

7E-9

組み込み型分散システム用デバッガ*

植木 克彦・安田 剛・横澤 知嘉子†

株式会社東芝 研究開発センター システム・ソフトウェア生産技術研究所‡

1 はじめに

最近の組み込みシステムは、マルチタスク・リアルタイム OS を用いてアプリケーション・プログラムを制御するようになってきている。さらに、アプリケーション・プログラムの規模の拡大にともない、一つのマイクロプロセッサでは処理が困難になり、複数のマイクロプロセッサを用いた分散型のシステム構成も増えつつある。また、クロスソフトウェア開発では、開発対象である分散システムのソフトウェアを遠隔からデバッグするためのリモートデバッグ環境が必要不可欠である。これに対して、我々は、分散システムのリモートデバッグ環境の実現とデバッグ手法の評価のために、EWS(ワークステーション)をホストマシンとしてターゲットシステムをデバッグできるリモートデバッグシステム「MTD(Multi Target Debugger)(仮称)」を作成している。

本稿では、この MTD の特徴と設計について報告する。

2 分散システム用デバッガに望まれる機能および性質

分散システムをデバッグする場合の主な問題点として、以下の2点が挙げられる [1]。

- 分散システムの動作の非再現性
- タスクの相互関係の理解の困難さ

動作の非再現性のために、テスト時に発見したバグをデバッグ時に再現できないなどの問題が生じている。この問題に対しては、実行時のイベントの履歴を記録してこれを基に動作を再現する、リプレイ機能が有効である。

タスクの相互関係の理解の困難さのために、タスク間通信などに起因するバグの原因の追跡が難しくなっている。これに対しては、タスクで発生したイベント(通信のシステムコールや外部割り込みなど)の相互関係をグラフィックスで表示する機能が助けになる。

マルチタスク OS を採用したことで、新たに、タスクの排他制御の不備が原因のバグが発生するようになった。この問題に対応するためには、システムコールを使ったタスク間通信の監視機能や、タスクの状態(Run, Ready など)の表示・変更機能のような、マルチタスク OS に対応した機能が必要と思われる。

以上より、組み込み型分散システム用デバッガは以下の機能を持つことが望ましいと考えられる。

- リプレイ機能
- イベント履歴の表示機能
- マルチタスク OS 対応機能

この他にも、快適な操作性実現のための GUI を持つこと、ターゲットシステムの資源をできる限り使わず負担を軽くするための、コンパクトなデバッガリモート部を用意すること、様々な OS・マイクロプロセッサに対応できる、高い移植性があること、など、デバッガに期待される性質がいくつかある。

3 MTD の特徴

この節では、MTD に採用した特徴的なデバッグ機能を紹介する。

3.1 MTD の構成とリモートデバッグ環境

MTD は、ターゲットシステム上にあつて基本的なデバッグ操作を行なうデバッガリモート部、ホストマシン上にあつてデバッガリモート部を制御し、ターゲットシステムを調べてデバッグ情報の収集を行なうデバッガ基本部、ユーザが操作できるホストマシン上にあり、集めたデバッグ情報をユーザに示してユーザからの操作命令をデバッガに送るユーザインタフェース部、の3つの部分からなる。図1では、4つのマイクロプロセッサ(80386)を VMEbus で結合したターゲットシステムをデバッグする MTD の構成を示している。

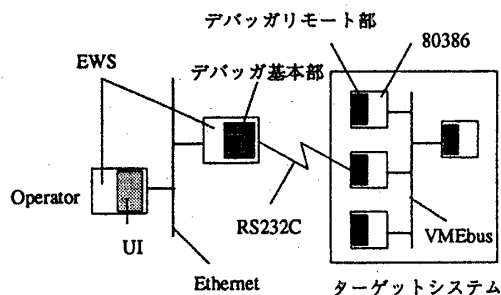


図 1: MTD の構成

ターゲットマシン上のデバッガリモート部はお互いに、デバッグ情報とデバッグ操作命令を伝えるために、アプリケーションプログラムが使う通信回線を使って通信を行なっている。これにより、ホストマシンをいずれか一つのターゲットマシンに接続することで、分散型システムのリモートデバッグ環境の構築が行なえ、ターゲットマシン毎にホストマシンと物理的な通信回線を用意する必要がなくなる。

3.2 イベント履歴を使った機能

分散システムは動作の把握が困難であり、動作をトレースするのも容易ではない。

そこで、プログラムの実行時に各タスク毎に記録したイベント履歴の情報を用いて、各イベントの相対的な順序関係をグラフィックスで表示するコンカレンシーマップ表示機能、

* A Debugger for Embedded Distributed Systems

† Katsuhiko UEKI, Takeshi YASUDA, Chikako YOKOZAWA

‡ R & D Center, Systems & Software Engineering lab., Toshiba Corp.

イベントの発生順序が前に記録したイベント履歴と同じになるように分散システムの動作を制御することで再実行時に記録した動作を再現するリプレイ機能、を実現する。

その他に、デバッグに有効な機能として、プログラムの各命令の実行時に変化する内部状態の元の状態を記録しておき、この記録を使って分散システムの内部状態を戻すことで、時間が逆転してシステムの状態が元に戻ってゆくような動作をさせる逆実行機能を用意する。

3.3 マルチタスク OS 対応機能

ターゲットシステム上の OS に対応して、システムコールによるタスク間通信の監視機能、TCB(Task Control Block)情報の表示機能、タスク状態 (READY・WAIT・RUN など) の変更機能を実現する。

4 MTD の設計

4.1 イベント履歴の記録・表示機能

ユーザが定義したイベント (システムコール・割り込み、など) が発生した場合に、発生したマシン名、イベントの種類と内容を、マシン毎に記録する。全てのイベントを記録すると膨大な量となるため、マシン間およびタスク間の通信に関するイベントなどのリプレイに最小限必要なイベントに限定して記録する。このイベント履歴を基に、マシン間およびタスク間の通信などのイベント相互の順序関係を表示するコンレンシーマップを作成する。

4.2 リプレイ機能

イベントの発生順序さえ同じなら同じ動作を行なったとみなせるような、比較的リアルタイム性の低いシステムを対象とした、分散システムのリプレイ機能を用意する。この機能の実現のためには、マシン間の通信や各マシンへの入出力のタイミングを操作するために、マシンを制御する機能と、各マシン内でプログラムの動作を再現する機能とを実現することが必要である。再実行時のイベントの発生順序が実行時に記録したイベント履歴と同じになるように分散システムを制御するために、イベント履歴に記録された次のイベントが発生するマシンを選び、順番に実行してゆく (図 2)。

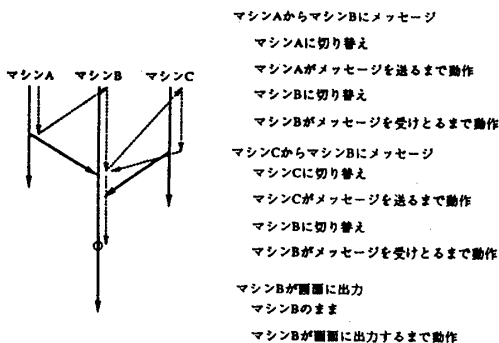


図 2: リプレイの例

4.3 マルチタスク OS 対応

今回の設計では、 μ ITRON 仕様 [2] の特定の OS に対応した機能を準備する。ターゲット上のデバッグリモート部は、優先度の高いデバッグタスクとして実現する。

タスク間通信の調査機能は、タスク間通信のシステムコールが呼び出される毎にデバッグタスクに制御を移して処理を行なえるようにすることで実現する。

タスク状態の変更機能は、デバッグタスクが発行するシステムコールによりタスク状態を変更することで実現する。

TCB 情報の表示機能は、デバッグタスクが直接 TCB を見ることで実現する。この OS ではタスクの生成が行なわれず (μ ITRON の仕様)、TCB の位置はコンパイル時に確定する (この OS の仕様) ため、デバッグタスクはコンパイル時に TCB の位置を知ることができる。

4.4 その他

逆実行機能を実現するには、一命令実行する毎に変化した内部状態を記録しなくてはならない (この記録を、ここでは実行履歴と呼ぶ)。この記録処理は、リアルタイムシステムの動作に対するオーバーヘッドが大きい。そこで、通常は実行履歴を記録せず、逆実行が必要となった時にイベント履歴を基にリプレイして実行履歴を記録する方式を採用する。

分散システムのリモートデバッグ環境実現のためには、ターゲットシステム上にデバッグ用通信網が必要であるが、デバッグ用通信網の設計法については前回の発表 [3] で触れており、ここでは述べない。また、ユーザインタフェースは現在仕様を検討中である。

5 効果

リプレイ機能によって、サイクリカルデバッグ手法によるデバッグが容易となり、テスト時には分散システムの動作への影響を考慮するため行なえなかった、動作の詳細な記録や調査が行なえるようになる。

通常デバッグ時には、結果から原因を探る思考法に従って、不正な動作を発見した場所からプログラムの動作をさかのぼって調べて、バグの原因を探しており、そのためにサイクリカルデバッグ手法によって徐々にシステムの状態の前に戻しながら調査している。このため、プログラムの状態を前の状態に戻すために再実行しているが、プログラムを逆向きに実行する逆実行機能があれば、結果から原因まで一度の逆実行でたどれることも期待できる。

6 おわりに

現在、この設計に基づいて、プログラム言語 C++ を使って MTD を作成している。今後は、今回設計した機能を順次実装して、評価を行なってゆく。

参考文献

- [1] 真鍋 義文, 青柳 滋己, "分散プログラム用デバッグ ddbx-p の試作開発", 信学技報 IN91-105 (Sep.1991).
- [2] μ ITRON 仕様書 Ver.2.01.00.00, 社団法人トロン協会 1989年7月
- [3] 村上 知嘉子, 安田 剛, 植木 克彦, 中村 英夫, "分散環境におけるデバッグ方式の検討", 情報処理学会第 45 回全国大会予稿集, Mar 1992.