

# 汎用OSテスト支援システムOSTD(2) — 制御方式 —

7V-5

吉澤康文\* 佐藤孝夫\* 森山浩\*\* 橋本忠雄\*

\*(株)日立製作所 \*\*日立マイクロコンピュータエンジニアリング(株)

## 1. はじめに

汎用大型計算機システムの中核であるオペレーティング・システム(OS)の高信頼化のため開発したOSテスト支援システム(OSTD:OS Test Driver)は、会話処理形態で被テストOSの実行やテスト、デバッグ情報の収集、蓄積を可能にしたものである。これを実現するために、会話処理機能とデータの蓄積機能を持つ汎用OS(VOS3/TSS)と、複数のOSを1台の計算機で同時に実行し、かつそれらOSの実行を監視する機能を持つ仮想計算機システム(VMS:Virtual Machine System)を結合した。さらに、一台のTSS端末でOSのテスト、デバッグ作業を連続して行うために、応答性、操作性を考慮した被テストOSコンソール・シミュレーション方式と端末制御方式を開発した。ここでは、OSTDシステムの構造と、その制御方式について述べる。

## 2. OSTDシステムの構造

OSTDシステムは、OSのテスト、デバッグ作業をTSSによる会話形式で実施することを可能とする。これによりテスト情報の収集、結果の確認、不良原因の究明といった一連のテスト、デバッグ作業を一台のTSS端末を介して一貫して行うことができる。この機能を実現するため、以下に示すような構造を採用した。(図1)

- (1) 仮想計算機技術により、一台の実計算機上で被テストOSとその動作を監視する汎用OS(VOS3/TSS)を同時に実行させる。
- (2) VOS3/TSS端末で被テストOSのコンソール装置、被テストOSが動作する仮想計算機の制御コンソールをシミュレートする。
- (3) 仮想計算機の制御および被テストOSコンソールの入出力制御は、仮想計算機の制御プログラムであるVMCP(Virtual Machine Control Program)によって行われる。したがって、被テストOSの制御情報をVOS3/TSSとVMCPの間で交信する仮想計算機(VM)間の情報転送路を用意する。

OSTDシステムは、図2に示すようにVM間交信伝送路を中心に、VMCPとVOS3/TSSの双方にプログラムコンポーネントを組み込んでいる。各コンポーネント間のインターフェースは、それぞれの階層単位にプロトコルを定めることにより簡単化し開発効率を上げた。

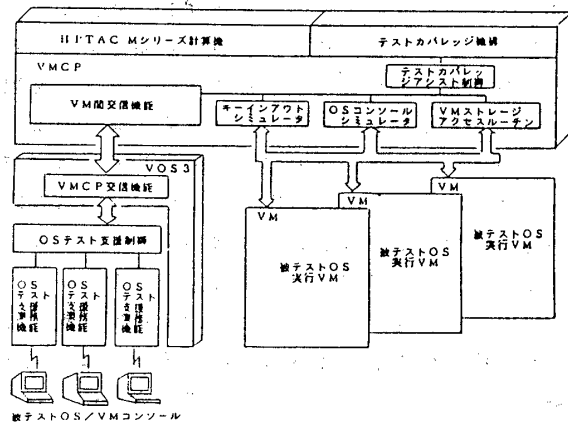


図1 OSTDシステムの構造

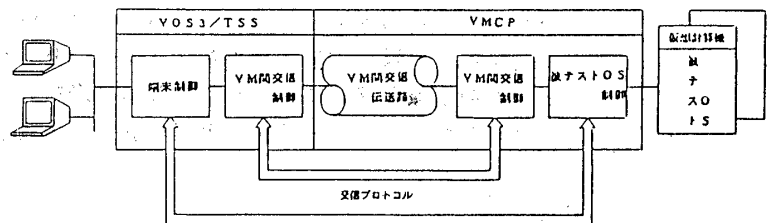


図2 OSTDシステム プログラム構造

Operating System Test Driver (2) - Control Logic -

Yasufumi YOSHIKAWA, Takao SATOH, Hiroshi MORIYAMA, Tadao HASHIMOTO  
HITACHI, Ltd.

### 3. 制御方式

次にOSTDシステムの制御方式について述べる。

- (a) VM間交信制御方式：VOS3/TSSとVMCPの間では以下の情報が転送される。

- ・被テストOSコンソール情報
- ・仮想計算機制御情報
- ・各種テスト、デバッグ情報

これらの情報の交信は頻繁に行われるが、特にOSが出力するメッセージやオペレータが入力する応答メッセージなどの被テストOS

コンソール情報の転送が多い。被テストOSコンソール情報とは、OSがコンソールに出力するメッ

ッセージとオペレータが入力するOSコマンドおよびメッセージに対する応答である。従ってVMCPからVOS3/TSSに対するデータ転送量は、逆方向の転送量よりはるかに多量であり被テストOSが連続して実行されるとデータバッファが短時間のうちに不足状態となる。このため各被テストOSが動作する仮想計算機に許す最大の交信バッファの容量を定め、このバッファが一杯になったらその仮想計算機の実行を中断する方式を採用した。

- (b) 非同期コンソール入出力シミュレーション方式：被テストOSがコンソールに対して発行した入出力要求は、VMCPによりインタセプトされ、VOS3/TSSに転送されTSS端末によりシミュレートされる。しかし、被テストOSが発生するメッセージ出力量は膨大であり、この出力要求を一つ一つVOS3/TSSによりシミュレーション処理を行うと、多大なCPU能力が必要となる。このシミュレーションを効率的に行うため、個々の出力要求に対してはTSS端末により実際にシミュレーションされる前に入出力完了を報告してしまう方式を採用した。さらに、仮想計算機間の交信オーバーヘッドを抑えるため、複数のメッセージ出力要求をまとめて転送する方式を採用することとした。しかし、通常のブロック転送方式のように、ある一定の出力要求が発生するまで転送を保留しておくメッセージ出力が遅れてしまう。そこで、VOS3/TSSがアイドル状態となった場合（メッセージ出力可能状態）に、メッセージ転送要求を発行し、この要求に対し保留されている複数のメッセージを一度に転送し、TSS画面に表示する方式を採用した。しかしコンソール処理系のテストを行う場合は、一つ一つのメッセージの出力を確認しながらテストを実行する必要がある。このため、ユーザの指定によりコンソール出力要求を端末に同期出力する方式と非同期に出力する方式を選択可能とした。

- (c) 画面制御方式：OSTDシステムでは1台のTSS端末で、被テストOSコンソール、仮想計算機制御コンソール、およびテスト、デバッグ作業を行うためのOSTD端末を実現する必要がある。ここで使用するTSS端末は、HITAC Mシリーズ計算機のコンソール装置と同じ性能であるため、被テストOSは一般にコンソールをシミュレートするためには全画面領域を必要とする。したがって、これら3つの情報を同時に表示することは出来ない。このため、複数の画面イメージを内部的に保持し、図3に示すようにキー操作によりそれぞれの画面を切り換える方式を採用した。

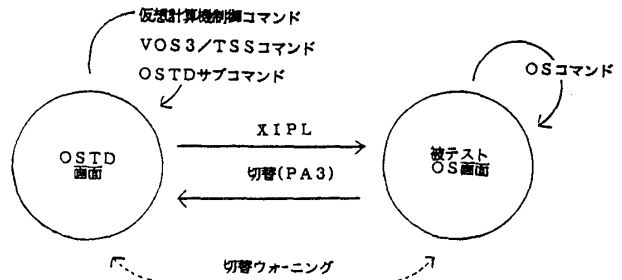


図3 OSTD画面切替方式

### 4. おわりに

ここで示した仮想計算機間交信方式、端末制御方式は、OSTDシステム構築のための要素技術である。OSTDシステムは、これら要素技術により基本システムを構築し、これに各種テスト、デバッグ機能を付加した構造をとっている。この構造をとることにより会話性が良くテスト機能の豊富なシステムを構築することができた。

### 参考文献

David Abramson : "Hardware Support for Program Debuggers in a Paged Virtual Memory"

ACM SIGARCH CAN (1983-6) p.8-16