4F−10

分散環境におけるデバッグ方式の検討

村上知嘉子　安田剛　植木克彦　中村英夫
(株)東芝　システム・ソフトウェア技術研究所

1. はじめに

最近の組み込みシステムでは、リアルタイムOSを用いてマルチタスクのアプリケーション・プログラムを制御するようになってきている。しかし、アプリケーション・プログラムは大規模化の傾向にあり、ひとつのマイクロプロセッサでは処理が困難になっている。そこでターゲットシステムの処理能力を高めるために、複数のマイクロプロセッサを使用し、その上でアプリケーション・プログラムを分散処理するという方法がとられている。

そのため、従来から行われているクロスソフトウェア開発手法において、複数のマイクロプロセッサから構成されているターゲットシステムについてもホストマシン上からデバッグを可能にする必要がでてきている。

そこで、このようなデバッグ環境を整えるために、必要な機能について報告する。

2. 小規模システムのデバッグ環境の現状

小規模なシステムには、ROM,RAMサイズの制限や、ハードウェアの絶対的な処理能力が低いといった特有の問題がある。そのためリアルタイムOSを用いることも難しかった。これは、OSの機能自体を縮小化することと、開発するシステムに合わせて不要なOSの機能を削減できるようにしたμITRON仕様準拠OSの出現によって解決した[1]。しかし、このμITRON仕様も現在のところカーネルのみの仕様であるため、デバッグ環境についてはあまり整備されていない。また今後μITRON仕様にOSベースでの機能の追加を行うので、分散処理も可能になる[2]。このようなことから、従来はあまり存在しなかった小規模な分散処理システムがこれから増加するものと思われる。

3. 分散型デバッガの必要性

複数のマシンを接続してターゲット・システムの任意のマシン上のデバッガをホストマシンから遠隔操作する場合、ホストマシンを含め各マシンは高度な通信機能を備え、全てのマシンアドレスを把握している必要がある。EthernetやTCP/IPを用いることで、これらの環境は整えることができる。しかし、前述したとおり小規模システム特有の問題があるため、EthernetやTCP/IPのような規模の大きな通信機能を備えることはできない。その結果、階層構造を持つ小規模なターゲット・システムに対してホストマシンから遠隔操作することはできなくなる。また、異なる通信回線でマシンが接続されている場合にも高機能な通信機能を備えることはできないので、同様にホストマシンからの遠隔操作はできない。そのため、従来の大規模な通信機能を用いることができない図1のような構成のシステムにおいてデバッグをする際には、マシンごとにICEを用いたり、各マシンにパソコンを接続してメモリダンプを取って結果を解析するなどの方法が用いられている。

そこで、このようなシステムにも、ホストマシンから任意のマシンに対して遠隔操作が可能になるデバッガ(分散型デバッガと呼ぶ)の開発が必要である。

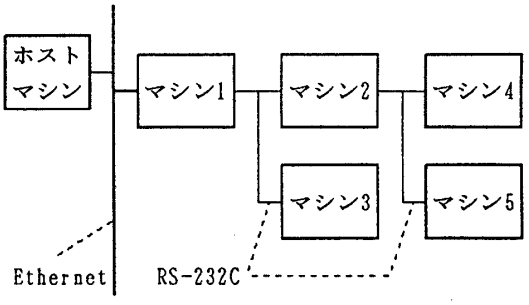


図1　システム構成例

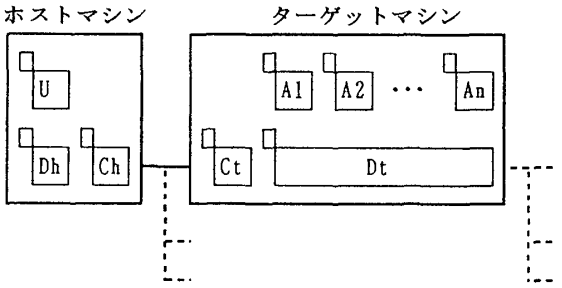
4.分散型デバッガの概要

具体的に分散型デバッガを構成する機能には、実際にデバッグを行う部分(デバッガ部)、通信部、ユーザー・インタフェース部の3つがある。分散型デバッガの適応範囲と考えられるターゲット・システムは、システムを構成しているハードウェア自身の規模が小さく、処理能力もさほどないというところに根本的問題がある。そのため、実現の際にはターゲット・システムを構成する各マシンの負荷を極力少なくすることが必要である。

そこで、ホストマシンを最大限に利用することにし、ターゲットマシン上で実行しなくてはならないことのみターゲットマシン上で実行し、それ以外のことはホストマシン上で実行させる方針をとることにした。そのため、デバッガ部と通信部はさらにホストマシン上で実行する機能とターゲット・システム上で実行する機能に分けてそれぞれのマシン上で実行する。また、ユーザー・インタフェース部に関しては、ホストマシン上のみで実行する。この構成を図2に示す。なお、それぞれの機能はOS上

An Implementation of Debug System in Distributed Environment
Chikako MURAKAMI, Takeshi YASUDA, Katsuhiko UEKI, Hideo NAKAMURA
Toshiba Corporation

でタスクとして実行する。



U ：ユーザー・インタフェース部　　A1～An ：アプリケーション
Dh ：デバッガ部(ホストマシン側)　Dt ：デバッガ部(ターゲットマシン側)
Ch ：通信部(ホストマシン側)　　Ct ：通信部(ターゲットマシン側)

図2　分散型デバッガの構成

分散型デバッガを実現する上で、処理を分散させるためマシン間の通信機能が不可欠なものになることから、ターゲットマシン上の通信機能をいかにコンパクトにし、従来行うことのできなかった通信を行えるようにするかが特に重要になる。ターゲットマシン側の通信部は小さくするため、メッセージを送受信する機能、受信したメッセージが自分宛のものかどうかを判断する機能、デバッガとのデータの受け渡しを行う機能に押さえた。また、ターゲット・システムを構成する各マシンはシステム全体の構成は認識しておらず、自分の上位(親)、下位(子)、同列(兄弟)の内1台のたかだか3台を把握しているに過ぎないものとする。この環境において、ホストマシンに入力したデバッガコマンドをマシンからマシンへと伝搬し、指定したマシンへ伝えることで任意のマシンのデバッガの遠隔操作を行う。

具体的な通信の動作は以下のようになる。図1のシステムにおいて、マシン4のデバッガをホストマシンから遠隔操作する場合について述べる。

(1)ホストマシンにマシン4を指定してコマンド入力する。
(2)ホストマシンはマシン1にメッセージを送る。
(3)マシン1の通信部は、受信したメッセージが自分宛のものでないことを判断し、マシン2(子)へメッセージを再送信する。
(4)マシン2の通信部は受信したメッセージが自分宛のものでないことを判断し、さらに接続しているマシン3(兄弟)とマシン4(子)へメッセージを再送信する。
(5)マシン3の通信部は受信したメッセージが自分宛のものでないことを判断後、さらに接続しているマシン(兄弟と子)がないことを認識するので、何もしない。
(6)マシン4の通信部は受信したメッセージが自分宛のものであることを判断し、デバッガ部にコマンドを渡す。(マシン5にメッセージの再送信はしない。)
(7)マシン4のデバッガ部は動作終了後、その結果をマシン4の通信部へ渡す。通信部はデバッガ部のコマンド実行終了のメッセージをホストマシン宛でマシン2(親)送信する。
(8)マシン2からマシン1を経てホストマシンまで、接続している親マシンへ順番に各マシンの通信部がメッセージを再送信していく。

5. 分散型デバッガの効果

4章で述べたような方法により、通信機能の小型化を行い実現した分散型デバッガの効果として、次のことが期待できる。

(1)小規模な階層化システムにも適応可能
ターゲットマシン側の通信部を小さくし、各マシンが全てのマシンアドレスを把握しなくてもマシン間で通信が行える。そのため、メモリ容量の少ないマシンで階層的に構成したターゲットシステムにも適応できる。
(2)異なる通信回線で接続したシステムに適応可能
送信側は接続していると判断できるマシンには一方的にメッセージを送信し、メッセージを受信した側で解釈する方式をとることによって、階層構造の中に異なる通信回線が使用されていても適応できる。
(3)ユーザー・インタフェースの充実
ホストマシンの豊富な資源をターゲットシステムに依存すること無く使用できるので、ユーザー・インタフェースを充実させることができる。
(4)システム完成後のデバッグ環境再構築が容易
マシンアドレスを把握しなくとも、マシン間で通信が行えることから、ターゲット側のデバッガ部、通信部を残しておくことで、システム完成後にもホストマシンをターゲットシステムの任意の箇所に接続して、分散型デバッグ環境を再び構築することができる。

6. おわりに

現在、分散型デバッガは試作中である。本稿中に述べた方法で、ホストマシンからターゲット・システムの任意のマシンに対して遠隔操作を行える環境を構築した。しかし、課題として、通信部のインプリメント法、ホストマシンとターゲットマシンの双方のデバッガ部の実現方式、どのようなユーザー・インタフェースを持たせるかといったことがある。これらの課題について検討・開発を行い、より充実したデバッグ環境の構築を今後引き続き行っていく予定である。

7. 参考文献

[1]μITRON仕様書 Ver.2.01.00.00, 社団法人トロン協会 1989年7月
[2]H.Takada and K.Sakamura, "Implementation of Inter-processor Synchronization/Communication and Design Issues of ITRON-MP", in TRON Project 1991, pp.44-pp.56, 1991.