

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

発表内容

1. 開発の背景
2. 開発ステップ
3. CPUへの追加機能
 - 3.1 M32R-SCM試作チップ
4. カーネル移植
 - 4.1 同期機構
 - 4.2 CPU間通信
 - 4.3 起動処理
5. 現在のステータス

1. 開発の背景

- 組込み機器の高機能化に伴い、CPUの性能向上が要求されている
- 組込み用途CPUは、低消費電力でなければならない→クロックを上げられない



- 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) 技術が有効

2. 開発ステップ

- ① Linuxがマルチプロセッサ動作するために、利用しているCPU機能を調査
 - CPUへ機能追加
- ② ソフトマクロコアに対してLinuxを移植
 - カーネル移植
 - ツール/ライブラリ移植
 - ソフトウェア・パッケージの作成
- ③ 実チップに対してLinuxを移植

3. CPUへの追加機能(1)

- 排他制御命令(lock/unlock命令)
 - lockからunlockまでの間バスを占有
→同期機構の実装に利用
 - i386: xchg, "lock" prefix + add, adc, and, btc, ...
 - mips: ll, sc (load-linked, store-conditional)
- CPU間割り込み(IPI: InterProcessor Interrupt)
 - 他のCPUへ割り込み(8要因)を発生させる
→CPU間通信/起動処理に利用
 - i386: Local APIC

3. CPUへの追加機能(2)

- 割り込みの分配
 - 全CPUに割り込みを発生させて、最も早く受け付けたCPUが処理を行う
 - i386: I/O APIC + Local APIC
- キャッシュコヒーレンシ
 - 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)
 - Dcacheのみ
 - IcacheのコヒーレンシはS/W制御

3.1 M32R-SCM試作チップ

- CPUコア: 2基
- 動作周波数: CPU 600MHz, BUS 100MHz
- 内蔵SRAM: 512KB
- キャッシュ: 命令8KB、データ8KB
 - 2way セットアソシアティブ
- TLB: 命令32エントリ、データ32エントリ
 - フルアソシアティブ

4. カーネル移植

- アーキテクチャ依存のコード
 - arch/m32r, include/asm-m32r以下を開発
- 似ているアーキテクチャのコードを参照
 - キャッシュ方式、MMU構造等
- インライン関数が多用されている
 - 不具合の原因追求が困難
- マルチプロセッサに対応していないドライバもある
 - 無線/有線LANドライバ等

4.1 同期機構

- クリティカルリージョンを保護するために
利用する

– スピンロック、セマフォ、アトミック操作

```
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
```

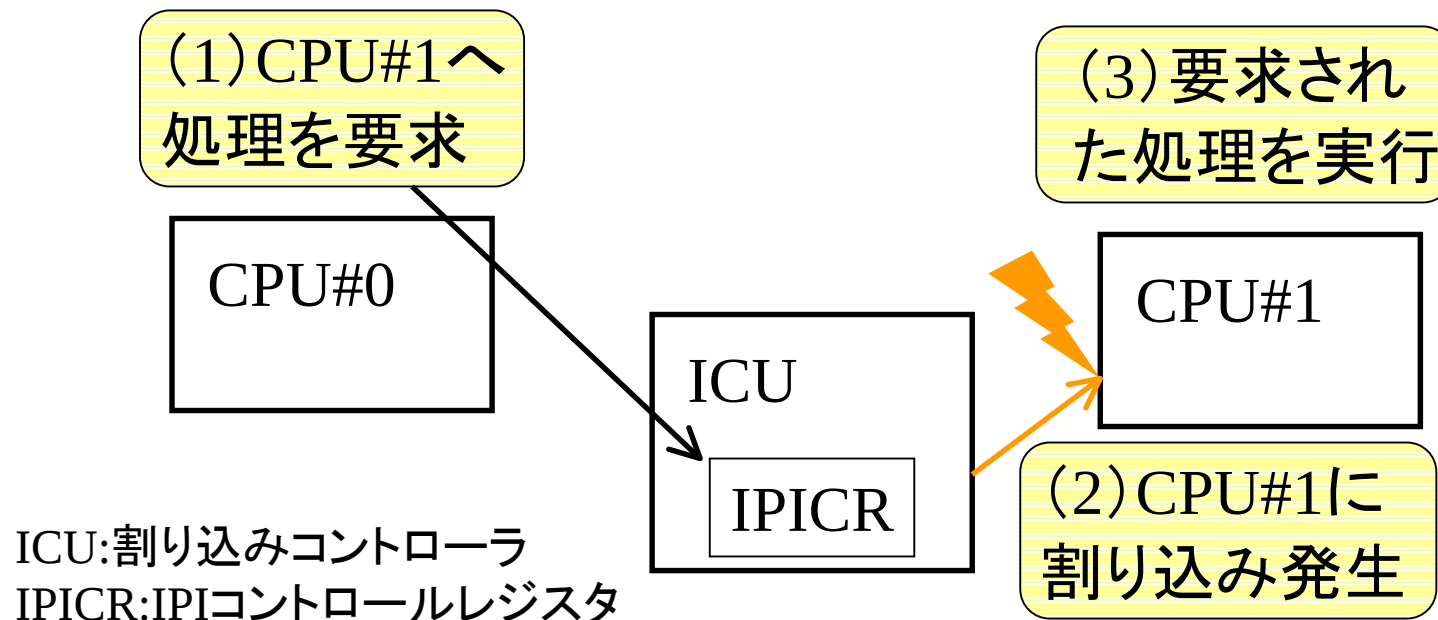
```
.subsection 1
.text.lock
```

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

ビジーループのコード
は、別セクションへ配置

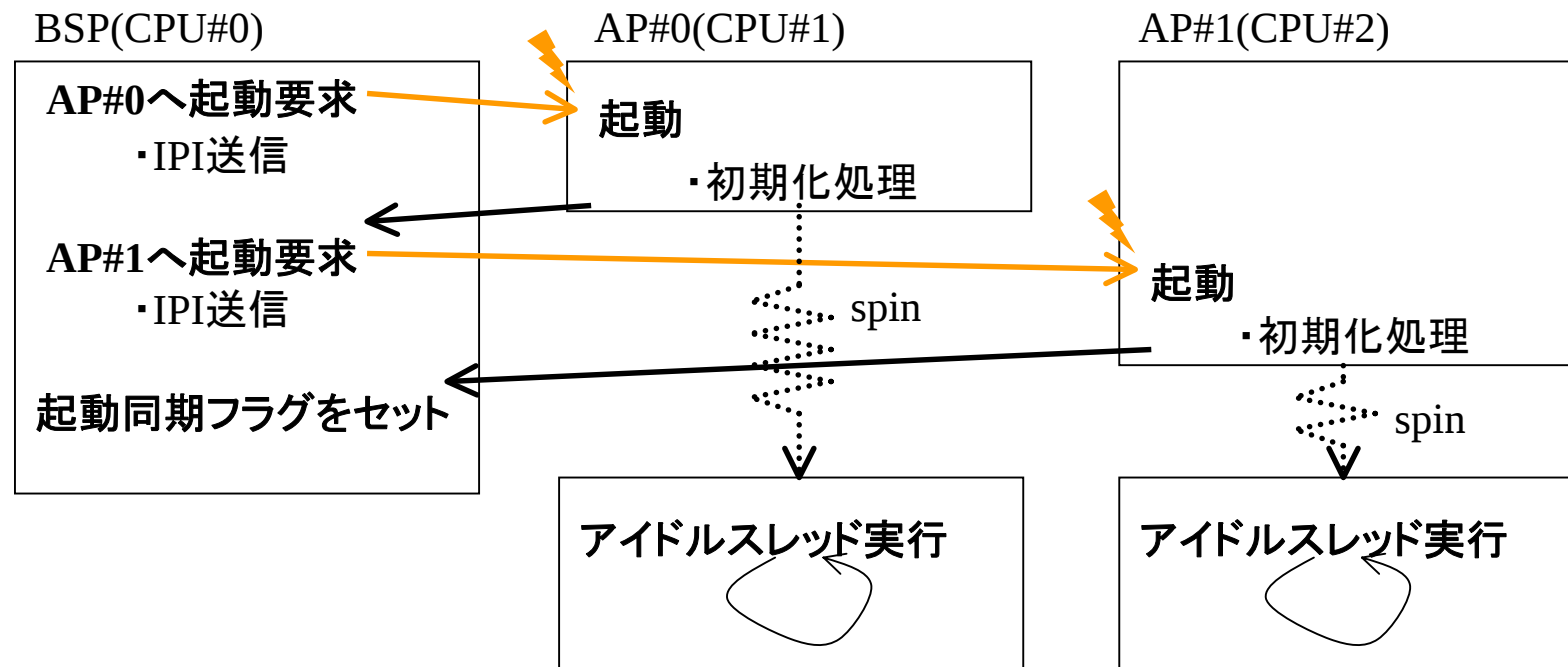
4.2 CPU間通信

- 他のCPUへ処理を要求する
 - 再スケジューリング、キャッシュフラッシュ、TLBフラッシュ、ローカルタイマ、CPU起動/停止



4.3 起動処理

- システム中の全CPUを起動する
 - BSP (Bootstrap Processor) が、IPIを使用してAP (Application Processor) を起動



3種の神器

- tags/TAGSファイル
 - vi/emacsでタグジャンプ
 - カーネルのトップディレクトリで、
"make tags/TAGS"
- cvs/cvsweb
 - バージョン管理およびカーネルのdiff
- lxr (Cross-Referencing Linux)
 - <http://lxr.linux.no/source/>

cvs/cvsweb

[Return to mmap.c CVS log](#)
[Up to \[Linux/M32R\] / linux-2.6 / mm](#)

Diff for /linux-2.6/mm/mmap.c between versions 1.1.1.7 and 1.1.1.8

version 1.1.1.7, 2003/10/27 03:12:26	version 1.1.1.8, 2003/12/19 00:18:55
Line 19	Line 19
#include <linux/hugetlb.h> #include <linux/profile.h> #include <linux/module.h>	#include <linux/hugetlb.h> #include <linux/profile.h> #include <linux/module.h> #include <linux/mount.h>
#include <asm/uaccess.h> #include <asm/pgalloc.h>	#include <asm/uaccess.h> #include <asm/pgalloc.h>
Line 474 unsigned long do_mmap_pgoff(struct file	Line 475 unsigned long do_mmap_pgoff(struct file
struct rb_node ** rb_link, * rb_parent; unsigned long charged = 0;	struct rb_node ** rb_link, * rb_parent; unsigned long charged = 0;
if (file && (file->f_op !file->f_op->mmap)) return -ENODEV;	if (file) { if (file->f_op !file->f_op->mmap) return -ENODEV;
	if ((prot & PROT_EXEC) && (file->f_vfsmnt->mnt_flags & MNT_NOEXEC)) return -EPERM; }
if (!len) return addr;	if (!len) return addr;

Diff format:

Removed from v.1.1.1.7
changed lines
Added in v.1.1.1.8

5. 現在のステータス

- Linux 2.4.19, 2.6.0が動作
 - カーネルおよびツールはLinux/M32Rプロジェクトのホームページからダウンロード可能
<http://www.linux-m32r.org/>
- M32Rソフトマクロコア
 - 大規模集積システム設計教育センター
(<http://www.vdec.u-tokyo.ac.jp/>) からUP版が研究教育用に無償公開されている
- 複数のプロジェクトにて採用

おわり