

UDC 681.302

航空宇宙技術研究所資料

TECHNICAL MEMORANDUM OF NATIONAL AEROSPACE LABORATORY

TM-384

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

末松和代・中村絹代・吉田正廣
原田公一・三好甫

1979年6月

航空宇宙技術研究所
NATIONAL AEROSPACE LABORATORY

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

末松和代** 中村綱代** 吉田正廣***
原田公一** 三好甫**

1. 緒 言

航技研計算センタ(以下センタと略称する)において昭和53年4月より運用を開始したFACOM 230-75アレイプロセッサ(以下APと略記する)は我が国で初めてのアレイプロセッサであり、FACOM 230-75(以下CPと略記する)と共に非対称マルチプロセッサシステムを構成している。APは科学技術計算に頻発するベクトル計算を高速に実行するための種々のハードウェア機能を有し、^{注1), 2)} その演算能力は最高 22 MFLOPSに達する。

APのハードウェアの特殊な機能、およびそれを使いこなすためのプログラム技術については既に述べたが、CPと合わせてシステムとしての能力を發揮させるためにはプログラム技術の向上だけでは十分でなく、APとCPがそれぞれ十二分にその能力を発揮できるように、ハードウェアの構成と配置、オペレーティングシステム(以下OSと略記する)の機能および運用を整備しなければならない。これはAPシステムのような非対称マルチプロセッサシステムに関しては、特に重要であると考えられる。

本資料は、APシステムに適合した運用系を検討し、OSの問題点を発見し、改善策を検討するために、航技研計算センタが作成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)の機能および概要について報告するものである。

2. JPSの必要性と目的

一般に計算機システムの処理能力の向上を図る場合に^{4), 注2)} 検討すべき問題点には以下のものがある。

- (1) 利用者のプログラム技術の問題。
- (2) ハードウェア資源の量的バランスおよび配置の問題。

(3) システムの運用法の問題。

(4) ソフトウェア資源の配置の問題。

(5) OSのオーバヘッドの問題。

(1)は科学技術計算を対象とするバッチ処理システムの実質的な処理能力の向上に関する重要な問題であるが、緒言で述べたように航技研資料において既に論じた。³⁾ (4)は、OSプログラムの常駐部と非常駐部への振り分け、および大記憶上の種々のファイルの配置法の問題である。これらが適切でないと入出力頻度の急激な増大や入出力待ち時間の増大を招く結果となり、システムの処理能力は低下する。(5)は、OS機能の過度の一般化とOSプログラム自身の非効率性によるOSオーバヘッドの増加の問題である。これはシステムの実質的な処理能力の低下をもたらす。(4)および(5)の問題は、システムのハードウェアおよびソフトウェアモニタ等による計測と、その計測データの解析により問題点の摘出を行い、それに基づいて解決策を検討するのが能率的であると考えられるので、本資料の対象としない。

(2)および(3)は、利用者ジョブのシステム資源に対する要求とその要求量に見合った一定のターンアラウンドタイム(以下TATと略記する)に対する要求を保証した上で、APおよびCPの稼動率の向上を計ろうとする場合、システムのハードウェア構成およびOSの機能は十分であるが、また運用系ジョブクラス系とジョブクラス毎の各種優先権は適切であるかという問題に帰着する。ハードウェア、特に主記憶またはチャネルの能力と配置にボトルネックが存在すれば、APおよびCPの利用率は低下する。また、システムに投入されるジョブ集団

注1) MFLOPSとはMega-Floating-operation-per-secondの略

注2) 文献4) pp123~147

注3) 磁気ドラム装置、磁気ディスク装置等を総称していう。

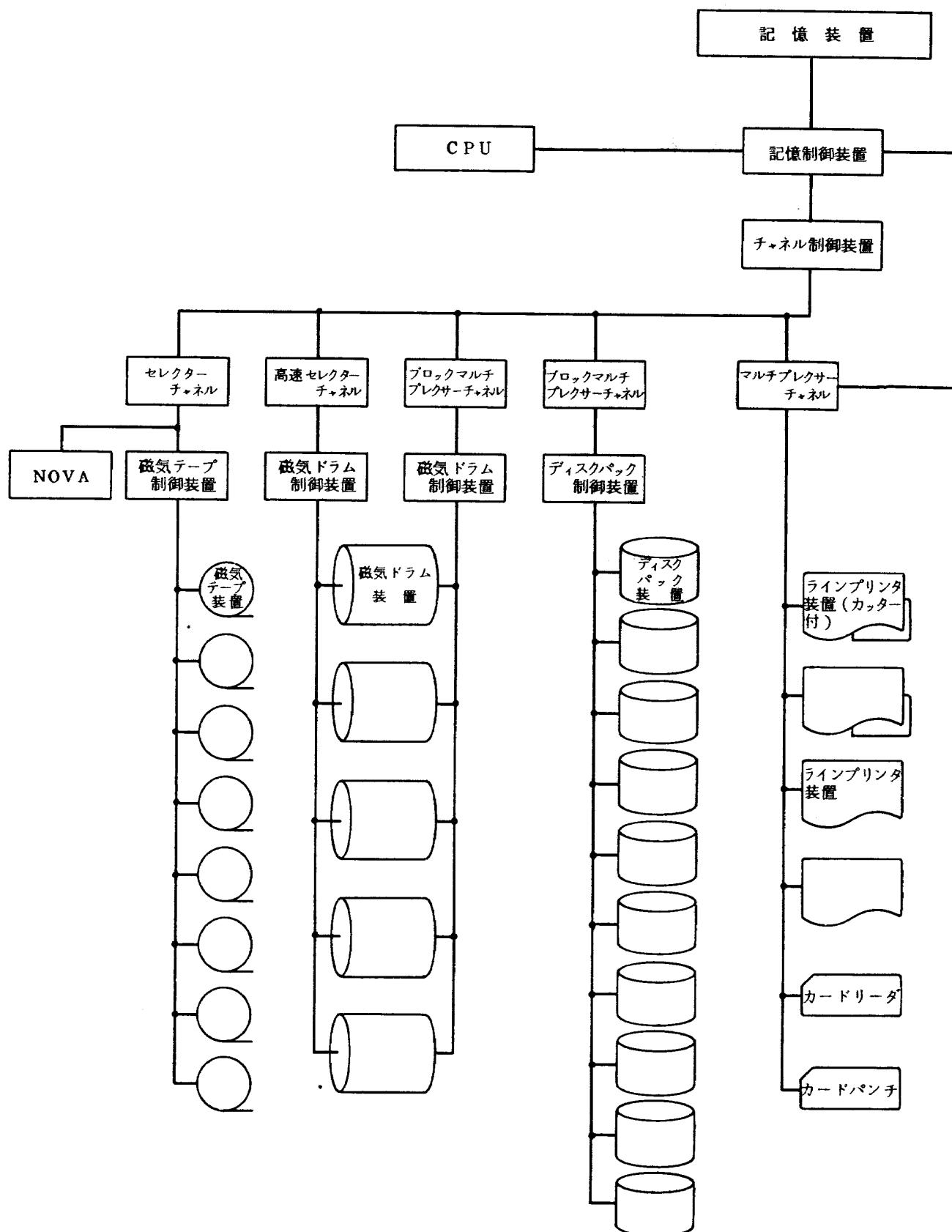
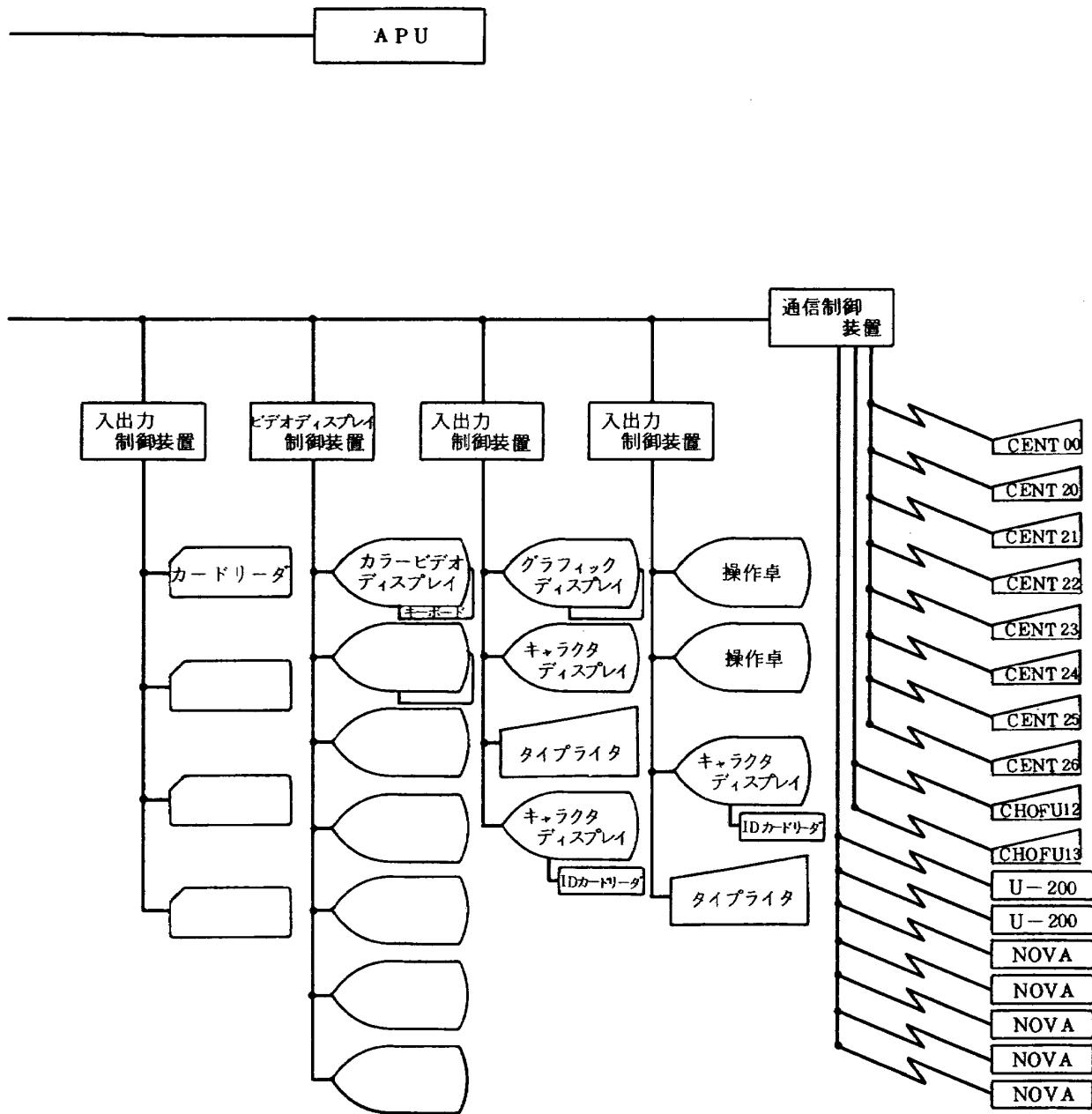


図 3.1.1 A.P.



に対して、OS の外部、内部スケジューラ^{注1)}、および運用系がマルチジョブ効果によるシステム資源の最適利用を図りうる機能と形態をもたなければ、システムの処理能力はやはり低下する。特に AP システムは非対称であり、AP は AP プログラム、CP は CP プログラムしか実行しないため、ハードウェアにボトルネックがあったり、運用系による各種優先権の設定に誤りがあったりすると、AP と CP の負荷のバランスがくずれ、AP および CP の両方の稼動率の向上は望めないことになる。したがって、(2)および(3)の問題は、通常の対称マルチプロセッサ系より重要であると考えられる。

今、アカウントデータの分析により、システムに投入されるジョブ集団が定量的に把握できており、システムの動作の論理が把握できている場合、上述のシステムの問題点を見出し適切な改善策を論じるための手法としては、次の 3 つが考えられる。

- (1) 数学的手法
- (2) シミュレーションによる手法
- (3) 計測データの解析による手法

(1)の手法は、最も経済的でありますマートでもあるが、外部・内部スケジューラ、および運用系といった、複雑なストラテジーを含んだ計算機システムを取り扱うには、システムの核心部分だけに対象を限定した場合でも不向きである。(3)の手法は最も直接的であり、手堅いようであるが、必要とされる計測データの量と項目に経済的な問題があり、計測手段にも制限があること、各項目の計測データ相互の相関関係の解析が容易でないこと、サンプリングの手法が明確でないこと、さらにシステムの改善策が OS の改良を含むような場合には実際にいろいろなシステムを作つて検討することは困難であること等を考慮すると、この手法も主要な手段とする訳にはいかない。(2)の手法は、計算機システムの動作を支配する論理を抽出し、それにより構成されたモデルを、種々の計測データに基づいて推定された計量モデルを媒介として、計算機システムの動作の模擬を行う手法であつて、経済

性では(1)の手法に、手堅さでは(3)の手法に劣るが、システムの動作を支配する論理をかなり詳細に体现するモデルを構成することが可能である。また経済性も(3)の手法よりもはるかに優れている。その上、種々の改善策を容易に比較検討しうるという点からも、他の手法より優れている。

以上の考察から、我々は AP システムにおけるハードウェア構成、OS の外部・内部スケジューラ、および運用の適・不適を検討し、改善策を検討するための JPS の作成を行うこととした。

3. AP システムと JPS

3.1 AP システムの概要とバッチ処理の流れ

AP システムの概要を図 3.1.1 に示す。

センタの入出力機器は、カード読取機 5 台、ラインプリンタ装置 4 台、およびカードせん孔器 1 台があり、他に、リモートバッチシステムが 7 セットとタイプライタ端末が 10 台ある。^{注2)} 入出力機器のうち、カード読取機、ラインプリンタ装置、カードせん孔機各 1 台以外は全てオープン使用である。これらの機器は全てマルチプレクサチャネルを介して処理装置に接続されている。

次に、AP システムにおけるバッチ処理の流れを示す。図 3.1.2 は、バッチサブモニタの構成の概要である。図中、矢印はジョブ処理の流れの方向を示す。ジョブはカード読取機から入力制御リーダ部によりプリスタックファイルに読み込まれ、その後、そのジョブの属するジョブクラス毎の待ち行列につながれる。次に入力制御解釈部は、待ち行列のスケジューリングアルゴリズムに従つてこの待ち行列からジョブを取り出し、ジョブ制御文等の解釈を行い、大記憶上の SYSIN ファイルにジョブを移す。ジョブ開設部は、このファイルからスケジューリングアルゴリズムに従つてジョブを取り出し、ジョブステップ開設部によりファイルの割り当て等が終了したジョブステップは、スーパーバイザの主記憶管理ルーチン^{注3)}のアルゴリズムに従つて主

注 1) 外部スケジューラは、システムに投入され蓄積されているジョブ集団から AP または CP で実行させるジョブを取り出すアルゴリズムをもち、内部スケジューラは、外部スケジューラから取り出されたジョブ集団に、AP・CP・主記憶・チャネル等を割り当てるアルゴリズムをもつ。

注 2) センタの特殊な入出力機器として、グラフィックディスプレイ 1 台、オペレーション用入出力機器 6 台、オフラインの X-Y プロッタ装置 2 セットがある。

注 3) 開設部とはイニシエータをいい、閉設部とはターミネータをいう。

注 4) 内部スケジューラの 1 つを含む。

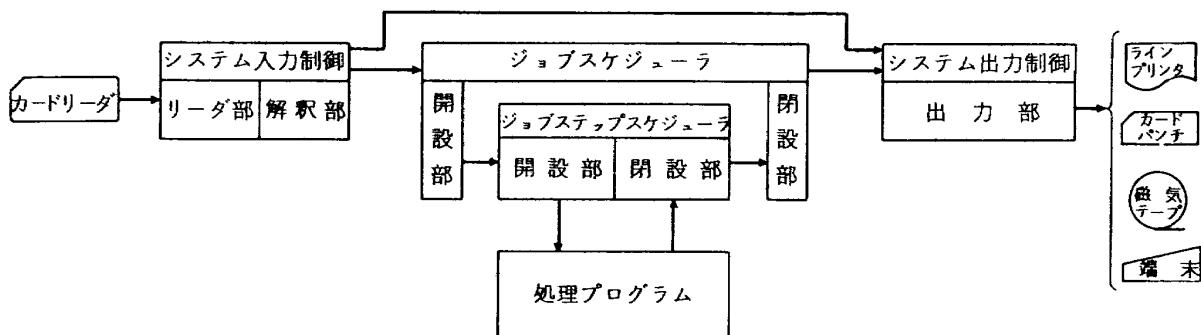


図 3.1.2 バッチサブモニタの構成の概要

記憶上にロードされ、AP または CP で実行される。実行の終了したジョブステップは、ジョブステップ閉設部に渡され、ファイル閉鎖等の終了処理を受ける。ジョブステップ開設部からジョブステップ閉設部までの処理サイクルは、そのジョブを構成するジョブステップの数^{注1)}だけ繰り返される。このようにして全てのジョブステップの実行が終了すると、ジョブはジョブスケジューラの閉設部に渡されジョブ終了の処理がなされる。ジョブが DPS ジョブ^{注2)}でなければ、ジョブの出力は、出力制御出力部により大記憶上の出力ファイルからラインプリンタ装置等に出力される。またジョブが DPS ジョブであれば、利用者からの要求があった時点で同様の経過を経て出力される。^{注3)}

次にジョブステップ実行の概要について述べる。開設されたジョブステップは、タスクという資格を与えられて実行される。以後開設されたジョブステップを便宜的にタスクと呼ぶことにする。図 3.1.3 は、タスクの状態とそれぞれの状態への遷移の条件について示している。図 3.1.4 は AP システム側から見たマルチジョブ処理の概要である。スーパバイザの主記憶管理ルーチンのアルゴリズムにより主記憶に割り付けられたタスクは、実行可能状態にあれば処理装置の前に実行待ち行列を作る。AP システムの場合は、異った機械命令系をもつ 2 つの処理装置 AP と CP により構成されているので、実行待

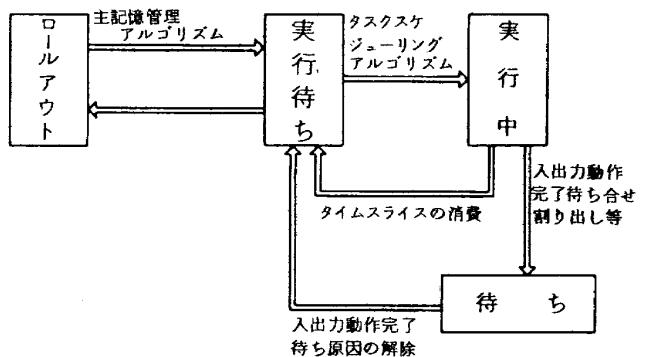


図 3.1.3 タスクの状態

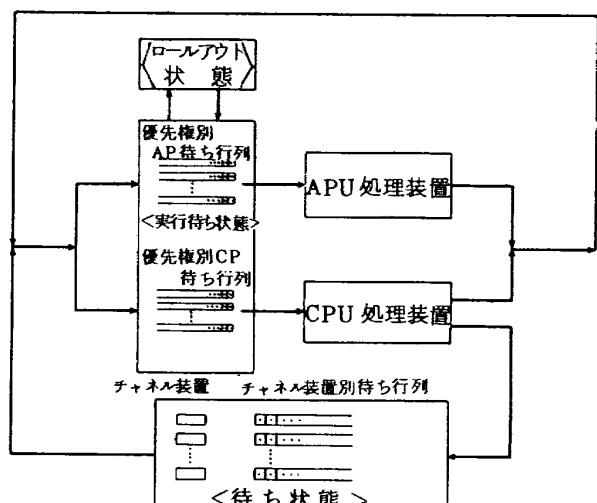


図 3.1.4 マルチジョブ処理の概要

注 1) ジョブステップがスキップされたり、全部終了しないでジョブが終了する場合は、実行されるジョブステップの数をさす。

注 2) DPS はディマンドプリンティングシステムの略。文献 5) 参照。

注 3) この流れは典型的な場合について説明している。

注 4) タスクの本来の定義とは異なる。

ち待ち行列は AP サービス^{注1)}に対するものと CP サービスに対するものの 2つができる。処理装置スケジューラ^{注2)}は、スケジューリングアルゴリズムに従って実行待ち待ち行列からタスクを取り出し、AP または CP にサービスを行わせる。したがって、AP または CP で実行中のタスクはある瞬間をとればシステム中に最大 2 個しか存在しないことになる。実行中のタスクはタイムスライス^{注3)}（AP システムでは 100 ミリ秒）が切れるごとに実行待ち状態となり、再び処理装置の前の実行待ち行列につながる。実行中のタスクが入出力を行う等の理由で待ち状態になった場合や、タイムスライスが切れた場合には、処理装置スケジューラはスケジューリングアルゴリズムにより実行待ち待ち行列から次に実行すべきタスクを取り出して実行する。もし実行待ちタスクがなければ処理装置はアイドルタスクを実行する。待ち状態となったタスクは、待ち原因が解除になると実行待ち待ち行列につながる。入出力命令の発信により待ち状態となったタスクは、チャネルによる入出力サービスが完了すると実行待ちとなる。1 つのチャネルに異ったタスクから要求が発せられた場合には、チャネルに対する待ち行列が形成される。この様な AP, CP, およびチャネルによる実行サイクルがジョブステップの終了まで繰り返される。また、サイクルの途中で主記憶管理ルーチンのアルゴリズムにより、タスクは主記憶からロールアウトファイルへ追い出されたり、逆にロールアウトファイルから主記憶へロールインされることもある。

3.2 JPS のためのシステムモデル

モデルの構成は、2 つの互いに矛盾する要請—“モデルはできる限り正確にシステムを表現しなければならない”および、“モデルはできるだけ計算コストがかからない様簡素に作成すべきである。”—に応えなければならない。これら 2 つの要請のバランスは、シミュレーション

の目的に基づいて図られる必要がある。

航技研のシステムはバッチ処理を主体としているため、会話型処理は副次的な位置にある。したがって、システムモデルを経済的なものにするため、会話型処理は JPS の対象としない。また、シミュレーションの目的は前章にも述べた様に、ハードウェア諸資源の量と配置、OS の諸機能、および運用系等における問題点の摘出と解決策の検討であるから、システムモデルの構成もこの観点から行う。

システムモデルは次の 4 項目から記述される。

- (1) システムに対する負荷。
- (2) システム構成。
- (3) ジョブ処理の流れ。
- (4) スケジューリングおよびシステム資源割り付けのアルゴリズム。

(1) は、システムに投入されるジョブ集団の記述と、システムへの投入規則の決定に帰着する。

我々のシミュレーションの目的は、バッチ処理システムにおけるマルチジョブ効果に重点がおかれてるので、ジョブの記述もそれを考慮してなされる必要がある。我々はジョブをジョブステップ構成と下記に示すジョブステップ毎のデータにより定義する。^{注4)}

- a. AP 使用時間。
- b. CP 使用時間。
- c. ファイルアクセス頻度。
- d. ファイルアクセス 1 回当たりのチャネル使用時間。

ジョブステップ毎にシステム諸資源の使用量を規定するのは、ジョブステップ毎にこれらの量が大巾に変動することを考慮したためである。次に、ジョブの AP および CP 総使用時間、主記憶使用量等に段階的な制限値を設定し、この制限値によりジョブクラスを定義する。こうして定義されたジョブクラス系に種々の運用系—各ジョ

注 1) AP サービスまたは CP サービスとは、AP または CP タスクが実行開始してから待ちまたは実行待ち状態になるまでに受けとる AP または CP の連続的な処理をいう。したがって、1 つのジョブステップの受けるサービス時間の和がそのジョブステップの AP 使用時間または CP 使用時間に達すれば、そのジョブステップは終了することになる。

注 2) 内部スケジューラの 1 つである。

注 3) タスクが連続的に処理装置から受けられるサービス時間の上限値。

注 4) 会話型システムのシミュレーションにおいては、これと異なったジョブの記述がなされる。たとえば、N.R. Nielsen⁷⁾はジョブを、(1) ファイルへ入出力を行う間の CP 使用時間、(2) 端末通信に必要な CP 使用時間、(3) ファイルアクセス 1 回当たりの入出力量、(4) 利用者の思考時間、(5) ジョブの総 CP 使用時間、(6) 必要主記憶量 (7) オーバレイセグメントの大きさ、の 7 つの量により定義している。

注¹⁾
プロセス毎のジョブクラス多重度¹⁾ および各種優先権²⁾ を対応させる。したがって、運用系を決定し、ジョブ集団のジョブクラス構成比率、各ジョブクラス内のジョブの各種ジョブステップ列(たとえばFORTRAN-LI ED-RUN の3ジョブステップから成るジョブステップ列等)に関する構成比率を決定し、各ジョブステップのAP およびCP 使用時間、ファイルアクセス頻度、ファイルアクセス1回当たりのチャネル使用時間の分布を決定すれば、システムに投入されるジョブ集団の記述は完結する。

システムへのジョブ投入時系列は、ジョブ投入の時間間隔の分布(たとえばポアソン分布)を決定すれば、あとはジョブクラス構成比率による振り分けを行うことにより決定する。以上により、システムに対するジョブ負荷モデルは完全に定まる。

(2)のシステム構成については、図3.1.1における入出力機器、磁気ディスク、磁気ドラム、および磁気テープ装置に対するアクセスはチャネルに対するアクセスと考えることにしたため、システム構成から除いた。すなわち、チャネルの下にはブラックボックスが結合されており、そのブラックボックスに対するアクセスに対してはレスポンスタイムの分布を考えることにした。したがってこのモデルでは、たとえば磁気ディスク上のアクセス要求のあったファイルの位置、アームの動き等はレスポンスタイムの分布の中に含まれていることになる。

シミュレーションモデルを現実のシステムに対して簡素化したのは、次の4つの理由による。まず第一にモデルの経済性である。第二に、入出力機器のレスポンスタイムの分布が計測データから推定でき、この分布を用いることが可能だからである。第三に、大記憶装置のレスポンスタイムの改善は本資料の第2章で述べた様に、システムの計測による方がかなり直接的であり、その改善によるシステム能力に対する寄与は、レスポンスタイムの分布を変化させてシミュレーションを行うことにより得られるからである。また第四に、現在入出力機器の能力、台数にボトルネックが存在しないことが外部からの観察により明らかである。

(3)のジョブ処理の流れに対しては、図3.1.2におけるジョブスケジューラの開設部から閉設部までのモデルを作成する。このモデルでは、プリスタック部から解釈部までと、出力ファイルから出力までの過程を省略している。その理由は、システム全体に占めるこの部分の比重の軽さとモデルの経済性である。したがって、システムに投入されたジョブはジョブスケジューラの前に起動待

ち状態の待ち行列を形成し、ジョブスケジューラによるジョブ閉設処理後、システムから退去すると考えられる。ジョブスケジューラの開設部から閉設部までは第2章に述べたジョブ処理の流れをそのまま実現することにした。

(4)については、(3)のジョブ処理のモデルにおけるジョブスケジューラ、ジョブステップスケジューラ、主記憶管理ルーチン、処理装置およびチャネルのスケジューラのアルゴリズムを、FACOM 230-モニタ⁵⁾ に従って定めた。

以上によりJPSのシステムモデルが決定した。

3.3 シミュレーション言語

JPSの目的は、前節で述べたシステムモデルをJPSの論理に従って計算機上で模擬することであるが、いかなる言語を用いてJPSプログラムを作成するかということは、シミュレーションを実行する場合重要なことである。

我々がJPSを作成するにあたって使用可能な言語としては、フォートラン、アルゴル等の一般高級言語の他に、GPSS⁸⁾, SOL⁹⁾ という2つのシミュレーション言語があった。D. Teichroew & J. F. Lubin¹⁰⁾ は、22のシミュレーション言語についてその概要を述べ、そのうち代表的な6つの言語: GPSS II, SIMSCRIPT, CLP, CSL, GASP, SOLについての比較検討を行っている。また、T. H. Naylor, J. L. Balintfy, D. S. Burdick, およびK. Chu¹¹⁾ はフォートラン、GPSS II, およびSIMSCRIPTについて比較を行っている。これらの文献での比較検討の結果は次のとおりである。

(1) プログラム作成の容易さでは

GPSS II > SOL >> フォートラン

(2) プログラムディバッグの容易さでは

GPSS II > SOL >> フォートラン

(3) プログラム作成上の柔軟さでは

フォートラン > SOL > GPSS II

(4) 実行速度では

フォートラン >> SOL, GPSS II

その他、SOLを使いこなすためにはアルゴルの知識が要求される。実行速度に関して我々はGPSSとフォートランの比較をほぼ同一の計算機システムモデル(GPSSのモデルの方が少しばかり簡単なモデルであった。)について行ったが、その結果、フォートランはGPSSに対して最低でも10倍の実行速度を示した。これらの検討結果と、JPSは一度作成すればその後はサブルーチンレベルでの追加、修正、交換を行えば良いという利点と、そ

注1) ジョブクラス多重度とは、システム内で許容されているジョブクラス毎の多重度の上限である。

の目的から見て頻繁に使用されるであろうことを考慮して、フォートランを用いて作成することにした。

3.4 JPS の論理と JPS における事象

我々の JPS は、3.2 節のシステムモデルを可変時間增加モデル¹⁰⁾としてフォートランにより記述したものである。

3.2 節で述べられたシステムモデルでは、事象は時間とともにシステム中に多数発生することになる。ここで事象とは、たとえばジョブのシステムへの投入、ジョブステップの開設、ジョブステップに対する主記憶の割り付け・解放、AP タスク^{注1)}の実行、CP タスク^{注2)}の実行、チャネルによる入出力の処理等である。

システムモデル中に存在する事象の列を

$$\{E_j^i\} ; \quad j = 1, 2, \dots, \quad i = 1, 2, \dots, N$$

とする。ここで i は事象の種類を示し、 j は同じ事象の時間的な発生順序を示す。いま時刻 T で各事象列毎に事象 $E_{j(i)}^i$ までがすでに発生し、次の事象 $E_{j(i)+1}^i$ が $ET_{j(i)}^i$ 時間後それぞれ発生するとする。ここで

$$ET = \min_{1 \leq i \leq N} \{ET_{j(i)}^i\} \quad (1)$$

$$I = \{i ; ET_i = ET\} \quad (2)$$

とおく。これはシステム内で最も近い将来に発生する事象は第 I 番目の事象列の事象 $E_{j(I)+1}^I$ であることを意味する。そこでシステムの時刻を ET だけ進め、全ての事象列について次の事象発生までの時間を ET だけ減ずる。すなわち、

$$T + ET \rightarrow T \quad (3)$$

$$ET_i - ET \rightarrow ET_i \quad (i = 1, 2, \dots, N) \quad (4)$$

とする。したがって $ET_I = 0$ となり、新しい時刻 T において I 番目の事象列の事象が発生することになる。そして I 番目の事象の発生に従って発生する事象があれば、その発生論理に従って発生するまでの時間が計算され、その事象の属する事象列へ加えられる。また、 I 番目の事象列において $E_{j(I)+1}^I$ の次に発生する事象 $E_{j(I)+2}^I$ が発生するまでの時間を計算し、 ET_i とともに

$$j(I)+1 \rightarrow j(I) \quad (5)$$

とする。すなわち、 I 番目の事象列においてすでに発生している事象は $E_{j(I)}^I$ ということになり、新しい時刻 T で各事象列毎に次の事象 $E_{j(i)+1}^i$ が E_i 時間後に発生することになり、再び式(1), (2)により ET と I が計算さ

れる。この様にしてシステム状態は次々と変化する。たとえば、 I 番目の事象がジョブステップ開設終了の事象であり、 $ET_I = a$ ミリ秒であるとすると、システムの時刻は a ミリ秒だけ進められる。そして各事象列において次の事象の発生までの時間が a ミリ秒ずつ減らされ、ジョブステップ開設終了の事象が発生する。そして、このジョブステップの主記憶への割り付けが可能か否かを調べ、可能であれば主記憶割り付けに要する時間を計算し（4.5 節参照のこと）、主記憶割り付け終了の事象を発生する。もし、主記憶への割り付けが不可能であればロールアウトされる時間が計算され、ロールアウトの事象が発生する。また、どちらの場合も主記憶管理ルーチンのロールイン／ロールアウトアルゴリズムにより、他のジョブステップがロールインやロールアウトされることがあるが、この場合もジョブステップ毎にその事象発生までの時間が計算され、事象が発生する。そして、次に発生するジョブステップ開設終了の時間が ET_I に入れられ、この事象列での事象発生回数 $J(I)$ が 1 増やされる。ここで再び式(1), (2)が使用され、次の ET と I が求められる。

この様にして、ある事象が発生することによりシステム状態を次々に変え、その都度式(1), (2)により次に発生する事象と発生するまでの時間が計算されて時刻が進められ、あらかじめ決められたシミュレーション終了時刻まで実行を続ける。以上が我々の JPS プログラム論理である。

次に JPS 内部で発生する事象列について説明する。以下事象列 $\{E_{j(i)}^i\}$ を記号的に E^i で示す。

(1) E^1 ；シミュレーション結果出力

E^1 はシステムモデルの動作と関係のない仮想的事象であって、シミュレーションにおける結果を周期的に出力させるための事象である。

(2) E^2 ；ジョブ発生^{注2)}

E^2 は一定時間間隔、または特定の確率分布法則に従った時間間隔で発生する。

(3) E^3 ；ジョブ投入

E^3 はジョブ発生の一定時間後に発生する。

(4) E^4 ；ジョブ開設

(5) E^5 ；ジョブ閉設

注 1) CP で実行されるタスクを CP タスク、AP で実行されるタスクを AP タスクという。一般的には 1 つのジョブから CP タスクも AP タスクも発生する。

注 2) ここでいう一定時間とはランダムではないという意味で使っている。また、ここでいう時間は全てシミュレーションモデルにおける時間である。

E^4 , および E^5 は、ジョブスケジューラ起動後、スケジューラの指示に従って、それぞれ一定時間後(4.3節参照のこと)に発生する。

- (6) E^6 ; ジョブステップ開設
- (7) E^7 ; ジョブステップ閉設
- (8) E^8 ; ジョブの最終ジョブステップ閉設
- E^6, E^7 , および E^8 は、ジョブステップスケジューラ起動後、スケジューラの指示に従って一定時間後(4.4節参照のこと)に発生する。
- (9) E^9 ; タスクの主記憶への割り付け
- (10) E^{10} ; タスクの主記憶からの解放
- (11) E^{11} ; タスクの主記憶へのロールイン
- (12) E^{12} ; タスクの主記憶からのロールアウト
- E^9, E^{10}, E^{11} , および E^{12} は、主記憶管理ルーチン起動後、主記憶管理ルーチンの指示に従って、それぞれ一定時間後(4.5節参照のこと)に発生する。

(13) E^{13} ; AP サービス開始
 (14) E^{14} ; CP サービス開始
 (15) E^{15} ; チャネルサービス開始

E^{13} , および E^{14} は、AP およびCP の前にそれぞれ実行待ちタスクがあり、AP およびCP が使用可能であれば常に発生する。また E^{15} は、チャネルの前に待ちがあり、チャネルがサービス可能の状態であれば常に発生する。

- (16) E^{16} ; AP サービス終了
 - (17) E^{17} ; CP サービス終了
 - (18) E^{18} ; チャネルサービス終了
- E^{16} は E^{13} の後に、 E^{17} は E^{14} の後に、 E^{18} は E^{15} の後に、特定の分布法則に従った時間間隔後に発生する。

(19) E^{19} ; ジョブステップ終了

E^{19} は、そのジョブステップのCP またはAP 使用時間を使い切ったとき発生する。

(20) E^{20} ; プログラム制御上の仮想事象(a)

(21) E^{21} ; プログラム制御上の仮想事象(b)

E^{20}, E^{21} は

$$ET_1 = \text{MIN}\{E^1 \sim E^8, E^{10}, E^{19}\}$$

$$ET_2 = \text{MIN}\{E^9, E^{11} \sim E^{18}\}$$

としたとき、それぞれ $ET_1 < ET_2$, $ET_1 > ET_2$ のとき発生する(詳細は4.6節で述べる)。

JPS は、上述の事象列の発生する順番を決定し、発生した事象に応じてシステムの状態を変化させる。したがって、JPS の構造は、最も近い将来に発生する事象を決定するメインルーチンと、事象に応じてシステムの状態を変化させるサブルーチンと、シミュレーション結果を出力させるサブルーチンから成る。

図 3.4.1 は JPS のプログラム構成と、事象によるプ

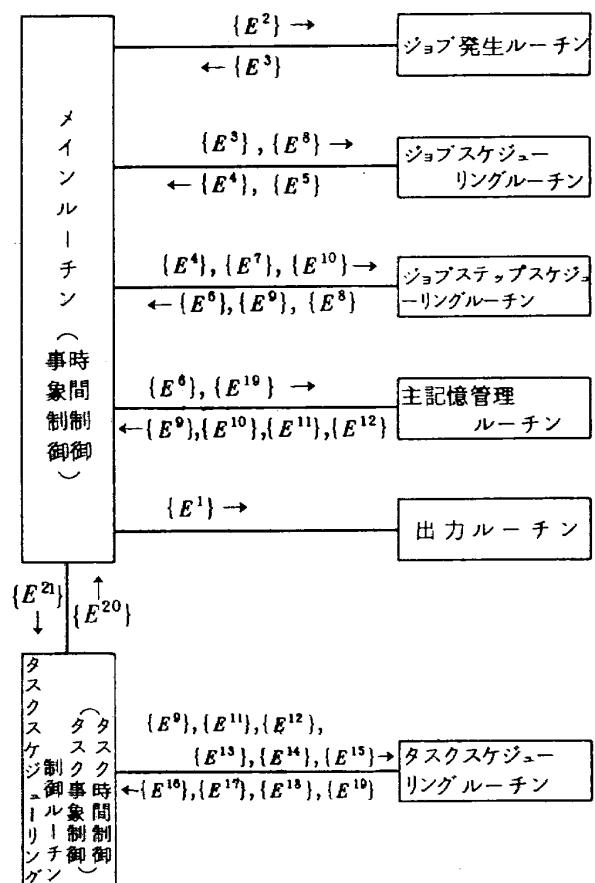


図 3.4.1 JPS のプログラム構成と
プログラム制御の流れ

ログラム制御の流れを示したものである。これらのサブルーチンは、出力ルーチンとジョブ発生ルーチンを除くと、OS のモジュールとほぼ対応している。この図で ' $E^i, E^{i'}, \dots$ ' はサブルーチンが呼ばれる原因となる事象を示し、「 $\leftarrow E^i, E^{i'}, \dots$ 」はサブルーチンによるシステム状態の変化により発生する事象を示している。

図 3.4.2 は、1つのジョブにおける事象発生の流れを示したものである。

4. JPS プログラムの概要

JPS は図 3.4.1 の様にシステムの時刻と事象に応じてシステムの状態を変更する各種サブルーチンおよびシステムに投入されたジョブを記述するジョブ発生ルーチン、シミュレーション結果を出力する出力ルーチンより構成されている。

本章ではこれらのルーチンの概要について述べる。

4.1 メインルーチン

メインルーチンは、事象 $E^1 \sim E^8, E^{10}, E^{19}, E^{21}$ の 11 種

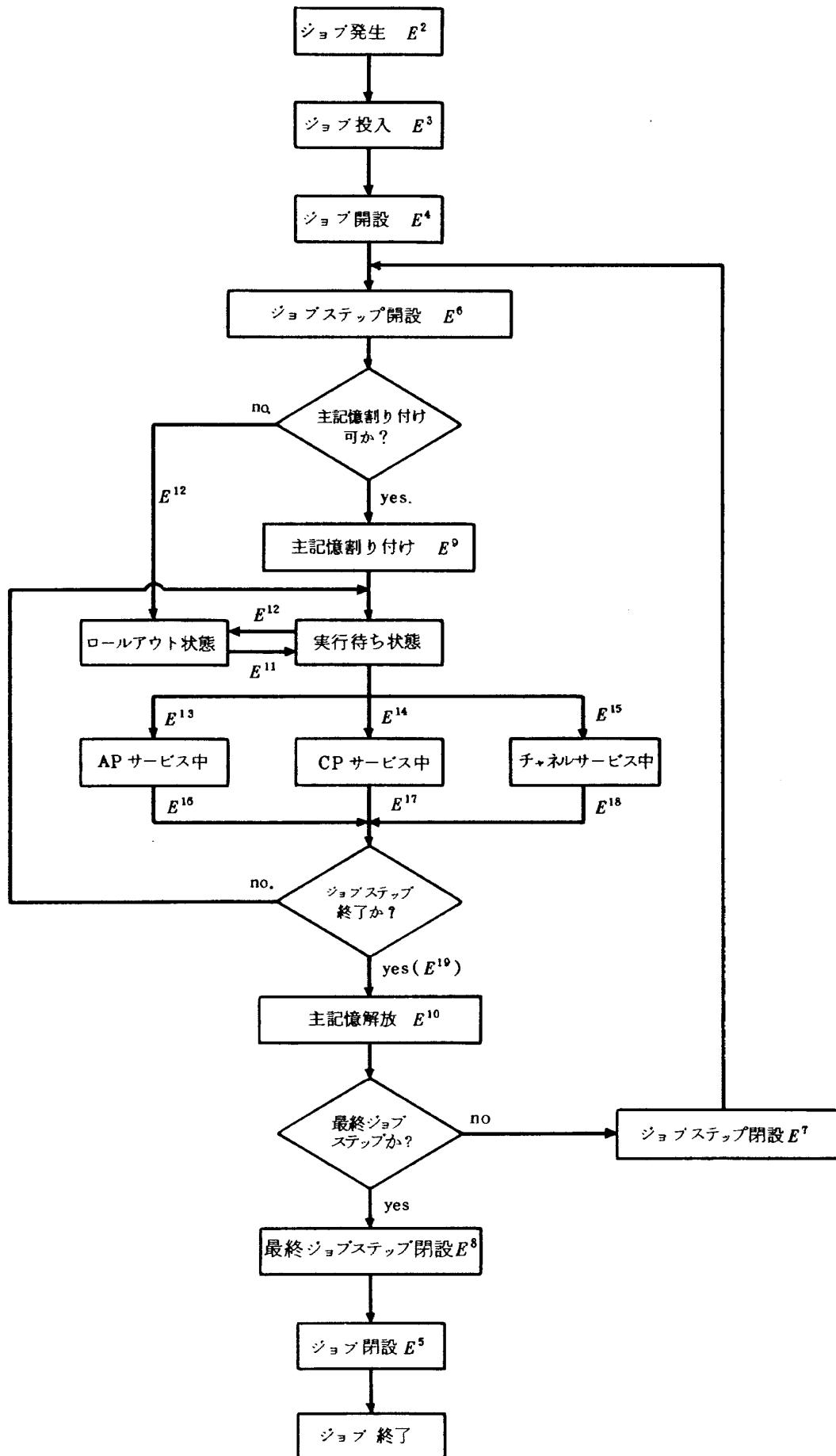


図 3.4.2 1 つのジョブにおける事象発生の流れ

表4.1.1 メインルーチンと各サブルーチンとのインターフェース

サブルーチン名	事象	emainルーチンからの引数	応答	サブルーチンからの引数
出力ルーチン (LPOUT)	結果出力 (E ¹)			
ジョブ発生ルーチン (GNRT)	ジョブ発生 (E ²)	IAT; ジョブ発生番号		IT; 次のジョブ発生間隔
ジョブスケジューリングルーチン (JCON)	ジョブ投入 (E ³)	(IP=1)	ジョブ開設不可	(IP=0)
	最終ジョブステップ閉設 (E ⁸)		ジョブ開設可	(IP=1) IT; ジョブ開設終了時間 (=C ₁) IJ; ジョブ名
	ジョブ閉設 (E ⁵)		開設ジョブなし	(IP=0)
ジョブステップスケジューリングルーチン (JS CON)	ジョブ開設 (E ⁴)	(IP=1)	開設ジョブ有	(IP=1) IT; ジョブ開設終了時間 IJ; ジョブ名
	ジョブステップ閉設 (E ⁷)		次にジョブステップなし	(IP=2) IT; ジョブステップ閉設終了時間
	主記憶開放 (E ¹²)		次にジョブステップ有	(注) つづいて E ¹² の処理を行う
主記憶管理ルーチン (CALT) (注 ₁)	ジョブステップ開設 (E ⁶)	(IP=1) IJ; ジョブ名	割りつけ不可	(IP=0)
	ジョブステップ終了 (E ¹⁰)		割りつけ可	(IP=1)

(注₁) 主記憶管理ルーチンからの受け渡しデータのうち、ロールイン・ロールアウトに関するものは、テーブルの形にし、別に設けた。

類の事象制御を行うが、連続して処理できる事象は1つの事象とみなすと、実際に扱う事象は、E¹, E², E⁴, E⁶, E⁸, E¹⁰, E²¹の7種類となる。メインルーチンはこれらの事象のうち最も近い将来に発生する事象を3.4節の論理に従って決定し、システムの時間をすすめ、各事象発生までの時間を更新する。

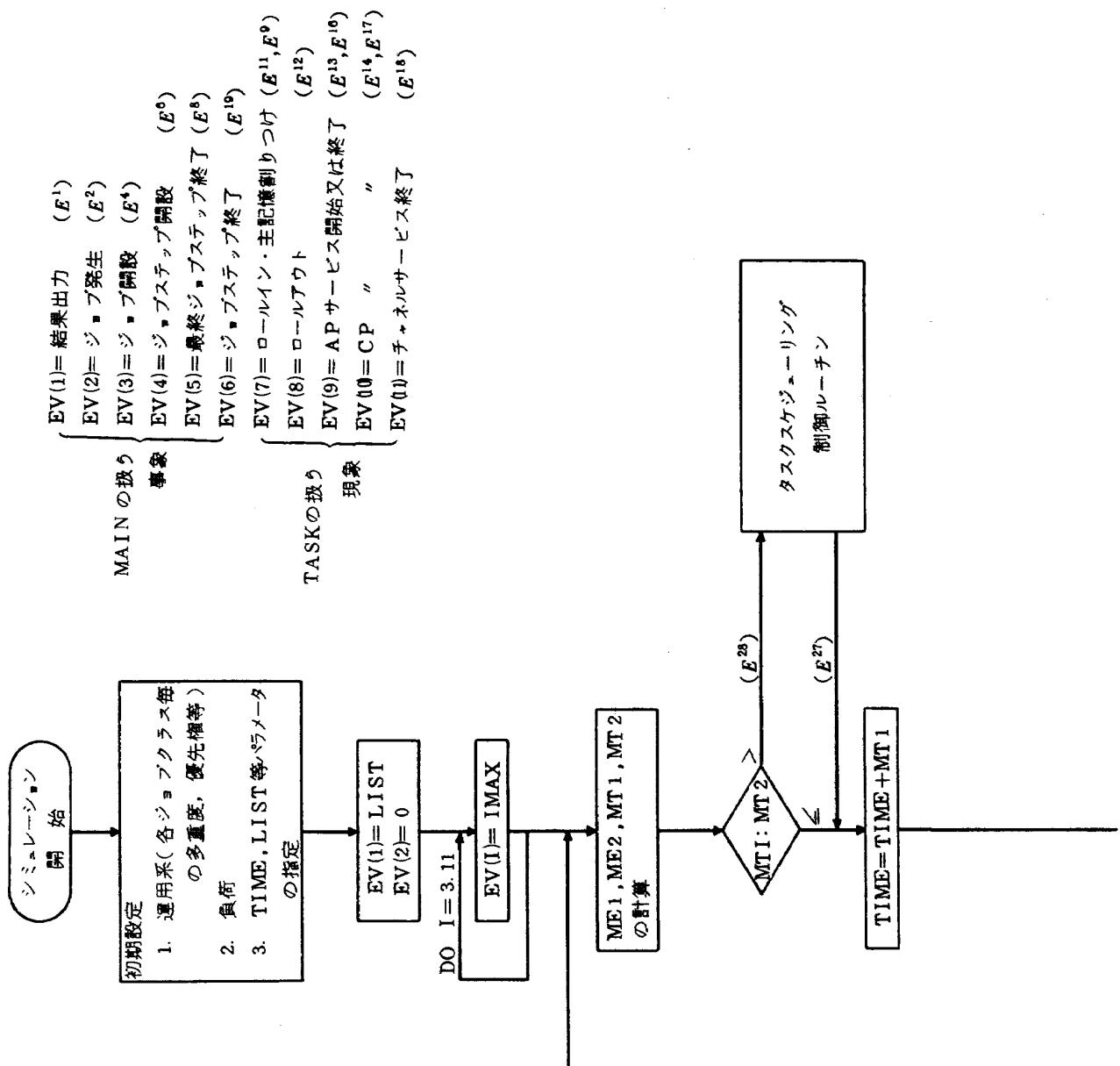
発生する事象が決定するとメインルーチンはシステムの状態変更の依頼を事象に対応するサブルーチンに対して行なうが、これらの処理を各サブルーチンへ依頼する際、表4.1.1に示す様な引数が使用される。ここで、メインルーチンから各サブルーチンへ渡される引数IPは事象内容を示し、各サブルーチンからの持ち帰り引数IPはその事象に対する応答を示す。図4.1.1にメインルーチンのフローチャートを示す。

4.2 ジョブ発生ルーチン(サブルーチンGNRT)

シミュレーションにおいてシステムに投入されるジョブ集団は3.2節に示した様に規定される。GNRTはジョブ発生事象によりメインルーチンからジョブの発生を依頼されるとジョブ集団内のジョブクラス構成比率、各ジョブクラス内のジョブステップ列構成比率に従い乱数により発生したジョブのジョブクラス、ジョブステップ列を定める。各ジョブステップ毎のAPおよびCP使用時間、主記憶使用量、AP、CPおよびチャネルサービス1回当たりの平均時間を前もって定めた確率分布法則に従って決定し、シミュレーションに必要なジョブの属性を定める。この様に定まった発生ジョブの属性は表KJOBお

表4.2.1 KJOB

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



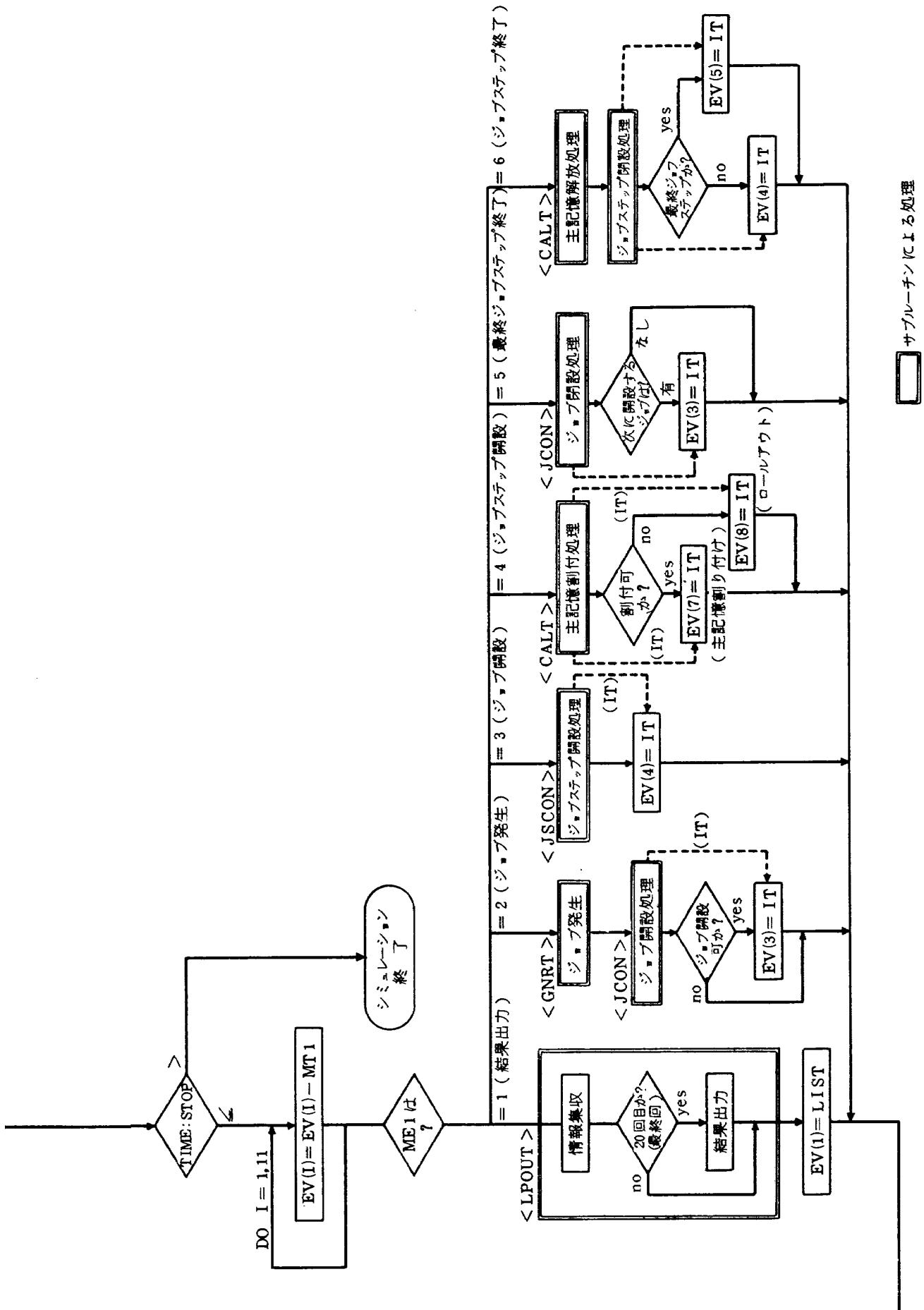


図 4.1.1 メインループ処理フロー

サブルーチンによる処理

表 4.2.2 I ; システムに滞在可能なジョブ数に対応
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

より KJOBS に記入される。この表は他の全てのルーチンで参照可能である。

4.3 ジョブスケジューリングルーチン (サブルーチン JSON)

このルーチンの機能説明に先だって JPS で採用したジョブスケジューリングアルゴリズムとボーキングについて説明しておく。

(1) スケジューリングアルゴリズム

このルーチンでは、FACOM 230 M-VI/VII システムに合わせてアンリミテッドジョブクラス^{注1)}を 7 種類、リミッテッドジョブクラス^{注2)}を 8 種類扱うことが可能である。また、スケジューリングアルゴリズムも同様に、アンリミテッドジョブクラスとリミッテッドジョブクラス別多密度の指定の有無により、下記の 4 つのモードの選択が可能である。

- ① クラス別多密度の指定がないとき。
 - ② アンリミテッドジョブクラスのクラス別多密度が指定されているとき。
 - ③ リミッテッドジョブクラスのクラス別多密度が指定されているとき。
 - ④ アンリミテッドジョブクラスとリミッテッドジョブクラスのクラス別多密度が指定されているとき。
- ここではモード④のスケジューリングアルゴリズムを説明する。（他のモードについては文献 5 を参照。）
- スタックされたジョブの中から、まず指定されたリミッテッドジョブクラスのジョブを選択し、ジョブ優先権の高い順にクラス別多密度まで起動する。クラス別多密度まで起動したがバッチ多密度に達しないとき、および、指定されたリミッテッドジョブクラスのジョブはすべて起

動したがクラス別多密度に満たない場合は、指定されたアンリミテッドジョブクラスのジョブをジョブ優先権の優先順位に従って指定多密度まで起動する。アンリミテッドジョブクラスのジョブをクラス別多密度まで起動してもバッチ多密度に達しないとき、および、指定されたアンリミテッドジョブクラスのジョブがクラス別多密度に満たない場合は、残りのアンリミテッドジョブクラスのジョブクラスは一切意識せずにジョブ優先権に従ってバッチ多密度まで起動する。

(2) ボーキング

JPS ではプログラムの関係上、起動待ちジョブとして保持できるジョブを、ジョブクラス別に 199 まで、システム全体で 500 までに制限している。したがって、これ以上になった場合の到着はボーキング（立ち去り）として扱かわれ、システム内には投入されない。

(3) JSON の機能

このルーチンは、論理的にはジョブ投入 (E^3) とジョブ閉設 (E^5) および最終ジョブステップ閉設 (E^8) の発生によってメインルーチンから呼ばれなければならないが、ジョブ閉設は最終ジョブステップ閉設に伴って発生し処理されるため、実際にはジョブ投入と最終ジョブステップ閉設によってのみメインルーチンから呼ばれることがある。その処理において、シミュレーション初期設定時に定められた次の値を用いる。

C_1 ; ジョブ開設に要する時間

C_2 ; ジョブ閉設に要する時間

以下にこれらの事象に応じた処理手順を示す。

i) ジョブ投入 (E^3) の場合

ジョブ投入の場合は、まずボーキングかどうかを調べもしボーキングであればボーキング表を更新して引数を $IP = 0$ (投入されたジョブの開設不可) としてメインルーチンへ復帰する。ボーキングでなければ、起動待ち行列のその時点までの統計計算を行った後に、投入されたジョブをその起動待ち行列へつなぐ。そして投入されたジョブのデータ、表 KJOB と KJOBS からこのルーチンで使用する表 IJA と IAJOB を作成し、ス

表 4.3.1

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
-----------	--------	----------

注 1) バッチ多密度に余裕があれば、指定されたクラス別多密度を越えてバッチ多密度まで開設できるジョブクラス。

注 2) バッチ多密度に余裕がある場合でも、指定されたクラス別多密度以上に開設できないジョブクラス。

表 4.3.2 I ; システムに滞在可能なジョブ数に応じ
IAJOB(I,J) J ; ジョブステップ数に応じ

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

ケジューリングアルゴリズムによりジョブの開設が可能かどうかを調べる。可能でなければ引数を $IP = 0$ (開設不可)としてメインルーチンへ復帰する。開設が可能であれば、投入したジョブに対してジョブ名を発行し、そのジョブに関する表NJOBを作成する。この表は他の全てのルーチンから参照可能である。そして、引数を $IP = 1, IT = C_1$ としてメインルーチンへ復帰する。ここで IT はジョブ開設終了までに要する時間である。

表 4.3.3 I ; システムに滞在可能なジョブ数に応じ
NJOB(I,J) J ; ジョブステップ数に応じ

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

ii) 最終ジョブステップ閉設 (E^8) の場合

最終ジョブステップ閉設の場合は、ジョブステップスケジューリングルーチンで作成されたジョブステップの統計データ集収表ATABのデータを、ジョブクラス別の統計データ集収表IABJへ和と2乗和の形で累積し、ジョブ閉設 (E^5) を発生する。この事象によりジョブの新たな開設が可能となる。そこで、スケジューリングアルゴリズムに従って次に起動すべきジョブがあるか否かを調べ、起動するジョブがなければ引数を $IP = 0$ (ジョブ開設なし) としてメインルーチンへ復帰する。起動するジョブがあれば、待ち行列に関する統計計算を行い、

表 4.3.4

IABJ(K) K ; ジョブクラスの数に応じ

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

そのジョブにジョブ名を発行し、そのジョブに関する表NJOBを作成する。そして、引数を $IP = 1, IT = C_1 + C_2$ としてメインルーチンへ復帰する。ここで IT は起動ジョブが開設終了までに要する時間である。JCONフローチャートを図 4.3.1 に示す。

4.4 ジョブステップスケジューリングルーチン(サブルーチン JSCON)

このルーチンは、論理的にはジョブ開設 (E^4) とジョブステップ閉設 (E^7) および主記憶解放 (E^{10}) の発生によりメインルーチンから呼ばなければならないが、ジョブステップ閉設は、主記憶解放に伴って発生し処理されるため、実際にはジョブ開設と主記憶解放の場合のみこのルーチンはメインルーチンから呼ばれる。その際、シミュレーション初期設定時に定められた以下の値が用

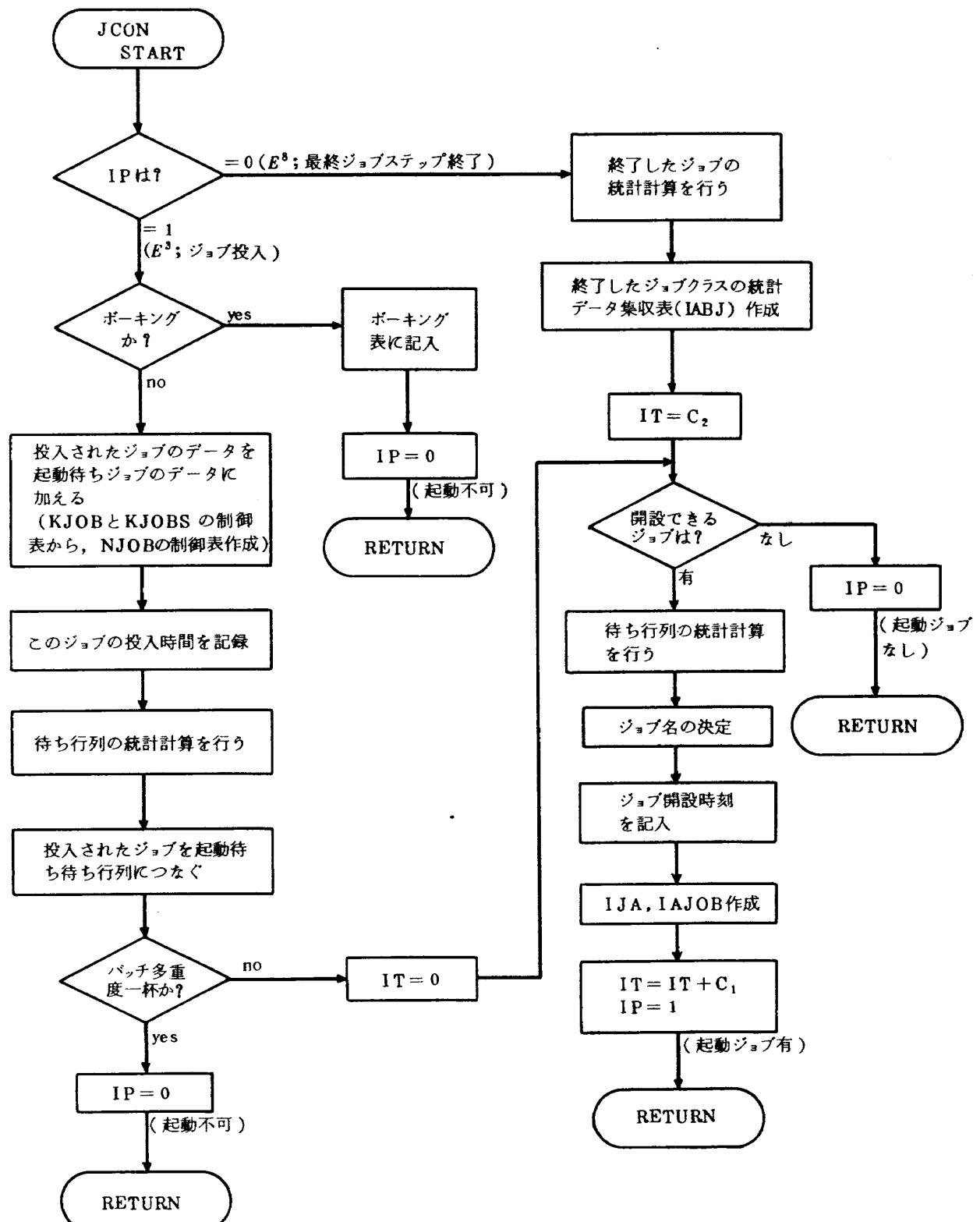


図 4.3.1 JCON フローチャート

いられる。

C_3 ; ジョブステップ開設に要する時間

C_4 ; ジョブステップ閉設に要する時間

以下にこれらのこと象に応じた処理手順を示す。

i) ジョブ開設 (E^4) の場合

ジョブ開設の場合は、JCONで作成された表NJOBから、主記憶の割り付けの際必要な表IJOBSと、タスク処理の際必要な表IJOBCを作成する。また、このジョブステップの開設時刻、このジョブの総ジョブステップ数および実行するジョブステップのジョブステップ順位を表IJOBSQに記入する。これらの表は他の全てのルーチンから参照可能である。そして、引数を $IP = 1$ (ジョブステップ開設有), $IT = C_3$ にしてメインルーチンへ復帰する。ここで IT はジョブステップが開設終了までに要する時間である。

ii) 主記憶解放 (E^{10}) の場合

主記憶解放の場合は、主記憶管理ルーチンとタスクコントロールルーチンで集収されたデータと表IJOBSQ

表 4.4.1

IJOBS (M) M; バッチ多度に対応

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)

表 4.4.2

IJOBC (M) M; バッチ多度に対応

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

表 4.4.3

IJOBSQ (M) M; バッチ多度に対応

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

から、そのジョブステップに関する統計データ集収表ATABを作成し、プログラムに関する統計データ集収のためIC, このジョブステップに属するプログラムの統計データ集収表TABP IC, 和と2乗和の形で累積する。ジョブに関する統計データも、表ATABを基にして、JC

表 4.4.4

ATAB (M, I) M; バッチ多度に対応
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	チャネル使用時間

表 4.4.5

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

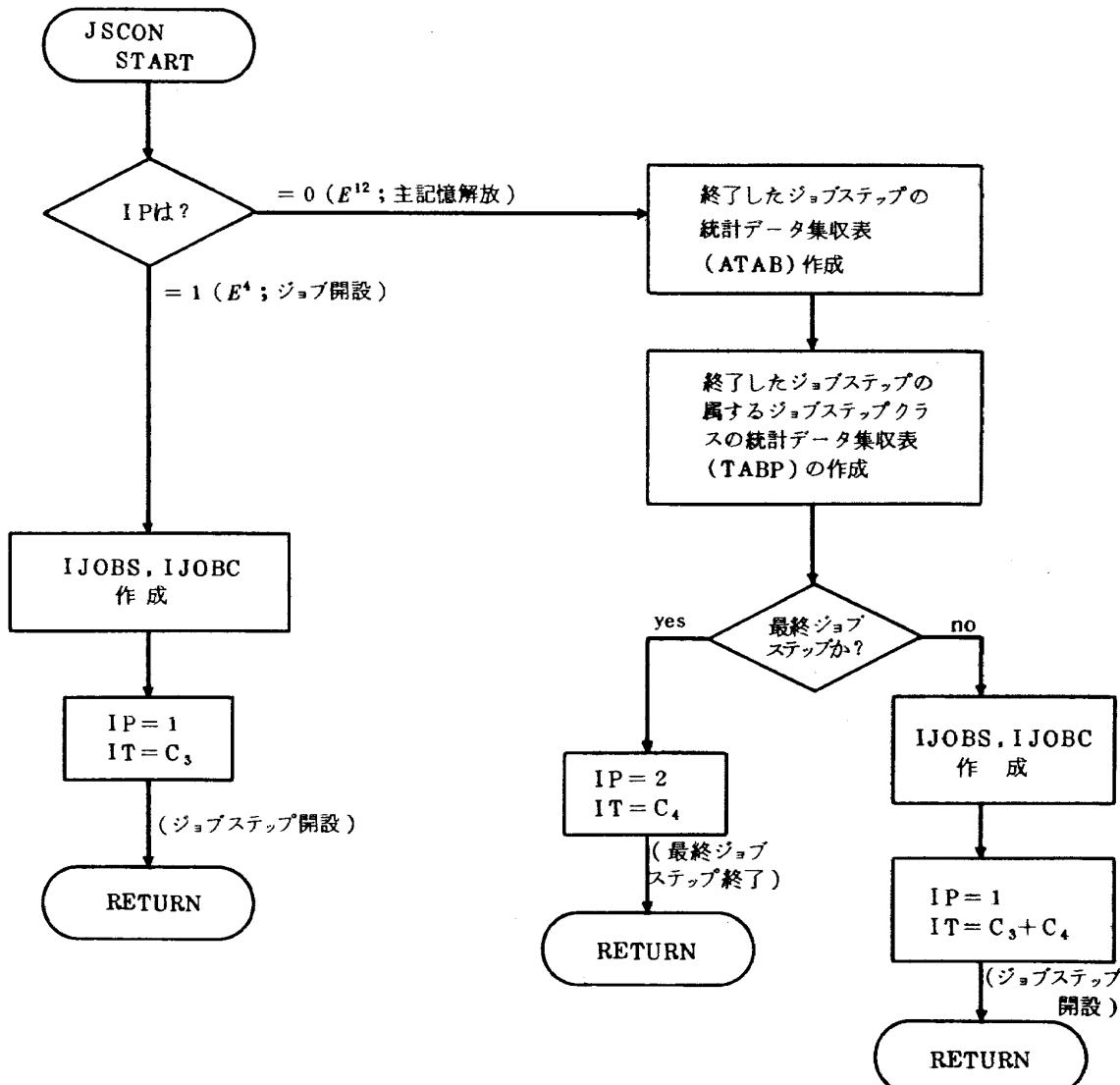


図 4.4.1 JSCON フローチャート

ON で処理される。そしてこのジョブステップが最終ジョブステップであるか否かを調べ、最終ジョブステップであれば、引数を $IP = 2$ (最終ジョブステップ)、 $IT = C_4$ にしてメインルーチンへ復帰する。ここで IT はジョブステップが閉設終了するまでに要する時間である。又、最終ジョブステップでない場合は、ジョブステップ閉設となり後続のジョブステップのために表 IJOBSQ を更新し、引数を $IP = 1$ (ジョブステップ開設有)、 $IT = C_3 + C_4$ にしてメインルーチンへ復帰する。ここで IT は後続のジョブステップが開設終了するまでに要する時間である。

JSCON のフローチャートを図 4.4.1 に示す。

4.5 主記憶管理ルーチン(サブルーチン CALT)

このルーチンの機能説明の前にここで採用した FACOM 230 M-V / VII の主記憶割り付けアルゴリズムについて説明しておく。

(1) 領域のモードと割り付け

領域は 256 K語境界にまたがって配置することのできない CNP モード⁵⁾ と 256 K語境界にまたがって配置することのできる CP モード⁵⁾ に論理的に分類される。航技研では、ユーザーの実行形式プログラムが CP モードであり、その他が CNP モードである。

割り付けは、CP モード、CNP モード共にハイアドレスからロー アドレス方向へ行われる。その割り付け手順を図 4.5.1(a), (b) に示す。CNP モードの割り付けは 256 K語境界毎に行われる(図 4.5.1(a)の①~③)又、CP モードの割り付けは、まず 512 K語境界からスタートして 512 K語境界をまたがない様に行われ(図 4.5.1(b)の①~②)，次に 512 K語境界から 256 K語境界をまたいで行われる(同③)。割り付け要求量が 256 K語以上のものに対してのみ、512 K語境界だけをまたいだ割り付け(同④)と、境界をすべて無視した割り付け(同

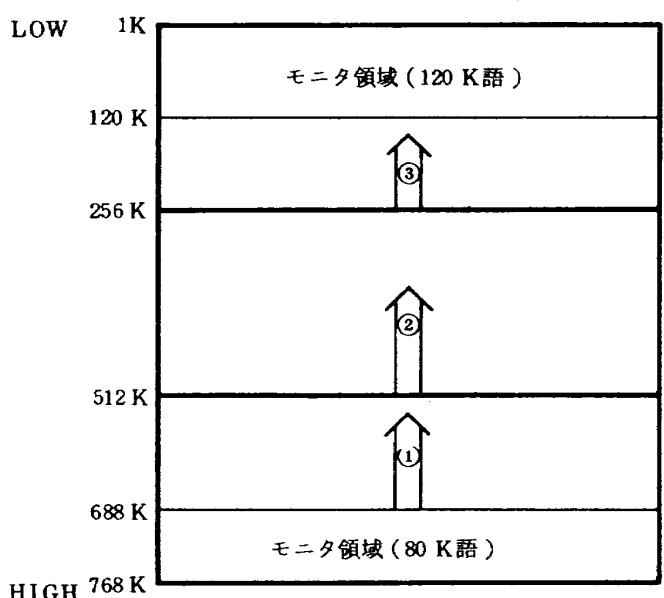
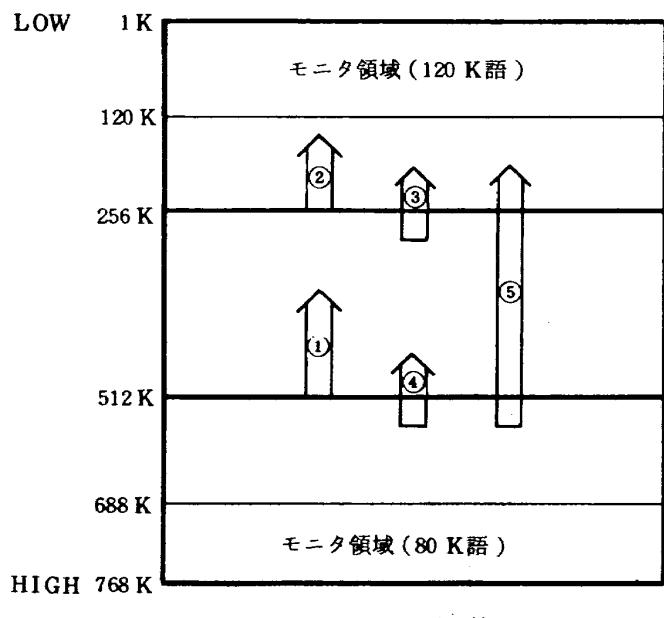


図 4.5.1

⑤) が行われる。

(2) 主記憶割り付けアルゴリズム

割り付けられる各ジョブステップは、表 4.5.1 に示される様に、2 ~ 4 個の領域から構成されており、割り付けは各々の領域に対して行われる。例えば RUN ジョブステップを考えた場合、まず CNP モードの 6 K語、次に CP モードであるユーザの実行形成プログラムの領域が割り付けられ、最後に CNP モードの 4 K語の領域が割り付けられる。もし、それらの領域が 1 つでも割り付けできなければ、その時点で割り付けは打ち切られ、ロールイン／ロールアウトアルゴリズムによる処理が行われる。

表 4.5.1 プログラム名と領域の大きさ

プログラム名	領域 ブロック ナンバー	1	2	3	4
APFORTRAN	6	48	14	8 + $\frac{CN}{15}$	
CPFORTRAN	6	28	16	5 + $\frac{CN}{100}$	
LIBD	6	10	32		
LIBE	6	13	13		
UTILITY	6	20			
APRUN	6	PS	4		
CPRUN	6	PS	4		

CN : カード枚数

PS : ユーザ・プログラムサイズ

単位 K語

(3) ロールイン／ロールアウトアルゴリズム

ロールアウトアルゴリズムによる処理は、割り付けを要求しているジョブステップの割り付けができない場合に行われ、まず、そのジョブステップより下位の主記憶優先権をもつジョブステップの使用している領域を主記憶よりロールアウトして割り付けを試みる。下位の主記憶優先権をもつジョブステップが複数ある場合には、その中で主記憶優先権の最も低いジョブステップより順に 1 つづつロールアウトして割り付けを試み、割り付けができた時点でロールアウト処理は終了する。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できない場合は、そのジョブステップをロールアウトする。割り付けていた場合は、ロールインアルゴリズムによる処理を行う。

ロールインアルゴリズムによる処理は、ロールアウトアルゴリズムによる処理終了後、又はジョブステップの主記憶解放後に行われる。まず、ロールアウトされているジョブステップの主記憶優先権が最も上位のものから順次割り付けを行い、割り付けができなくなった時点で終了する。

(4) CALT の機能

このルーチンへは、ジョブステップ開設 (E^8) およびジョブステップ終了 (E^{10}) の発生によりメインルーチンから呼ばれる。このルーチンでは、以下に示す事象処理の他に各事象処理に応じて、主記憶に関する統計データの集収を行う。事象処理においてシミュレーション初期設定時に定められた次の値を用いる。ただし、この値は領域の分割数・大きさ等によりその都度計算される。

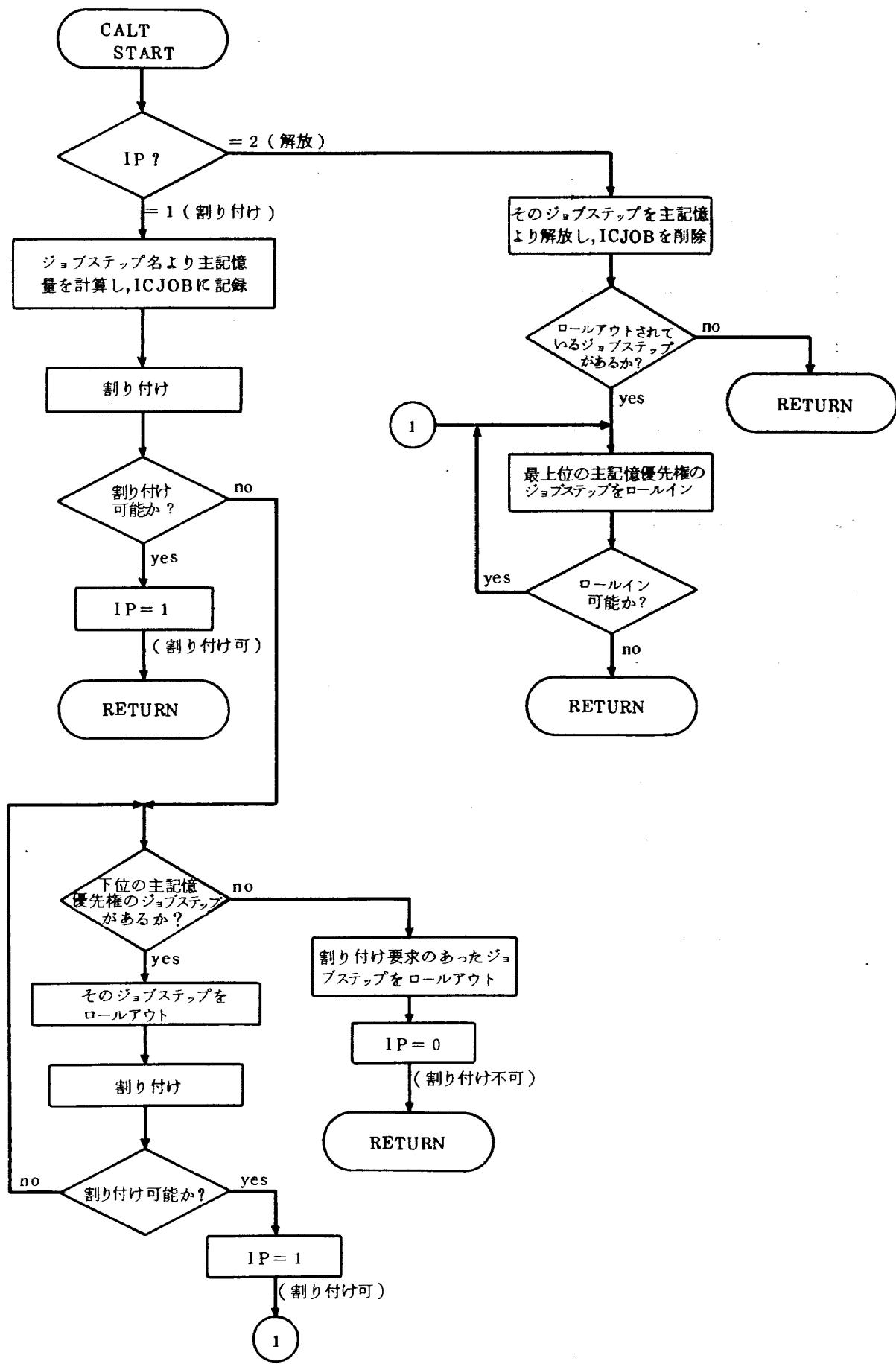


図 4.5.2 CALT フローチャート

表 4.5.2

MJOB

1 事象発生状況
2 ロールインジョブステップ数
3 ロールアウトジョブステップ数

MJOBN (N) N ; 発生事象数に対応

事象発生のジョブ名
発生時間間隔
発生事象名

 C_5 ; ロールイン又はロールアウトに要する時間。 C_6 ; 割り付け処理に要する時間。 C_7 ; 解放に要する時間。i) ジョブステップ開設 (E^9) の場合

ジョブステップ開設の場合は、そのジョブステップの要求する領域の割り付けを主記憶割り付けアルゴリズムに従って試み、割りつけできない場合はさらにロールイン／ロールアウトアルゴリズムに従って割り付けを試みる。その際、要求されている領域の数や大きさ、主記憶優先権は、表 I JOBC を参照する。そして割り付けが可能であるか否かに従って、 $IP = 1$ (割り付け可), $IP = 0$ (割り付け不可) の応答と共に、処理の際に生じたロールアウト、ロールイン事象があれば、表 MJOB と MJOBN に記入し、メインルーチンへ復帰する。

ii) ジョブステップ終了 (E^{10}) の場合

ジョブステップ終了の場合は、そのジョブステップの領域を全て主記憶から解放し、解放後ロールアウトされているジョブステップがあれば、ロールインアルゴリズムによる処理を行い、その結果ロールインされるジョブステップがあれば、表 MJOB と MJOBN に記入し、メインルーチンへ復帰する。

図 4.5.2 に CALT のフローチャートを示す。

4.6 タスクスケジューリング制御ルーチン

タスク関係のサブルーチンは他と異なり 2 重構造になっている。タスク関係の事象はミリ秒単位で発生し、一方、他の事象は分、秒単位で発生する。したがって、タスク関係の事象が発生する度にメインルーチンへ復帰していくにはシミュレーション時間がかかりすぎる。そこで、事象 E^{21} が発生し、一旦制御がタスクスケジューリング制御ルーチンに移れば、事象 E^{20} が発生しない限りメインルーチンへは復帰しないこととする。したがって、タスクスケジューリング制御ルーチンは、事象 E^{20} の発生の監視とタスク関係の事象の制御（最も近い将来に発生する事象の発見とその事象に基づくシステム状態の変更

をタスクスケジューリングルーチンに指示する。）を行う。

このルーチンの扱う事象は、 $E^9, E^{11} \sim E^{18}$ であり、事象 E^{21} の発生によりメインルーチンから呼ばれ、事象 E^{20} の発生によってメインルーチンへ復帰する。

4.7 タスクスケジューリングルーチン（サブルーチン TASK）

このルーチンの機能について述べる前に、その機能を理解するために必要な事項の説明を行う。

(1) 主記憶割り付けからジョブステップ終了までのタスクの動き。

説明に際して便宜上タスクを次の 3 種類に分類する。

S_1 ; AP 又は CP のサービスタスク。ただしタイムスライス切れで再び AP または CP サービスを受ける場合は S'_1 とする。

S_2 ; 入出力のための CP タスク。

S_3 ; チャネルサービスタスク。

また次の様に変数を定義する。

ST_1 ; ジョブステップの AP 又は CP 使用時間の残り時間。

ST_2 ; タイムスライスの長さ。（タイムスライスはジョブステップ開設時とタイムスライスを使い切った時にその最大値が与えられる。）

ST_3 ; 発生された AP 又は CP サービス時間。タイムスライス切れの場合はその未処理時間が次の AP 又は CP サービス時間となる。

ST ; AP 又は CP サービスの長さ。

I ; AP 又は CP サービス終了原因

$I = 1$ の時、ジョブステップ終了

$I = 2$ の時、タイムスライス切れ

$I = 3$ の時、入出力処理

SCP ; 入出力処理のための CP サービス時間。

SCH ; そのジョブステップの 1 回当たりの平均チャネル使用時間。

なお、上記の ST と I は次の様に計算される。

$$ST = MIN(ST_i) \quad i=1, 2, 3$$

$$I = (i; ST_i = ST)$$

これらの定義に基づいて、主記憶割り付け (E^9) からジョブステップ終了 (E^{10}) までのタスクの動きを図 4.7.1 に示す。ただし、実際のシミュレーションではこれに加えてロールイン、ロールアウトの事象がある。

まず、主記憶を割り付けられたタスク（状態 S_1 ）は、AP タスクか CP タスクかの決定（本節(2)参照）に従い AP 又は CP の実行待ち行列へつながれ（本節(3)参照）実行を待つ。このタスクのサービス開始の事象の発

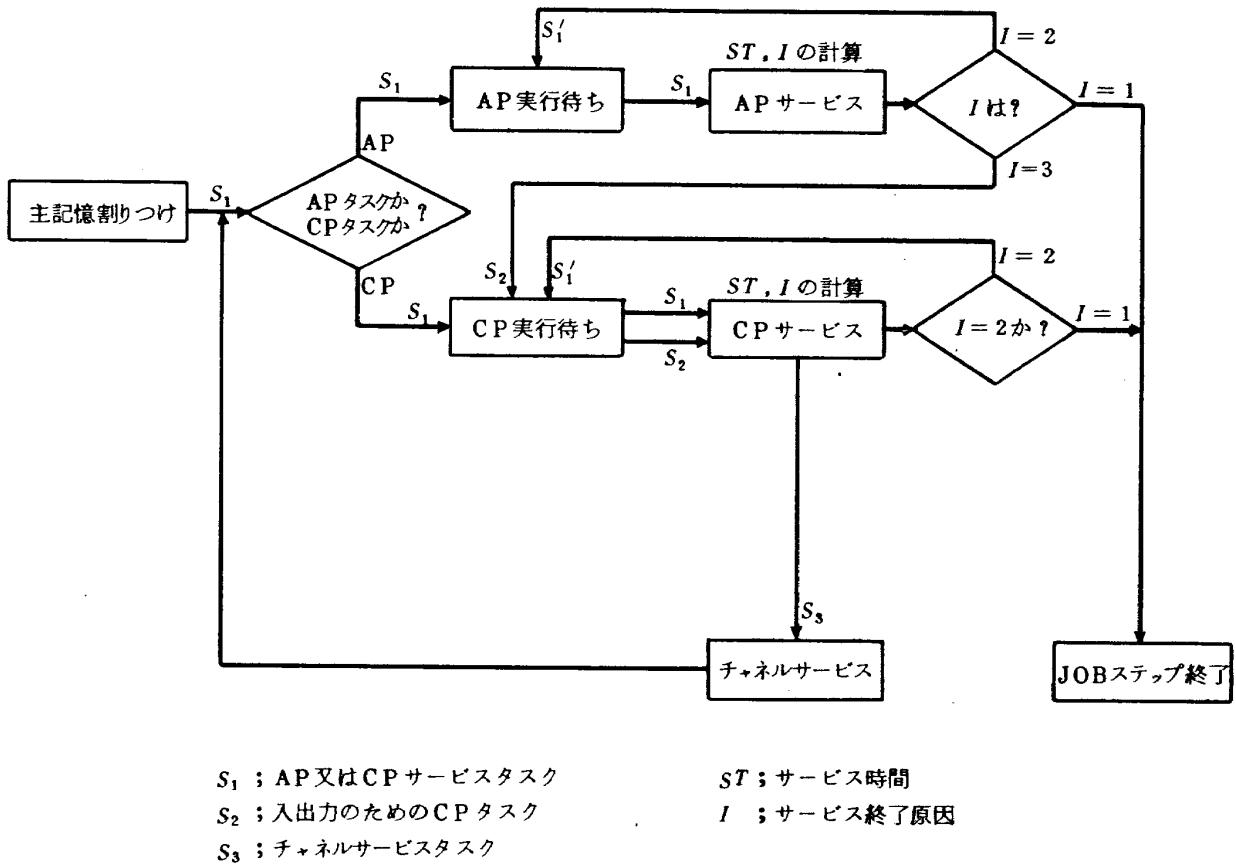


図 4.7.1 タスクの動き

生により、 ST_3 が発生される。 ST_3 はそのジョブステップの平均APサービス時間又は平均CPサービス時間を平均値とする、指数乱数によって求まる。ここでサービス時間 ST とサービス終了原因 I が前に述べた方法で計算され、 ST 時間経過後、 I の値によって次の様に処理される。

ST_3 のサービスがタイムスライス切れのため中断されると($I=2$ の場合)，再びAP又はCPの実行待ち状態(S'_1)となり、 ST_3 のサービスが全て終了するまでくり返しAP又はCPサービスを受ける。 ST_3 のサービスが終了すると($I=3$ の場合)，APタスクの場合には入出力処理のためのCPタスク(S_2)が発生し、CP実行待ちとなり、このタスクのCPサービス開始の事象の発生により SCP 時間のサービスを受ける。そして、このサービスが終了すると、チャネルサービスタスク(S_3)となり、ただちにチャネルサービスが開始される。CPタスクの場合は、 ST_3 のサービスが終了すると入出力処理のためのCPタスクは発生せず、チャネルサービスタスク(S_3)となり、ただちにチャネルサービスが開始される。チャネルサービスが終了すると再び状態は S_1 となり、APタスクかCPタスクかに従って実行待ち行列に

つながれる。この様にして S_1, S_2, S_3 のサイクルは、ジョブステップのAP又はCPサービス時間が全てサービスされるまでくり返され、 $I=1$ となったときジョブステップは終了する。

(2) AP タスクと CP タスクの決定

AP タスクが発生するのは、RUN ジョブステップだけであり、RUN ジョブステップ以外のジョブステップは CP タスクを発生する。しかし、AP RUN ジョブステップでの CP 使用を許す運用、CP RUN ジョブステップでの AP 使用を許す運用等が考えられる。AP と CP の混合使用を許さない運用では、AP RUN ジョブステップは全て AP タスクを、CP RUN ジョブステップは全て CP タスクを発生することになる。また、ある割合で混合使用を許す場合は、許された割合に従ってどちらのタスクであるかを決定する。その決定は、シミュレーションにおいては一般には乱数を用いて行うが、シミュレーションの経済性の観点からあらかじめ定めた一定周期毎に交互に使用する様に定めることもできる。

(3) 実行待ち待ち行列へのつなぎ方

タスクを実行待ち待ち行列へつなぐ場合には、そのタスクの実行優先権を調べ、その優先権毎の待ち行列の最

後につなぐ。ただし、ジョブステップ開設後はじめてAP又はCPサービスを受ける場合や、タイムスライスを使い切らずに入出力サービスを受けた場合は、その待ち行列の先頭につなぐ。

(4) 事象に応じた処理手順

以下にタスクスケジューリングルーチンで処理される事象とその処理手順を示す。

i) 主記憶の割り付け(E^9)、ロールイン(E^{11})、チャネルサービス終了(E^{18})の場合

これらの事象の場合は、まずAPタスクかCPタスクかに従って実行待ち待ち行列につなぎ、つないですぐ実行可であればすぐにAPサービス開始(E^{13})又はCPサービス開始(E^{14})を発生する。

ii) ロールアウト(E^{12})の場合

ロールアウトの場合は、ロールアウトするタスクの実行優先権と待ちの種類(APかCPか)を調べ、その優先権のAP又はCP実行待ち待ち行列からはずし、ロールインに備え、待ちの種類を記録しておく。

iii) APサービス開始(E^{13})の場合

APサービス開始の場合は、APの実行優先権の高い待ち行列から順次タスクをさがし、最初に見つけたタスクを実行する。もし、APサービス終了時に待ち行列にタスクが無い場合はこの事象は発生しないこととするため、この事象の発生の場合は常に実行すべきタスクが存在することになる。次にタスクの種類を調べ、 S_1 であれば ST_3 を発生し、 S'_1 であれば未処理の ST_3 を ST_3 とし、 ST と I とを計算する。そして ST 時間後にAPサービス終了(E^{16})を発生する。

iv) CPサービス開始(E^{14})の場合

CPサービス開始の場合は、APサービス開始の場合と同様に、CPの実行優先権の高い待ち行列から順次タスクをさがし、最初に見つけたタスクを実行する。次にタスクの種類を調べ、 S_1 であれば ST_3 を発生し、 S'_1 であれば未処理の ST_3 を ST_3 として、 ST と I とを計算する。ただし、 $I=3$ の場合は、連続して入出力のためのCPタスクが実行されるので、入出力処理に要する時間(SCP)を ST に加算する。従ってこのタスクは、CP処理終了後、 S_3 となる。又 S_2 の場合は、 $ST=SCP$ とする。そしてどの場合も、 ST 時間後にCPサービス終了(E^{17})を発生する。

v) チャネルサービス開始(E^{15})の場合

チャネルサービス開始の場合は、 $ST=SCH$ とし ST 時間後にチャネルサービス終了(E^{18})を発生する。

vi) APサービス終了(E^{16})の場合

APサービス終了の場合は、APサービス開始の際計

算した I に従い処理を行う。 $I=1$ であれば、すぐにジョブステップ終了(E^{19})を発生し、 $I=2$ であれば、 S'_1 になるため、再びAP実行待ち待ち行列につなぐ。 $I=3$ であれば、 S_2 になるため、CP実行待ち待ち行列につなぐ。この場合、すぐ実行可であれば、CPサービス開始(E^{14})を発生する。そして今終了したタスクのかわりにAPを使用するタスクがあれば、APサービス開始(E^{13})を発生し、APを使用するタスクが無い場合には、空状態となりAPサービス開始事象は発生しない。

vii) CPサービス終了(E^{17})の場合

CPサービス終了の場合は、CPサービス開始の際計算した I に従い処理を行う。 $I=1$ であれば、すぐにジョブステップ終了(E^{19})を発生し、 $I=2$ であれば、 S'_1 になるためCP実行待ち待ち行列につなぐ。又 S_3 の場合はすぐにチャネルサービス開始(E^{15})を発生する。そして今終了したタスクのかわりにCPを使用するタスクがあれば、CPサービス開始(E^{14})を発生し、CPを使用するタスクが無い場合は、空状態となりCPサービス開始事象は発生しない。

図4.7.2にTASKのフローチャートを示す。

4.8 出力ルーチン(サブルーチンLPOUT)

このルーチンは、結果出力(E^1)の発生の際呼ばれる。結果出力(E^1)は、シミュレーションの実行におけるシステム状態の遷移を調べるために等間隔で20回発生し(この回数は可変である)、発生の度に各サブルーチンで集収した統計データ集収表IABJ, TABJ等をもとに、システムに関する情報、ジョブクラスに関する情報、ジョブステップに関する情報を集収する。このルーチンで使用するデータの多くは、いつでも平均値と標準偏差が計算できる様に、和と2乗和の形で累計されている。そして20回目の事象発生の際は、上記の処理を行うと同時に、今まで集収した情報をまとめて、結果出力をを行う。

JPSの結果出力は以下の項目について行う。

i) システム稼動状況に関するもの

- (1) CP稼動時間、および稼動率
- (2) CP空き時間
- (3) CPモニタオーバヘッド時間
- (4) AP稼動時間、および稼動率
- (5) AP空き時間
- (6) チャネル使用時間
- (7) 主記憶利用率
- (8) 平均ロールアウトジョブ個数
- (9) 主記憶空き領域の個数の平均値
- (10) 主記憶空き領域の大きさの平均値
- (11) 主記憶空き領域の標準偏差

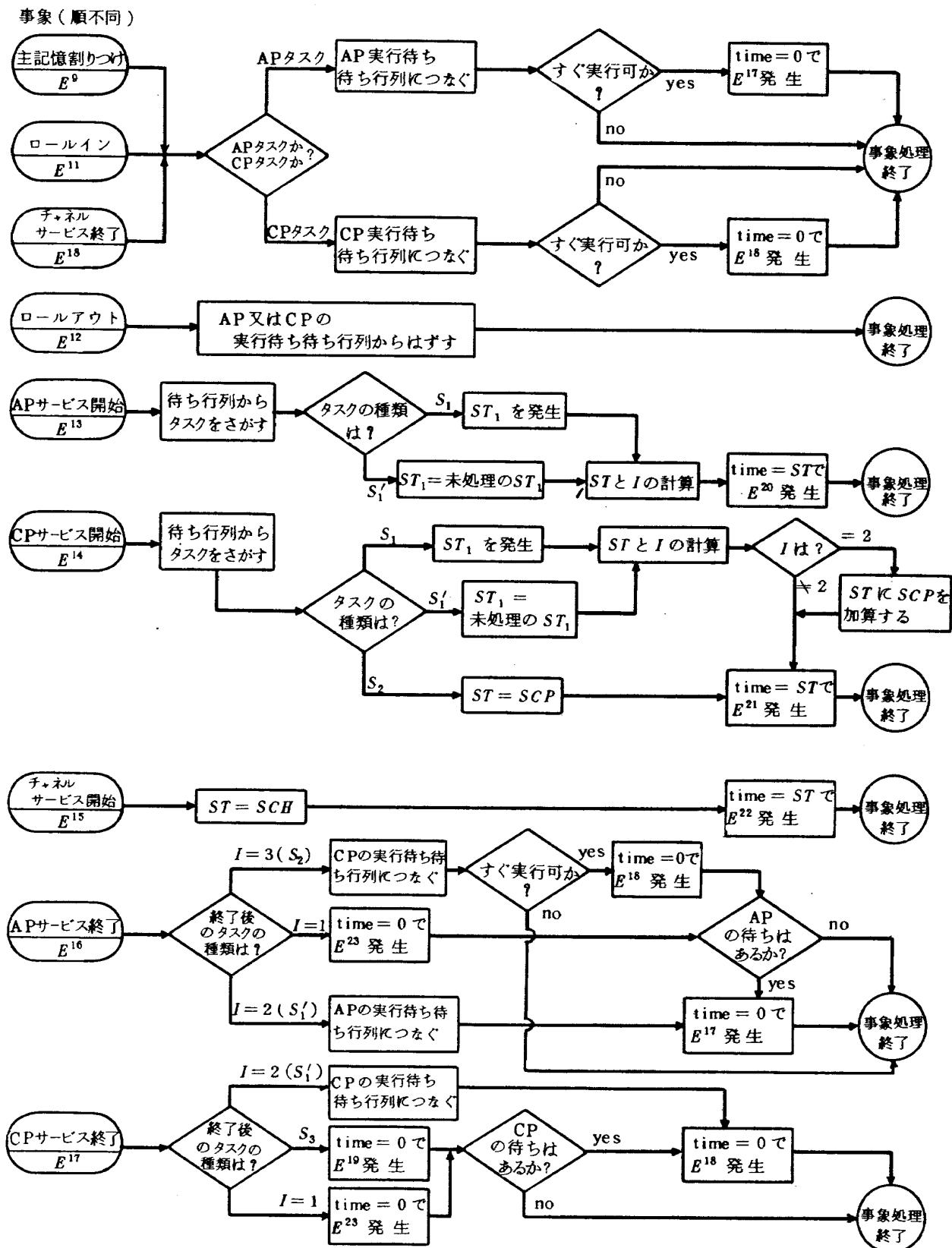


図 4.7.2 TASK フローチャート

- ⑩ ロールアウト, およびロールイン回数
 ⑪ ジョブクラスに関するもの(ジョブクラス別)
 (1) 平均待ち行列の長さ
 (2) 处理件数
 (3) 経過時間(ジョブ開設から閉設までの時間)の平均値と標準偏差
 (4) ターンアラウンドタイム(ジョブ投入より閉設までの時間)の平均値と標準偏差
 (5) 主記憶時間の平均値と標準偏差
 (6) ロールアウト時間の平均値と標準偏差
 (7) ロールアウト回数の平均値と標準偏差
 (8) AP, CP, およびチャネル使用時間の平均値と標準偏差
 (9) JOB時間(AP, CP, チャネル使用時間の和)の平均値と標準偏差
 (10) 主記憶時間/JOB時間の平均値と標準偏差
 (11) 経過時間/JOB時間の平均値と標準偏差
 (12) ターンアラウンドタイム/JOB時間の平均値と標準偏差

iii) ジョブステップに関するもの(ジョブステッププログラム毎、プログラム名は例えばAPFORTRAN, LIED, AP-RUN, CP-RUN等をさす。)

ジョブステップに関しては、ジョブステッププログラム名毎に ii) の(1), (4), および(12)を除く各項目について、出力する。

4.9まとめ

以上が我々の JPS の概要であるが、初めに説明した様に、JPS の各サブルーチンは、OS のモジュールに対応して作成されているため、OS の各種モジュールの改良、スケジューリングアルゴリズム変更のシミュレーションによる検討のためには、それに対応するサブルーチンの部分を変更するだけで良い様になっている。

メインルーチンとサブルーチンの受け渡しは、事象の受け渡しにより行っていることと、その事象の種類と一般性は十分大きく取ってあることにより、JPS の改良と付加もまた、容易に行い得る様になっている。

システムに対するジョブ負荷の変更、運用系の変更は、ジョブ発生ルーチンの一般性から全く容易に行い得ることは明らかであろう。また、シミュレーション結果の検討のための出力情報は、JPS の目的に合わせて多くの項目をとってあるので十分であろう。

最後に JPS の時間単位であるが、これは 1 ミリ秒とした。これはシミュレーションにおける事象の発生間隔から考えるならば、十分な精度であるといえる。

5. シミュレーションの実行

本章では、前章に述べた JPS プログラムが正しく動作するか否かを検査するために行った AP システムのシミュレーション結果について簡単に報告する。

5.1 ジョブ集団と運用系の決定

シミュレーションを行うためには、まず AP システムの運用系—ジョブクラス系とその多重度と各種優先権—to を決定しなければならない。AP システムに対して、どの様な運用系を採用するかは現在定まっていないので、とりあえずジョブクラス構成は現在の、2-CPU システムに対するジョブクラス構成を延長して使用することとし、各ジョブクラス毎の多重度および優先権等は簡単化されたモデルを用いた予備的なシミュレーション実験^{注1)}^{注2)}

表 5.1.1(a) ジョブクラスとその制限値

ジョブクラス	主記憶制限値	処理装置(AP または CP)使用時間制限値	ジョブクラス多重度
S H	60 K語	50 秒	1
D T	80 K語	180 秒	1
C M	150 K語	1200 秒	1
A M	150 K語	1200 秒	2
A L	300 K語	3600 秒	1

注) K語 = 1024 語

表 5.1.1(b) ジョブステップ別優先権

ジョブステップ名	優先権	主記憶優先権	実行優先権
SH-RUN ジョブステップ	9	9	9
DT-RUN ジョブステップ	8	9	9
CM-RUN ジョブステップ	7	8	8
AM-RUN ジョブステップ	10	9	9
AL-RUN ジョブステップ	6	10	10
APFORTRAN, CPFORTRAN, LIED LIBE, UTILITY ジョブステップ	10	10	10

注 1) AP システムに対する運用系は、JPS による実験に基づいて定める予定である。

注 2) システムから AP だけをぬきだした簡単なシステムモデルについてのシミュレーション実験を、本実験の前に行った。

に基づいて定めた。

運用系を表 5.1.1(a), (b) に示す。表 5.1.1(a) のジョブクラス中、AM および AL は AP で RUN を行うジョブクラスである。表 5.1.1(b) の優先権はジョブステップに対して与えられているが、これは AP システムの非対称性を考慮している。ジョブステップにはこれ以外のものもあるが、他は航技研において無視できる程少ないので省いた。

表 5.1.2 ジョブクラス構成比率

ジョブクラス	構成比率 (%)
S H	63.0
D T	4.2
C M	5.9
A M	23.6
A L	3.3

た。

次にこの運用系に対するジョブクラス構成比率、ジョブステップ列構成比率を表 5.1.2, 5.1.3(a), (b) に示す。各ジョブステップにおけるシステム資源の要求量は表 5.1.4 および 5.1.5 に示す。表 5.1.4 において各ジョブステップ名毎の確率分布量の計算は、1 つのジョブの 1 ジョブステップ毎に共通である。また表 5.1.5 において主記憶使用量と AP および CP 使用時間 ACPT は独立に計算する。表 5.1.5 において 1 ジョブ当たりのファイルアクセス回数 FLAC と平均 CP サービス時間 MCPS が記載されていないが、これは RUN ジョブステップのアカウントデータの回帰分析から 1 秒当たりのファイルアクセス回数 FLCP が大記憶に関しては

$$FLCP = 10^{\{-0.464 \times \log_{10} ACPT + 1.9093\}}$$

磁気テープに関しては

$$FLCP = 10^{\{-0.887 \times \log_{10} ACPT + 3.6695\}}$$

表 5.1.3(a) SH ジョブクラスのジョブステップ列構成比率

ジョブステップ別 No.	AP 又は CP FORTRAN	LIED	LIBE	UTILITY	RUN	総ジョブステップ数	構成比率 (%)
1	1					1	19.52
2			1			1	5.41
3			2			2	4.97
4			2	2		4	3.91
5	1	1				2	2.59
6	1			2		3	1.39
7	1	1			1	3	54.76
8					1	1	4.55
9	1	1	2		1	5	2.05
10	1	1		2	1	5	0.99

表 5.1.3(b) DT, AM, AL, CM ジョブクラスのジョブステップ列構成比率

ジョブステップ別 No.	AP 又は CP FORTRAN	LIED	LIBE	UTILITY	RUN	総ジョブステップ数	構成比率 (%)
7	1	1			1	3	87.8
8					1	1	7.3
9	1	1	2		1	5	3.3
10	1	1		2	1	5	1.6

表 5.1.4 ジョブステップ毎システム資源の要求量

ジョブステップ名	CPFORTRAN	APFORTRAN	LIED	LIBE	UTILITY
主記憶使用量 (CORE) 単位; K語	$50 + (5 + \frac{M}{100})$ Mは平均 451.1 の 指數分布に従う	$68 + (8 + \frac{M}{15})$ Mは平均 451.1 の 指數分布に従う	48(固定)	32(固定)	26(固定)
ファイルアクセス 回数 (FLAC) 単位; 回	平均 136.3 の指數分布に従う	平均 136.3 の指數分布に従う	平均 440.1 標準偏差 178.9 の正規分布に従う	平均 29.2 の指數分布に従う	109(固定)
CP 使用時間 (CPT) 単位; ミリ秒	平均 3825.7 の指數分布に従う	平均 7651.4 の指數分布に従う	平均 1434.8 標準偏差 701.5 の正規分布に従う	平均 323.8 の指數分布に従う	944(固定)
平均 CP サービス 時間 (MCPS) 単位; ミリ秒	CPT/FLAC	CPT/FLAC	CPT/FLAC	CPT/FLAC	CPT/FLAC
ファイルアクセス 1回当たり チャネルサービス 時間 単位; ミリ秒	30	30	30	300	300

表 5.1.5 各ジョブクラス RUN ジョブスラップのシステム資源要求量

ジョブステップ名 項目	SH-RUN	DT-RUN	CM-RUN	AM-RUN	AL-RUN
主記憶使用量 (CORE) 単位; K語	$10 + S$ Sは6~60の一様 分布	$10 + S$ Sは6~80の一様 分布	$10 + S$ Sは25~150の一 様分布	$10 + S$ Sは25~150の一 様分布	$10 + S$ Sは25~300の一 様分布
AP+CP 使用時 間 (ACPT) 単位; 秒	0~50 の一様 分布に従う	0~180 の一様 分布に従う	50~1200 の一様 分布に従う	50~1200 の一様 分布に従う	50~3600 の一様 分布に従う
ファイルアクセス 1回当たりのチャネ ルサービス時間 単位; ミリ秒	30	30	30	30	30

表 5.2.1 ジョブクラス構成比率比較

項目	ジョブクラス名	SH	DT	CM	AM	AL
構成比率 理論値 (%)	63.0	4.2	5.9	23.6	3.3	
構成比率 発生値 (%)	61.6	3.8	6.4	25.1	3.1	
発生件数(件)	762	47	79	311	38	

と回帰式が求まっているので、ACPTが求まれば

$$FLAC = FLCP \times ACPT$$

$$MCPS = 1 / FLCP$$

と計算できることによる。

以上により、今回のシミュレーション実験においてシステムに投入されるジョブ集団が定まる。このジョブ集団からシステムへのジョブの投入間隔は混雑時を想定して平均90秒の指指数分布とした。このジョブ集団を規定するデータは2-CPUシステム運用時のアカウントデータの分析から推定して得たものであり、特にAPに関するデータは論理的なもの(APFORTRANに関するもの。)と、全く推定によったもの(AMおよびALジョブクラスに関するもの。)との混合である。

最後に、ジョブおよびジョブステップの開設開始から終了までの所要時間と、閉設開始から終了までの所要時間をそれぞれ次の様に定めた。

$$C_1 = 133 \text{ ミリ秒}$$

$$C_2 = 727 \text{ ミリ秒}$$

$$C_3 = 1734 \text{ ミリ秒}$$

$$C_4 = 812 \text{ ミリ秒}$$

またジョブステップのロールイン・ロールアウトに要する時間、主記憶割り付けに要する時間、および主記憶解放に要する時間も次の様に定めた。

$$C_5 = 3.75 \times X + 10 \times Y \text{ ミリ秒}$$

$$C_6 = 2.44 \times Z + 26.6 \text{ ミリ秒}$$

$$C_7 = 500 \text{ ミリ秒}$$

ここで、X、Y、およびZの値は下記のとおりである。

X ; 転送領域の大きさ(K語)

Y ; 領域分割個数(個)

Z ; プログラム領域の大きさ(K語)

これらは全てOSの走行ステップ数とソフトウェアモニタによる計測値から推定したものである。

5.2 シミュレーション結果の検討

シミュレーションにおいて、その打ち切り時間の決定は重要である。これが短かすぎるとシミュレーションは十分な定常状態に達しないし、長すぎると不経済である。そこで種々の性格のジョブ集団と運用系に対して、シミ

ュレーションの時間的な特性を検討したが、その結果、今回のJPSによるシミュレーションが十分な定常状態に達するためには実際の運用時間に換算して20時間のシミュレーションを行えば十分であるとの結論に達した。

まず、前節でシミュレーションに対して与えたジョブ母集団と、シミュレーション中に発生したジョブ集団(母集団からのサンプルと考えられる。)の比較を行う。

表5.2.1および5.2.2はそれぞれジョブクラス構成比率とジョブステップ列構成比率を示す。理論値(母集団値)と発生値は表からかなり良い一致を示していることがわかる。表5.2.3および5.2.4は、各ジョブステップ毎のシステム資源利用量に関するものであるが、これも良い一致を示している。ファイルアクセスに関するデータは示さなかったが、いずれもこの表に記載されたデータと関数関係にあるので、同程度の一致を示すであろう。

図5.2.1は20時間を4時間毎に区切り、その時間間隔の間に処理が終了したジョブの件数をジョブクラス毎にプロットしたものである。図は明らかに定常に達していることを示す。この結果における総処理件数および各ジョブクラス毎の処理件数は、シミュレーションに対して

表 5.2.2 ジョブステップ列構成比率比較

項目 ジョブ ステップ列No	構成比率 理論 値 (%)	発生件数 (件)	構成比率 発生 値 (%)
1	12.40	133	10.75
2	3.44	43	3.43
3	3.03	37	2.99
4	2.48	28	2.26
5	1.64	34	2.74
6	0.88	14	1.13
7	66.99	811	65.56
8	5.56	78	6.30
9	2.51	45	3.63
10	1.22	14	1.13

表 5.2.3 各ジョブスラップ毎のシステム資源要求量比較

項目名 ジョブステップ名	発生件数 (件)	平均主記憶使用量(CORE) 単位; K語		平均処理装置使用時間 (CPT) 単位; ミリ秒	
		理論値	発生値	理論値	発生値
CPFORTRAN	620	60	60	3825.7	4007.10
APFORTRAN	431	107	106	7651.4	7407.98
LI ED	904	48 *	48 *	1434.8	1414.27
LIBE	217	32 *	32 *	323.8	314.91
UTILITY	158	26 *	26 *	944 *	944 *

* は一定値で与えるので当然一致する。

表 5.2.4 RUN ジョブステップシステム資源要求量比較

項目名 ジョブステップ名	発生件数 (件)	平均主記憶使用量(CORE) 単位; K語		平均処理装置使用時間 (ACPT) 単位; ミリ秒	
		理論値	発生値	理論値	発生値
SH-RUN	473	33	32	25	20
DT-RUN	47	43	46	90	92
CM-RUN	79	88	97	625	625
AM-RUN	311	88	84	625	580
AL-RUN	38	163	181	1225	1326

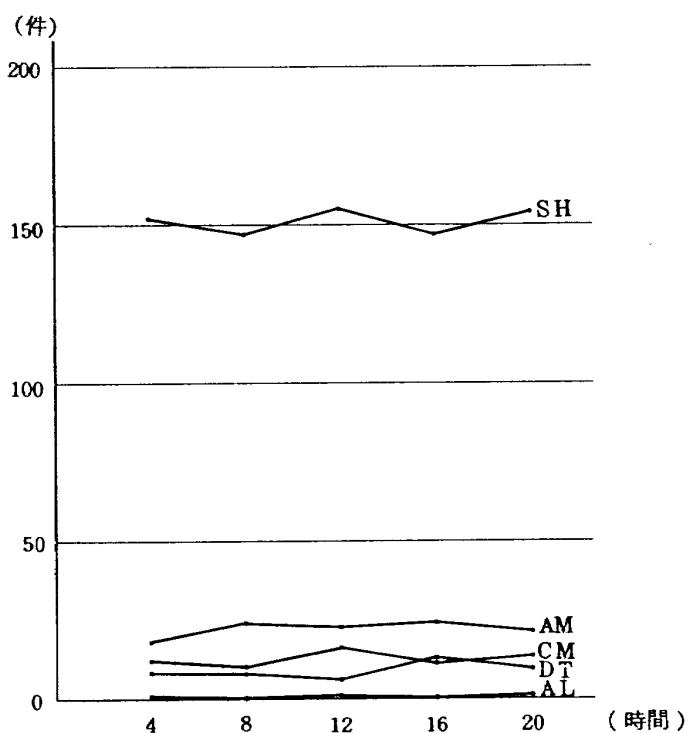


図 5.2.1 ジョブクラス別処理件数(4時間毎)

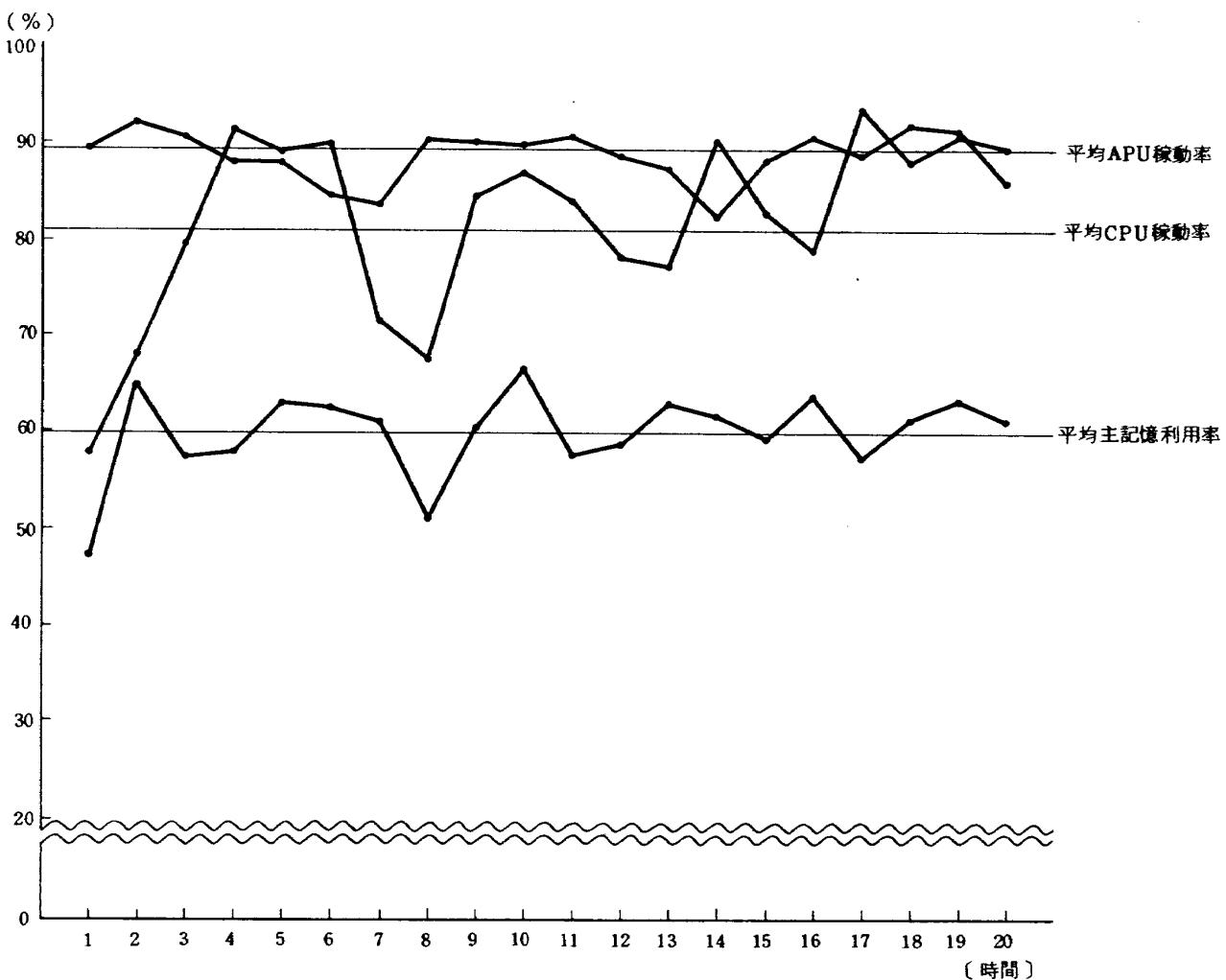


図 5.2.2 AP・CP 稼動率および主記憶利用率 (1 時間毎)

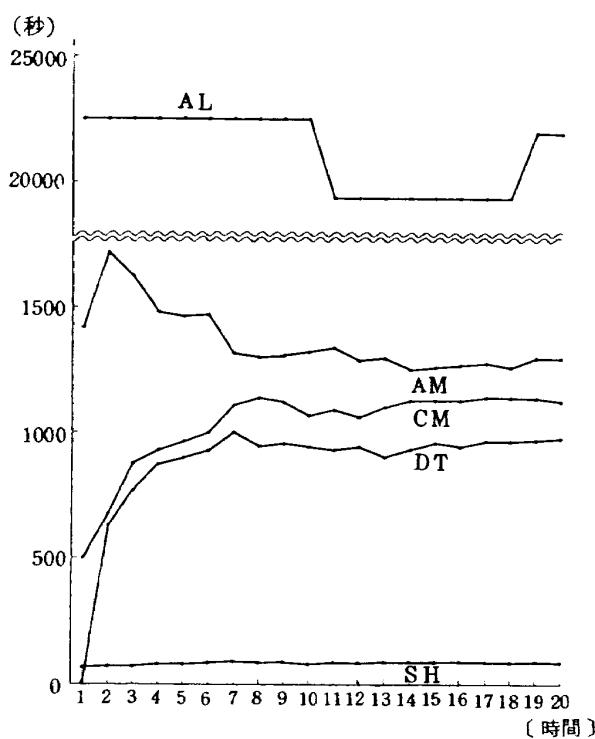


図 5.2.3 ジョブクラス別経過時間の推移

与えたジョブ集団のジョブ構成比率を考慮すれば、AL クラスの処理件数を除いては妥当なものといえる。

AL クラスの処理件数に関しては、与えられたジョブ集団が AP および主記憶に関してシステムの能力を上回っていることと、AL クラスの主記憶優先権が低いことに原因があり、これはこの運用系のもつ問題点を指摘すると同時に、後に示す様に主記憶管理アルゴリズムにも大きな欠点があることを示している。

図 5.2.2 にシステムの AP, CP, および主記憶の 1 時間毎の利用率を示し、図 5.2.3 に各ジョブクラス毎の処理されたジョブの平均経過時間の推移を示す。

図は両方ともシミュレーションがこれらの変数に関して十分な定常状態に達していることを示している。

この結果において注目すべきことは、

(1) CP の稼動率の振動の幅が AP のそれに比較して大きいこと。

(2) AP の利用率の増大減少の傾向と CP のそれとの

間に相反関係が見られること。

- (3) 主記憶の利用率が小さいこと。
 - (4) AL ジョブクラスの経過時間に大きな変動が見られること。
- である。

(1)において AP 稼動率が安定している理由として、まず AL ジョブクラスのジョブは主記憶要求量が大きく、かつ主記憶優先権が低いためロールアウトされやすいためかわらず、ジョブクラス多重度が 2 である AM ジョブクラスのジョブの主記憶優先権が高いため、主記憶上に AP プログラムがほぼ 2 重複確保されていること、次に AM ジョブクラスの AP サービス時間とチャネルサービス時間の期対値の比が 5.7 : 1 であるために、主記憶上に AM ジョブクラスのジョブが 1 本しかない場合でも 80 % 程度の AP 稼動率は保たれると考えられることが挙げられる。その反面、SH, DT, および CM の場合の CP, サービス時間とチャネルサービス時間の期対値の比はそれぞれ 1.8 : 1, 0.3 : 1, および 8.1 : 1 であり、そのうち最も CP を使用する割り合いの高い CM が AL についてロールアウトされやすいため、CM のロールアウトに伴い CP 稼動率は大きく振動することとなる。したがって、AP, CP 稼動率の示した値は、シミュレーションの妥当性を間接的に示唆していると言える。

(2) はシステムの非対称性に原因があり、これもシミュレーションの結果の妥当性を間接的に示唆しているが、この現象ができるだけ押えることが運用系の設定にあって最も重要なことである。

(3) は現在の 2-CPU システムにおけるソフトウェアモニタリングによる計測値と同程度の数値であって、これも統計的にシミュレーションの結果の正しさを裏書きしているが、これは OS の主記憶割り付けアルゴリズムを改良する必要があることを示している。

(4) は処理件数の所で述べたことが原因である。

以上でシミュレーション結果の検討は終るが、最後に 20 時間のシミュレーションに要した時間は 753.6 秒であり、これは実時間のおよそ 96 倍の実行速度であって、シミュレーション言語にフォートランを選んだ利得は十分にあったことを付け加えておく。

6. 結 言

本資料は AP システムの動作をシミュレートするための JPS プログラムの機能と概要について説明したものである。

この JPS は、JPS に対するシステムモデルの説明により明らかな様に、AP システムのバッチ処理の様相を

かなり詳細に実現しており、このプログラムが正しく動作していることは第 5 章に示した実験例からも明らかなる。

JPS の作成により、今後計算センタにおいて、AP システムに対する運用系の適・不適を検討すること、およびシステムの問題点を摘出し、その解決策を検討することがかなり容易になった。事実、本資料の実験例でも明らかなる如く、OS の主記憶割り付け法に問題があることが既に指摘された。これはこの問題の解決策の検討にも JPS が有力な道具となることを示唆するものであると考える。

JPS の作成に際して、FACOM 230-MVI/VII についての詳細資料を富士通(株) 計画部瀬田嘉勝氏、ならびに LP 部棚倉由行氏に提示いただいたことを感謝して稿を閉じることにする。

文 献

- 1) 三好、末松; FACOM 230-75 アレイプロセッサーについて I, 航技研資料 TM-325, 1977 年 3 月
- 2) 富士通(株); アレイプロセッサハードウェア解説書
- 3) 末松、中村、石塚、吉田、福田、三好; FACOM 230-75 アレイプロセッサーについて II, 航技研資料 TM-369, 1978 年 10 月
- 4) Infotech state of the Art Report 18 pp 123 ~ 147
- 5) 富士通(株); FACOM 230 M-V/VII 解説書
- 6) Infotech state of the Art Report 18 pp 229 ~ 262
- 7) N. R. Nielsen; An Analysis of Time-Sharing Techniques Communications of the ACM, vol 14, No. 2, pp 79 ~ 90, 1971
- 8) 富士通(株); FACOM 230 M-V/VII GPSS 解説書
- 9) 富士通(株); FACOM 230 M-V/VII SOL 解説書
- 10) D. Teichroew & J. F. Lubin; Computer Simulation - Discussion of the Technique and Comparison of Languages, Communication of the ACM, vol 19, No. 10, pp 723 ~ 741, 1966
- 11) T. H. Naylor, J. L. Balintfy, D. S. Burdick & K. Chu; 水野幸男、小柳芳雄、培風館, 1971

航空宇宙技術研究所資料 384号

昭和 54 年 6 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 藏 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 4 - 1 - 7 (羽 田 ビ ル)

Printed in Japan