超大型機のふるまいのグラフィック・モニタリング
── 4-CPU マルチ・プロセッサシステムの事例 ──

東京大学大型計算機センター
石田晴久・野本征子

§1 はじめに

計算機が大型化されてくるにつれ，システムがどのように動いているかをとらえることは難しくなってくるが，それを知ることはシステム構成を評価し，その改善策および次期システムのあり方を検討する上で重要な情報を与える。我々はHITAC 8800, 8700 各2台から成り，仮想記憶方式を採用しているマルチ・プロセッサ・システムの性能を調べることを目的として，システムが保有している情報をとり出し(一部はOSにパッチを入れてとり出し)，編集を行ない，結果をグラフィック・ディスプレイに表示した。現在，ディスプレイ上に表示される情報は次のようなものである。

(1) 4台のCPUのアイドル状態。
(2) 主記憶，仮想記憶領域の使用状態(どのようなタスクが占めているか，システムのどの部分が主記憶上に存在しているか等)。
(3) ページ・フォールトとスワッピングの状態。
(4) リエントラント・プログラム(コンパイラ，実行時ルーチン，リンケージ・エディタ等)の共用数。
(5) I/O 装置の使用状態(I/O発行回数，I/O待ち行列の数)。
(6) ドラム(スワップ・イン，スワップ・アウト用)の使用状態。
(7) ドラム(スワップ・イン，スワップ・アウト用)にREAD/WRITEしたページ数(チェンジビットの効果)。
(8) タスク・スイッチの累積値。
(9) 状態別タスク数(ランニング，レディ，ウェイト，ページング，ブロック，ペンディングの各状態について)。
(10) TSSレスポンス時間の状態(入力，レスポンス，出力，連続出力，思考，連続入力)。
(11) 上記 (1), (2), (3), (5), (6) について1日(30分間隔)，1時間(5分間隔)の状態変化。

なお，(11)以外はシステムの状態を実時間でディスプレイ上に表示するものである。

§2 本システムの概要

本システムは4台のCPUをもつ，マルチ・プロセッサ・システムであり，主な構成要素はH-8800, H-8700 CPU各2台，主記憶3MB，スワッピング用高速磁気ドラム(4MB) 4台，ファイル用磁気ドラム(4MB) 2台，超大型磁気ディスク(100MB) 12台，磁気ディスク(29MB) 32台，磁気テープ10台，カード・リーダ6台，ライン・プリンタ12台，カード・パンチャー5台等である。

本システムでは仮想アドレス方式をとっており，仮想空間をベース，セグメント，ページと呼ぶ単位に分け 1ページ=4096B，1セグメント=64ページ，1ベース=64セグメントである(図1)。また，実アドレスは24ビットより成り，図1の32ビットの論理アドレスをアドレス変換機構により変換して求めている。主記憶と高速磁気ドラムの間のスワッピングはページ単位に行なう。

ジョブの形態は①バッチ・ジョブ，②会話(TSS)ジョブ，③リモート・バッチ・ジョブ，

④実時間ジョブの4つに分けることができるが，これらの形態によってCPUを分けることはせず，どのCPUを使用するかは不定である。現在は，H-8800 2台は主に演算，H-8700 2台は主に入出力およびOSの動作を行なうようにしている。

利用形態としては，①オープン・バッチ方式(do it yourself I/O)，②クローズド・バッチ，③リモート・バッチ(15ステーション)，④TSS(22端末)，⑤グラフィックスがある。①のオープン・バッチ方式は当センター特有のもので，ユーザはライン・プリンタ，カード・パンチの装置の前にとりつけてあるトークン・リーダにトークン・カードをさし込む事によって任意の時刻に自分のジョブをとり出すことができるようになっている。

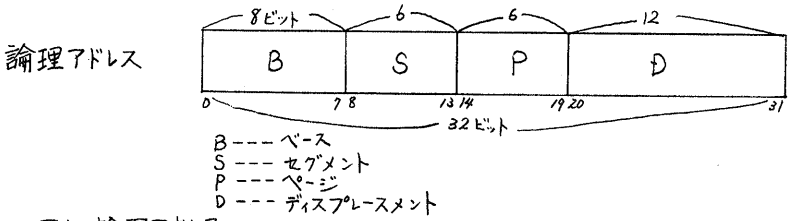

図1 論理アドレス

§3 ソフトウェア・モニタのテクニック

次に我々が行なっているソフトウェア・モニタの概略を述べる。

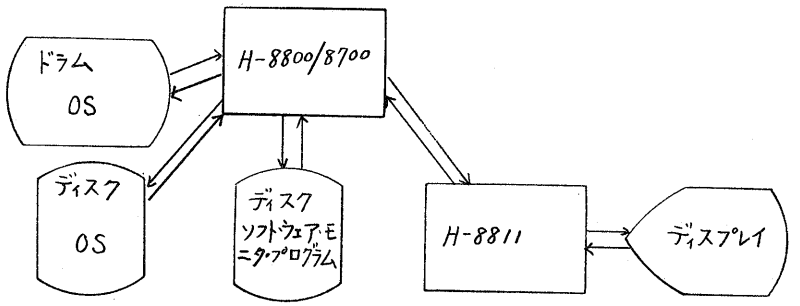

図2 ソフトウェア・モニタの概略図

システム領域に格納されている必要な情報をプログラム中にとり込み，H-8811グラフィック・システムのディスプレイ上に表示する(図2)。H-8811は8K語の主記憶を持ち，H-8800/8700とは100KBのデータ交換チャネルで結ばれている。

次に一例として，I/O発行回数を求めるルーチン，およびリエントラント・プログラムの共用数を求めるルーチンの内容を示す。

(a) I/O発行回数を求めるルーチン

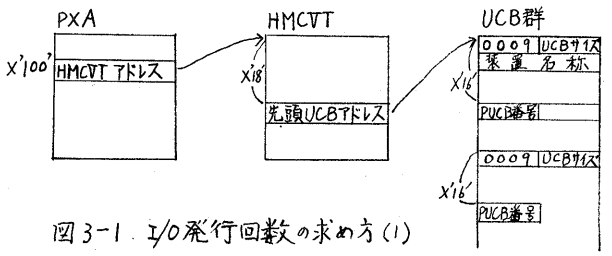

図3-1 I/O発行回数の求め方(1)

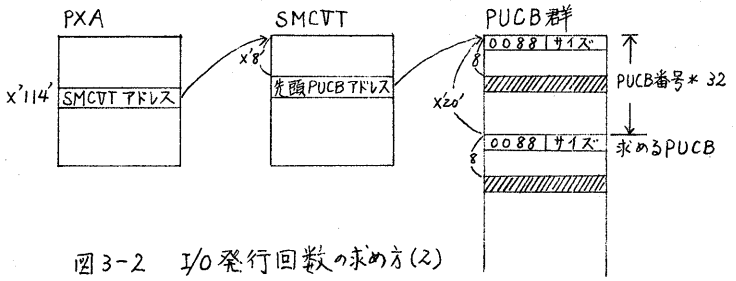

図3-2　I/O発行回数の求め方(2)

PXA(プリフィクス・エリア)は0～10000番地までを占め，各CPU固有のエリアで，CPUが働くために必要な情報を格納しておく部分である。まず，図3-1のようにポインタをたどり，UCB群を捜し，その中から自分の捜したい装置名称を捜し，そのPUCB番号を求める。別に図3-2のようにポインタをたどりPUCB群を捜し，その先頭からPUCB番号＊32バイト(PUCBの先頭)＋8バイト目にあるI/O発行回数を求める。

(b) リエントラント・プログラムの共用数

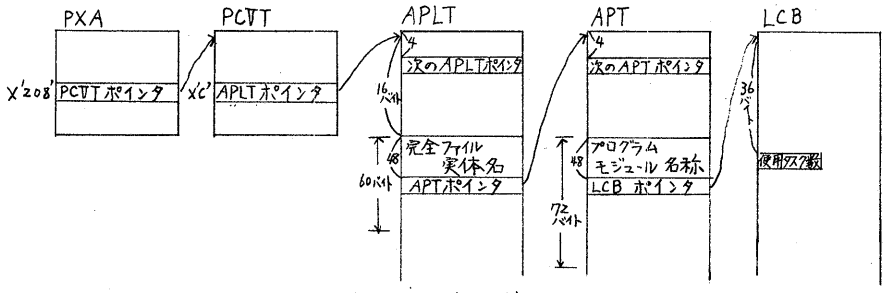

図4　リエントラント・プログラムの共用数の求め方

図4のようにポインタをたどり，APLTを求め共用数を求めたいプログラム・モジュールが含まれるプログラム・ライブラリ・ファイルを完全ファイル実体名をもとに捜す。そのAPTポインタをたどり，求めるプログラム・モジュールを捜し，LCBポインタをたどり使用タスク数を求める。APLT, APTで最後のブロックにおけるポインタは0がセットされている。

I/O発行回数を求めるプログラムは名称で1つ1つサーチするためにかなり時間がかかる。又，H-8800/8700とH-8811のデータ伝送に時間がかかり一つの画面を出すのに3～10秒位かかる。

情報を捜し出すには上記のようにポインタを追っていくため，実行途中にポインタが切れてしまうことがあり得る。このため，ポインタが切れた場合を考え，プログラム・チェック割込みを登録しておき，再試行できるようにした。又，各表示ルーチンにWAITマクロ(一定時間WAITして再び実行を再開する)を使用し，一定時間間隔で，順次，その時刻の状態を表示するようにした。

H-8800/8700は機密保護のために，8段階のリング・レベルを設けており，他のレベルから読み出し，書き込み，実行を許さないようなプロテクションをかけることができる。システム領域のある部分は読み出し禁止になっており，我々が必要としたデータの入っている領域にも読み出し禁止のプロテクションがかかっていた。このため，我々はそれらを読み出すために，日立側により特別に用意された

LOOKマクロを使用した。LOOKマクロはシステム・プログラマだけに許されたマクロであり，これを使用することにより，読み出し禁止になっている領域からも読み出しを行なうことができる。LOOKマクロを使用するときは，各レジスタに次の値をセットしておく。

GR14-- 読み出したい領域の先頭アドレス。
GR1-- 読み出した情報を格納する領域の先頭アドレス。
GR0-- 読み出したい情報の長さ(バイト数)。

又，仮想アドレスと実アドレスの対応にあたっては，LRA (Load Real Address) 命令を使用した。この命令は第2オペランド・アドレスを実アドレスに変換し，第1オペランドで指定した汎用レジスタにロードする命令である。

我々のソフトウェア・モニタは情報を採取してくる部分ではアセンブラを用い，ディスプレイの部分はFORTRANのグラフィック・サブルーチン・パッケージ(GSP)を使用している。採取して編集する情報にはスナップ・ショット型のものと累積値とがある。次にそれらを少し詳しく述べる。

(1) 4台のCPUごとのアイドル・タイム。それぞれのCPUはアイドル状態の時，アイドル・タスクをランさせている。そこでそのアイドル・タスクをランさせている時間の累積値をアイドル・タイムとしている。

(2) 主記憶，仮想記憶の使用状態。主記憶768ページ(3MB)のうちシステム常駐部分，システム非常駐部分，ユーザ領域，未使用領域が占める割合，又，システム領域中でどういう部分が主記憶に存在するか。

(3) ドラムの使用状態。ドラム(各1040ページ≒4MB)4台のうち，実際に使用している部分はどの位あるか。

(4) ページングの回数。システム領域およびユーザ領域について，それぞれ次の3つの値を表示する。
(a) 領域を参照してページ・フォールトを発生した回数。
(b) 領域を参照してページ・フォールトを発生し，ドラムをREADした回数。
(c) 領域を参照してページ・フォールトを発生し，当ページに割当てるための主記憶が不足したり，又，当ページはページ・インかページ・アウトのI/O動作中で完了を待つためにページング状態となった回数。

(5) リエントラント・プログラムの共用数。コンパイラ，実行時ルーチン等はほとんどがリエントラントな構造をしており，現時刻にどのようなモジュールがタスク間で共用され同時にランしているかを表示する。

(6) ディスク(入力スタック，出力スタック，WORK用のファイル，システム・ファイル)，ドラム，ライン・プリンタ，カード・リーダ，磁気テープ，紙テープ等100台以上の装置のそれぞれについてI/O発行回数と装置使用の待ち行列数を表示する。

(7) スワッピング用ドラムに対するREAD/WRITEのページ数を表示する。主記憶にとり込まれ，次にそれをスワップ・アウトする場合，もしこのページに書き換えが生じておらず(この場合，当ページのC(change)ビットは0のままになる)，ドラムに全く同じものが存在する場合は実際にはドラムにWRITEされない。このような現象がどの位起きているか，すなわちCビットの効果がどの位あるかが分る。

(8) タスク・スイッチの累積値。タスク・スイッチとはシステム・タスク，ユーザ・タスク，アイドル・タスク間の変化の事である。シングルタスクでもタスクT_1 → アイドル・タスク → タスクT_1と変化するとカウンタには2が加えられることになる。

(9)状態別タスク数。タスクにはランニング，レディ，ウェイト，ページング，フロック，ペンディングという状態があるが，これらを分析し，タスクの状態を表示する。

(10) TSSレスポンスの分布。入力，レスポンス，出力，連続出力，思考，連続入力，のCPUタイムとETIMEをヒストグラムとして表示する。

(11) (1), (2), (3), (4), (6)の1日又は1時間の変化。

§4 結果と考察

以上のような情報より，我々は次のような知見をえた。これらのうち，(2), (4), (8)などはこれまで恐らくは実測されたことのない情報だと思われる。

(1) H-8800 1台，H-8700 2台の3-CPUで運転した場合はアイドル率は5%～10%と低い値を示しているが，4台のCPUで運転した場合は30%～40%位アイドルしている(図5, 6)。これは，4台のCPUに対しては3MBの主記憶では少なすぎるということを示している。なお，74年9月～75年1月にCPUの改造を行なったが，改造を行なう前は4-CPUでアイドル率は2%～7%であった。

又，本システムではH-8800 2台は主に演算用に使用し，H-8700 2台は主に入出力を含むOS動作用に使用しているが，アイドルは2台のH-8800の方が常に少なく，25%～35%であり，2台のH-8700の方は35%～50%のアイドル率である。

なお，75年9月頃より主記憶を1MB増し，4MBにすることになっているので，それが実現できればアイドル率は大幅に減るものと期待される。

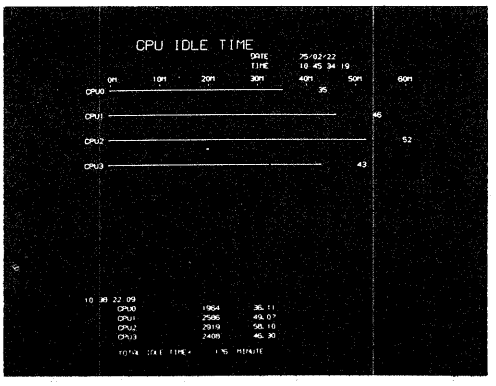

図5 アイドル状態

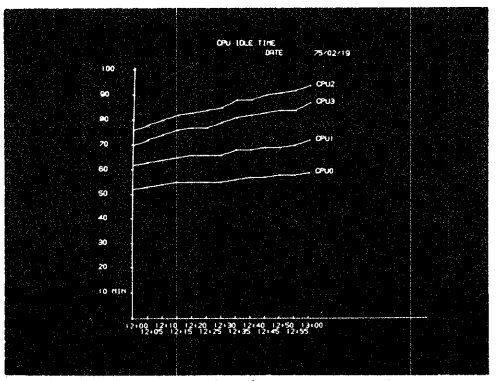

図6 1時間のアイドル時間の変化

(2) OSの占める領域(SYSRES, SYSLIB, PUBLIB, テーブル類)が時として50%(1.5MB)になっているが，OSの計画当初の考えではOS領域は23%(700KB)位にするはずであった。現在はユーザが使用できる領域が少なすぎると思われるが，主記憶を1MB増やせばOS領域を30%～35%までに減少できると考えられる。図7はシステム領域が50%近い例で短い縦線がSYSRES(タスク管理，ジョブ管理，データ管理その他)，長い縦線がSYSLIB(コンパイラ，ライブラリ，I/Oルーチン，ユーティリティー，テーブル類)であり，空白の部分がユーザプログラム又は空きエリアである。

又，図7からSYSRES, SYSLIB, ユーザプログラムがページ毎にランダムにとり込まれていることがわかる。0～255ページの先頭の短い縦棒がかたまった部分がPXA(プリフィクス・エリア)で真中より少し左に短い縦棒がかたまった部分はSYSRES常駐部分であり，いつもこの位置にとり込まれる。

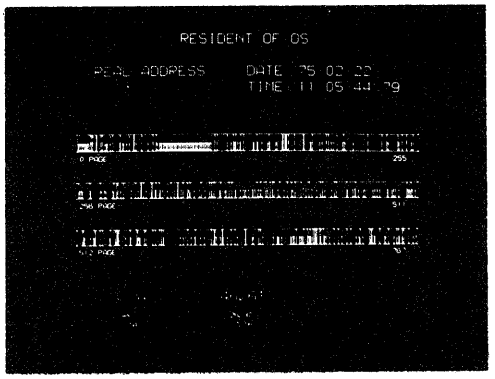

図7 主記憶の状態
短い縦線 SYSRES
長い縦線 SYSLIB
空白 ユーザ領域,未使用領域

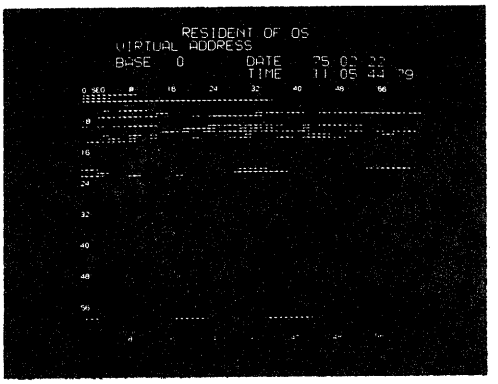

図8 仮想記憶上の状態(BASE 0)
白く光っている部分が主記憶上に存在する。

(3) 図8は仮想記憶上を表示したもので白く光っている部分が主記憶に存在する部分でFORTRANコンパイラ, FORTRAN実行時ルーチン(BASE 0の20～23セグメント)がいつも主記憶に存在することがわかる。

(4) ページングの状態は図9,図10により見ることができるがページの呼びもどし率は6%位ある。OSのページングの設計方針はFINUFO(First In Not Used First Out)というアルゴリズムで行なわれており,ページの変化状態は図11の通りである。主記憶にとり込まれ,次にスワップアウトされる時,Cビット(Change bit)を調べ,そのページが書き換えられていたら,スワップアウトされ,再びドラムに書き出されるが,Cビットが0の時はそのページは書き換えられていないということでドラムに書き出すことはしない。この状態は図10で見ることができ,Cビットの効果が高い事が分る。

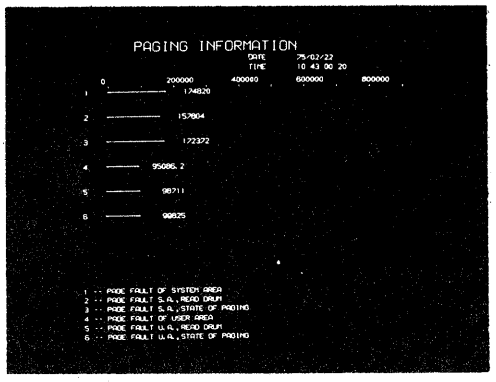

図9 ページングの情報
§3(4)のa,b,Cのシステム領域に対する値を1,2,3, ユーザ領域に対する値を4,5,6として示してある。

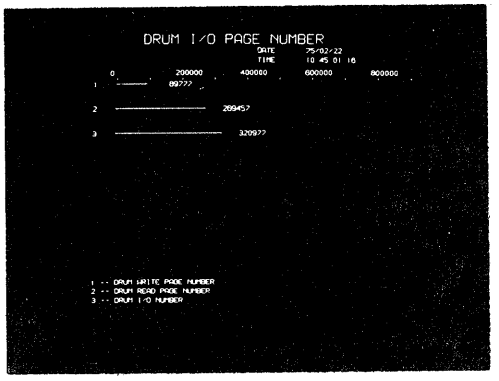

図10 ドラムにREAD/WRITEしたページ数
1はドラムWRITEのページ数
2はドラムREADのページ数
3はドラムI/O数

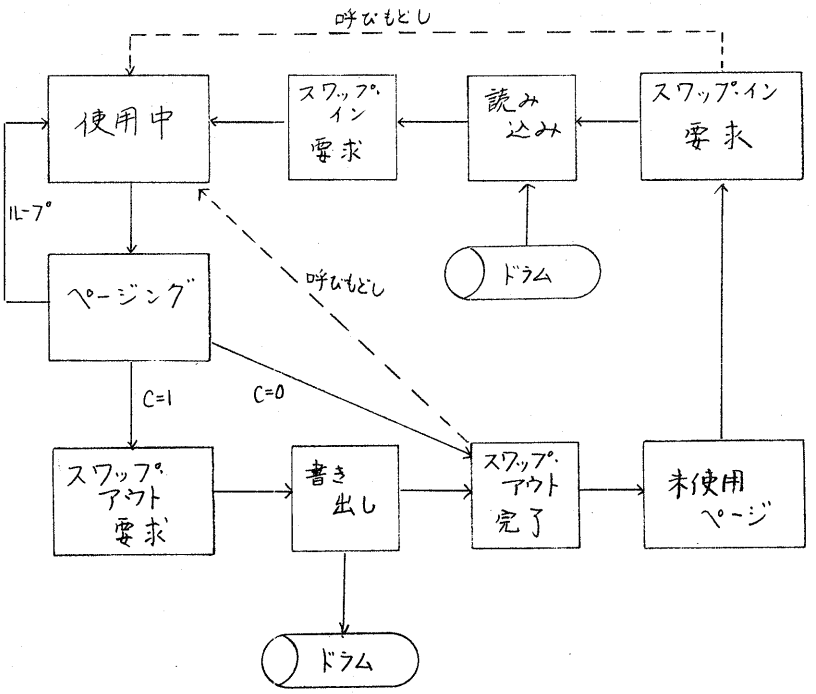

図11　ページの変化図

C=1の時はメモリが書き換えられているのでスワップ・アウトされドラムに書き出されるが，C=0の時は書き換えが行なわれておらず，メモリとドラムの内容が一致しているためスワップ・アウトは行なわれない。

5) I/O発行回数(図12, 13, 14)より，少しの差はあるがカードリーダ，ライン・プリンタはほぼ平等に使用されていることがわかる。又，kk1(ディスク)にはユーザの名簿および　カタログ(ユーザのファイルの名簿)が入っているが，これが非常によく使用されていることがわかる。

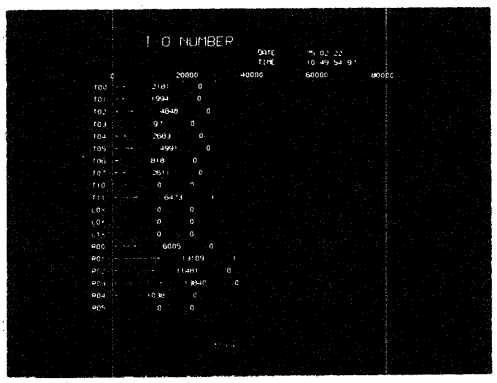

図12　I/O発行回数と待行列(1)

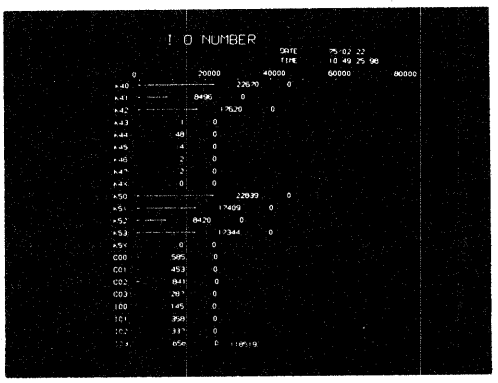

図13　I/O発行回数と待行列(2)

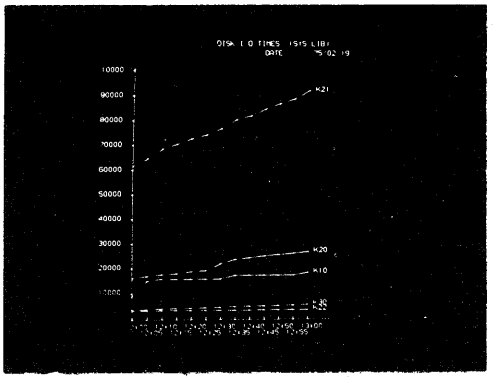

図14 I/O発行回数の変化(1)

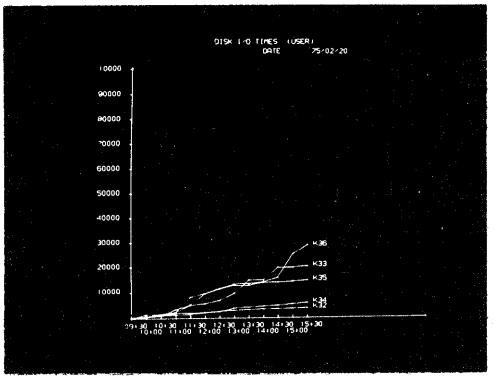

図15 I/O発行回数の変化(2)

(6) I/O装置への待ち行列はほぼ0か1でラインプリンタ,カードリーダに並ぶ程度であり,OSによる装置割当てがうまくいっていることがわかる(図12,13)。

(7) ドラムの実際に使用されているページは28%～38%で4台のドラムが均等に使用され,スケジューリングがうまく行なわれていることがわかる。しかし,現状ではドラムはほぼ90%近く領域的には使用されており,実際に使用ページとはなっていないが領域確保は行なわれている。ユーザにより領域指定が行なわれ実際は未使用であるページがドラムの55%～65%もあるのは少し多すぎるのではないかと思われ,領域指定の方法につき検討すべき点があると思う(図16)。

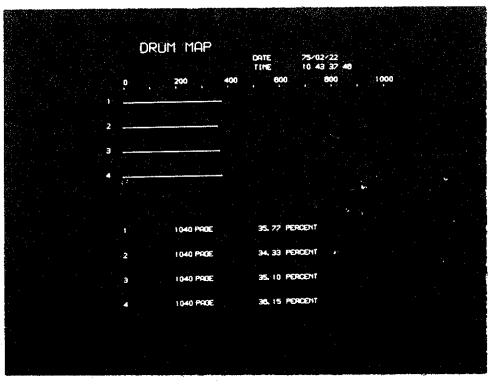

図16 ドラムの使用状態

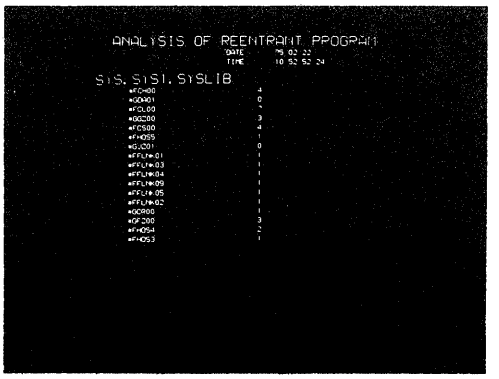

図17 リエントラント・プログラムの共用数
右の数字がリエントラント・プログラムを現在使用しているタスク数を示す。

(8) リエントラント・プログラムの共用数について調べてみると,モジュールを使用しているタスク数は ユーザ・タスクの多重度がバッチ8, TSS 15 計23の時次のようになっている(図17)。
#FCH00(FORTRANコンパイラ)が2～5, #FCL00(FORTRAN実行時ルーチン及び基本外部関数等)が4～7, #FCS00(FORTRANサービス・ルーチン,デバッグルーチン)が1～4でFORTRAN関係の3つのルーチンは,主記憶からスワップ・アウトされることなく常に主記憶上に存在していることがわかる。又,#GDA01(リンケージ・エディタ)は0～2, #GDZ00(シンボリック・ライブラリ保守)0～2,その他

COBOL, PL/I の実行時ルーチンは0～1である。

(9) 図17および図18よりシステムの常駐部分は時間とともにほんの少しづつ増し11%～12%の領域を占め，一度主記憶にとり込まれると追い出されることは絶対にない。システムの非常駐部分は時間によって，かなりの差があり，多い時は主記憶の40%を占め，少ない時は20%程度である。未使用領域はほぼ5%以内であるが，朝10時までと，昼頃等，比較的ユーザが少なくなった時に増える傾向にある。

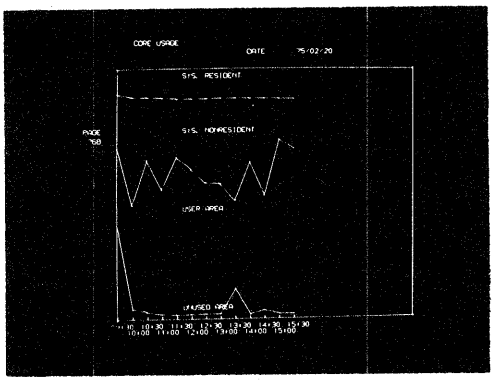

図17 主記憶の使用状態（1日）

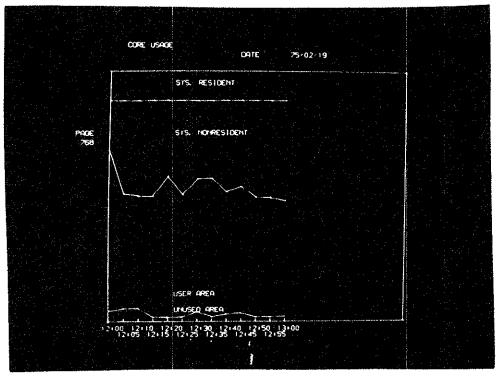

図18 主記憶の使用状態（1時間）

§5 結論

本論でのべたソフトウェア・モニタリングの手法は主記憶の状態，ページングの状態，OSの使用状態，タスクの状態，周辺機器の状態，リエントラント・プログラムの共用数等を測定し，仮想記憶ページング方式等について，現在の状態を知り，さらに次期システムへの資料を得るのに役に立つ。従来，計算機システムの真価はCPU速度と記憶装置等，ハードウェアにあると考えられてきたが現システムのように大きなシステムになると，それだけでなく，記憶領域への割りつけ方や周辺機器の構成にも影響をうける。周辺機器に対するネックなどは我々の採取情報によりかなり知ることができたと思われる。OSによる記憶領域の割り付け方式にはまだ検討すべき点が残されており，よりよい方式を考えることが必要であろうと思われる。

ここで作製したソフトウェア・モニタでは，システムを評価するための情報は比較的少ないオーバヘッドで(TSSレスポンス・タイム測定はオーバヘッドを少しとる)採取できたが，このように大きなシステムになると，将来はOSで採取できる情報をもっと増やすとともに，必要なデータをハード的に測定できるようなハードウェア・モニタの機能を内蔵すべきだと思われる。

なお，このソフトウェア・モニタはアセンブリでコーディングした部分が1400ステップ，FORTRANでコーディングした部分が2900ステップである。

最後に本ソフトウェア・モニタの開発に当り，OSの若干の改造，LOOKマクロやTSSヒストグラム測定用パッチの提供，種々のシステム情報の採取法の教示などに快よく応じて下さった日立ソフトウェア工場システムプログラム部の大西勉氏を中心とする技術陣に心から感謝する。

参考文献

1. 石田：東大超大型コンピュータシステム，情報処理学会誌，Vol. 15, No. 7, pp. 534-541 (1974).
2. J.H.Saltzer: The instrumentation of Multics, Comm. ACM, Vol. 13, No. 8, pp. 495-500 (1970).
3. J.M.Grochow: Real-time graphic display of time-sharing system operating characteristics, Proc. of AFIPS FJCC, Vol. 35, pp. 374-386 (1969).
4. T.Masuda: Optimization of program organization by cluster analysis, Proc. of IFIP Congress 74, Vol. 2, pp. 261-265 (1974).