

## SR2001におけるカーネルデバッグ

2H-4

山本伸夫\* 西門隆\* 横山法明\*\*

\*(株)日立製作所システム開発研究所 \*\*\*(株)日立製作所ソフトウェア開発本部

## 1. はじめに

一般に、カーネルのデバッグ時には、デバッグが必要とする割込み処理や入出力処理を提供するカーネル自身をデバッグする形となるため、ユーザプログラム用のデバッグと同等レベルの機能を持ったツールを使わず、1カーネルのデバッグでも多大の時間を要するという問題がある。並列計算機では、さらに、互いに絡み合った複数カーネルを同時にデバッグしなければならず、これまでの低レベルのデバッグ機能では、十分なデバッグが不可能である。

そこで、並列計算機SR2001[1]では、ユーザプログラム用デバッグと同等レベルのデバッグ機能を持ち、さらに、複数のノード上のカーネルを一つの端末から同時にデバッグすることを可能とするカーネルデバッグを開発した。

本稿では、SR2001用カーネルデバッグの概要について述べる。

## 2. 構成

SR2001のカーネルデバッグRODEO(RemQte DEbugger fOr kernel)は、デバッグ対象となるプログラムが動作する計算機(ターゲット)を、何らかの通信路で接続された別の計算機(ホスト)からリモートアクセスしてデバッグを行なう、いわゆるリモートデバッグ方式を用いている。

RODEOは図1に示す通り、ホスト計算機(UNIX<sup>注1</sup>)

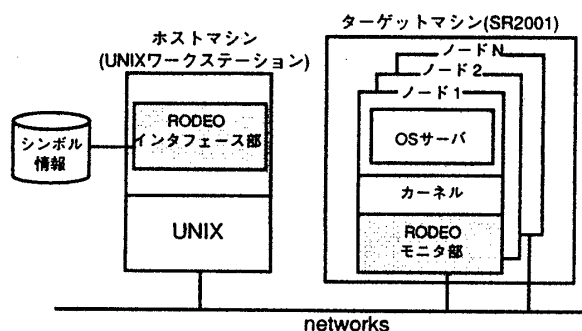


図1. RODEOのシステム構成

Kernel Debugger for SR2001  
Nobuo YAMAMOTO, Takashi NISHIKADO,  
Noriaki YOKOYAMA  
Systems Development Laboratory and Software Development  
Center, Hitachi, Ltd.

<sup>注1</sup> UNIXは、X/Open Company Limitedがライセンスしている米国ならびに他の国における登録商標です。

ワークステーション)上で動作するRODEOインタフェース部と、ターゲット計算機上で動作するRODEOモニタ部から構成される。RODEOインタフェース部は、ユーザとの対話処理を担当し、その要求をネットワークを介してRODEOモニタ部に伝えてデバッグ処理を行う。一方、RODEOモニタ部はRODEOインタフェース部からのデバッグ要求に従って、カーネルに対してデバッグ操作を行なう。

## 3. RODEOインタフェース部

RODEOインタフェース部は、ユーザとの対話処理を行い、ユーザからの要求をモニタ部へ伝達する。

本インタフェース部は安定したホスト計算機上のUNIXアプリケーションプログラムとして動作するため、デバッグ自身の動作が保障されるばかりでなく、ファイルI/O処理機能や様々なUNIXコマンドの実行機能の使用が可能である。また、プログラムのサイズに制限されることがないため、必要な機能をすべて組み込むことが可能である。

RODEOインタフェース部は、ユーザプログラムデバッグと同等レベルのデバッグ機能を持ち、さらに、複数プログラムの同時デバッグや様々なOSへの適用を考慮した汎用性を備えることに重点を置いて開発した。RODEOインタフェース部の具体的な機能を以下に示す。

## (1) シンボリックデバッグ機能

ターゲットプログラムの外部名称をシンボリックに指定する機能を提供する。また、構造体フィールド等のシンボリックな指定も可能とする。さらに、複数ノード、複数プロセスや共用ライブラリのデバッグに対応するため、複数オブジェクトのシンボル(シンボルセット)を同時に扱い、ターゲットに合わせて自動的にシンボルセットを切り替える機能を提供する。

## (2) コマンド実行・式の評価機能

組み込み、ユーザ定義、UNIXの3種類のコマンドの実行をサポートする。また、コマンドへのパラメータは、演算式とし、演算子としては、C言語と同等レベルをサポートする。また、if、while等のコマンド実行制御コマンドを組み込みでサポートする。

## (3) ユーザマクロ、変数定義機能

複数のOS、ダンプ解析等、様々な対象を同じデバッグで扱うため、強力なマクロ定義や変数定義機能を提供する。

## (4) データ、レジスタの表示・更新機能

値の簡易表示機能や、フォーマット表示機能をサ

ポートする。さらに、物理だけでなく論理の世界でのメモリやレジスタの内容の参照・更新を可能とする。

#### (5) 入出力サポート機能

コマンドファイル入力、入出力のロギング機能、ファイルへの出力変更、UNIXの"more"機能、入出力履歴による入力変更等をサポートする。

#### (6) エラー処理およびコマンドのデバッグ機能

コマンド入力の構文エラーや実行字エラーに対して、デバッガ内でエラー回復処理を行なうと共に、ユーザによる例外処理の記述を可能とする。さらに、コマンドの実行トレース機能を提供する。

#### (7) 複数ターゲットの実行制御機能

各ノード、プロセス、スレッドに対して、ブレークポイント設定やシングルステップ実行を行なう機能を提供する。

#### (8) 複数ターゲット指定機能

複数のノード上のOSに対して、一つのコマンドで同一のデバッグ操作を行なうことができる。

RODEOインタフェース部は、マシンやターゲットに依存する部分を分離した構造を持つ。従って、モニタ部とのインタラクションを行なう部分だけを変更することで、ユーザプログラムのデバッグ、ダンプ解析、システム動作モニタリング、他プロセッサ用OSのデバッグ等に簡単に適用可能であり、統一したインタフェースをユーザに提供できる。

## 4. RODEOモニタ部

モニタ部はSR2001を構成する各ノード上で動作するプログラムであり、RODEOインタフェース部に対してデバッグに必要なプリミティブ(メモリ参照・変更、レジスタ参照・変更、ブレークポイント設定・解除等)を提供する。

図2に示す通り、各モニタは、デバッグ対象のカーネルとは独立の形で実現されており、特殊ケースを除きカーネルの全動作のデバッグが可能である。また、各モニタは、そのノードに対するデバッグプリミティブの実行機能を持つ点は共通であるが、RODEOインタフェース部との関係において、以下のマスタおよびスレーブの2種類ある。

#### (1) マスタモニタ

マスタモニタは、RODEOインタフェース部からの要求を直接受け、デバッグ対象のスレーブモニタに要求を転送する役割を持つ。本モニタは、イーサネット制御デバイスが装着されている一つのノード上に配置され、RODEOインタフェース部とUDPパケットを使って通信する。マスタデバッガモニタが使用するイーサネット制御デバイスはモニタが専有するため、カーネルはその制御デバイスを使用できないが、他のイーサネット制御デバイスはカーネル

が使用可能である。

#### (2) スレーブモニタ

スレーブモニタはマスタモニタを経由してRODEOインタフェース部と通信する。RODEOインタフェース部から発行されるあらゆるノードに対するデバッグ要求は、イーサネットを介してマスタモニタに送られ、要求に示されたノード番号に基づいて対応するノードのスレーブモニタに転送される。逆に、要求に対する応答はスレーブモニタからマスタモニタに転送された後、イーサネットを介してRODEOインタフェース部に送られる。その際の、マスタモニタとスレーブモニタ間の通信は、保守・デバッグ用に設けられたネットワークを通して行なわれる。

RODEOモニタ部を以上の構造にすることにより、各ノードにイーサネットを接続することなく、しかも、簡単なプログラム構造で複数ノードの同時デバッグを可能とした。

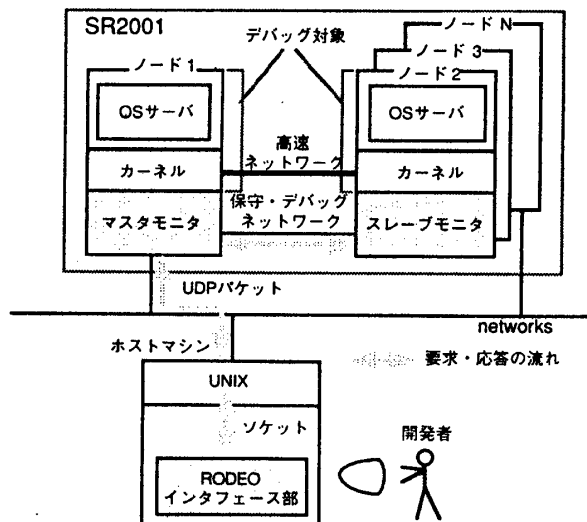


図2. インタフェース部とモニタのインタラクション

## 5. おわりに

本稿では、SR2001のカーネルデバッガについて報告した。本デバッガを用いることにより、割り込み処理やメモリ管理といったカーネルの基本パートから、OSサーバにいたるあらゆる場面のデバッグを効率的に行なうことを可能とした。

## 参考文献

- [1] 西門、他：SR2001 OSの開発コンセプト、  
情報処理学会第50回全国大会論文集 (1995)