

汎用OSテスト支援システムOSTD(1)

7V-4

— 概要 —

久保隆重 吉澤康文 佐藤孝夫 井村淳一 宮本和靖
(株)日立製作所

1. はじめに

計算機システムの普及に伴いその信頼性に対するニーズは益々高くなっている。特に大型計算機においては、その使用範囲が広く信頼性の低下は社会問題に及ぶ。計算機システムの信頼性は、ハードウェアとソフトウェアにより支えられるべきものである。たとえば、ハードウェアの一部の機能が故障した場合その機能がシステムの急所で無い限りソフトウェアによって機能の切り離しが行なわれ、システムの運転が継続される。現在、この種の耐故障性強化技術は一般化している。同様な技法はソフトウェアの構造のなかにもある。しかしながら、この種の耐故障性強化技術を全面的に適用できない部分がいくつか存在する。その良い例がオペレーティング・システム(OS)の中核部分である。つまり、計算機システムの中核にソフトウェア不良が存在していると、システム異常停止の発生確率が高くなり信頼性が低下してしまう。高品質なソフトウェアを効率的に開発するためにはプログラミング環境が整備されていなければならない。一般の応用プログラムに対しては、ドキュメントの作成から、プログラムの作成、修正およびテストといった作業を一貫しておこなうワークステーション環境が実現されている。これに対し、大型汎用OSは数百万ステップからなる膨大なソフトウェアにより構成されており、かつそれらのほとんどが特権モードで実行されるため、OSのテストは実計算機または仮想計算機を独占して行わなければならなかった。そのため一般の汎用プログラムのようなプログラム環境が整備されておらず大型汎用OSの開発には膨大な計算機リソースを必要とし、その開発効率は必ずしも良くなかった。このため、OSの開発作業を一貫して行い、かつ高度なテスト環境を提供しテスト・デバッグ工程を省力化、効率化するOSテスト支援システム(OSTD:OS Test Driver)を開発した。

2. OSTDの概要

従来OSの開発では、プログラムの作成、修正、コンパイルといった作業は、汎用OSによって行なわれるが、そのテストは、特権モードでの実行比率が高いため実計算機または仮想計算機を独占して行なっていた。このため、以下のような問題点があり、テスト、デバッグの効率低下を招いていた。

- (1) OSの開発システムとテストシステムが別のシステムであるため作業効率が悪い。
- (2) OSのテストを行なう実計算機または仮想計算機上では、高度なテスト支援機能が実現出来ない。

これらの問題点に対しては、過去様々なテストツールが開発されているが、表1で示すような問題点があった。OSTDは、既存のOSテストツールのこのような問題点を克服し、特権プログラム比率が高いOSのテスト・デバッグに必要な諸機能を体系的に提供するシステムである。

OSTDは、従来独立のシステムであったOS開発用システムであるVOS3/TSSと、OSのテスト用システムである仮想計算機システム(VMS)を有機的に結合した構造(図1)をとっている。この結合を実現するため、テスト制御用仮想計算機

表1 OSテスト・ツールの機能概要

項番	ツール名 主な機能	VOS3	VMS	デバッグ 支援
		TEST		
1	テストデータの作成	●	△	○
2	データの書込み	○	○	○
3	中断点、監視領域	○	○	●
4	メモリ内容の表示	○	○	○
5	記号定義・マップ	○	X	△
6	実行トレース	X	△	○
7	テスト手順の作成	△	X	○
8	テスト充分性の評価	X	X	X
9	CPUパネル模擬	X	●	X
10	特権プログラムへの適用	X	●	●

Operating System Test Driver (1) - Overview -

Takashige KUBO, Yasufumi YOSHIKAWA, Takao SATOH, Jun'ichi IMURA, Kazuyasu MIYAMOTO
HITACHI, Ltd.

(VM) と被テストVM間のVM間交信機能、被テストOSコンソールとOSコンソールのシミュレーション機構などを開発した。さらに高度なOSのテスト、デバッグ作業のために、表2に示す機能を盛り込んだ。以上により次の機能を持たせることができた。

- (a) OS設計者用ワークベンチ(ステーション)の提供: 会話型でOSのテストとデバッグを1台の端末で実行可能にする。
- (b) OS(特権プログラム)のテスト: 仮想計算機(VMS)機能を用いて複数のOSを同時に走行させ、被テストOSの監視、実行制御を可能とする。
- (c) テスト・デバッグ情報のオンライン収集と処理: TSS機能を用いてダンプ情報解析、端末操作情報、ストレージダンプ収集を可能とする。
- (d) テスト充分性評価をOSのテストに適応するためテスト・カバレッジ・アシスト(TCA)を開発し、サポートした。
- (e) 被テストOSの高度な実行制御のための拡張実行制御(EEC): 多重コントロール・ブレーク・ポイント及び多重データ・ブレーク・ポイント
- (f) ユーザの操作性向上のための記号定義、記号アクセス(SYMBOLIC interface)機能
- (g) 多種、多様な端末負荷環境及び障害発生環境を実現する多端末シミュレータ(TTCS)

3. 効果

OSTD方式は、従来のテスト方式に比べ1台の端末で高度なOSのテスト、デバッグが可能となる。つまり、表3に示すように、従来のように2台のコンソール装置と端末装置でなく、1台の端末だけを用いてOSのテスト、デバッグ作業が連続的に実行可能となった。また、従来OSのテストは、計算機室で行なっていたが、OSTDにより、設計者の机上で行なうことができる。さらに、テストの充分性の定量的評価及び多様な端末環境が、実現可能となり、合理的で無駄のないテスト、デバッグができる。

4. おわりに

本報告では、汎用オペレーティング・システムのテスト、デバッグの作業効率向上のために開発したOSテスト支援システム(OSTD)の概要を示した。OSTDシステムは、汎用OSと仮想計算機技術を応用し、そこに各種テスト支援ツールを組み込んだシステムである。

参考文献

David Abramson: "Hardware Support for Program Debugging in a Paged Virtual Memory"
ACM SIGARCH CAN (1983-6) p.8-16

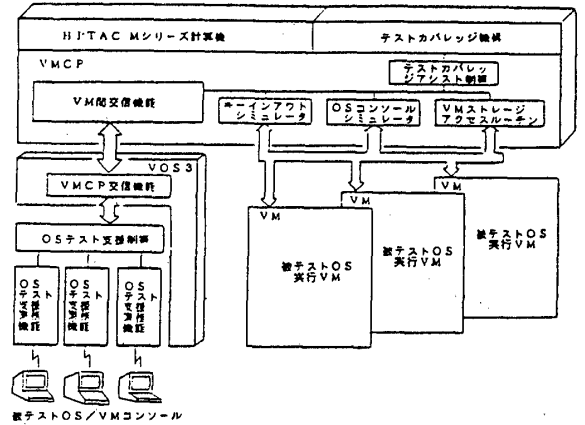


図1 OSテスト支援システムの機能概念図

表2 OSTDのテストデバッグ機能

項番	項目	機能
1	被テストOS 実行制御	中断点設定
		被テストOSへのメモリ番込み表示
2	テスト デバッグ 情報収集、解析	ストレージダンプ情報収集
		ダンプ解析
		端末オペレーション情報の保存と編集
		スプール情報の保存と編集
3	プログラム修正 テスト条件	テスト充分性評価情報の採取
		ソースプログラム修正、コンパイル
		ロードモジュールの作成、システム生成
		テストデータ/ジョブの作成
4	操作性の向上	テストの定型オペレーションの自動化
		パッチ、モジュール入れ換え
		ストレージマップ表示
		記号定義、記号アクセス
5	実環境 シミュレーション	テストデバッグオペレーションのカタログ化
		端末障害シミュレータ 端末負荷シミュレータ

表3 OSTDの効果

項目	方式	従来方式	OSTD方式
テストコンソール台数/OS		1	1
デバッグ端末台数/OS		1	(TSS端末)
テスト・デバッグ連続作業		不可	可