

シングルチップマルチプロセッサへの Linux の適用

山本 整[†] 高田 浩和[‡]

三菱電機(株)[†] (株)ルネサステクノロジ[‡]

1. はじめに

組み込み機器の高機能化に伴い、搭載される CPU に性能向上が要求されてきている。組み込み用 CPU の場合、低消費電力も要求される。これらの要求を満たすために、クロック周波数の高速化に依存した性能向上だけではなく、複数の CPU コアを内蔵したシングルチップマルチプロセッサ (SCM: Single Chip Multi-processor) 技術が注目されている。このような背景のもと、ルネサステクノロジオリジナルの組み込み用 32 ビット RISC マイコンである M32R の SCM を試作することになった。

ソフトウェアプラットフォームは、Linux を採用した。Linux はソースコードが公開されており、組み込み OS としての採用事例が多くなってきている。既に対称型マルチプロセッサ (SMP: Symmetric Multi-processor) をサポートしていたことも Linux を採用した理由の 1 つである。

Linux システムを動作させるために、カーネルの移植だけではなく、ツールおよびライブラリも移植したが、本論文ではこのうちカーネル移植について述べる。

2. M32R アーキテクチャ

M32R ソフトマクロに対して Linux を移植し、ハードウェアおよびソフトウェアの動作を検証した後に、実チップを試作した。マルチプロセッサとして動作させるために、既存の M32R ソフトマクロに対して追加した機能のうち主なものを以下に示す。

排他制御命令 (lock/unlock)

ある CPU が lock 命令により変数を読み出してから unlock 命令を実行して変数を書き込むまでの間、システム中の他の CPU がその変数を書き換えることを禁止する。カーネル中では、スピンドックやアトミック操作の実装に使用される。

CPU 間割り込み (IPI: InterProcessor

Interrupt)

他の CPU に対して割り込みを発生させる。カーネル中では、再スケジューリングやキャッシュフラッシュなどの要求を他の CPU に対して発行する場合に使用される (CPU 間通信)。

割り込みの分配

外部端子や内蔵周辺 I/O からの割り込み入力は、全 CPU に分配され最も早く受け付けた CPU がその割り込みを処理する。

キャッシュコヒーレンシ

CPU はそれぞれ個別にキャッシュを持つ (プライベートキャッシュ) ため、データのコヒーレンシを保つ必要がある。コヒーレンシを保つために、オペランドアドレスをスヌープし、キャッシュラインの状態を管理する (MESI プロトコル)。

3. カーネル移植

Linux は既に多くの CPU アーキテクチャに対応しており、アーキテクチャ依存のコードは切り分けられている。今回「arch/m32r」および「include/asm-m32r」ディレクトリ以下のファイルを開発した。

移植は他の CPU の実装を参照しながら行う。Linux は PC/AT 互換機をメインターゲットとして開発されている OS であることから、主な部分は IA-32 (i386) を参照した。ただしアーキテクチャの差異が問題となる機能の場合は、できるだけ似ているアーキテクチャを参照する必要がある。例えば、キャッシュ制御は M32R と同じ物理アドレスキャッシュ方式である PowerPC (ppc) を参照した。

アーキテクチャ依存のコードは、プリミティブな部分が多く、それらはアセンブリ言語により最適化されている場合が多い。アセンブリ言語で記述されたコードを異なるアーキテクチャに移植する場合、機械的に 1 対 1 に置き換えることができない命令も多く、コンパイルを通すために内部動作を理解しないで書き換えてしまうと、後々デバッグで苦労する。Linux のヘッダファイルは、インライン関数を多用しているた

Porting Linux to a Single Chip Multi-processor

[†]Hitoshi YAMAMOTO Mitsubishi Electric Corp.

[‡]Hirokazu TAKATA Renesas Technology Corp.

め、不具合箇所の特定はより困難である。従って移植は、書き換え対象のファイルに関するカーネル内部の動作を理解しながら作業を進めることが望ましい。

3.1. マルチプロセッサ機能の移植

マルチプロセッサをサポートするために、同期機構、CPU間通信および起動処理を実装した。

同期機構

Linux は同期機構として「スピンロック」、「セマフォ」および「アトミック操作」を使用している。このうち「スピンロック」は UP(Uni-processor)では使用していないため新たに作成した。「セマフォ」および「アトミック操作」は UP でも使用しているが、MP(Multi-processor)に対応するための修正を行った。

リスト 1 にロック変数アドレスがレジスタ r0 の場合のスピンロックを示す。ラベル 2 から始まるコードは、スピンロックがビジーの時だけ実行される(ビジーループ)。このコードが実行される確率は低いため、通常のプログラムと異なる領域にまとめるように最適化してある。この最適化により Lmbench ベンチマークの Local Communication Latencies テストで約 10%の性能向上が見られた。

```
1: mvfc    r5, psw
   clrpsw #0x40    #割り込み禁止
   lock   r4, @r0  #変数操作を
   addi   r4, #-1  #アトミックに
   unlock r4, @r0  #行う
   mvtc   r5, psw
   bnez   r4, 2f   #ビジーの場合
                        #ビジーループへ

   .subsection 1
   .text.lock
2: ld     r4, @r0  #ビジーループ
   blez   r4, 2b
   bra    1b      #再度ロック獲得へ
   .previous
```

リスト 1: スピンロック

CPU 間通信

他の CPU へ処理を要求する場合に CPU 間通信を使用する。CPU 間通信は IPI を用いて実装した。他の CPU へ要求する処理を以下に示す。

- 再スケジューリング
- TLB フラッシュ
- キャッシュフラッシュ
- ローカルタイマ
- CPU 停止

起動処理

システム起動時に動作する CPU は 1 つである。この CPU を BSP(Bootstrap Processor)、他の CPU を AP(Application Processor)と呼ぶ。どの CPU が BSP となるかはハードウェアが決定する。起動時 AP は、スリープ状態で待機する。

BSP はハードウェアおよび Linux の初期化を実行し、初期化の最後で AP の起動を行う。BSP は順番に各 AP を起動し、最後に同期用フラグを立てることにより AP をアイドル状態(idle スレッド実行)とする。

4. おわりに

現在 M32R SCM 試作チップ上で、Linux カーネル 2.4.19 および 2.6.0 が動作している。今後、更なる性能向上を行う予定である。

本研究にご協力いただいた関係各位にお礼を申し上げます。

参考文献

- [1] Linux/M32R ホームページ、
<http://www.linux-m32r.org/>
- [2] Hirokazu Takata, Naoto Sugai, Hitoshi Yamamoto, "Porting Linux to the M32R processor", Ottawa Linux Symposium 2003
- [3] 高田浩和、作川守、坂本圭、山本整、稲岡一弘、近藤弘郁、清水徹、「Linux を搭載した M32R アーキテクチャ研究開発用プラットフォーム」、Linux Conference 2002
- [4] S.Kaneko, K.Sawai, N.Masui, K.Ishimi, T.Itou, M.Satou, H.Kondo, N.Okumura, Y.Takata, H.Takata, M.Sakugawa, T.Higuchi, S.Ohtani, K.Sakamoto, N.Ishikawa, M.Nakajima, S.Iwata, K.Hayase, S.Nakano, S.Nakazawa, O.Tomisawa, T.Shimizu, "A 600MHz single-chip multiprocessor with 4.8GB/s internal shared pipelined bus and 512kB internal memory", Digest of Technical Papers, 2003 IEEE Int'l Solid-State Circuits Conf., pp.254-255, San Francisco, USA, Feb.2003
- [5] 大規模集積システム設計教育センター (VDEC) ホームページ、
<http://www.vdec.u-tokyo.ac.jp/>
M32R ソフトマクロ(UP 版)は、オープンソース化され VDEC にて研究教育用に無償公開されている。