

共有メモリからデータを取得する割り込みハンドラの実行

2014/9/30

藤田将輝

1 はじめに

割り込み元 OS から共有メモリにデータを格納し、IPI を送信すると割り込み先 OS で割り込みハンドラが動作し、共有メモリからデータを取得できることを確認した。本資料ではこの流れを示す。

2 割り込みハンドラが動作するまでの流れ

割り込み元 OS で共有メモリにデータを格納してから割り込み先 OS で割り込みハンドラが動作し、共有メモリからデータを取得するまでの流れを図 1 に示し、以下で説明する。

- (1) AP2 が割り込み先 OS へ、コア 1 のベクタ表に割り込みハンドラを登録するシステムコールを発行する。
- (2) システムコールにより、割り込み先 OS がコア 1 のベクタ表に割り込みハンドラを登録する。
- (3) AP1 が割り込み元 OS へ共有メモリにデータを格納するシステムコールを発行する。
- (4) システムコールにより、割り込み元 OS が「fujita」という文字列を格納した配列を共有メモリに格納する。
- (5) AP0 が割り込み元 OS へ IPI を送信するシステムコールを発行する。
- (6) システムコールにより、割り込み元 OS がコア 0 へ IPI の送信要求を行う。
- (7) コア 0 がコア 1 へ IPI を送信する。
- (8) 割り込み先 OS の占有しているコア 1 が IPI を受信し、割り込みハンドラが動作する。
- (9) 割り込みハンドラにより、割り込み先 OS が共有メモリから「fujita」という文字列を格納した配列を取得する。

3 使用したシステムコール

3.1 割り込みハンドラの登録

割り込みハンドラの登録に使用したシステムコールについて以下に示す。

【形式】 `asmlinkage int request_ipi_irq(int vector)`

【引数】 `int vector`: ベクタ番号

【戻り値】 成功:割り込みハンドラを登録した IRQ 番号
失敗:-1

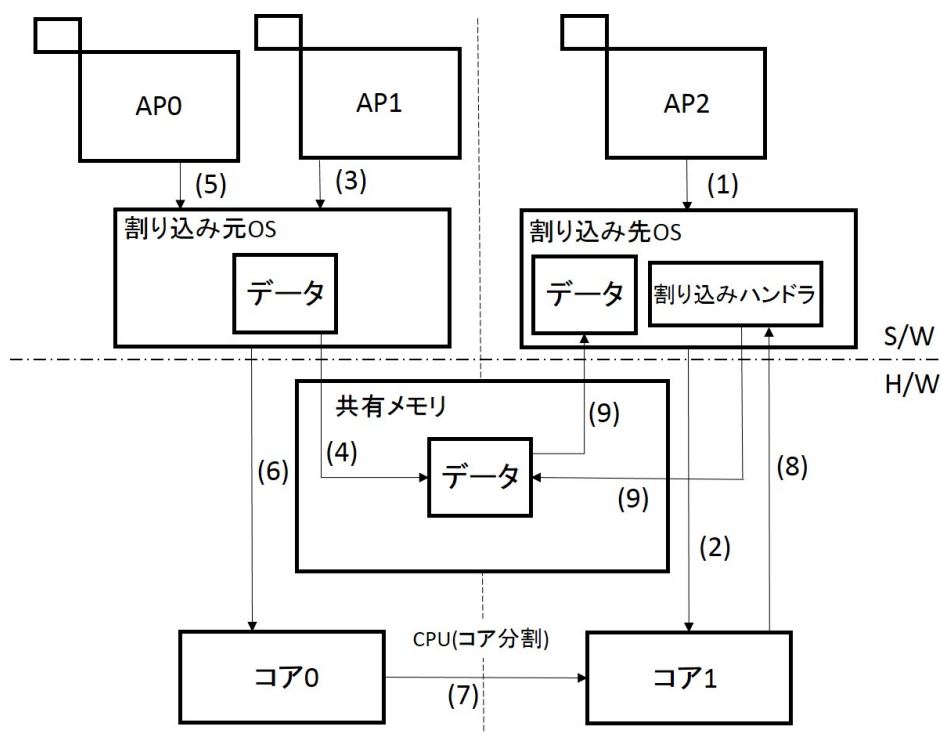


図 1 割り込みハンドラ動作までの流れ

【機能】登録可能な IRQ 番号 `irq` を探し，`irq` に割り込みハンドラ `fujita_ipi_handler` を登録する．その後，各コアのベクタ表 `vector_irq` のベクタ番号 `vector` のエントリに `irq` を登録する．

3.2 共有メモリへのデータの格納

共有メモリへのデータの格納に使用したシステムコールについて以下に示す．

【形式】`extern int sys_mem_test(void)`

【引数】なし

【戻り値】成功:1

失敗:0 以外

【機能】Mint の共有メモリに「fujita」という文字列を格納した配列を格納する機能を持つ．今回の実験では `0x1000020` に「fujita」という文字列を格納した配列を格納した．

3.3 IPI の送信

IPI を送信するシステムコールについて以下に示す．

【形式】`asm linkage void send_yamamoto_ipi(int core_id, int vector, int n, int interval)`

【引数】`int core_id`:IPI 送信先のコア ID

int vector:ベクタ番号
int n:IPI 送信回数
int interval:IPI 送信間隔

【戻り値】なし

【機能】core_id のコア ID を持つコアへベクタ番号 vector の割り込みハンドラを実行させる IPI を n 回連続で送信する．この際，IPI の送信間隔は interval である．

4 割り込みハンドラ

登録した割り込みハンドラについて以下に示す．

【形式】irqreturn_t fujita_ipi_handler(int irq, void *dev_id)

【引数】int irq:割り込みハンドラを登録する IRQ 番号
void *dev_id:デバイス ID

【戻り値】var 配列の cpu 要素

【機能】Mint の共有メモリから buffer に文字列を取得し，表示する．

5 おわりに

本資料では割り込み元 OS で共有メモリにデータを格納し，割り込み先 OS に IPI を送信後，割り込みハンドラによって共有メモリからデータを取得する流れを示した．今後は NIC ドライバの割り込み処理に必要なパケットの構造の調査と NIC ドライバの改変すべき箇所の調査を行い，実装する．