

UTS/VS・TSS機能の追加について

高橋勝彦 (三菱電機)

1. はじめに

最近におけるオペレーティング・システムに対するユーザの要望は“使い易さ”にかなりの評価の重点を置いており、従来に増してコンピュータを“単なる道具”として位置づける姿勢が強くなってきたように思われる。今後もこの傾向は続き、多くの努力が必要であろう。

TSSの使い易さについてはサービスを受ける立場と管理する立場で評価は異なり、同様にユーザの使用目的や業務内容及び個々の利用者の能力によっても評価も要望も差があるのが当然である。

一般的にTSSの使い易さを論議する場合には幾つかの評価項目がある。

- (1) 多重処理下、多端末処理時の応答時間はどうか！
 - (2) 利用者からみたコマンド等の操作性と学習性はどうか！
 - (3) 利用者としてのシステム制御のしやすさはどうか！
 - (4) サポートされているTSS機能、アプリケーションはどうか！
 - (5) 利用できるシステム資源の内容及範囲はどうか！
 - (6) 他の処理モードとの互換性とコミュニケーションはどうか！
 - (7) TSS機能、性能及びユーザ管理の柔軟性と拡張性はどうか！
 - (8) 利用できる端末の種類、仕様及び使用形態はどうか！
- etc

夫々のテーマごとに多くの議論があり、UTS/VSオペレーティング・システムにおいても多くの貴重な意見が反映されて現在に到っている。

本文ではUTS/VSにおける“TSS機能、性能の管理と活用”という面に焦点を絞ってユーザ(本文では“管理者”と“利用者”に分離)に対してどのような機能を提供されているかについて紹介する。

2. 機能、性能の管理と活用の概要

UTS/VS・TSSの最大の特長はその設計思想にあり、主な特色ある項目として

- (1) 完全な互換性……バッチ、TSS、リアルタイム処理等は夫々その処理の特質を持ちながら完全に互換性を有し基本的には言語なども含めて内容に差がない

- (2) 共用プロセッサ、ライブラリ 処理モードに関係なく同一プロセッサやライブラリを同時に利用する場合は共用可能である
- (3) 応答時間の良さ 上記の共用プロセッサやダイナミック・テーブル・ドリブンによるスケジュール方式により高速応答を保证する
- (4) 自己拡張性 ロードモジュールの作成(言語によらない)やコマンド・プロセッサの作成は直ちに新規コマンドの登録となる
- (5) コントロールのしやすさ 利用者はサービスを自由に制御可能であり、システムマクロ命令により R/S 機能を自由に活用できる

などがあり、TSS の使い易さの面でも高い評価を得ているが特に幅の広いユーザのニーズに対応できる機能の拡張性や性能のコントロールに抜群の能力を発揮している。

次頁の「UTS/VS の TSS 機能、性能の管理概念図」で示される中で太枠で囲まれた部分がユーザに与えられた管理機能である。

一般の利用者に対してはこのうち

- (1) 利用者アカウント内のサービス・プロセッサ管理
- (2) システム・マクロ命令の活用 etc.

が使用可能である。

一方、サービスの管理者に対しては

- (1) システム・サービス・プロセッサの管理
- (2) R/S のユーザ・エクジットの活用
- (3) ユーザの管理
- (4) システム変数の管理 etc.

がある。

上記(2)の機能のみ R/S 生成時のみ有効で、他はシステムの稼動中にも使用できる。

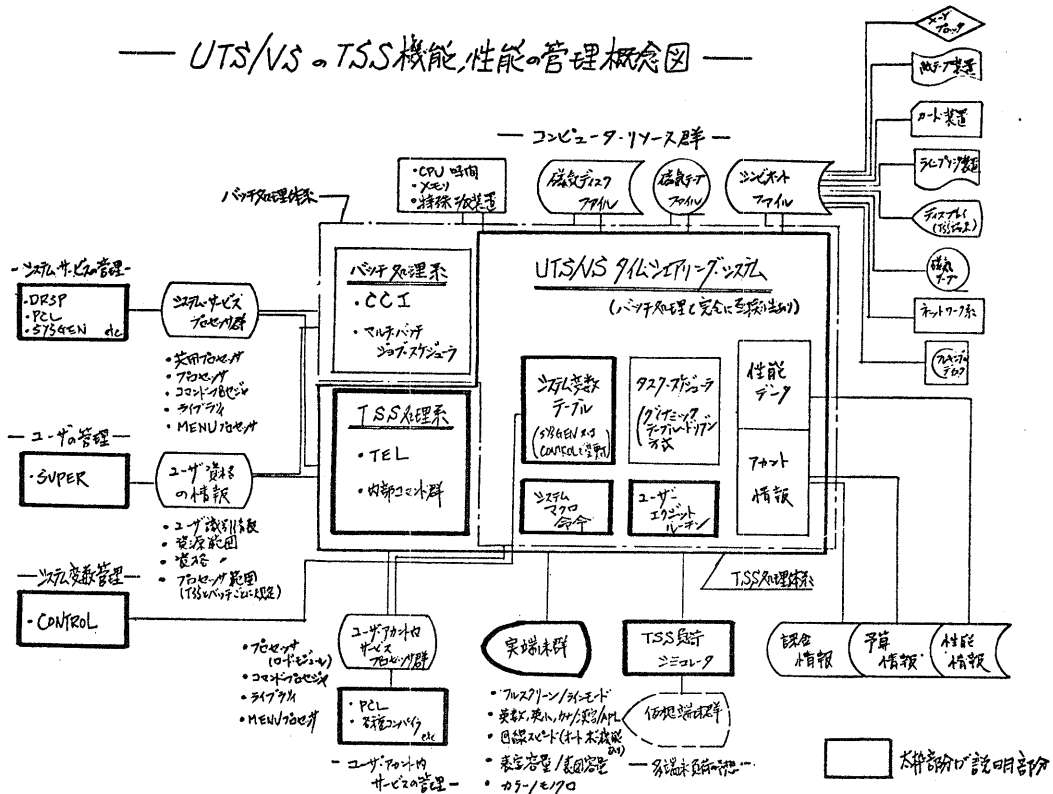
3. 一般利用者の機能

3.1 サービス・プロセッサの管理

UTS/VS の TSS では自己のアカウント内に自由にシステム提供のコマンドとは異なるコマンドを設定できる。

方法としては二つの方式が提供され、ロードモジュールを言語に関係なく作成するか(他のアカウントのモジュールのコピーや参照でもよい)又は言語形式のコマンド・ランゲージ(CIL)によって定義することにより可能となる。

—— UTS/VS. TSS 機能、性能管理概念図 ——



その他独自のライブラリ、作成やMENU画面の設定が可能であるが本文では省略する。

(1) ロードモジュール方式 ……特に特別の手段はなくロードモジュールを作成すれば手続きは完了する。(又はコピーを参照)

即ちUTS/VSのコマンド(プロセッサ)は以下の文法で定義され、システムのコマンドとの違いは“.”で続くアカウント名、パスワードの有無のみである。自己及び他人のコマンドを呼ぶ場合のみ必要によりアカウント名、パスワードを付加すればよい。

この原則は一般のファイルにも適用されている。

* 装置はデバ仮イデバデバで
論理装置名を多く用いている

<コメントの一般形式>

!コメント[.「アカウント」(.「パスワード」)] {「ソースファイル」 [{ON OVER} オブジェクト 能力] [, 「リンク」 能力] }

(21) シーファイル → M:SI
 オブジェクト → M:BX
 プリンター → M:LD

＜例＞ ! FORT Source ON Object, LP → FORTRANのコンパイル・ソース・オブジェクトは共に Source Object 形式
! PROG. Data → 自己のパート内の "PROG" を実行する。データは Data と呼ばれる。 ランタリングへ

- (2) コマンド・プロシージャ方式・・・言語形式のコマンド・ランゲージ(CSL)と一般コマンドを組合せて一連の新しいマクロコマンドをファイル上に定義することができる。
 コマンド名は"ファイル名"で呼びかたは一般形式と同一である。このCSL中には条件式、演算式、システム変数等と任意に用いられるのでバッチ処理とTSS処理のコマンドを完全に一致させることもできる。

<例>

```
1.000 !! PROC PROG
2.000 !! DCL (PROG,S,SOSU)
3.000 !! IF &SYSMODE = BATCH THEN GOTO EX
4.000 !PORT S:&PROG OVER B:&PROG
5.000 LS,BD,NS,BC
6.000 !RUN B:&PROG OVER L:&PROG
7.000 !! WRITE * PROGRAM IS ---> &PROG
8.000 !EX: EXIT
```

上記のファイル名をFCLGとすると

! FCLG. PRO を実行すると

S:PROをコンパイル

B:PROとL:PROのモジュールを自動的に作成する。

3.2 システム・マクロ命令の活用

会話形のアプリケーションを作成する上では端末制御や割り込み処理など関連するシステム・マクロ命令の利用が必須となる。

以下では特にTSS処理がらみの幾つかの命令を紹介する。利用可能言語はアセンブラとのインタフェースを持つすべての言語で、ライブラリとして定義できる。(命令の使用範囲は処理モードよりは利用者の資格による)

- (1) 割り込みとエグジット・・・ブレークによる割り込み(M:INT), トラップの割り込み(M:TRAP), アボートによる割り込み(M:XCON)をコントロール可能であり、またリターンも任意のレベルへ制御できる
- (2) 端末の制御・・・プロンプト・キャラクタ(M:PC), アクティベーション・キャラクタ(M:AC), 端末のタイプの定義(M:CT)など多くの命令が用意されている。
- (3) 回線の制御・・・すべてのTP(トランザクション・プロセッシング)のマクロ命令を用いることができる。
 回線のゲット命令とトランスペアレント・モードでの回線入出力を組合せると自己のホストをパススルーしてさらに上位ホストのTSSの端末となるサービスも可能となる。
- (4) その他・・・多くの命令の中で特色あるものを紹介する。

M:LINKロード・モジュールへのリンク(実行)とリターン

M:LDRCLロード・モジュールのロードとトランスファ

M:WAIT指定時間のウェイト

M:LDEV論理装置ストリームと物理装置の結合と属性の定義

M:JOBバッチ処理へのジョブの投入と状態の確認

4. サービス管理者の機能

サービス管理者の主要な役割りは機能の管理、パフォーマンス管理とユーザ管理である。これらの管理機能はユーザ管理を除いてOS自体が最適運営を行なうので絶対必須という意味ではない。

OSのカスタマイズや積極的な運営管理を行なう場合には重要な機能であろう。

4.1 機能の管理

(1) システム・サービスプロセッサの管理……基本的には先に述べた一般利用者と同一である。

たゞシステム関係では共用プロセッサ、ライブラリ等の管理機能(DRSPなど)があり、アッセンブラやFORTRAN等で作成したロードモジュールを登録管理できる。多重に利用するプロセッサはシェアードとしパフォーマンスを高める事が重要である。

その他一般コマンドの登録(ロードモジュールやコマンドプロセッサ)、MENUなどについては省略する。

(2) ユーザエグジツト機能……OSの解放はシステム目的に合致する様、構造やパラメータを指定するわけであるが、キメ細かい部分についてはOSに手を入れたいとの要望も出てくる。

UTS/VsではTSs関連については三箇所ユーザエグジツトを設けユーザによるカスタマイズを許している。

M:ACINIT……ログオン操作完了後コントロールが渡される。

M:ACTERM……セッション・サービス完了後コントロールが渡される。

M:AUTOLOG……利用者のログオン要求後直ちにコントロールが移る。

これらの機能を活用すると以下の様な例が可能となる。

- ① 利用者への持主へのメッセージやお知らせを出す
- ② 持主の予算管理が可能となる。
- ③ アカント・ログの内容を自由に換えられる
- ④ 特定の回線の自動ログオンを可能とする

4.2 パフォーマンス管理

UTS/Vsのスケジューラ自体は与えられたシステムパラメータ、資源の総容量やタスクの属性等をみながら最適スケジュールを自動的に行なうた

の管理の主体はシステム・パラメータと後に述べるタスク即ち利用者への管理となる。

- (1) システムパラメータの制御……システム・パラメータは“CONTROL”によって稼働中にダイナミックに可能であり、また“STAT”によって詳細なパフォーマンス・データを入手できる。
- さらに仮想端末を定義できる“TSS負荷シミュレータ”と組み合わせて用いれば、実負荷と同時処理させて多端末システムでのパフォーマンスの測定とチューニング・アップが可能である。

CONTROLプロセッサのTSS関連主要パラメータ

QUM ……最大のTSSユーザ数
QUAN ……CPUバウンドなタスクのための最大タイム・クォンタム
QMIN ……最小保証されるタイム・クォンタム
SQUAN ……主記憶に存在することを保証された最小スワップ・クォンタム
QPRI0 ……ベース・プライオリティ（ここを基準に動的に変化する。）
QXMF / QIMF ……入出力動作量でタスクを制御するパラメータ
TB / UB ……端末出力時間でタスクを制御するパラメータ

- (2) ユーザの管理……ユーザの管理は“SUPER”によって定義され、結果的にはパフォーマンスと密接な関連を持つ。

- ・ ユーザの識別と資格の定義
 - ・ 主記憶の実空間、仮想空間の既定値と最大値*
 - ・ CPU時間の既定値と最大値*
 - ・ ディスク容量の利用可能最大値
 - ・ 周辺装置の既定値と最大値* *バッチ処理に有効
- などに加えて以下の定義等も可能である。
- ・ 使用できる、できないプロセッサ名
 - ・ ログオン終了後自動コールするプロセッサ名
 - ・ ディスク・ファイルの保持期間の既定値と最大値^{dc}

5. おわりに

日本におけるTSSの実績はこの数年急速に伸び、次第に日本的な発想のもとで吸収され変化しようとしているように思われる。

使い易さという面でも今後次第に要求が高まるであろう日本語化によってまた違ったTSSが生まれてくることも予想される。

TSS機能の拡張性についてもさらに自由な形式となることを目指したい。