

《解 說》

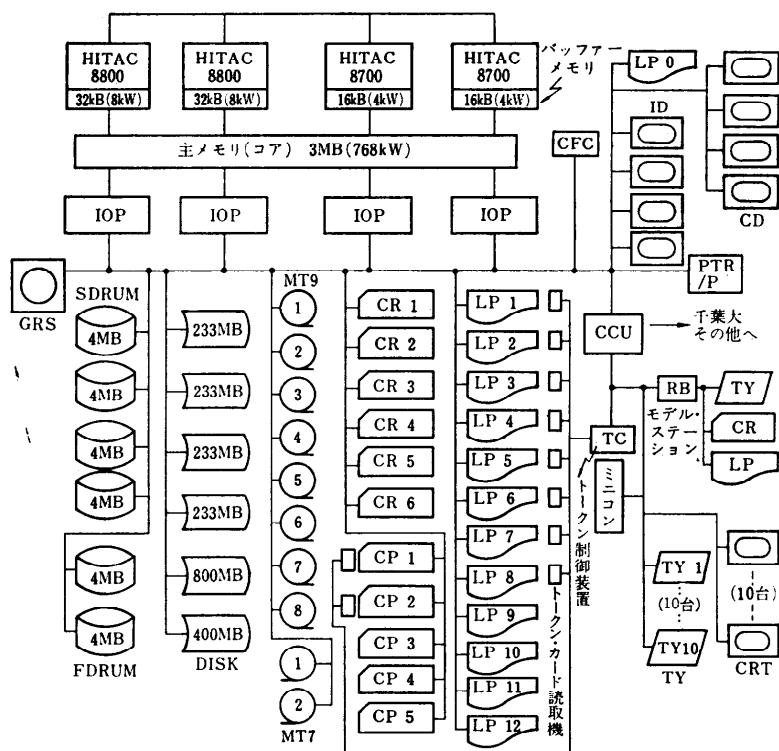
東京大学超大型コンピュータ・システム

石田 晴久*

1. はじめに

全国の大学関係研究者の共同利用施設である、東京大学大型計算機センターでは、1973年1月から、新しいコンピュータ・システムが稼動している。このシス

テムは、図1のように、CPU 4 台、主記憶（コア）3 MB（メガバイト）=768 kW（キロ語）、ページング・ドラム4台（計16 MB）、集団ディスク6台（計2,132 MB）、その他多数の周辺機器からなる、巨大なコンピュータ・コンプレックスである。



SDRUM	スワップ用ドラム	平均アクセス	10 m 秒	CCU	通信制御装置	256 回線
FDRUM	ファイル用ドラム	平均アクセス	10 m 秒	TC	トークン制御装置	200 ビット/秒
DISK	集団ディスク	平均アクセス	30/84 m 秒	RB	リモート・パッチ・ステーション	2,400 ビット/秒
MT 9	9トラック磁気テープ		240 kB/秒	TY	タイプライタ	20 字/秒
MT 7	7トラック磁気テープ		120 kB/秒	CRT	文字ディスプレイ	1,000 字/画面
CR 1~6	カード読取機	1,470 枚/分		GRS	グラフィック・サブシステム	1,024×1,024 点
CP 1~5	カードセン孔機	100 枚/分		PTR/P	紙テープ・リーダー (パンチ付)	1,000 字/秒
LP 1~11	ラインプリンタ	1,200 行/分		CD	コンソール・ディスプレイ (プリンタ付)	1,000 字/画面
LP 0	コンソール・プリンタ	300 行/分		ID	補助コンソール・ディスプレイ	1,000 字/画面
IOP	入出力プロセッサ	8 MB/sec		CFC	構成制御卓	

図 1 東大センター・システムのハードウェア構成

* 東京大学大型計算機センター

これは、単一のシステムとしては、現在、国内最大規模のものである。大学に関する限り、アメリカでも UCLA (カリフォルニア大) の IBM 360/91 に 4 MB の主記憶がある点を除けば、このような超大型機はない。この事情は日本独特の国策による集中投資の結果であり、関係者は超大型機の実現をもちろん喜んでいるが、日本の大学におけるコンピュータ・パワーが、一般にはまだまだ貧しいことは否めない。

ところで、図 1 の東大システムの主力機である HITAC 8800 は、1966 年から 1972 年にわたる通産省の大型プロジェクトで、国費約 100 億円をかけて開発された「超高性能電子計算機」をプロトタイプとし、それをさらに改良した商用機である。この CPU の速度は、IBM 370/168 より速く、FACOM 230/75 と並ぶかなりの高速機であるが、世界最高速の IBM 370/195 や CDC 7600 には、まだ及ばないと評価されている。一方、システム・ソフトウェアの方は、HITAC 8800/8700 のために新規に開発されたこれまた巨大な OS で、OS 7 と呼ばれている。プログラミング言語は FORTRAN, COBOL, PL/I, BASIC が主であり、ALGOL や APL はまだない。

上述のように、大型プロジェクトの流れをくむだけに、東大システムには、現代のコンピュータ技術の粋を集めたといつてよい位に、新しい技術がふんだんに取り入れられている。主なことはだけあげても、パイプライン制御、バッファメモリ、仮想記憶、記憶のリング保護機構などの機密保護機能、非対称多重プロセッシング、ダイナミック・リンク、リエントラント・コーディング、FORTRAN コンパイラの最適化機能、オープン・バッチ処理などがそれである。

図 2 には、東大システムにおける処理形態を示す。この汎用システムでは、4 台の(しかも性能の異なる) CPU による多重プロセッシングが行なわれている。

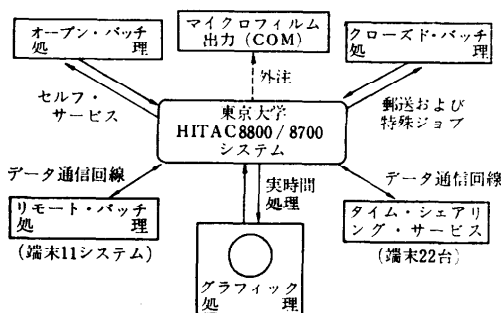


図 2 東大システムにおける処理形態

従来行なわれてきた 2 台の CPU による同時処理に比べると、3 台さらに 4 台の CPU による同時処理となれば、2 台の CPU 間のやりとりの間に第 3 者および第 4 者の CPU が介入するので、こうした 4 CPU による多重プロセッシングでは、それだけ制御が複雑になる。しかし、東大システムであえてこの構成をとっている理由は、次のとおりである。

(1) 多数のユーザ (約 2,200 人) の要求に応じられるよう、スループットを上げてターンアラウンド時間の短縮をはかる。

(2) OS をひとつにし、全主記憶を一体として扱って、メモリを大きく使う。(仮想記憶の採用で、メモリ空間はさらに広がっている。)

(3) ファイル (ディスク) などのリソースを、全処理形態で共通に使う。

(4) オペレーション上および管理上の省力化をはかる。

2. 4 CPU による多重プロセッシングのためのハードウェア機能

東大システムの第 1 の大きな特徴は、4 台の同種・異速度の CPU による多重プロセッシングを行なっていることである。このためハードウェア機構として HITAC 8800/8700 に用意されているのは、CPU 間通信のためのダイレクト・コントロール、バッファ・メモリのキャンセル、主メモリの共有、構成制御などである。つぎに、これらの主なものについて述べる。

(1) ダイレクト・コントロール機構

これは、CPU 間の直接通信を行なうのに使われるもので、CPU 間のインターフェースには、つぎの 3 種がある。

データ転送	Write Direct (WRD) 命令による。
	Read Direct (RDD) 命令による。
構成制御	バッファ・メモリ・キャンセル指令
	Load/Store Multiple Address Registers (LMA, STMA) 命令による。
	Set/Insert Configuration Control Registers (SCFR, ICFR) 命令による。

これらのうち WRD 命令は、実際には後述のように、相手 CPU に伝えるべき情報を (共通の) 主メモリに登録しておいて、相手 CPU に割込みをかけたり、相手をスタートさせたりするのに使われる。また構成制御命令は、CPU・主メモリ・入出力プロセッサ間の接続関係の変更使用される。

(2) バッファ・メモリのキャンセル機能

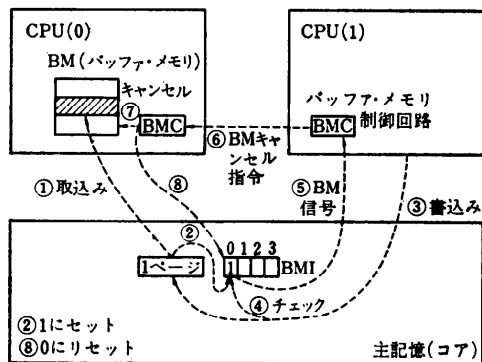


図 3 バッファ・メモリの制御

バッファ・メモリをもつ系では、バッファ・メモリはプログラマからはみえないようにし、命令でメモリへの書き込みが行なわれると、バッファ・メモリへ書き込むと同時に、主記憶への書き込みを行ない、バッファ・メモリと主記憶との内容の一致をとるようにするのが普通である。これを、ストア・スルー (store-through) 方式という。

ところが、この方式の複数 CPU で多重プロセッシングを行なうと、ある CPU (A) が自分のバッファに取り込んでいる主記憶領域に、たまたま他の CPU (B) が書き込みを行なったときに、バッファと主記憶の一致がとれなくなるという事態が生ずる。これを避けるのに、HITAC 8800/8700 では、新しい試みとして、図 3 のように、B から A にバッファ・キャンセル信号を送り、それを受けると、A 側のバッファ・メモリの内容の 1 ブロック (64B) または全体が自動的に消されるようになっている。こうなると、A の方では、改めて主記憶からバッファへ必要な情報の転送をしなければならないので、若干のスピードダウンになるが、こんな事態はめったに起きないので、実用上はほとんど問題にならない。

3. OS 7 の多重プロセッシング機能

すでにみたように、東大システムでは、速度の異なる CPU による同時処理を行なっている。これを非対称多重プロセッシングと呼ぶが、これについて、OS 7 でとくに工夫しているのは、つぎの点である。

(1) 高速の 8800 にはユーザ・タスク (演算実行が主) を、8700 にはシステム・タスク (入出力処理など) を優先的に割り当てる。これは両 CPU の速度差を生かすためで、実際には、タスクの優先度を表す数値に、両 CPU で異なるゲタをはかせて優先度

をきめている。

(2) 8800 を遊ばせないようにするため、8800 が万一アイドルになると、8700 の処理しているタスクを取り上げて 8800 に割り当てる。

こうした CPU 間の連絡を行なう場合、OS 7 では、主記憶が共有であることを利用して、まず呼ぶ側の CPU が主記憶にアクション (割込み処理の指定) を登録して、それから相手の CPU に WRD 命令で割込みをかけると、相手 CPU は主記憶に登録されているアクションを拾うという単純な方法をとっている。8800 が 8700 のタスクを取り上げるやり方を例にとれば、これは、つぎのように行なわれる。

a) 8800 がアイドルになると、タスク・ディスパッチャ (OS 7 の核の部分) により、ラン中の 8700 を探す。ラン中の 8700 があれば、それに WRD 命令を出し、自分はアイドルのまま待つ。

b) 8700 は、割込みを受けると、アクション・スキャナ (OS 7 の核) を経由して、タスク・ディスパッチャで、8800 がアイドルかどうかを調べる。これは、状態が a から b へ移る間に、第 3 の CPU から他のタスクが生まれて、8800 が再びラン状態に入る可能性があるからである。

c) アイドルな 8800 があれば、8700 はその 8800 に WRD 命令を出し、自分はアイドルになる。

d) 8800 が割込みを受け付けると、アクション・スキャナを経由して、タスク・ディスパッチャで、先に 8700 が放棄したタスクを取り上げて実行する。ただし、CPU が 4 台 (一般には 3 台以上) あると、c から d へ移る間に、第 3 者の CPU がそのタスクを取り上げてしまい、アイドルな 8800 のやるべきタスクがなくなってしまうということもありうる。このあたりは、2 台の CPU による多重プロセッシングに比べて、4 台の CPU による多重プロセッシングの方がむずかしい理由のひとつである。

同じように、上記 (1) の制御でも、8800 が 8700 に入出力処理を渡す場合は、WRD 命令が使われる。したがって、一般には、8700 から 8800 へのダイレクト・コントロールよりも、8800 から 8700 へのダイレクト・コントロールの方が多い。

(3) CPU 時間の制御をする目的で、8700 の CPU 時間に一定の比 (東大では 0.25) をかけて 8800 CPU 時間に加える。これを、非対称タイマ制御という。この比は、実際にはジョブの性格に左右されるので、この制御は、単なる便法である。

(4) 緊急を要するジョブは、有資格者が指定すれば最優先で、8800 のみにより処理する。この機能は、処理速度の測定にも使われる。

(5) どれかの CPU が故障した場合、構成制御機能により、自動的にその CPU の切離しを行なう。

こうした多重プロセッシングの問題点としては、全システムの一斉ダウンが、時として生じうるという点がある。また、個々のジョブが、どの種の CPU にどの位ずつ処理されるかが場合により異なるため、CPU 時間の 1 義的な測定がユーザ・レベルではできず、したがって、計算料金も場合によって少し異なるという問題もある。

なお、CPU が 4 台あることから、ユーザ・レベルで、ひとつのジョブを 4 つのタスクに分けて同時並行処理させることも可能である。このマルチタスキングの目的で、FORTRAN から呼べるサブルーチンとして、¥GETECB (ECB は event control block), ¥ATTACH, ¥DETACH, ¥POST, ¥WAIT, ¥ENQ, ¥DEQ, ¥STOP などのサブルーチンが用意されている。

4. 東大システムにおける仮想空間の構造

つぎに、東大システムのメモリ系では、バッファ (cache)・メモリ と仮想記憶とが同時に採用されていて、メモリがバッファ・メモリ、主記憶、ドラム、ディスク*と階層構造をなしている。この結果、バッファ・メモリのおかげで処理速度が向上し、仮想記憶のおかげで、ユーザの使えるメモリ領域が格段に大きくなった。しかし、その反面、実行速度がジョブの性格に大幅に左右され、バラつきが大きく、また後述のように、とくに大型計算の場合、真に効率的なプログラムを作るには、このメモリ・ハイアラキーを意識しなければならないという、新たな問題も生んでいる。

図 4 に、東大システムの仮想空間の構造を示す。HITAC 8800/8700 では、論理アドレスは 31 ビット (2,048 MB 分) あって、仮想空間はベース (アドレス 7 ビット)、セグメント (6 ビット)、ページ (6 ビット)、バイト (12 ビット) に分かれている。とくに、1 ページは $4,096 \text{ B} = 4 \text{ kB} = 1 \text{ kW}$ である。このうち、ベースは、システム空間のいろいろな領域をユーザ空

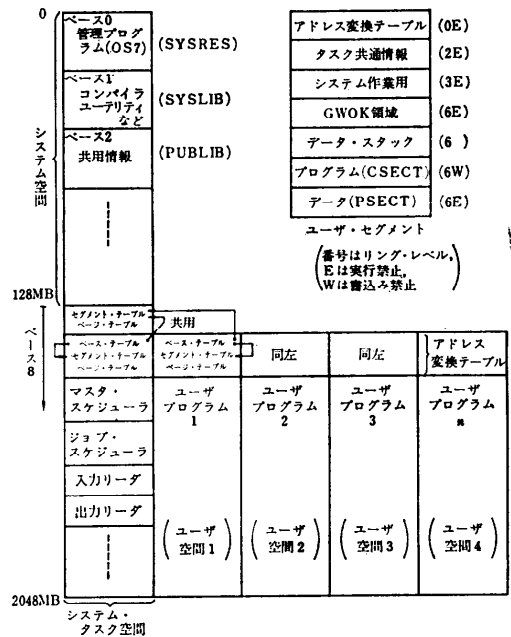


図 4 仮想空間の構造

間と分離し、セグメント・テーブルの共用を可能にすることによって、図 4 のシステム空間を共用できるようにする、という目的で、新たに導入された概念である。

図 4 で重要なのは、ユーザがある番地を指定したときに、その番地はあくまでも自分の仮想空間内の番地でしかないため、他人の領域を侵犯することは絶対にできないということである。しかし、これでは逆に、端末間交信のようなユーザ相互の情報交換ができなくなる。そこで、共用のシステム空間内に PUBLIB という共用情報領域を設けて、それを可能にしている。

図 4 に示したように、仮想空間は、多重プログラミングの多重度を n (東大では 8 ないし 28 程度) とすると、 n 人についてそれぞれ 2,048 MB とたいへん広い。しかし、実際には、仮想空間はとびとびのアドレスのところだけが使われるだけで、現実には、システム空間とシステム・タスク空間および n 個のユーザ空間とをつめた大きさが、「ドラム容量 (16 MB) と主記憶 (3 MB) とを合わせた」実空間の大きさ以下という制限が出てくる。このため、東大の普通のユーザの仮想記憶使用量の上限は、60 kW ないし 160 kW 程度にしている。しかし、とくに指定すれば、1,500 kW 程度のプログラムも実行できる。

* ただし、MIT の MULTICS とは異なり、OS7 では、ディスク・ファイルは仮想空間とは別個の存在で、ユーティリティ・プログラムを通して使うようになっている。

図4の右上に示したように、ユーザ・プログラムは、8番目のベース内の、最低7個の、保護属性の異なるセグメントを使って実行される。この際、ページ(1kW)が後述のメモリ保護の単位となっているため、各セグメントの最小サイズは1kWだから、どんな小さなプログラムでも7kWは使うことになる。このうち、5kWがシステム・オーバーヘッドである。

つぎに、このような仮想空間上に書かれたプログラムは、主記憶の中ではある大きさのコア領域(これを主記憶枠と呼ぶ)の中で実行され、これをはみ出す分は、ページごとに、ドラムとの間で転送されて使われる。このオンデマンド・ページング方式では、主記憶枠は固定ではなく、ダイナミックに変化するが、これは、OS7の大きな特徴である。

5. オープン・バッチ処理方式

オープン・バッチ処理方式は、ジョブのターンアラウンド時間を短くし、センター側の省力化をはかるために、東大センターで考え出した独特の方式である。この方式では、処理結果を、センターの人手で取り出して棚(ビジョン・ボックス)に並べる(クローズド・バッチ方式)代わりに、集団ディスクの中にためておき、ユーザからトークン・カード(IDカード)によるリクエストがあって初めて、ラインプリンタやカードパンチに出力し、前者の場合は、終りに用紙を自動的にカットしている。これを、デマンド出力と呼ぶ。

オープン・バッチ処理に必要なディスク容量は、入力カード枚数が平均500枚として、ジョブ当たり40kB、1,000ジョブ分で入力用に40MB、また平均印字行数1,300行として、ジョブ当たり170kB、1,000ジョブ分で印字出力用に170MBとなる。これはたいへんな容量のようでもあるが、100MBのバックなら、入力ファイルに1バック、出力ファイルに2バックだから、超大型機にとってはたいした量ではない。

このオープン・バッチ処理方式は、実施してみるまでは、関係者の間に一抹の不安があったが、今日までの状況をみると、大成功であった。ユーザのバッチ・ジョブの実に98%までがこれで処理され、郵送されてくるジョブは、わずか2%しかない。

6. TSS とリモート・バッチ処理

図2にも示したように、東大システムでは、ひとつのシステムの中で、TSS サービスやリモート・バッチ処理も行なわれている。ファイルやコマンドも、も

表1 リモート・バッチ・ステーション

設 置 場 所	端末システム	通 信 方 式
千 葉 大 学	MELCOM 9100/30 F	調歩同期式 2,400 bps
一 橋 大 学	FACOM 230-25	"
信 州 大 学	HITAC 10 II (5台)	"
東大土木工学科	OKITAC 4300 C	"
電気通信大学	HITAC 8250	同 期 式 2,400 bps
法 政 大 学	OKITAC 4300 C	調歩同期式 2,400 bps
横浜国立大学	HITAC 8250	同 期 式 4,800 bps

(bps はbits/sec の略。上記のほか、センター内に2組、東大電子工学科、東大電気工学科にそれぞれ1組のテスト用端末がある。)

ちろん共通である。リモート・バッチの方は、調歩同期(非同期)方式は2,400 bps(ビット/秒)の選択コーリング方式のみ、SYN同期方式では、2,400 bpsから48,000 bpsまでの通信速度が使えるようになっている。現在使用中または準備中(法政、横浜)の端末システムを、表1に示す。

一方、TSSの制御は、つぎのように行なわれている。

(1) TSSタスク全体のCPU使用率を、システム生成時に指定する。たとえば、TSS対バッチを30%対70%というようにすれば、タスクはこの比を守るよう取り出される。

(2) TSSのタイム・スライス値を2段階にし、端末入出力のためブロックされていたタスクが入出力完了で起動された直後だけは、すぐに0.05秒程度だけサービスし、その後は、0.5秒程度のタイム・スライスでラウンド・ロビン式にサービスする。これは応答を速めるためで、レスポンス制御と呼ぶ。なお、東大システムでは、短いバッチ・ジョブのターンアラウンドを早めるため、バッチでも、2秒程度のタイム・スライスによる実行制御を行なっている。

(3) バッチとTSSのジョブ(タスク)の多重度を、別々にオペレータ・コマンドで指定できる。現在は、それぞれ8と20程度である。このように多重度が高くとれるのは、主記憶が大きいせいでもあるが、仮想記憶系の特徴でもある。

TSS端末としては、現在、センター内に2,400 bpsの文字ディスプレイが10台、200 bpsのタイプライタ(カナ付)が10台、それに、東大の船舶工学科と化学科にタイプライタがそれぞれ1台ずつ設置されている。オープン・バッチのターンアラウンドが速いせいか、今のところTSSの利用度はそう高くはなく、TSSジョブ件数は10%程度である。

なお、このTSSでは、公衆回線はサポートしていないが、近い将来、音響カップラ内蔵端末機を、電話

経由で使うことも可能にする予定で、準備を進めている。今後の課題としては、この TSS の操作性をさらに向上させ、会話型ソフトウェアの充実ははかることが残されている。

7. 情報の共用と機密保護

東大システムのような、多数のユーザをかかえる巨大なシステムでは、システムや情報の共用と保護は、きわめて重要な問題である。センターで利用している OS 7 の機能には、つぎのようなものがある。

(1) 図 5 に示すように、HITAC 8800/8700 には、MIT の MULTICS で開発されたリング保護の概念が取り入れられている。図 3 にも示したように、プログラム (CSECT) には書き込み禁止の保護がかかり、データ部 (PSECT) には、実行禁止の保護がかかる。読出し禁止の指定もできる。

(2) 主記憶内で新しいページを使うときに、余白はあらかじめクリアする。これは、実空間上でも、他人の情報が盗めないようにするためである。

(3) ユーザを 5 段階のハイアラーキーに分け、それによって、アクセスできる情報のレベルを区別する。

(4) バッチでは文字のパスワード、TSS では乱数による数字パスワードを利用する。バッチのパスワードは、ユーザがとくに指定すれば、ラインプリンタで印字されるのを防ぐことができる。

(5) システム内のユーザ登録簿には、パスワード

のほか、計算料金の予算額といままでの実算額（累積額）を常時記録しておき、実算額が予算額を越えているときは、処理を行なわない。

(6) ユーザ個人用ファイルは、PERMIT, SHARE, INHIBIT, TRANSFER, RECEIVEF などのコマンドにより、いろいろな形の共用（個人間、グループ間などで）ができる。

(7) PUBLIB の利用により、ユーザ・プログラム間の通信ができる。

(8) MAIL, RECEIVE 両コマンドの利用により、ユーザ間で各自 7 通までの電報（ただし長さ 132 B）のやりとりができる。

(9) ジョブ実行開始時に、前回いつ使い、実算額がいくらだったかというメッセージが出るので、他人による不正利用がすぐ分かるようになっている。これは、センター組込みルーチンで実現した。

以上のような機能を利用しているため、パスワードが盗まれないかぎりには、東大システムは、他人による妨害や悪用には、非常に強いシステムになっている。

8. システムの運用と評価

東大システムのような大きなシステムになると、その運用の準備もかなり大がかりになるが、その主なものは、つぎのとおりである。

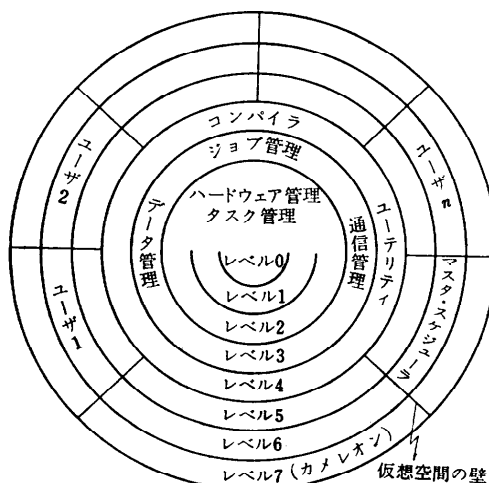
(1) センター・OWN・コーディング・ルーチン
ジョブ・スケジューリング、タスク優先度決定、ジョブ属性チェック、図 4 のオープン・バッチ用各種ディスプレイ、料金計算、花文字（アカウント）ページ打出しなどを行なうルーチンを用意して、OS 7 へ組み込んでいる。

(2) システム・ファイル

OS 7 自体を入れるファイルのほかに、ユーザ登録簿、アカウント情報、ファイル・カタログ（登録簿）、各種キュー（待行列）、入力、出力、作業領域、システム・メッセージ、障害情報など計 29 種のシステム・ファイルについて、いろいろなパラメータを設定し、システムに組み込んでいる。

(3) システム生成

システム生成は、OS 7 の新版が出るごとに行なわれるが、タスクやメモリやジョブのスケジューリング、論理チャネルや入出力チャネル、各 CPU・メモリ・周辺機器の特性、各データ通信機器など計 82 項目について、多数のパラメータを設定してセンター用の OS 7 を作成するため、ほう大な作業となる。



（内から外へはアクセスできるが、逆は原則としてできない。）

図 5 リング保護の実際

表 2 平均ジョブ像などの推移

平均 期 間	1966 年	1969 年	1974 年 2 月
主 計 算 機	H 5020	H 5020 E	H 8800/8700
平均命令実行時間	13 μ sec	1.7 μ sec	0.2/0.8 μ sec
仮想記憶使用量	なし	なし	52 kW*
コア使用量(主記憶枠)	—	24 kW	38 kW*
スワップ回数**	なし	なし	651(73%)***
コンパイル時間	—	—	3.7 秒 (90%)
リンク 時 間	—	—	0.4 秒 (82%)
実 行 時 間	—	—	37.8 秒
CPU 時間合計	144 秒	77 秒	41.5 秒
入力カード枚数	—	440 枚	476 枚
ソース・レコード数	—	—	415 (89%)
出力印字枚数	23 枚	30 枚	23 枚 (各 55 行)
個人ファイル	なし	なし	236 kB (29%)
年間処理件数	71,898 件	123,705 件	280,181 件 (初年度)
年間利用者数	1,648 人	2,191 人	2,182 人

* OS 使用分 5 kW を含むが、実行時ルーチンやコンパイラは含まない。

** コアからドラム、またはドラムからコアへの 1 ページ (1 kW) 分の転送を、それぞれ 1 回とした。

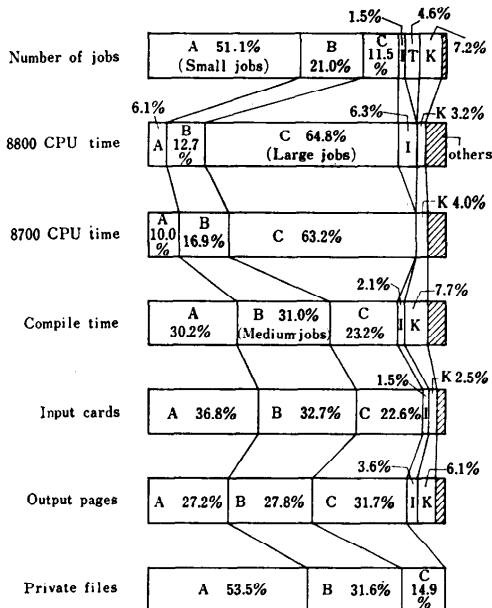
*** カッコ内の % は、その項がゼロでないジョブの比率で、その前の数値は、それらゼロでないものについての平均値。

ところで、センター運用のためのこうした多数のパラメータの最適値をきめるには、システムがどう動いているかを正確に把握する必要がある。そこで、OS7 で、1 ジョブごとに約 450 B ずつ記録される詳細なアカウント・データを、東大センターでは、いろいろな角度から分析して、管理情報として役立てている。

表 2 は、そうした分析情報の一部 (全ジョブの平均値など) を過去のデータと比べた例である。この表から速度差を考えると、CPU 時間からみてジョブは数倍に大きくなっているが、個人ファイルが使えるので、入力枚数はふえていない。印字枚数の方も、用紙不足対策として改ページを極力少なくしているため、1 枚 66 行分の紙に平均 55 行印刷されるほど密に使われているので、増えていない。

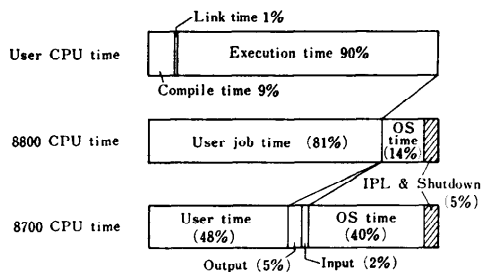
一方、メモリ使用量もまだそう大きくはなく、センターの設定した主記憶枠をそれほどはみ出さないで、スワップ回数も少なく、仮想記憶の害はほとんどなく、むしろ、オーバーレイの考慮をすることなく、大きな配列が扱えるのが、ユーザに喜ばれている。

つぎに、図 6 に示すのは、東大システムでのリソースの使われ方である。小さいジョブほど、8700 で処理される率の高いことがみてとれる。同図(b)は、ユーザに CPU 使用時間として課金できる比率が非常に高く、システム・タスクが、主に 8700 で処理されていることを示している。また、図の OS 時間には CPU のアイドル時間も含まれるが、この OS 時間の内訳を



* (Note) K : Centre jobs I : MT jobs
T : TSS jobs

(a) ジョブ・クラス毎のリソース使用率



(b) CPU 時間の使われ方

図 6 東大システムにおけるリソースの使われ方 (1974 年 2 月の合計)

知るには、ソフトウェア・モニタなどの測定手段が必要であり、その分析は、今後の課題である。なお、IPL (Initial Program Load) とシャットダウンの時間が目立つのは、2 月でも、平日 10 時間程度しか運転していないためである。日本の大学で、アメリカ並みに 1 日 24 時間、週 7 日の運転体制がとれるようになるかどうかは、今後の課題であろう。実際、毎分 4 件もの高いスループットをもつシステムを、人手不足のゆえに休ませながら使うというのは、もったいない話である。

さて、以上のべた東大システムについては、ソフト

ウェア、ハードウェアともに、いっそうの改良・拡充が続けられている。研究面では、学術データ・ベースの試用、北海道・東北・東京・名古屋・京都・大阪・九州の7大学の大型計算機センターを、48,000 ビット/秒程度の回線で結ぶコンピュータ・ネットワークの検討などが、徐々に進められている。

最後に、従来できなかったような大型計算を可能にする超大型コンピュータ・システムが、東大センターにおいて実現できたことについて、(株)日立製作所および東京大学関係の方々の、長期間にわたる非常な努力に謝意を表わしたい。本稿で詳しくのべられなかつ

た OS7 の構造については、大西氏らの論文を参照していただければ幸いである。

参考文献

- 1) 大西ほか: HITAC 8700/8800 オペレーティング・システム (OS7), 情報処理, Vol. 14, No. 10, pp. 669-777 (1973).
- 2) 石田: 超大型コンピュータの技術 (1)~(7), bit, Vol. 5, No. 11 (1973)~Vol. 6, No. 4 (1974).
- 3) 石田: 東大センター TSS の構造と機能, 電子通信学会電子計算機研究会資料 (1974. 4. 30).
(昭和 49 年 3 月 15 日受付)