4P-7

密結合マルチプロセッサ用OS MUSTARD の耐故障性の強化と障害処理機構

川口 浩美 **、広屋 修一 *、渡辺 明子 **、宮地 利雄 *
* 日本電気（株）、** 日本電気技術情報システム開発（株）

1. はじめに

マルチプロセッサシステムでは，構成要素数が多いため，システム全体での故障発生確率は，シングルプロセッサよりも高くなる傾向がある．しかし，一部のプロセッサが故障しても，他の部分が正常に動作しているならば，正常部分だけで，システムとしての動作が継続可能である．

MUSTARD [1] [2] は、V60 / V70 で構成される密結合型マルチプロセッサ・システムを対象にした、組込みシステム用リアルタイム OS である。**MUSTARD** は，交換機システムなどでの利用実績をもつ RX 616 [3] に対して上位互換のアプリケーション・インタフェースを提供するとともに、RX616同様、多様なハードウェア構成のもとで稼働させることができる。

本稿では，**MUSTARD** の，障害処理機構について述べる．

2. システム構成と故障モデル

MUSTARD は，組込みシステム用としてシステム・ユーザが開発する多様なハードウェア構成に対応できるように設計されている．特に，耐故障性に関しては，次のようなハードウェア機構への対応を想定している．

(a) 各プロセッサボードを，ソフトウェア制御で，共通バスからの切り離すことができる．

(b) 論理的に，1 個のプロセッサを，プロセッサ 3 台からなる三重多数決で構成し，不一致発生後の再同期化のためのソフトウェアによるローカルなハードウェア・リセットが可能．

なお，これらは，どちらも，**MUSTARD** をインストールするための不可欠な条件ではない．

後者は，地上に比べて，放射線によるプロセッサ内でのビット反転等の，回復可能な一時的故障の発生頻度が高いとされている，宇宙空間において，利用されるものである．

MUSTARD の耐故障性についての機能的特徴は，以下の通りである．

1) RX616 のシステム再開処理機能に上位互換．

2) システム起動時に，故障して動作できないプロセッサが存在する場合にも，正常なプロセッサのみで，縮小されたシステムとして運転を開始できる．

3) (b) のシステムで，いずれかのプロセッサでプロセッサの一時障害による不一致が発生しても，システム全体を止めることなく，ローカルに再同期し，処理を継続して行うことができる．

4) 頻繁に，回復可能な一時故障がおこる場合や，保守のためにプロセッサボードの脱着を行う場合を想定して，任意のプロセッサをシステムから動的に切り離したり，組入れたりするための機能を提供している．

3. 耐故障OSカーネル機能

3.1 OS 起動時のプロセッサ認識

MUSTARD では，システムの起動時に，故障して動作できないプロセッサが存在する場合にも，正常なプロセッサのみで動作することができる．これは，次の方式の採用により実現されている．

(1) 正常なプロセッサには，システム起動時に，0 から順に，論理的なプロセッサ ID が付けられるが，起動を始めてから一定時間後に起動してくるプロセッサへの論理プロセッサ ID の割付けは，マスタプロセッサ(後述)によって，停止される．

(2) プロセッサに，次の 2 種類の状態，

- 稼働状態である [**ALIVE**]
- 稼働停止状態である [**DEAD**]

を定義し，論理プロセッサ ID が割付けられると，システムを運転する **ALIVE** なプロセッサになる．論理プロセッサ ID が割付けられなかったときは，**DEAD** なプロセッサになる．

論理プロセッサ ID が割付けられなかったプロセッサは，**DEAD** を変更するシステムコールが発行されるまで，システム構成に影響を与えない．

(3) 最小の論理プロセッサ ID を持つプロセッサ(マスタプロセッサ)だけが，システムワイドな共有資源の初期化を行う．

3.2 システム動作中におけるプロセッサリセット

RX616 では，システムの障害に対する，システム再開処理機能を実現するため，システムコール **rst_sys** を提供している．**rst_sys** は，障害の重度に応じた再開処理レベルを引数として，呼ばれる．

RAS mechanism of the Multiprocessor Operating System **MUSTARD**
Shuuichi HIROYA*, Akiko WATANABE**,
Hiromi KAWAGUCHI**, Toshio MIYACHI*
* NEC Corporation, **NEC Scientific Information System Development Ltd.

RX616 は A〜D の 4 レベルの再開処理呼出し機能を提供しており，例えば，レベル A では，システム全体の再ロードからの再開レベル D では，予備系への切り換えのための CPU のコンテキストの退避と回復が行われる．これらに加えて，MUSTARD では，1 台のプロセッサだけで，CPU コンテキストの，退避と回復を行う，レベル E の再開処理を提供している．
これは，(b) の構成を想定したものである．

以下にレベル E の再開処理の概要を述べる．

ある 1 台のプロセッサで，三重化多数決の不一致が起こった場合，障害の発生したプロセッサには，レベル E の再開処理呼出しが起こる直前のCPUコンテキストの退避を行った後，ハードウェア・リセットがかかるが，このとき，システム内の他のプロセッサでは処理を継続している．

ハードウェア・リセット後に，CPU コンテキストの回復を行い，処理を続行する．

また，MUSTARD はカーネル全体を排他制御しているため，カーネル内を走行できるシステムコールは通常一つだけであるが，レベル E 再開処理システムコールは，プロセッサ毎に並列動作可能である．

システム運転中での，レベル E 再開処理の動作を示す(図1)．

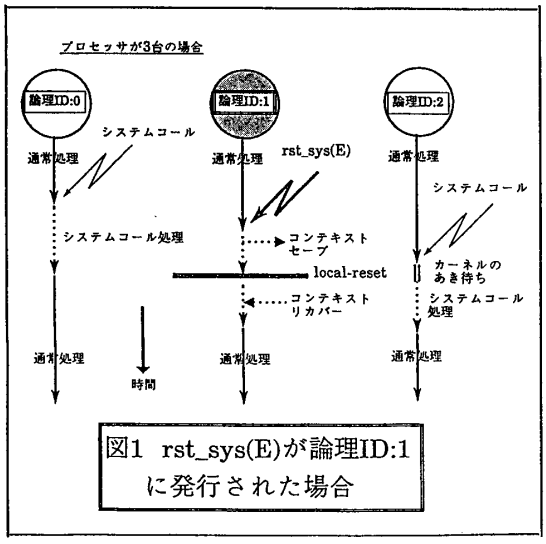

図1 rst_sys(E)が論理ID:1に発行された場合

3.3 動的なプロセッサの削除/追加

MUSTARD では，指定したプロセッサの状態を，稼働状態である ALIVE から稼働停止状態である DEAD へ，または逆に遷移させるための，次の，2 つのシステムコールを提供している．

(1) プロセッサ・アンマウント要求システムコール
umt_prc(物理プロセッサID)

(2) プロセッサ・マウント要求システムコール
mnt_prc(物理プロセッサID)

稼働状態のプロセッサに，umt_prc システムコールが発行されると，アンマウント要求をされたプロセッサはALIVEから，DEAD へ遷移し，システムから切り離される．

なお，システム運転中に，新しいプロセッサボードが装着された直後には，そのプロセッサの状態は，DEAD である．

DEAD のプロセッサは，mnt_prc システムコールが発行されることにより，ALIVE になって，リセットエントリから処理を始め，システムの処理動作に加わる．(図2)

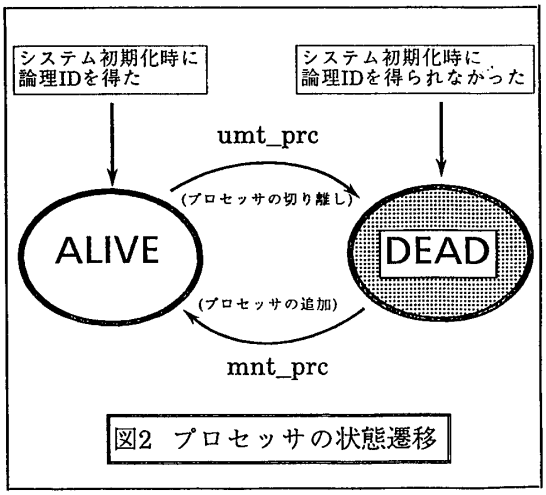

図2 プロセッサの状態遷移

umt_prc と mnt_prc により，頻繁に一時故障が発生するプロセッサボードをシステムから論理的に切り離したり，システムを稼働させたまま，プロセッサボードの保守や交換が可能となった．

この umt_prc(物理プロセッサID) と mnt_prc(物理プロセッサID) の，2 つのシステムコールは，プロセッサローカルなリセット等の，ハードウェア的なサポートがない場合も有効であり，例えば，頻繁に故障を起こすプロセッサに対し，umt_prc(物理プロセッサID) を発行すれば，そのプロセッサがシステムに影響を及ぼすことを防げる．また，前述の RX616 のレベル D の再開処理によって，予備系に切り換えたとき，予備系の切り換えだけでは，プロセッサ数に変化はないが，mnt_prc(物理プロセッサID) を発行することにより，システム内の稼働プロセッサを増やすことができる．

4. おわりに

本稿では MUSTARD の耐故障性の強化，及び障害処理機構について述べた．

これらの実装は，これまでに，ほぼ完了している．

参考文献

[1]広屋他，"種々のハードウェアに適用可能な密結合マルチプロセッサ用OS MUSTARD"，情処第39回全国大会，1989.
[2]渡辺他，"密結合マルチプロセッサ用OS MUSTARDにおけるプロセッサ間通信方式"，情処第39回全国大会，1989.
[3]下島他，"V60/V70リアルタイムOSにおけるフォールトトレラント機構"，情処第33回全国大会3V-3，1986.