

マルチプロセッサ向けクロス試験環境の構築

2M-7

林 聡子、坂本 康広
富士通(株)

斎藤 誠、美馬 泰人
富士通第一通信ソフトウェア(株)

1. はじめに

近年、情報通信の多様化に伴い、要所を構成する機器も急速に多機能化する方向にあり、マルチプロセッサのシステムも少なくはない。それらに搭載されるファームウェアの開発量も膨大なものとなっており、高品質、高効率な開発を行うためにはファームウェア開発工程の中でも、最も難しいとされるシステム試験(ハードウェア・ソフトウェア結合試験)工程の改善は避けられない課題である。本課題に対して、従来のシステム試験工程の作業を、実機を用いることなくソフトウェアのみの試験段階で可能としたシステム試験シミュレーション環境を構築した。これにより、従来、実機システムの開発線表、試験作業の難しさ等に大きく依存していたシステム試験作業の前倒しが可能となり、大幅な開発工数削減を行なうことができたので、ここにその成果を発表する。

2. システムの概要

システム試験工程での主な作業は、実機搭載リアルタイムOSとのインターフェース確認、実機ハードウェアとのタイミング試験、総合ランニング試験等からなる。

本システムではこれらの試験動作を実現するにあたり、次の点にポイントを置いている。

- 1) 熟練を要しないやさしいHMIを提供。
- 2) 実機システムと同じ動きを実現。
- 3) 多人数で同時並行デバッグを実現。

これらの実現手段として本システムでは、図1に示すように、UNIXの標準装備デバッガ(sdb)をベースとしたHMI配下に、実機搭載プログラム(実機用プログラム+リアルタイムOS)と、実機ハードウェアの動作を補うプログラムをリンクしたものを一つのプロセスとして捉えることにより、実機上での動作を開発マシン上で実現している。

本システムは、次の部分より構成される。

- (1) シミュレータマネージャ部
ユーザタスクへのデバッグ及び、外部割り込み動作を行なう。
- (2) 疑似CPU部
実機搭載CPU、ハードタイマ等のハードウェア処理をソフトウェアで実現する。

(3) リアルタイムOS部

実機搭載リアルタイムOSの実アドレスアクセス部分を書き換え、そのまま搭載する。

(4) オンラインデバッガ部

ユーザタスクと同レベルで走行するデバッガ。実機稼働後はそのまま実機上に組み込む。

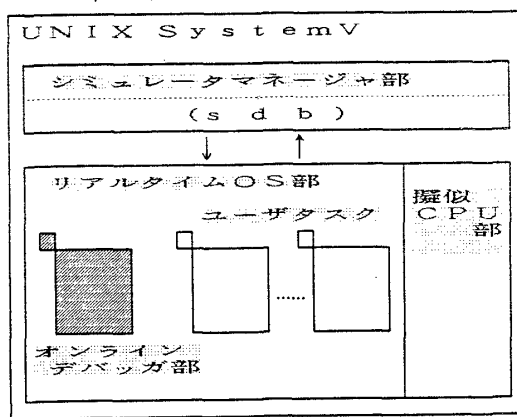


図1 シミュレータ構成図

3. シミュレーション機能

本環境では、実機と同様な次のシステム試験作業を行うことが可能である。

(1) タスク動作試験

実機用リアルタイムOS配下のユーザタスクの動作(タスクスイッチング、タスク間通信等)を実機上と同じタイミングで、シンボリックにデバッグ、確認できる。

(2) 割り込み試験

一般外部、タイムアウト、特急等の割り込みをHMIで任意の場所に発生させ、割り込み試験ができる。

(3) 状態表示・トレース

各タスクの状態、各種トレース(割り込み、システムコール等)をオンラインで表示できる。

(4) オンラインデバッグ

オンラインデバッガ部による、ユーザタスク代行、動的メモリアクセス、システムコール発行等がオンラインで行える。(実機両用)

Cross Simulation Environmnet for Multi-processor System

Satoko HAYASHI, Yasuhiro SAKAMOTO
FUJITSU LIMITED

Makoto SAITOH, Yasuto MIMA
FUJITSU COMMUNICATION SOFTWARE LIMITED

さらに、s d bの一括実行機能を応用した自動試験機能により、付加装置なしで自動ランニング試験が可能である。

4. 試験イメージ

次に実際の利用例を示す。

(1) 割り込み試験 (図 2)

- 1) 動作確認箇所にブレークポイントを設定する。
- 2) プログラムを実効する。最もレベルが高いAがまず起動される。
- 3) 被試験タスクが待ち状態になった時点で、次にレベルが高いタスクBが起動される。
- 4) ここで割り込みをかける。割り込み発生手順として、発生・解除の指示、ベクタ番号、割り込み種別を与える。この時点で、最も優先度が高い割り込みハンドラが起動される。
- 5) 割り込み処理終了後、3)と同様の手順で割り込み解除を行なう。
- 6) 割り込みハンドラよりのメッセージ受信で待ちが解除された後、再び被試験タスクAが走行する。

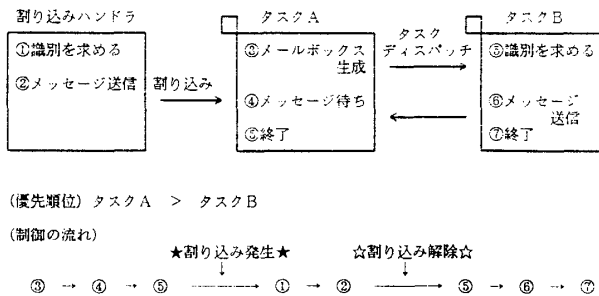


図 2 割り込み試験

(2) オンラインデバッグ併用試験 (図 3)

- 1) 動作確認箇所にブレークポイントを設定する。
- 2) プログラムを実効する。最もレベルが高いAがまず起動される。
- 3) 被試験タスクが待ち状態になった時点で、次に存在しないタスクの代わりにオンラインデバッガaが起動される。
- 4) 別のオンラインデバッガbでタスクAがアクセスしたエリアのダンプ、現在の各タスクの状態を別画面にて確認する。
- 5) オンラインデバッガaよりHMIでタスクAにメッセージ送信する。
- 6) メッセージを受信したタスクAが再び走行する。

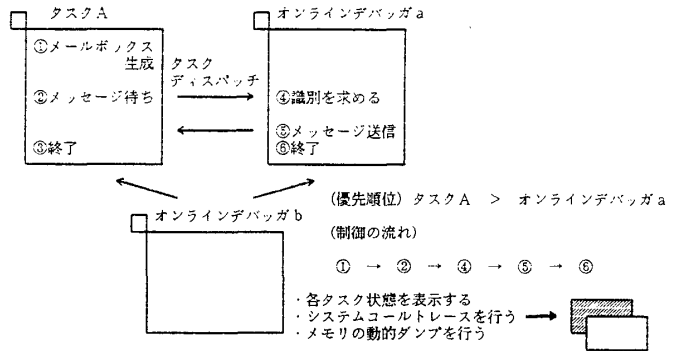


図 3 オンラインデバッグ機能併用試験

5. 評価

現在、本システムは当社製WSであるA-50 (UNIX System V) 上で稼働しており、主として通信制御関連製品のファームウェア開発に使用している。システム試験工程で最も問題発生率が高い部分の確認を、実機システムを用いることなくかつ、多人数での同時並行実行という形で可能にしたことにより、重大障害の早期発見・対処が可能になった。その結果、システム試験工程での障害発生を激減させることに成功した。(図 5) これまで、シミュレーションは超大型コンピュータの応用分野であったが、UNIXの持つ基本機能をうまく利用したことにより、WSクラスのマシンでも十分な効果を上げられることが明らかになった。今後は、この方向がファームウェア開発の一つの解答になることを強く確信している。

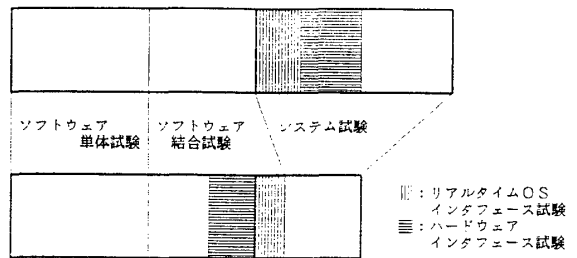


図 5 試験工程の短縮

6. おわりに

本システムは現状のところ、マルチプロセッサ/1システムの形態までの動作シミュレーションが可能であるが、さらなるシステムの多様化・複雑化に対応していくために、マルチプロセッサ/複数システムの形態を現在試行中である。また、基本的にモジュラ化が可能である疑似CPU部、リアルタイムOS部の多種類サポートが今後の課題である。

本システムの構築に当たって、御指導いただいた関係各位に深く感謝致します。