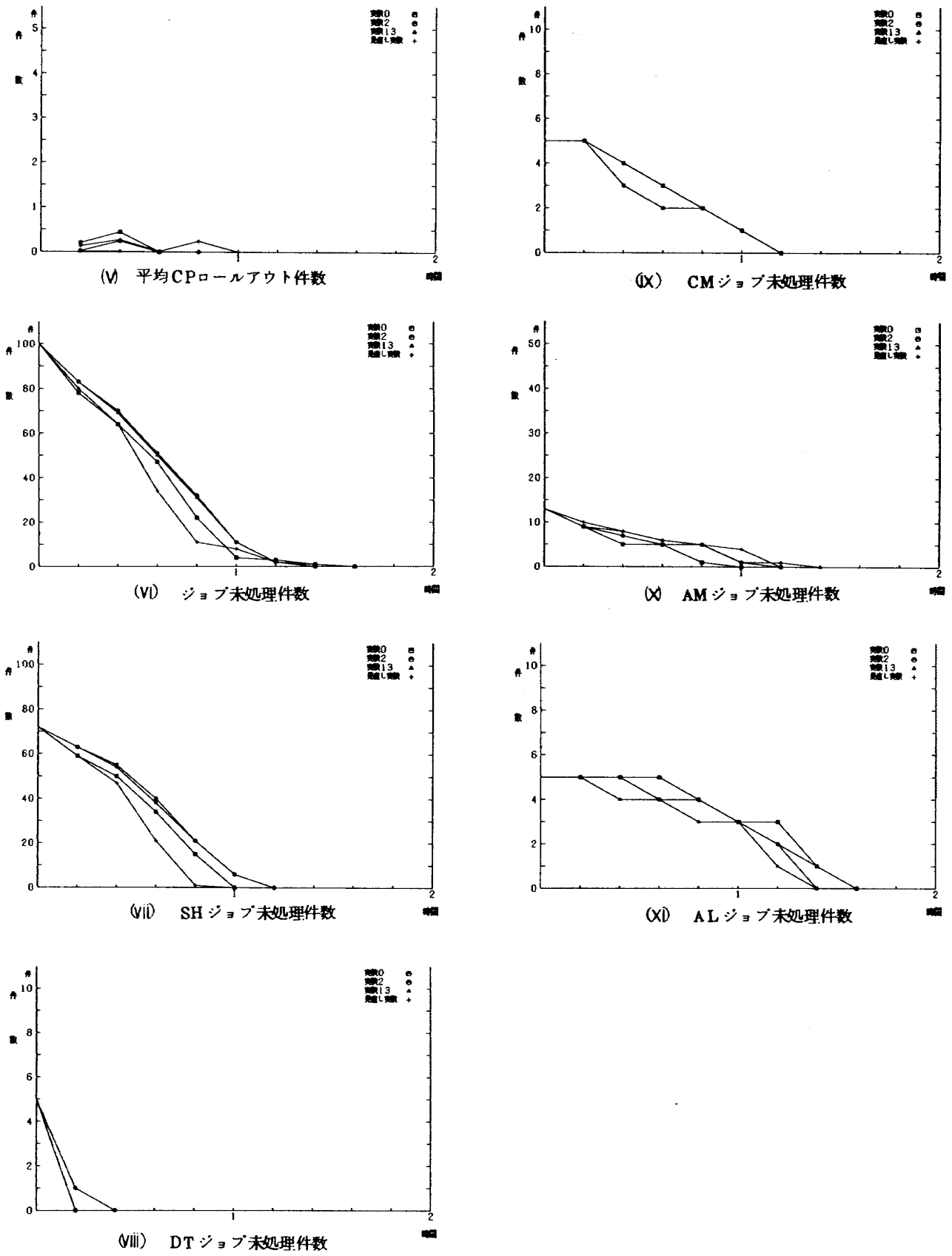
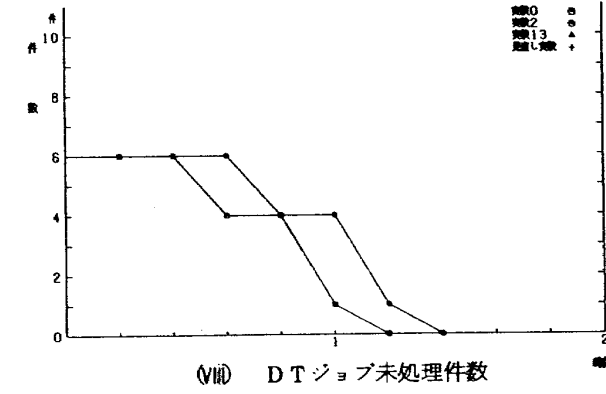
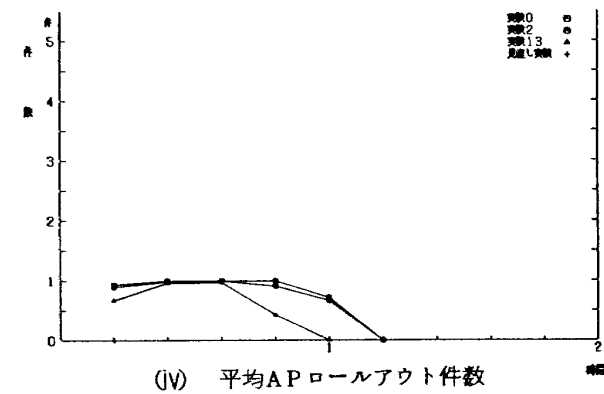
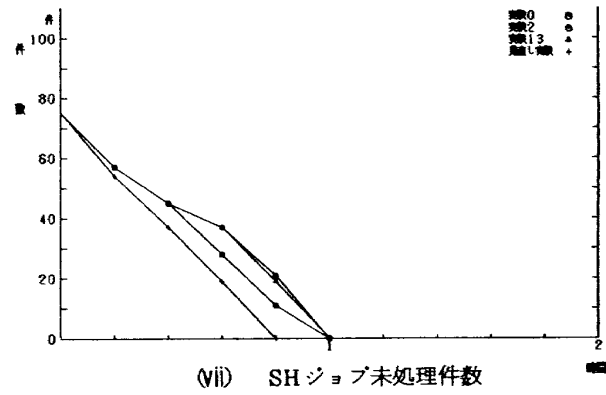
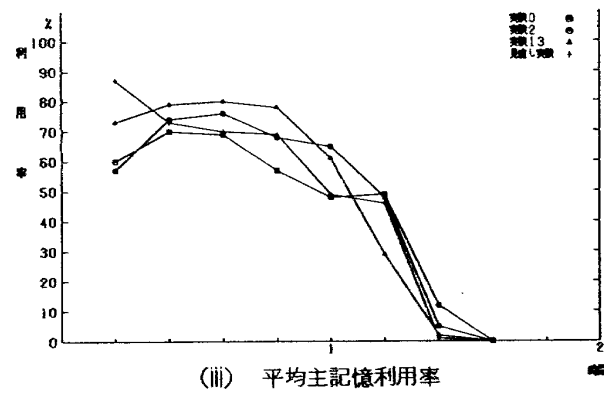
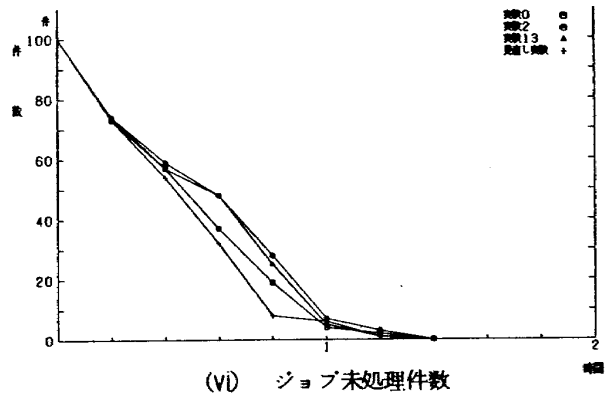
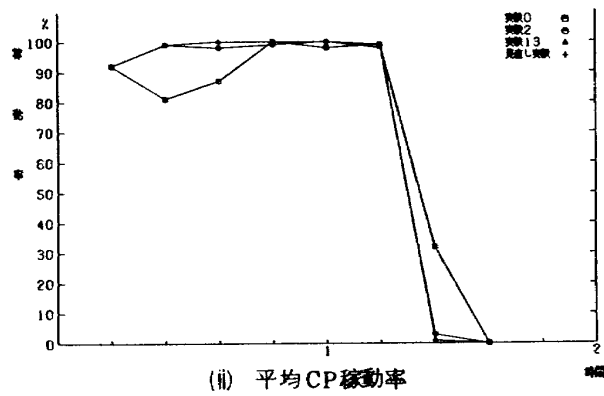
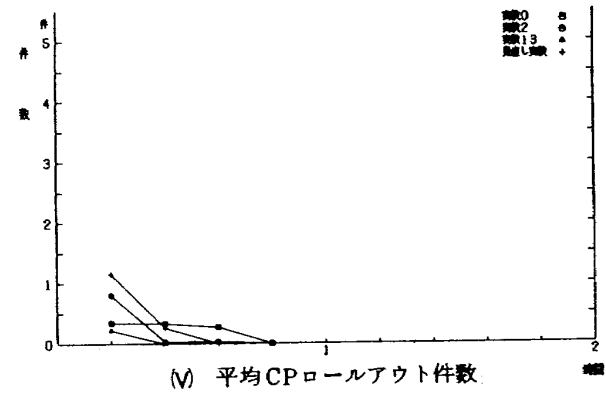
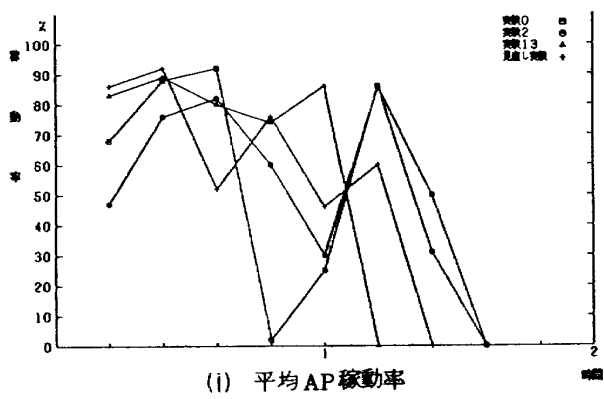


図 3.5 ラッシュアワー実験結果(中程度負荷の場合)つづき



注) 平均AP・CP稼働率, および平均主記憶利用率は12分毎の平均値を示す。

図 3.6 ラッシュアワー実験結果 (過大負荷の場合)

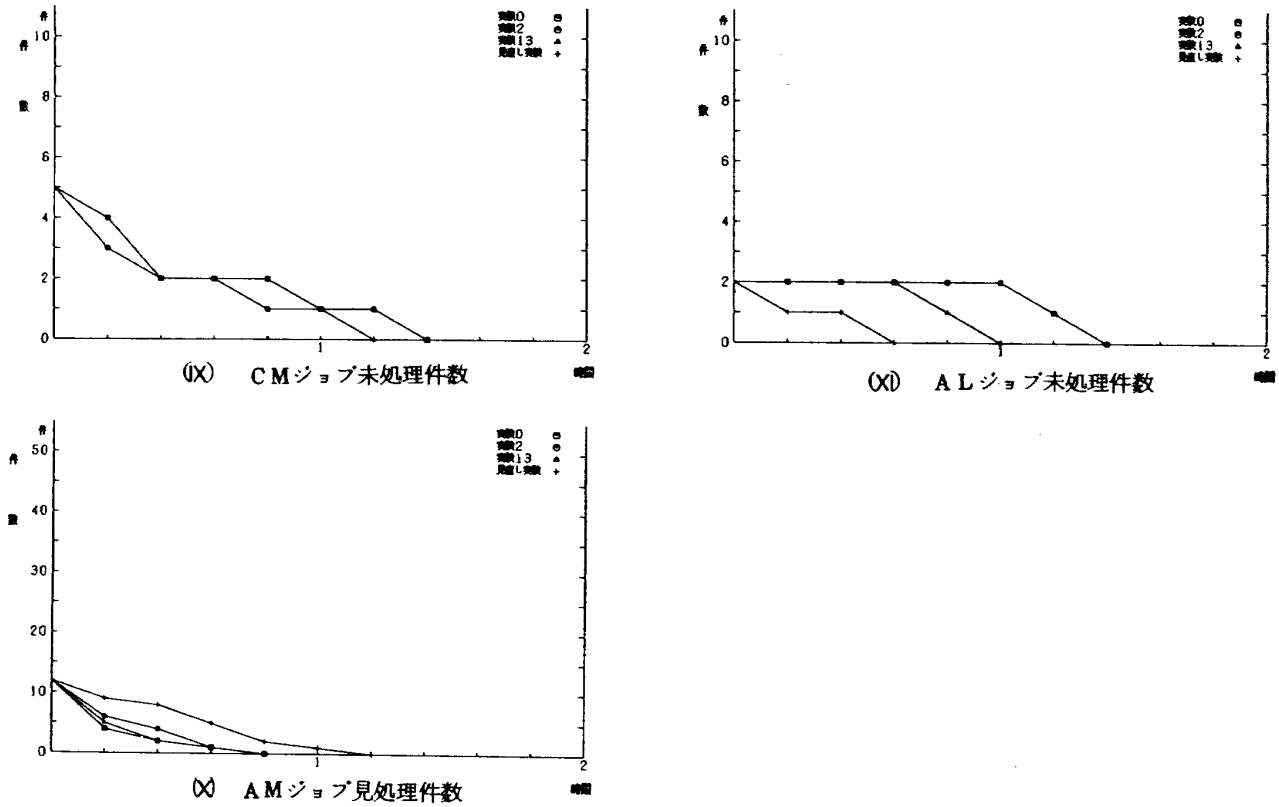


図 3.6 ラッシュアワー実験結果 (過大負荷の場合) つづき

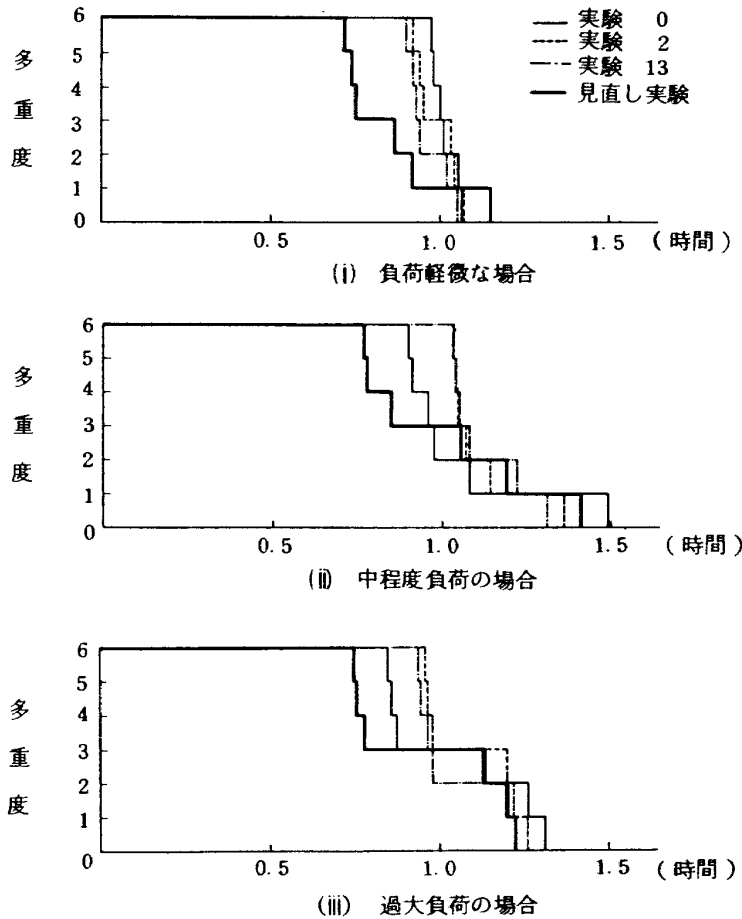


図 3.7 ラッシュアワー実験における多重度の推移

多重度が5になった時刻(即ち一つのジョブクラスのジョブ処理が全て完了した時刻)は増大している。これは主記憶の有効利用と共にA Pクラスのジョブ処理が優先されるような運用系を採用しているためである。

しかし、ジョブの混雑時にはその70%以上を占めるS Hクラスの処理を優先して、できるだけ早く混雑状態を緩和すべきことも重要である。そのためには運用系の見直しも必要となるが、それについては4.7節で論じる。図3.4~3.6と表3.3~3.4に載せた見直し実験についてはそこで詳しく述べる。

4. 考 察

現APUシステムの最大の問題は主記憶ネックの存在にあり、その解決は処理装置稼働率向上とジョブクラス間のバランスのとれたターン・アラウンド・タイムの保証に直結するものと予想される。本節では表3.3の実験結果に基づいて、各種の主記憶ネック解決手段による効果を検討する。システム資源有効利用の目安として、主記憶利用率の平均1%の向上は実装主記憶768KWの場合で平均約5.68KWの有効利用に、処理装置稼働率の平均1%の向上は20時間シミュレーションの場合で平均約12分のアイドル時間減少に相当する。しかし、シミュレーションの本質上、その結果の小さな数字の変動にはそれ程とらわれることなく、全体的傾向の把握に重点を置く。

4.1 ロールイン・ロールアウト処理方式の改善効果

新ロールイン・ロールアウト処理方式は、ロールアウト状態のジョブステップを可能なかぎりロールインする点に旧方式との違いがある。したがって、その改善効果は主記憶優先権の低いC MまたはA Lクラスのジョブの実行の機会を多少とも増すが、その効果は特に実行優先権の高いA LクラスのA P寄与率の向上となって表わることが期待される。

この効果を見るために実験0と1を行った。実験0は現有の主記憶管理方式の場合の実験であり、実験1はそこでロールイン・ロールアウト処理方式のみを改善した場合の実験であるが、確かにA LクラスのA P寄与率の向上が認められ、これはロールイン・ロールアウト処理方式の改善効果の裏付けとできる。ただし、ロールイン・ロールアウト処理方式の改善のみによって処理装置稼働率の大巾な向上を期待することはできない。

4.2 各種アロケータ効果の比較

各種割り付け方式の単体性能の比較はアロケータ0と1が中心である。アロケータ2は本来再配置機能と組合せて用いることを前提として考案したものであり、またアロケータ3はアロケータ1の細部検討用のものである。

アロケータ0~2の単体性能を比較するために、アロケータ以外は全く同一の条件の下で実験1,2,および7を行った。アロケータ1はアロケータ0が引き起す主記憶上の断片化をできるだけ克服することを目指している。実際、その目的は空領域の個数が実験1と2とでは平均0.5個程度減少していることから十分に達せられたといえる。これは主記憶利用率を10%も向上させる効果を生み、A Pジョブステップの平均ロールアウト件数は半減した。またそれによるA P稼働率の向上は10%近くにも達している。また、C P稼働率もわずかではあるが向上が認められる。このように、割り付け手順がアロケータ0よりも非常に簡素なことを考え合わせると、アロケータ1は十分に有用であると結論できる。

一方、実験7からも判るように、アロケータ2による断片化克服能力はアロケータ1に匹敵しているが、主記憶の利用率の向上は多少劣る。これはアロケータ2がC PとCNPモードによる割り付け領域の混在を許していないことから当然の結果といえる。主記憶使用量の大きいA Lクラスのジョブのロールアウト状態が実験2の場合よりかなり増大しているのはこのためである。したがってA P稼働率の向上はあまり望めない。しかし総じて言えば、アロケータ2はアロケータ0と比較して十分に有用であると判断できる。なお、アロケータ2をリロケータと組合せて使用した場合の効果は4.4節で述べる。

もし現計算機システムのデータ供給能力が主記憶空間のどこでも同一であり、かつ256KW境界という制限がなければ、アロケータ単体としては1または3を使用すべきである。このため、上記の想定のもとで実験3と5を行い、アロケータ1と3の間に割り付け効果の優劣が見出されるか否かを調べた。一般に、主記憶要求量が大きくばらつく場合、全ての領域を同一方向から割り付けていくと断片化が生じ易い。そこで要求量の小さい領域と大きい領域に分けて、それぞれを逆方向から割り付けし、できるだけ両者を混在しにくくした方が少くとも有利である。この理由からすれば、CNPモードによる割り付け領域はC Pモードによるものよりも平均的に小さい事が十分予想されるので、アロケータ3を用いるよりもアロケータ1を用いた方が好ましいことを示す実験結果が期待できる。実際、実験3と5の結果を比較すれば、上記の推論の正当性が裏付けされ、アロケータ1の優位性は明白に認められる。

以上のことから、単体としてはアロケータ1が最も有効であると断言できる。

4.3 主記憶装置の増設効果

主記憶ネックを解決する最も端的な方法は256KWの

主記憶の増設である。現 APU システムに対しては、これは単なる増設効果にとどまらず、更に主記憶空間上のデータ供給能力が一様になることから、アロケータ設計の際に要請された基本的拘束条件の一つが解除される。従って、主記憶が増設されればアロケータ 0 を考慮する必要性が全くなくなり、アロケータ 1~3 のみが実験の対象となる。

主記憶を 256KW 増設して、APU システムが持ちうる最大主記憶量 1 MW を装備したシステム構成を想定した実験は 4, 6, 9, および 12 である。これらの実験から総じて言えることは、主記憶の増設効果は絶大であって、どのアロケータを採用しようとも主記憶そのものの余裕から利用率の急激な低下を示しており、主記憶競合がほとんど無くなる。したがって、それぞれの実験における処理装置の稼働率はほぼそれぞれの限界を示している。これ以上の稼働率の向上は主記憶の増加だけでは計れるものではなく、運用系の見直しが必要となるが、これについては 4.7 節で論じる。

最後に、実験 4, 6, 9, 12 は主記憶を増設してもいかにその断片化が速やかに進行してしまうかを示している点に注目する必要がある。主記憶競合を完全に解決し、ロールアウトが全く行われなかった実験 15 については 4.6 節で述べる。

4.4 再配置機能の有効性評価

前節ではシステム資源を物理的に増大して得られる効果を調べたが、本節ではシステム資源の物理的増大に頼ることなく、再配置というソフトウェア手段によって最大どの程度まで主記憶の有効利用が計り得るかを調べてみる。なお、2.6.4 節で述べたように、この機能の有効性はアロケータ 2 に対応して設計したリロケータ 1 を用いて評価する。

4.2 節で見たように、アロケータ 2 の単体性能はアロケータ 1 のそれには及ばない。しかし、主記憶管理ルーチンに再配置機能を持たせた場合、アロケータ 2 は再配置処理を非常に簡単化する割り付け方法である。ただ処理が簡単になるだけでなく、以下に示すように、アロケータ 1 を組合せて得られる性能はアロケータ 1 の単体性能を遙かに凌駕するものと予測される。

まず、実験 10 の検討から始める。この実験は再配置を行うタイミングを、ジョブステップ開設処理時に主記憶割り付けができなかった場合と、ロールイン開始直前の時点とに限定して、リロケータ 1 の効果を調べるために行った。実験結果を見ると確かに、実験 7 に比べて空領域の個数は平均 2 個程度も減少しており、断片化克服の効果を良く示している。この効果は平均主記憶利用率の

大巾向上につながっており、特に空領域の平均サイズが 1.5 倍以上にも大きくなったことから、AL ジョブが主記憶上に割り付けられる機会がかなり増大している。その結果、AL クラスのジョブの AP 寄与率が大巾に増大し、経過時間指数も平均 2.5 倍以上も向上している。この様に、AP に関して大変良い結果を示す反面、CP 稼働率に最も影響度の高い CM ジョブのロールアウト時間は減少したものの、ロールアウト回数は倍増している。この現象は空領域の平均サイズが大きくなったことにより、AL ジョブは主記憶上にロードされる機会が一段と増したものの、まだ主記憶の有効利用が十分でなく、AL ジョブや CM ジョブのロールアウトされる機会がまだかなり残っているものと推測される。実験 10 を全体的に調べてみて、この主記憶管理の効果は、アロケータ 1 による実験 2 と同程度であり、決して満足できるものではない。

次に実験 13 の結果を調べる。この実験はリロケータ起動回数の増大は無視して、主記憶解放事象が発生する都度再配置を行って、可能な限り断片化を克服した場合の効果を見るために行った。表 3.3 から明らかなように、空領域の個数はかなり 1 に近くなっている。1 にならないのは 256 KW 境界のためである。この結果を見るとリロケータの効果は顕著であり、実験 10 に比べてロールアウト回数を半減させる結果をもたらし、AL ジョブの処理状況はさらに向上し、また処理装置の稼働率はほぼ限界近くまで向上している。限界に近いと推測する理由は、表 3.3 の主記憶利用率の伸びが処理装置の稼働率と比べて急速に鈍っており、80% 程度で飽和するものと予測できるからである。これは注目すべきことであり、実験 13 の主記憶管理のもとでは主記憶ネックの問題はほぼ解決されたと考えて良い。換言すると、これ以上 AL クラスのジョブのロールアウト回数を減少させるには主記憶の増設が必要であるが、そのようにしても本運用系のもとでは処理装置の稼働率をほとんど向上しえないであろう。したがって、主記憶の増設を行うよりは実験 13 における主記憶の余裕を SH ジョブの多重度を上げるために振り向け、CPU の処理能力を使い尽くす方が賢明である。このような運用系の見直しは 4.7 節で述べる。

なお、以上のように、本研究における目標はほぼ達したことが確認できたので、リロケータ 2 の実験は行わなかった。

最後に、再配置機能の使用に伴うオーバーヘッドの問題を検討する。もし、リロケーションの領域移動処理に要するオーバーヘッドを 1 MW 当り 1 秒だと仮定すれば、20 時間シミュレーション中に移動した主記憶総量の処理に

要するオーバーヘッドは、実験10の場合で約127秒、実験13の場合で約160秒となる。また、リロケーションの実験に当り全ての入出力処理の終了を待たなければならないと仮定する(付録A参照)。この場合、再配置ルーチン起動時点において出されている入出力命令の処理時間の最大値が問題となるが、航技研計算センタでは入出力命令の処理時間が約30msとはほぼ一定しているため、1回の再配置ルーチン起動当り30ms CPUがアイドルとなると考えて良い。この数値とシミュレーション実験におけるリロケーション回数から再配置処理のための総中断時間を算出し、領域移動処理に要する時間を加算すると、実験13の場合のオーバーヘッドは約293秒となる。これを処理装置の負荷率に換算すると約0.4%となり、この値は平均稼働率の上昇分に十分吸収される。

4.5 256 KW境界存在の影響度

航技研で使用しているOS, FACOM230 M-VIIでは、CNPモードによる主記憶割り付けを撤廃することができない^{注)}。このため、256 KW境界を有する割り付け領域が存在し、主記憶の断片化を進める一因となっている。本節ではその悪影響の程度を推測して、今後のための一資料に供したい。この目的から実験3, 8, 11, および14を行い、それぞれ実験2, 7, 10, および13の結果と比較した。なお、実験14は仮想的に実験13から256kw境界の制約を撤廃した場合であるから、明らかに、全く主記憶の断片化が生じない。

総じて言えることは、256 KW境界という制約が無ければ、確実に空領域の個数は減少するが、この効果は実験3と8ではまだ主記憶優先権の低いALジョブとCMジョブが主記憶上にロードされ易くなる程ではなく、このため処理装置稼働率の明らかな向上にはつながらない。しかし、このような状態は実験11では一変する。即ち、主記憶優先権の低いALジョブとCMジョブが主記憶上にロードされ易くなる程まで空領域の平均サイズが拡大していると、少しでも空領域個数が低下すれば処理装置の稼働率は明らかに向上する。なお、実験14においてその効果が低いのは前節で述べたようにシステム資源の利用率がほぼ飽和しているためと思われる。

以上のように、システム利用状況によっては256 KW境界の存在がかなりの悪影響につながるといえる。

4.6 主記憶管理の最良方策

今まで我々は現有システムに対する実験0から始めて、各種の改良機能を追加しながらシステム利用効率の向上

の程度を調べてきたが、最後にこれら全有効機能具备した場合の性能を見るために実験15を行った。この結果は本稿の考察範囲内では最高の性能を示している。実験0と15の結果を目安として、以下にこれまでの検討を総括しておく。

まず、ロールイン・ロールアウト処理の改善は当然必須である。次に、主記憶の割り付け方式に対する改良策がアロケータに限定される場合には、明らかにアロケータ1を採用すべきである。アロケータ0をアロケータ1に置き換えるだけでも、かなりの効果が期待できる。しかし、この改良レベルでは処理装置の能力を使い尽くすることはできない。また、再配置機能を利用する場合、アロケータ2とアロケータ1の組合せでこの機能を実限するのが容易かつ有効である。もしアロケータを主記憶解放事象の発生都度起動するならば、即ち、最大限に主記憶の断片化を防止するならば、実験13と15の比較から明らかなように、主記憶ネックの状態から脱することができ、主記憶の増設なくとも処理装置の能力を使い尽くすることができる。この場合に問題となる再配置処理に要する時間は決して小さいものではないが、総合的システム処理能力の上昇率はそれを吸収してもなお十分余りあるものと予測される。

また、256 KWの主記憶増設は航技研システムに対して絶大な効果をもたらすと考えられ、計算機システムに対する負荷が過大な時にはこの余裕が大変有効に作用するが、その効果はやはり主記憶断片化克服機能の併用によって絶大となる。このような過大負荷時におけるシステム処理能力の差異は20時間シミュレーションから直接明らかにすることはできないが、図3.6に示したラッシュアワー実験結果から明白にとらえることができる。

4.7 運用系の見直し

表3.3の実験結果1~15は全て3.2.4節で設定した運用系に基づいている。この運用系はもともと主記憶ネック状態を強く意識して設定したものである。特にAPUに対しては、AMクラス、ALクラスのジョブ多重度をそれぞれ2.1として合計3多重を許し、AMジョブの主記憶優先権を最高にし、ALジョブの主記憶優先権は最低だが実行優先権を最高にして運用している点に特徴がある。このように、APジョブがAPUリミットでありかつ主記憶要求量が大いにかかわらず、APUに対して3多重を許している理由は2つ挙げられる。第一に、AM、ALジョブ共に1多重とし主記憶優先権も共

注) 業者提供のサービスプログラムを使用する限りCNPモードは撤廃できない。しかし、CPモードのサービスプログラムをユーザが開発使用すれば、CNPモードの撤廃は可能である。

に最高にして運用した場合、APUの稼働率は保証されるが、その反面主記憶競合のためにCPUの稼働率が激減してしまう。第二に、AM、ALジョブ共に1多重にするがALジョブの主記憶優先権をCPクラスより低くして運用した場合には、APUの実質多重度が減少するために稼働率が激減してしまう。このような事情により、APU、CPU共にそれ相当の稼働率を維持しようとするれば、通常は2個のAMジョブを処理し、主記憶に余裕がある時にはALジョブを優先的に処理するように運用するという上記の妥協案が最も有力となる。事実、実験0の結果を見ると、APクラスの多重度は平均1近く低下しており、AP稼働率の8割以上もAMジョブの処理に占められているが、有効主記憶時間比はかなり高く、APクラスのジョブは主記憶を有効に利用しているといえる。当然のことながら、ALクラスのジョブの有効主記憶時間比は非常に高いが、その経過時間指数は極端に悪い。

この運用系のもとで主記憶利用率の向上を計れば、その効果がALジョブの処理状況の改善に向けられ、したがってCP稼働率を低下させることなくAP稼働率を向上できるであろう。アロケータ1の効果を示す実験2はその好例である。即ち、他の事項をほとんど変化させずに、平均AP稼働率を一段と向上させ、AMジョブ処理の割り合いがAP稼働率の6割まで低下してはいるが、AMジョブの経過時間指数をあまり高めることなくALジョブの経過時間指数を著しく減少させている。APおよびCPクラスの平均ロールアウト件数は実験0とはほぼ同じ位大きいから、主記憶はまだかなりの競合状態にあることがわかるが、実験2はアロケータ1を使用すれば、この運用系の所期の目標を十分達しうることを示している。

しかしながら、主記憶ネック状態を脱しうるまでに利用率を向上できた場合、または256KWの主記憶増設をした場合にはこの事情は一変し、上記のようなAPおよびCPクラスによる主記憶競合そのものがなくなり、APUおよびCPUそれぞれに対する実行優先権のみによってジョブは処理されて行く。このため、上記運用系のもとでは実験15が典型的に示すように(以下、実験4, 9, 12, 13でも同様のことが言える)、ALジョブ処理の割り合いがAPU稼働率の70%以上にも増大し、その結果ALジョブの経過時間指数が激減する一方で、AMジョブの有効主記憶時間比を著しく減少させ、経過時間指数を激増させている。このため、主記憶資源の有効利用の観点からすれば、AMジョブが2多重であることには無駄が多すぎるということが指摘される。

このように主記憶競合が解除された場合には、運用系の見直しが明らかに必要となる。運用系を考える場合にはそれ相当の処理装置稼働率を保証する一方、適度のスループットをも保証することも忘れてはならない。後者は全ジョブの70%以上を占めるSHジョブの処理件数を適度に保ち、ディバグジョブに対するターンアラウンドタイムを保証するための考慮である。また、システム資源使用量の大きいジョブクラスが小さいものより実行を優先されるべき根拠はなくなる。以上の3点を具体的指針にした場合、主記憶競合がほとんどない場合のシステムの運用系は次のものが適当と考えられる。

主記憶優先権 $SJ = AM_R = AL_R > CM_R > DT_R > SH_R$

実行優先権 $SJ = AM_R = AL_R > SH_R = DT_R > CM_R$

多重度 $SH = 2, DT = CM = AM = AL = 1$
 SJ : サービスジョブステップ
 SH_R : SHクラスのRUNジョブステップ

3.2.4節の運用系と異なる点は、第一に主記憶競合がなければAMジョブの多重度は1で十分としていること、第二にAPUではRUNジョブステップのみの処理であるから、AMジョブとALジョブの実行優先権を同等にしてラウンドロビン方式で処理すること、第三にAMジョブの1多重をSHジョブに振り向け、主記憶競合の緩和を計ると共にSHジョブのスループット向上を狙っていることである。

表3.3~3.4, 図3.4~3.6における見直し実験は、上記の運用系のもとで実験13を行った結果である。図3.4~3.6から明らかのように、実験13より多少処理装置の稼働率を犠牲にはするものの、ジョブ処理のスループットを確実に向上させる事ができ、表3.4が示すように多重度が5になるまでの時間を大巾に短縮してジョブの混雑をいち早く緩和していると共に、全ジョブが終了するまでの時間をかなり短縮している。

5. 結 言

本稿では現APUシステムにおける主記憶のネック状態に鑑み、この問題を解決すべく主記憶の有効利用のための各種方策を用意し、それらをAPUシステムへ適用した場合に得られる効果をシミュレーション実験によって予測した。

用意した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る。即ち、ロールアウト状態のジョブステップを可能な限りロールインするように処理の改善を

行うものである。第二は現有アロケータによって引き起されている主記憶上の断片化をできる限り克服し、かつ主記憶利用率の大巾向上につながるアロケータを設計することである。第三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。第四は物理的資源に頼らずに、再配置機能の利用によって主記憶の最大限の有効利用を計ることである。

シミュレーション実験はまず主記憶ネックを意識した運用系のもとで行い、ロールイン・ロールアウト処理を改善することによって、ALクラスのAPU寄与率を向上させる実験事例を見出した。アロケータの単体性能としては、割り付け領域の大きいものと小さいものに分けてそれぞれができるだけ互に混在しないように割り付ける方式が最良であることを見出した。再配置機能については、この機能を最大限に利用するならば主記憶の断片化をほとんど防止でき、その効果によってほとんど主記憶競合も解決しうることを見出した。主記憶の増設効果については言うまでもないが、この効果を真に発揮させるためにもやはり再配置機能を併用することが望まれる。

以上の方策を複合利用して主記憶ネックを解消したならば、自ずから運用系の見直しが必要となるが、その際の指針としてそれ相当の処理装置の稼働率を保証する一方、適度のシステムスループットをも保証することを忘れてはならない。この指針に基づいた新運用系を提案し、その有効性をシミュレーション実験によって検証した。

なお、現在使用している多重度および各種優先権が、ジョブクラス毎に固定されているいわゆる静的なジョブスケジューリング方式では、各ジョブが同時にサービスジョブステップの処理を行った場合、APUもCPUも見掛上、一時的に負荷が軽くなってしまうことが度々起りうる。このため、処理装置のアイドル率を5~10%以下にすることはなかなかむずかしい。この問題の解決にはRUNジョブステップのプールを可能とするいわゆる動的なジョブスケジューリングの方式が必要となるが、この方式の実現方法並びにその有効性の検討については稿を改めて報告する。

文 献

- 1) 中村, 石塚, 吉田; 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TM-344, 1978年3月
- 2) 末松, 中村, 吉田, 原田, 三好; FACOM 230-75 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TM-384, 1979年6月
- 3) 末松, 吉田, 中村, 三好; FACOM 230-APシステムのシステムシミュレーション, 航技研TR-590, 1979年11月
- 4) 富士通(株); アレイプロセッサハードウェア解説書
- 5) 富士通(株); FACOM 230 M-VI/VII 解説書

付録 A 再配置に伴う実際の諸問題

第一に、サービスジョブステップのような入出力頻度が多く、短時間に終了するジョブステップをも再配置の対象にすべきか否かの問題がある。

第二は、ある境界を越えて移動できない主記憶割り付け単位の存在である。例えば、FACOM 230 M-VI では CNP モードで割り付けられたものは 256KW 境界を越えて移動できない。このような拘束条件がある場合に、どれだけ再配置により断片化を克服できるかという問題がある。

第三に、なんらかの事象が完了するまでは再配置できないジョブステップが存在する問題である。特に入出力要求を出しているジョブステップは、その要求終了時点まで移動不可能としている OS が普通である。また、入出力実行中であるならば移動不可能であるが、入出力実行待ちならば移動可能としている OS もある。いずれにしてもこの問題は、再配置の実行を中断させる要因となる。

第四には、全アクティブタスクの実行を停止することなく再配置を行える機能が OS に具っているか否かの問題である。もしこの機能がなければ、第三の要因によって CPU はアイドルとなり、このために再配置のために必要とする時間が激増する可能性がある。特に、MT マウントや MT リワインド実行中のジョブステップは、再配置実行を長時間に渡って中断させる。

付録 B ジョブの属性の決定

以下に、昭和53年度のアカウント情報に基づいて得られた推定値を用いて、航技研ジョブ集団から模擬発生したジョブのジョブクラスの決定、そのジョブステップ列の決定、および各ジョブステップのシステム資源使用量の決定の過程を詳述する。

B.1 ジョブクラスの決定

表 B.1 はジョブ集団におけるジョブクラス構成比率と各ジョブクラス内での制御文エラー率を示している。発

生したジョブはまずこの構成比率に基づいてジョブクラスが決定される。次に、そのジョブクラス内での制御文エラー率に従って、このジョブが制御文エラーであるか否かが決定される。JPS では、制御文エラーのジョブはジョブ開設処理は行なわず、ただちにそのジョブを終了させている。

B.2 ジョブステップ列の決定

表 B.2 はジョブ集団におけるジョブステップ列構成比率を表わしている。B.1 でジョブクラスの定まったジョブは、この構成比率に基づいてジョブステップ列が決定される。ここで同表に示すごとく、SH クラスとその他のクラスとでは構成比率が異なることに注意する。また、FORTRAN ジョブステップが APFORTRAN か CPFORTRAN か、また RUN ジョブステップが APRUN か CPRUN かの判定は、そのジョブが AP クラスか CP クラスかによって決定される。

以上のジョブクラスとジョブステップ列等の決定は、全てジョブが発生する度に一様乱数を用いて行われる。

B.3 サービスジョブステップのシステム資源使用量の決定

表 B.3 はサービスジョブステップのシステム資源使用量を決定する分布関数とその母数の推定値とを示している。サービスジョブステップは全て CPU で処理されるが、それらの CP 使用時間は同表に示した確率分布に従って算出される。なお、これらの期対値は別途アカウントデータから推定されており表 3.1 に示した通りである。また、主記憶使用量の分布は和の形で表わしているが、これはそれぞれに対する領域が割り付けられるならば、連続して割り付けられる必要のないことを意味している。また α の値は一様乱数を使って決定される。ファイルアクセス 1 回当たりのチャネルサービス時間は、どのサービスジョブステップでも RUN ジョブステップでも 30ms に集中していることから、一定値 30ms としている。総ファイルアクセス回数 (FLAC) は、先に決定された CP 使用時間を使って同表の式から計算される。平均 CP

表 B.1 ジョブクラス構成比率、制御文エラー率、および各種制限値

ジョブクラス		項目	構成比率 (%)	制御文エラー率 (%)	処理装置使用時間制限値 (秒)	主記憶使用量制限値 (KW)
CP クラス	SH		70.2	5.0	120	60
	DT		5.9	10.1	180	120
	CM		7.3	8.0	1200	160
AP クラス	AM		13.4	5.8	1200	160
	AL		3.2	7.8	3600	300

表B.2 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, CM, 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B.3 サービスジョブステップのシステム資源使用量の分布

項目 ジョブ ステップ名	CP使用時間 \log_{10} (CP) の分布 (秒)	主記憶使用量 の分布 (KW)	ファイルアクセス 1回当りのチャネ ルサービス時間 (ミリ秒)	ジョブステップ当りの 総ファイルアクセス回数 (回)
CPFORTRAN	平均 0.062 標準偏差 0.601 の正規分布	$6 + 28 + 10 + 46$ = 90 (固定)	30 (固定)	$10^{\{0.783 \times (\log_{10}(\text{CP}) - 0.062) + 1.70\}}$
APFORTRAN	平均 0.554 標準偏差 0.775 の正規分布	$6 + 45 + 15 + \alpha$ 80%は $\alpha=40$ (固定) 20%は $\alpha=40 \sim 148$ の 一様分布	30 (固定)	$10^{\{0.601 \times (\log_{10}(\text{CP}) - 0.554) + 1.78\}}$
LIED	LC-1 LCは平均1.199の 指数分布	$6 + 11 + 30$ = 47 (固定)	30 (固定)	$10^{\{1.163 \times (\log_{10}(\text{CP}) - 0.194) + 2.66\}}$
LIBE	LC-1 LCは平均0.270の 指数分布	$6 + 13 + 13$ = 32 (固定)	30 (固定)	$10^{\{2.345 \times (\log_{10}(\text{CP}) + 0.730) + 1.68\}}$
UTILITY	LC-1 LCは平均0.481の 指数分布	$6 + \alpha$ 70%は $\alpha=20$ (固定) 30%は $\alpha=38$ (")	30 (固定)	$10^{\{0.800 \times (\log_{10}(\text{CP}) + 0.519) + 2.44\}}$

サービス時間は CP 使用時間を FLAC で割ることによって求まる。シミュレーション時間を進める一つの要因である CP サービス時間は、この平均 CP サービス時間を平均値とした指数確率分布に従って毎回決定している。

B.4 RUN ジョブステップのシステム資源使用量の決定

表 B.4 は RUN ジョブステップのシステム資源使用量を決定する分布関数とその母数の推定値を示している。CPRUN ジョブステップの各資源使用量の決定方法は B.3 節と同様であるが、APRUN ジョブステップに対しては少し説明を要する。

APRUN ジョブステップではまず総処理装置使用時間 (APCP) が同表の確率分布に従って決定される。アカウント情報の統計解析の結果によると、APRUN ジョブステップはこの値の 1.7% だけ CPU を使用する。こうして CP 使用時間 (= APCP × 0.017), および AP 使用時間 (= APCP × 0.983) が決定される。また、APRUN ジョブステップの平均 AP・CP サービス時間も、CPRUN ジョブステップの場合とは異った算出手順が必要である。アカウント情報の統計解析結果によると、APRUN ジョブステップは AP プログラム実行中に何回かの CP プログラムの呼び出し (以後 CP 通信回数という) を行

い、何回かの CP 通信回数につき 1 回の入出力要求を出す。1 秒当りの CP 通信回数 (CMN) は次の回帰式によって推定される。

$$CMN = 10^{\{-0.504(\log_{10}(AP)) - 1.820\} + 0.599} / 2$$

APU は入出力命令を持たないから、AP プログラムの入出力要求は全てそれが呼び出した CP プログラムから出される (図 3.2 (ii) 参照)。したがって、APRUN ジョブステップに対する平均 AP サービス時間 (APS) と平均 CP サービス時間 (CPS) は、

$$APS = \frac{AP}{APCP \times CMN}$$

$$CPS = \frac{CP}{APCP \times CMN}$$

と求まる。シミュレーション時間を進める APRUN ジョブステップの AP および CP サービス時間は、上記の値を平均値とした確率分布に従って毎回決定している。また、AP プログラムの入出力要求は、ジョブステップ開設時および入出力処理実行後に、

$$n = \lfloor CMN / FLCP \rfloor^{\text{注}}$$

の計算を行い、CP サービスを n 回受けた時に入出力要求が出たものとして処理している。

注) ガウス記号

表B.4 RUNジョブステップのシステム資源使用量の分布

項目 ジョブ ステップ名	APCP使用時間 (APCP)の分布 (秒)	主記憶使用量 (CORE)の分布 (KW)	ファイルアクセス 1回当りのチ ャネルサービス 時間 (ミリ秒)	ディスクへのファイルアクセス頻度 (FLCP)の分布 (回/秒)	MTへのファイルアク セス頻度 (MTCP) の分布 (回/秒)
SH-RUN	APCPは 平均 17.0 の指数 分布 CP = APCP AP = 0	$6 + 4 + \alpha$ α は $\Gamma(11.0, 5.09)$ 分布	30 (固定)	$10^{\{-0.613(\log_{10}(CP) - 0.485) + 1.06\}}$	
DT-RUN	APCPは 平均 25.4 の指数 分布 CP = APCP AP = 0	$6 + 4 + \alpha$ α は 平均 55.2 標準偏差 14.7 の正規分布	30 (固定)	$10^{\{-0.613(\log_{10}(CP) - 0.485) + 1.06\}}$	$\log_{10}(M)$ Mは 平均 1.652 標準偏差 0.491 の正規分布
CM-RUN	APCPは 平均 331.2 の指数 分布 CP = APCP AP = 0	$6 + 4 + \alpha$ α は 平均 71.6 標準偏差 23.6 の正規分布	30 (固定)	$10^{\{-0.613(\log_{10}(CP) - 0.485) + 1.06\}}$	
AM-RUN	APCPは 平均 208.2 の指数 分布 CP = APCP $\times 0.017$ AP = APCP $\times 0.983$	$6 + 4 + \alpha$ α は $\Gamma(7.0, 12.79)$ 分布	30 (固定)	$10^{\{-0.669(\log_{10}(AP) - 1.830) - 0.059\}}$	
AL-RUN	APCPは 平均 736.8 の指数 分布 CP = APCP $\times 0.017$ AP = APCP $\times 0.983$	$6 + 4 + \alpha$ α は 平均 127.7 標準偏差 53.1 の正規分布	30 (固定)	$10^{\{-0.669(\log_{10}(AP) - 1.830) - 0.059\}}$	

航空宇宙技術研究所資料419号

昭和55年7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 共 進
東京都杉並区久我山5-6-17
