

苫小牧高専における新電算機システム（その2）

（マクロ・コマンドの開発及びベンチマークテスト）

中 津 正 志*・三 河 佳 紀**

The New System of the Computer in Tomakomai Technical College (Part II)
(Development of Macro-Command Programs and Bench Mark Tests)

Masashi NAKATSU and Yoshinori MIKAWA

要 旨

本校電算機システムを効率よく使用するための各種マクロ・コマンドを開発した。これらは、繰返し行う処理、ある決まった手順で行う処理を簡単なコマンドで実行することが出来る。

又、このシステムの能力を調べるためにベンチマークテストを実施した。テストの主な内容は、TSS レスポンス、四則演算速度及び関数実行速度である。

Synopsis

In order to use efficiently the system of the computer in our college, the authors developed various macro-commands. Since they made regular procedures and monotonous and repetitive jobs into simplified commands, it is easy to use these commands.

Moreover, they carried out the bench mark tests to study an ability of this system. Principle contents of these tests are the response speed of Time-Shearing System, the time of arithmetic calculation and the speed of execution to functions.

1. は じ め に

前報¹⁾において新システムのハードウェア及びソフトウェアの概要、運用状態について報告した。その後、情報処理の授業や、実験、卒研、研究で使用する中で、次の事柄について開発や調査をする必要が出てきた。

1) 電算機にある処理をさせるには多くのコマンドを順序よく間違いなく入れなければならない。その際、キーインミスや手順ミスのために思うような処理ができず時間を労費してしまうことはよく経験することである。従って、多くのユーザが共通して行う処理はマクロ化し共通領域に登録することによって利用の便をはかることは急務

である。筆者等は電算室を利用するユーザの要望の中から使用上及び管理運営上頻度高く行なわれる処理に対するマクロ・コマンドの開発を行った。

2) 同時利用端末数が多い程 TSS レスポンスが遅くなることは自明のことであるが、具体的に使用端末数によってどれだけレスポンスが変化するかは知られていない。又、基本演算速度についても同様である。そこで、本校のシステム構成によるベンチマークテストを計画し実施した。

2. マクロ・コマンドの開発

CLI (Command Line Interpreter) マクロ・コマンドは一般に複数のコマンドを内容としたファイルであるが、引数を利用してユーザ独自のコマンドを定義したり、マクロ・コマンドの中で他のマクロ・コマンドや自分自身を再帰呼び出しして実行できるので、あらかじめ提供されている擬似

* 機械工学科 助教授

** 教育メディア開発センター職員

マクロと組み合わせることによって、ユーザの希望する機能を持ったフレキシブルな CLI マクロ・コマンドを作成することができる。表－1 は筆者等が開発したマクロ・コマンドの一覧である。表中の番号2 と5 についてはリストを巻末資料1 に掲載した。主なものについて述べると、

番号1 MTERM はホスト1 のオペレータ用のディレクトリ内にあり電算関係センター員しか使用できない。ターミネイトするのは CAI 室に設置した端末のプロセスだけである。前報にも述べたが授業終了時の学生の操作ミスで次の利用学生に迷惑をかけることが多かったが、このマクロ・

コマンドを使用することでこの種のトラブルはなくなった。

番号2 F77CLE は単にコンパイル、リンク、実行するだけでなく、処理中に作られる中間ファイルを用済後削除するのでユーザ領域の節約につながった。簡単なプログラムでも、実行後は 200 block (102.4 KB) 前後の中間ファイルがつくられる。不要ファイルの削除は個々のユーザのデリート作業を軽減し、ミスによる種々のロスを少なくする上で大切である。このマクロを紹介した後のホスト1 のユーザ領域の使用率が学生全体で 38.2% から 27.7% に下がるなどその効果が確かめら

表－1 マクロ・コマンド一覧

番 号	マ ク ロ 名	利 用 者	処 理 内 容
1	MTERM.CLI	電算センター員	・ CAI室にある端末だけを一齐にターミネイトする。 (HOST01から1 回実行するだけでHOST02,HOST 03につながっているCAI室の端末で走っているプロ セスもターミネイト出来る)
2	F77CLE.CLI	全ユーザ	・ ファイル名 .F77 のプログラムをコンパイル、リン ク、実行し結果をディスプレイする。 ・ 処理中に作られる中間ファイル (.OB .ST .PR .ERR) は処理後自動的に削除する。 ・ スイッチによって結果の出力先を変えられる。
3	MCRE.CLI	電算センター員	・ CLIマクロコマンドやユーティリティプログラムな どをHOST01～HOST03の: UTIL (ディレクトリ 名) の下に登録する。
4	XYP.CLI	全ユーザ	・ ファイル名 .F77 のXYプロッタプログラムをコン パイル、リンク、実行し結果をXYプロッタへ出力 する。(他の働きはF77CLEマクロに同じ)
5	MSPACE.CLI	電算センター員 授業担当者	・ ユーザ名で示されたユーザの使用ブロック数を調べ 集計し平均使用率を計算する。 ・ ユーザ名はテンプレートを使用出来る。
6	QPUR.CLI	電算センター員	・ バンドプリンタ、レーザプリンタ、XYプロッタ等 の出力キュー内にある不要なファイルを全て削除す る。
7	MQC.CLI	電算センター員	・ 不要な出力待ちのジョブ (全ユーザ) を削除する。
8	MQCA.CLI	全ユーザ	・ 自端末から自分の出力待ちのジョブをキャンセルす る (他のユーザの出力結果は削除する事は出来ない)
9	F77C.CLI	全ユーザ	・ ファイル名 .F77 のプログラムをコンパイルしエラ ーがある場合スイッチで指定した先ヘソースリスト とエラーメッセージを出力する。
10	MWK.CLI	全ユーザ	・ WORKの下に自分のディレクトリを作りその中に 指定したファイルを移す。 ・ 作業後、指定したファイルを自分のディレクトリに 移し WORK 下のファイルは抹消する。

れた。

番号3 MCREはマクロ・コマンドの登録作業を簡単にするために作成した。一般ユーザ向けのマクロは実際に試用してもらい、その使い勝手や要望によって機能を追加したり変更することが多い。従って1つのマクロについての登録作業は何回か必要であるが、そのたびに3台のホストコンピュータへ同種の操作をするのは大変であった。

番号5 MSPACEは学生、教官、卒業研究生など、ユーザの任意のグループのユーザ領域の使用状態を一覧表に出し使用率も計算するマクロである。授業担当者はクラス全体や学生個々の領域使用状態を把握することができる。又、今後、適正ユーザ領域を算定する場合の基礎資料を得る上でも有効である。

番号8 MQCAは自分のジョブをキャンセルするマクロである。学生は授業中何回も、修正したプログラムを実行するが、出力不要となった結果をQDISPLAYにより表示された中から、シーケンス番号をタイプインすることによって簡単にキャンセルできる。

マクロ・コマンドを開発し使用するなかで、次のような反省を得た。

1) マクロ・コマンドにたくさんの機能を持たせようとするとかえって使い方が複雑になり、ユーザが覚えなければならない事柄が多くなりマイナスである。特にスイッチや引数は、あらかじめ提供されているCLIマクロ・コマンドのスイッチや引数と混用されることが多いので注意が必要である。

2) キーインミスや誤用はつきものであるなのでその対策、たとえば、リトライできるようにしたり、画面上にメッセージを出して適切な指示を与えることも有効である。

3) マクロ名や使用方法に整合性を持たせ、できるだけ標準化したものとする。

4) あまり多くのマクロ・コマンドがあるとかえって使用しにくくなるので、ユーザの使用状態や希望を把握した後、内容をよく吟味して作成することが必要である。

5) 作成後も適当な試用期間を経てから登録するようにし、登録後の仕様変更はユーザに混乱をおこしやすいので極力さけるべきである。

3. ペンチマークテスト

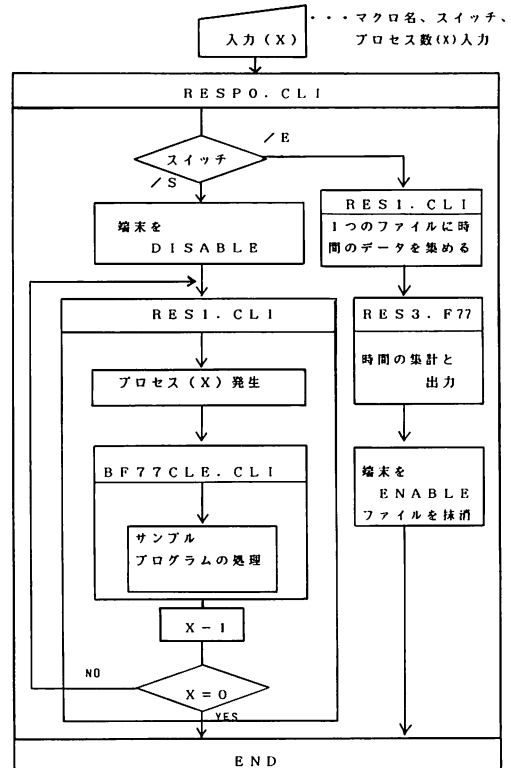


図-1 TSS レスpons調査用プログラム

3.1 TSS レスpons

図-1はTSSレスpons調査用プログラムの概要である。マクロ名 (RESPO) と同時走行プロセス数 (x) を入力することにより、一担全端末をディスエーブルし、次に RES 1 により各端末に1本ずつ x 本のプロセスを割り合て、あたかも x 人の学生が同時にプログラムを実行させたと同じ状態をつくる。F 77 CLE はコンパイルから実行までの各ステップの時間を計測する以外は表-1 F 77 CLE マクロ・コマンドと同じである。サンプルプログラムは行列 (50×50) の積計算をする簡単なもの (step 数 52) で実行時間 8 sec 程度のもを使用した。コンピュータの主な仕様は、

機種	日本データゼネラル
	MV/2000 DC (32 bit)
主記憶	5 MB
メモリ転送	320 nsec/2 リード
処理能力	0.96 MIPS
アドレス空間	1 ユーザ 2.1 Gbyte
使用言語	FÖRTRAN 77

表-2 TSS レスpons時間

単位: Sec

同時 プロセス数		C O M P I L E		L I N K		E X E C		T O T A L			
		A	B	A	B	A	B	A : OPT無			B : OPT3
		OPT無	OPT3					M I N	M A X	平均	平均
M	1	13.0	25.0	15.0	15.0	8.0	5.0	—	—	36.0	45.0
V	2	18.0	39.0	24.0	23.5	17.0	9.5	59.0	59.0	59.0	72.0
2	3	24.3	53.3	32.0	34.3	23.3	12.0	78.0	81.0	79.7	99.7
0	5	36.4	79.4	53.4	53.4	39.6	15.4	121.0	133.0	129.4	148.2
0	10	55.7	138.5	102.1	108.1	74.6	40.9	195.0	252.0	232.4	287.5
0	15	74.2	204.7	169.3	161.8	114.2	43.9	293.0	395.0	357.7	410.4
D	20	85.6	289.3	316.6	206.1	108.0	47.9	461.0	567.0	510.2	543.3
C	24	91.0	335.7	420.6	326.8	124.4	70.6	517.0	732.0	636.0	733.1
PC-9801VM		29.0	—	63.0	—	20.0	—	—	—	112.0	—

注) A: コンパイル時オブテマイズ無し

B: コンパイル時オブテマイズレベル3を指定

このコンピュータの最大接続端末数は24なのでプロセス上限値を24とした。なお、授業では44台の端末を3台のホストコンピュータに分散して接続しており、実験に使用したホストは16台がつながっている。参考までにRESPO及びRES3のリストを巻末資料2に示した。

表-2に実験結果を示す。表中Aはコンパイル時オブテマイズを働かせない場合で、通常この状態で授業を行っている。Bはオブテマイズレベル3(最高)の場合の結果である。下欄はパーソナルコンピュータで同じプログラムを実行した時の結果である。

Aの場合、プロセス数1と24ではコンパイル時7倍、リンク時28倍、実行時15倍、トータル17.7倍となった。同じくBの場合、13.4倍、21.8倍、14.2倍、トータル16.3倍でAに比しコンパイル時間の増加が大きい。Aの場合のトータル時間の最大最小を見ると、プロセス数5程度までは大きな差はないが、それ以上になると差が指数関数的に広がっている。パソコンによる処理時間はAの2.2倍(コンパイル)、4.2倍(リンク)、2.5倍(実行)、3.1倍(トータル)となった。

図-2はプロセス数によるレスポンスの変化をグラフにしたものである。コンパイルと実行時間はプロセス数が増してもそれほど増加しないのに対し、リンク時間は、プロセス数15あたりから急に増加している。トータル時間も15あたりからわずかに増加率が増しているようである。前述したように、授業での利用端末数は16台(ホスト3)であり、グラフで見る限りレスポンスは、この範囲

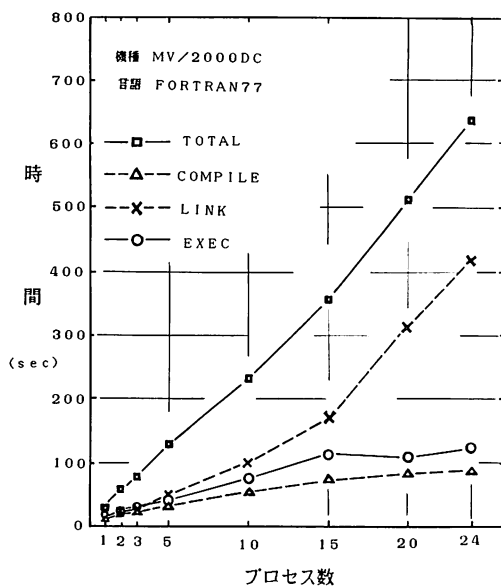


図-2 プロセス数によるレスポンスの変化

で、プロセス数に比例して長くなっている。実際の使用状態で同時にいくつのプロセスが走るかは、現在、推定するしかないが、群馬高専の調査³⁾によると、同時走行プロセス数は多くて全端末(20台)の1/2程度と見つめられている。本校にあてはめると16台中8台(8プロセス)としてトータルレスポンス約191secとなる。しかしこれはほんの短い間のピーク時であり、通常は、8割近くがエディット作業をしていると考えられる²⁾ので、その場合は約80sec(プログラム内容にもよる

表-3 比較表（最適化及びスケジューリング）

単位：％

プロセス数		C O M P I L E			L I N K			E X E C			T O T A L		
		A	B	C	A	B	C	A	B	C	A	B	C
M	1	100	192	92	100	100	100	100	63	113	100	125	100
V	2	100	217	114	100	98	100	100	56	100	100	122	104
2	3	100	219	102	100	107	104	100	52	107	100	125	104
O	5	100	218	102	100	100	100	100	39	104	100	115	102
O	10	100	249	121	100	106	105	100	55	111	100	124	111
O	15	100	276	214	100	96	96	100	38	112	100	115	126
D	20	100	338	271	100	65	71	100	44	157	100	106	123
C	24	100	369	367	100	78	72	100	57	159	100	115	131

注） A：オブテマイズ無し ヒューリスティック方式
 B：オブテマイズレベル3 ヒューリスティック方式
 C：オブテマイズ無し ラウンドロビン方式

が）くらいが平均的トータルレスポンスと推定される。しかし、授業中実際に何本のプロセスが同時に走るかは本校においても調査する必要があると考えている。

表-3は最適化及びスケジューリングの違いによりレスポンスがどうなっているか比較するためにA（オブテマイズ無し，ヒューリスティック方式）の場合を100％として計算したものである。MV/2000 DCのOSは特に指定しない限りヒューリスティック（Heuristic）方式のスケジューリングによって処理される。この方式は，コンパイル，リンクなどCPUを集中的に使うプロセスよりも会話型プロセスの方を優先するのでユーザプロセスにとって適した方式とされている。それに対しラウンド・ロビン（Round robin：Cの場合）は，スイッチに登録したタスクをサイクリックに実行していく方式である。紙面の関係上Cの場合の生データは掲載していないが％の形で載せた。

AとB（オブテマイズの有無）で比較すると，コンパイル時Bの方が1.9～3.7倍長い，リンク時はプロセス数15以下でほとんど差がない，実行時はBの時間が短かく0.38～0.6倍，各ステップ単純平均で0.5倍，つまりBは半分の時間で実行が終了する。

AとCを比較すると，プロセス数5以下では大きな差はないが，10以上になるとCの方が時間がかかることがわかる。特にコンパイル時間について著しく，最大3.67倍にも達している。リンク時間はプロセス数10くらいまではほぼ同じであるがそれ以上になると時間が短くなりプロセス数24で72％であった。トータル時間の最大と最小の

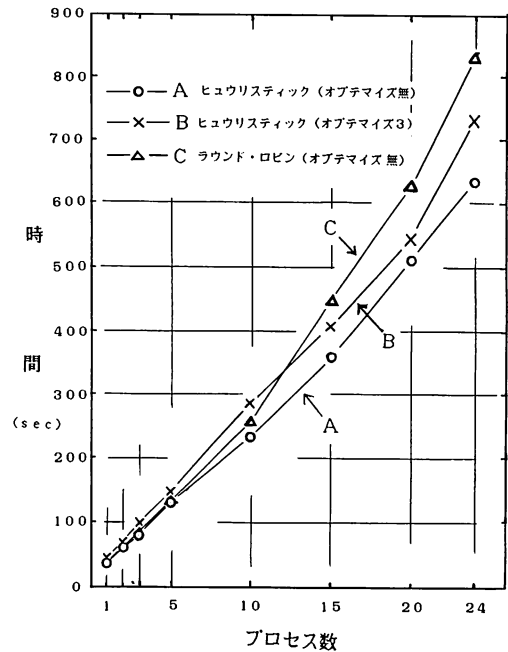


図-3 トータル時間の比較

差はAの場合プロセス数24で215 sec あるが（表-2 参照）C（ラウンド・ロビン方式）の場合，当然のことながら小さく41 secであった。

図-3は，A，B，C，3つのトータル時間（平均）を図にしたものである。Aは全プロセス数についてレスポンスが早い。Bの場合，実行時間だけを見ると約半分で早いがトータルで見るとむしろ時間が長い。従ってFÖRTRANの最適化は，あらかじめ実行形式にしたものをデータを変えて繰り返し処理する場合には時間が早く有利である

表-4 整数型四則演算速度

演算回数：10万回 単位：Sec

	MV / 2000DC			PC	PC (BASIC)
	OPT (無)	OPT (1)	OPT (3)		
空ループ	0.860	0.928	0.526	4.0	201.0
加 算	0.152	0.038	0.318	0.6	33.0
減 算	0.172	0.104	0.329	0.3	33.0
乗 算	0.415	0.395	0.655	4.3	33.0
除 算	1.038	0.970	1.199	4.8	87.3

注) OPT (無) : オプティマイズ無し
OPT (3) : オプティマイズレベル3
PC (BASIC) : パソコン (BASIC)

OPT (1) : オプティマイズレベル1
PC : パソコン (FORTRAN)

が、情報処理教育のように、実行時間が短かったり、コンパイルエラーが大きな比重を占めコンパイルから実行までを一貫して行なうことが多い場合はA(オプティマイズ無し)の方が能率が良いと言える。

C(ラウンド・ロビン)はプロセス数が10を越える付近から急激に時間が長くなっている。

3.2 四則演算速度

通常行なわれているテスト方法と同様に実施し加減乗除算10万回の時間をフォートランのISYS関数でミリsecで計測し空ループ時間を差し引いて各演算の時間を求めた。

表-4は整数型変数による演算時間である。除算がやはり大きく時間を要している。オプティ

ズ無と有(3)とでは、空ループ時間が40%近く短い。しかし四則演算のみでは、オプティマイズ無しの方が短くなっている。パソコンとの比較では、特にBASICの演算速度が極端に遅いことがわかる。

めやすとして加減乗除を単純に加えたものについて比較するとOPT(3)の演算時間は1.4倍、パソコンのFORTRAN5.6倍、BASICでは104.7倍であった。

表-5は単精度実数型変数及び倍精度実数型変数についての結果である。表-4と比べ加減算は時間が増しているが、乗除算は実数型の方が早かった。

オプティマイズによる演算速度はレベルが高い(3)の方が時間が長いことがわかった(四則全体の単純

表-5 実数型四則演算速度

演算回数：10万回 単位：Sec

	実 数					倍精度実数				
	MV / 2000DC			PC	PC BASIC	MV / 2000DC			PC	PC BASIC
	OPT(無)	OPT(1)	OPT(3)			OPT(無)	OPT(1)	OPT(3)		
空ループ	1.266	1.179	0.750	7.7	203.0	1.517	1.400	0.787	8.6	208.0
加 算	0.318	0.342	0.370	4.6	39.0	0.375	0.377	0.422	4.3	42.0
減 算	0.257	0.343	0.366	4.3	41.0	0.277	0.392	0.444	4.6	46.0
乗 算	0.321	0.488	0.416	4.8	40.0	0.507	0.738	0.582	5.1	46.0
除 算	0.814	0.900	0.941	5.9	41.0	1.354	1.469	1.519	5.9	63.0

注) OPT (無) : オプティマイズ無し
OPT (3) : オプティマイズレベル3
PC (BASIC) : パソコン (BASIC)

OPT (1) : オプティマイズレベル1
PC : パソコン (FORTRAN)

表－6 関数実行速度

演算回数：10万回 単位：Sec

	実 数					倍精度実数				
	MV/2000DC			P C	P C BASIC	MV/2000DC			P C	P C BASIC
	OPT(無)	OPT(1)	OPT(3)			OPT(無)	OPT(1)	OPT(3)		
S I N	3.750	3.755	3.967	46.0	67.0	11.864	11.916	12.216	46.3	84.0
E X P	3.862	3.946	4.157	34.0	63.0	13.498	13.615	13.911	36.0	79.0
ベキ乗	15.225	15.247	15.418	53.7	108.0	30.022	30.060	30.337	58.7	123.0
S Q R T	6.549	6.632	6.808	12.3	37.0	9.394	9.329	9.599	12.3	53.0

注) OPT(無) : オプティマイズ無し
OPT(3) : オプティマイズレベル3
P C (BASIC) : パソコン (BASIC)
OPT(1) : オプティマイズレベル1
P C : パソコン (FORTRAN)

比較で1.18倍)ただし、逆に空ループ時間は大幅に少ない(0.59倍)ので実際の実行時間として見る時はオプティマイズレベルの高い方が時間が短くなる。

倍精度(̄OPT 無)は単精度(̄OPT 無)の約1.46倍(加減乗除)となっている。単精度によるパソコンとの比較では、FORTRAN 11.5倍、BASIC 94.5倍であった。

3.3 関数実行速度

表－6は関数実行速度を示す。̄OPT(無)を基準に比較すると、̄OPT(3)は1.033倍、パソコンのFORTRAN 4.97倍、BASIC 9.36倍である。また、単精度(̄OPT 無)と倍精度(̄OPT 無)では2.204倍であった(四則の場合1.5倍)

4. お わ り に

電子計算機は日進月歩で発達しており性能の向上には見覚えましいものがある。長い間使われてきた旧電算機(H-8250)が、その末期においてはパーソナルコンピュータに主座を譲ったのもやむを得ないことであった、しかし、そのパーソナルコンピュータもまた、新電算機による学内ネットワークの実現によって端末として使うことが多くなるなど、大きく変わろうとしている。筆者等は新システムの利用や運用を簡便にするためのマクロ・コマンドを開発し、ほぼ目的の効果を上げることができたが、まだ開発しなければならないマクロは多数あり今後の課題としている。ベンチマークテストについては多数のデータを集積できたので今後の運用の基礎資料にしたいと考えてい

る。

最後に、本報及び電算機システムの運用にあたり、貴重なアドバイスを下さいました日本データゼネラル株式会社 丸山博人氏に感謝申し上げます。

参 考 文 献

- 1) 中津正志, 宇野克志, 林雄二: 苫小牧高専における新電算機システムについて, 情報処理教育研究発表会論文集, 第7号, p20 (1987).
- 2) 小宮山淑方, 浦井茂雄, 村本健一郎, 釣健孝: FACOM 320 Eの使い勝手, 情報処理教育研究発表会論文集, 第7号, p34 (1987).
- 3) 室賀進也: TSS 計算機システムにおける同種プロセス同時走行数の測定, 情報処理教育研究発表会論文集, 第7号, p32 (1987)
- 4) 木戸能史, 安達和年: 情報処理教育システムの新構想について, 情報処理教育研究発表会論文集, 第5号, p81 (1985).
- 5) AOS, AOS/VS, 解説書, CLI 編, 日本データゼネラル株式会社.
- 6) AOS/VS 解説書, システム管理者編, 日本データゼネラル株式会社.
- 7) FORTRAN 77 解説書, 日本データゼネラル株式会社.

(昭和62年11月30日受理)

資料 1

```

COMMENT *****
COMMENT *
COMMENT *      filename: F77CLE.CLI
COMMENT *
COMMENT *
COMMENT *****
PUSHB:PROMPT POP
CLASS2 IGNORE
WRITE .....**.,FORTRAN77 COMPILE LINK and EXECUTE...**
WRITE .....**.,filename *macro* : F77CLE.CLI.....**
(!NEQ.(%1%),( ))
F77/OPTIMIZE=1/E=%1%.ERR %1%
TYPE %1%.ERR
      (IULT,(ISIZE %1%.ERR) 140)
      DEL %1%.ERR
F77/LINK %1%
DEL/2-IGN %1%.LIST
LIST %1%.LIST
DEL %1%.OB
DEL %1%.ST
I %1%
LIST %1%.LIST
DEL %1%.PR
      (!NEQ,%0/XP%,/KP)
      (!NEQ,%0/LP%,/LP)
      (!NEQ,%0/BP%,/BP)
      (!NEQ,%0/CON%,/CON)
      (!NEQ,(%0/%),())
      WRITE .....WARNING : OUT PUT SWITCH ?? %0/% ?? ***
      WRITE *** /BP : BAND PRINTER.....**
      WRITE *** /LP : LASER PRINTER.....**
      WRITE *** /COM: CONSOLE DISPLAY ***
      WRITE * default:CONSOLE DISPLAY ***
      TYPE %1%.LIST
      (!ELSE):TYPE %1%.LIST
      (!END)
      (!ELSE):TYPE %1%.LIST
      (!END)
      (!ELSE):TYPE/L-0BP %1%.LIST
      (!END)
      (!ELSE):TYPE/L-0LP %1%.LIST
      (!END)
      (!ELSE):TYPE/L-0KP %1%.LIST
      (!END)
WRITE .....**.,%1%.OB %1%.PR and %1%.ST are deleted...**
WRITE .....**.,END of F77CLE.CLI.....**
      (!ELSE)
      (!EQU (!READ ***DO YOU WANT ERROR MESSAGE PRINT OUT? Y/N*** ) Y)
      TYPE/L-0BP %1%.ERR
      DEL %1%.ERR
      (!ELSE)
      DEL %1%.ERR
      (!END)
      (!END)
      (!ELSE)
      WRITE .....**.,FILE DOES NOT EXIST.....**
      WRITE .....**.,ABNORMAL END !!.....**
      (!END)
DEL %1%.LIST
POP
COMMENT *****
COMMENT *
COMMENT *      filename  NWK.CLI
COMMENT *
COMMENT *
COMMENT *      switch /S : START
COMMENT *
COMMENT *      /E : END
COMMENT *
COMMENT *****
      (!NEQ,(%1%),( ))
      (!NEQ,%0/S%,/S)
      (!NEQ,%0/E%,/E)
      WRITE .....SWITCH ERROR...**
      WRITE **./S : START...**
      WRITE **./E : END...**
      WRITE .....PLEASE INPUT A CORRECT SWITCH...**
      (!ELSE)
      COPY/D :UDD:({USERNAME}):%1% %1%
      (!NEQ,(%2%),( ))
      COPY/D :UDD:({USERNAME}):%2% %2%
      (!ELSE)
      (!END)
      DIR :UDD:({USERNAME})
      DEL/V :WORK:({USERNAME})
      (!END)
      (!ELSE)
      DIR :WORK
      CREATE/DIR ({USERNAME})
      DIR :WORK:({USERNAME})
      COPY %1% :UDD:({USERNAME}):%1%
      (!NEQ,(%2%),( ))
      COPY %2% :UDD:({USERNAME}):%2%
      (!ELSE)
      (!END)
      F/S
      (!END)
      (!ELSE)
      (!NEQ,(%0/%),())
      (!NEQ,%0/S%,/S)
      (!NEQ,%0/E%,/E)
      WRITE ..... switch ERROR.....**
      WRITE **./S : START .....**
      WRITE **./E : END.....**
      WRITE *** PLEASE INPUT A CORRECT SWITCH...**
      (!ELSE)
      DIR :UDD:({USERNAME})
      DEL/V :WORK:({USERNAME})
      (!END)
      (!ELSE)
      DIR :WORK
      CREATE/DIR ({USERNAME})
      DIR :WORK:({USERNAME})
      DIR
      (!END)
      (!ELSE)
      DIR :WORK:({USERNAME})
      DIR
      (!END)
      (!END)

```

資料 2

```

COMMENT *****
COMMENT *
COMMENT *      filename: RESPO.CLI
COMMENT *      TEST OF TSS RESPONSE TIME (TSS)
COMMENT *      switch /E : END of TEST (TSS RESPONSE)
COMMENT *      default : START of TEST(TSS RESPONSE)
COMMENT *
COMMENT *      programed by NAKATSU in 1987.9.7
COMMENT *****
PUSH: PROMPT POP
DEFACL + ONARE
CLASS2 IGNORE
VARO %IX
(IULE, (IVARO), 24)
(IUCE, (IVARO), 1)
  YARI (IOSU, 30, (IVARO))
  (INEQ, (SO/S), (I))
  (INEQ, SO/S, /S)
  (INEQ, SO/EX, /E)
  WRITE ***...switch /E : END OF TEST (TSS RESPONSE)...***
  WRITE ***... default : START OF TEST.....***
  (ELSE)
  CREATE RES_TESTPRO.DAT
  RES2.CLI
  PRINT/LP RES_TESTPRO.DAT
  F77CLE/LP RES3
  DELETE RES_TESTPRO.DAT
  DELETE RES_TESTPRO-.DAT
  CX ENABLE @CON(29,28,27,26,25,24,23,22,21,20,19,18,17,16,15,14,13,12,11,10,9,8,7,6)
  (IEND)
  (ELSE)
  CREATE RES_TESTPRO(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24).DAT
  CX DISABLE @CON(29,28,27,26,25,24,23,22,21,20,19,18,17,16,15,14,13,12,11,10,9,8,7,6)
  PAUSE 120.00
  RES1.CLI
  (IEND)
  (ELSE)
  CREATE RES_TESTPRO(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24).DAT
  CX DISABLE @CON(29,28,27,26,25,24,23,22,21,20,19,18,17,16,15,14,13,12,11,10,9,8,7,6)
  PAUSE 120.00
  RES1.CLI
  (IEND)
  (ELSE)
  WRITE ***... PLEASE KEY IN THE VARUE FROM 1 TO 24...***
  WRITE ***... PLEASE REPEAT THE MACRO OF RESPO.CLI...***
  (IEND)
  (ELSE)
  WRITE ***... PLEASE KEY IN THE VALUE FROM 1 TO 24...***
  WRITE ***... PLEASE REPEAT THE MACRO OF RESPO.CLI...***
  (IEND)
POP

```

```

C *****
C *
C *      filename RES3.F77
C *      TSS response program
C *      programmed by N.NAKATSU
C *
C *      on 16 sep 1987
C *****
INTEGER*4 TB(4,25),TW(4,25),TS(4,25),COMP(25),LINK(25)
INTEGER*4 EXEC(25),SEC(4),TOTL(25)
CCHARACTER*26 TITL(25)
CCHARACTER*70 CB3,CB4
DATA TB/100*0/,TW/100*0/,TS/100*0/,COMP/25*0/
DATA LINK/25*0/,EXEC/25*0/,SEC/4*0/
OPEN (1,FILE="RES_TESTPRO.DAT",RECFN="DS",FID="YES")
I=1
  READ(1,60,END=100) TITL(I)
  FORMAT(A26)
  DO 90 L=1,4
    READ(1,75) TB(L,I),TW(L,I),TS(L,I)
    FORMAT(I8,I2,I2,I2,I2,I2)
    SEC(L)=(TB(L,I)*60+TW(L,I))*60+TS(L,I)
  CONTINUE
  COMP(I)=SEC(2)-SEC(1)
  LINK(I)=SEC(3)-SEC(2)
  EXEC(I)=SEC(4)-SEC(3)
  TOTL(I)=SEC(4)-SEC(1)
  I=I+1
  GO TO 50
100  N=I-1
      AVC=0.0
      AVL=0.0
      AVE=0.0
      AVT=0.0
      DO 300 M=1,N
        AVC=AVC+COMP(M)
        AVL=AVL+LINK(M)
        AVE=AVE+EXEC(M)
        AVT=AVT+TOTL(M)
        WRITE(12,200) TITL(M)
        FORMAT(I8,A26)
        WRITE(12,220) COMP(M),LINK(M),EXEC(M),TOTL(M)
        FORMAT(I8,5X,"*COMP=",I4,3X,"*LINK=",I4,3X,"*EXEC=",I4,3X,"*TOTAL=",I4,1X,"sec")
      CONTINUE
      AVC=AVC/N
      AVL=AVL/N
      AVE=AVE/N
      AVT=AVT/N
      CB3="*****"
      WRITE(12,320) CB3
      FORMAT(I80,A57)
      WRITE(12,340) AVC,AVL,AVE,AVT
      FORMAT(I80," *COMP=",F5.1," *LINK=",F5.1," *EXEC=",F5.1," *TOTAL=",F5.1," sec")
C
      STOP
      END

```

