

Mintオペレーティングシステムにおける LinuxとAndroidの共存制御

平成25年2月15日

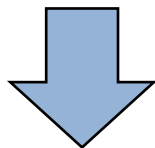
岡山大学 工学部 情報工学科

北川 初音

研究背景

<Mint>

- (1) 1台の計算機上で複数のLinuxを独立に走行させる方式
- (2) 32/64bit Linuxを混載可能



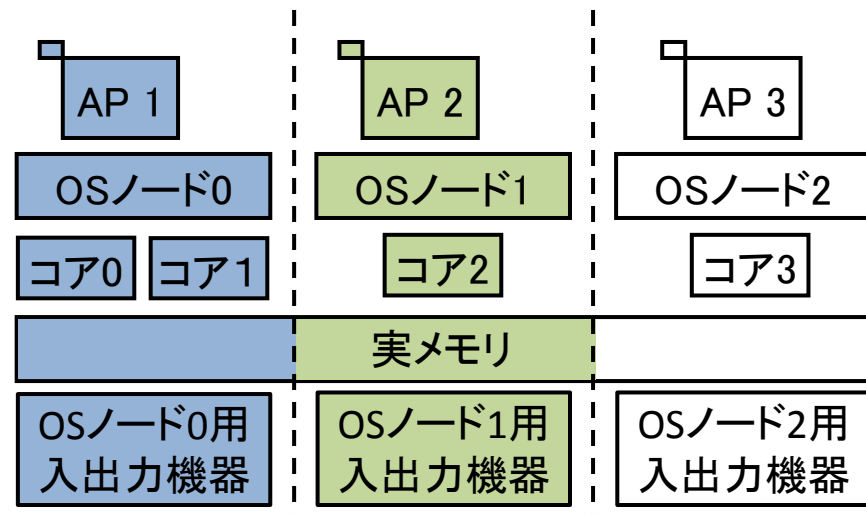
更なる試みとしてLinuxとAndroidの混載を実現

<LinuxとAndroidの混載の目的>

OS固有の特性を同時に使用

(利用例)

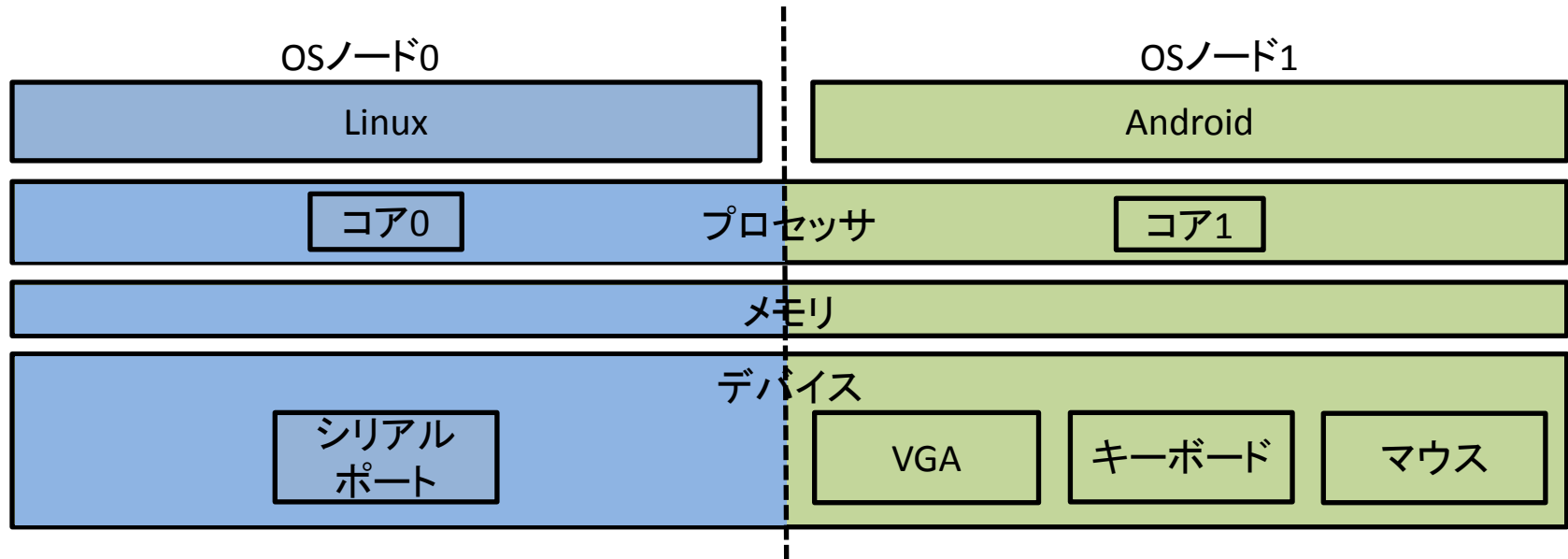
家電製品において、制御用OS(Linux)とUI用OS(Android)として利用



LinuxとAndroidの混載

<構成>

- (1) LinuxからAndroidを起動
- (2) OSノード0 (Linux)はシリアルポートを占有
- (3) OSノード1 (Android)はVGA, キーボード, マウスを占有



<問題>

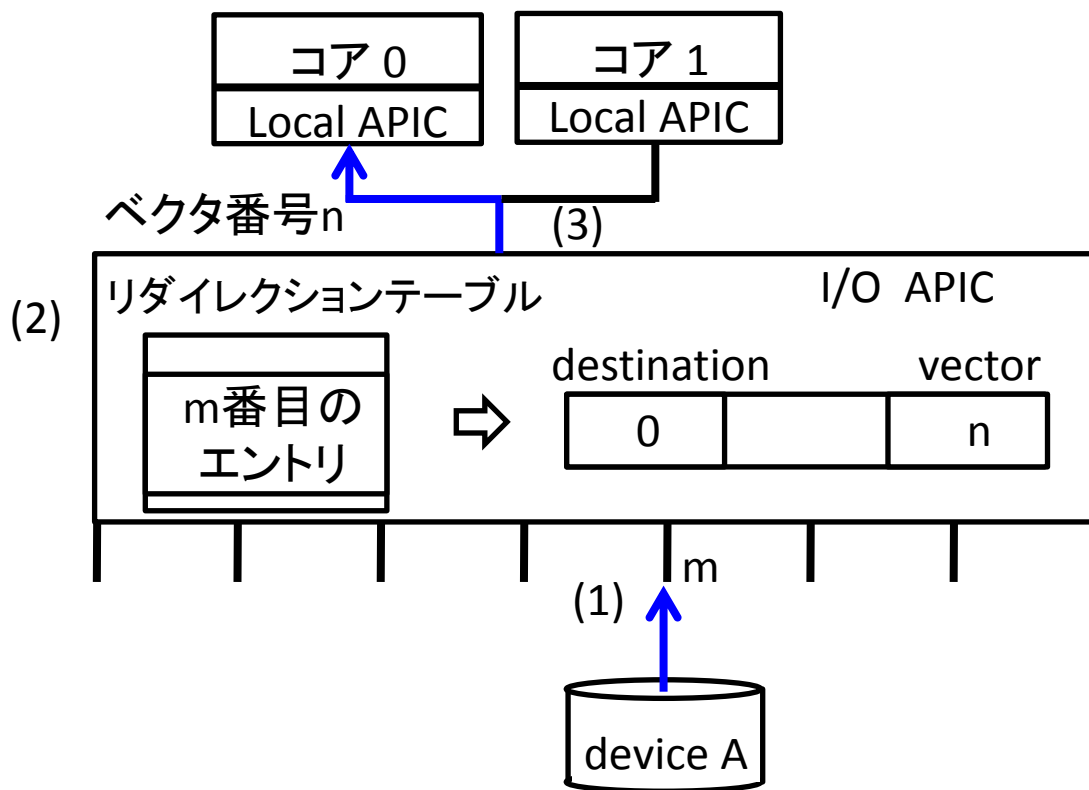
OSノード1 (Android)でキーボードが使用できない

原因: OSノード1の起動時の割り込みの設定

コアに割り込みが通知されるまでの処理

<ピン番号からベクタ番号への変換>

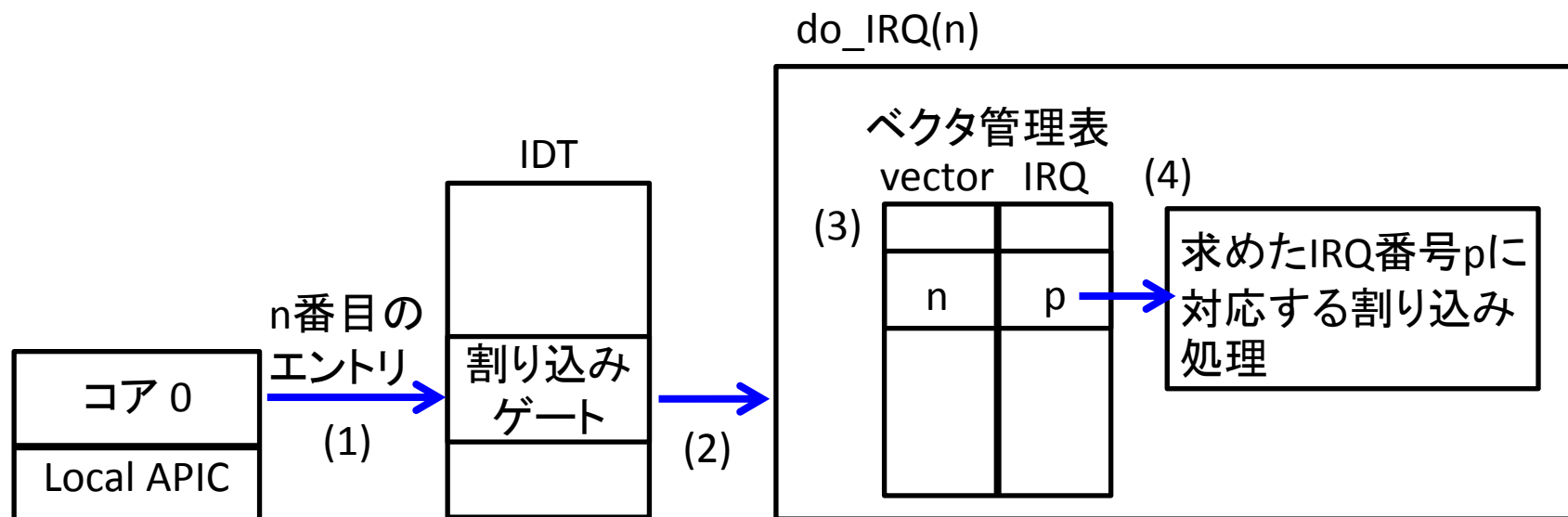
- (1) ピン番号mに割り込みが通知される
- (2) 通知先のコアを求め、ピン番号mをベクタ番号nに変換する
- (3) コア0にベクタ番号nを通知する



割り込み処理を呼び出すまでの処理

<ベクタ番号からIRQ番号への変換>

- (1) 割り込みゲートを呼び出す
- (2) do_IRQ()を呼び出す
- (3) ベクタ番号nからIRQ番号pを求める
- (4) IRQ番号pに対応する割り込み処理を行う

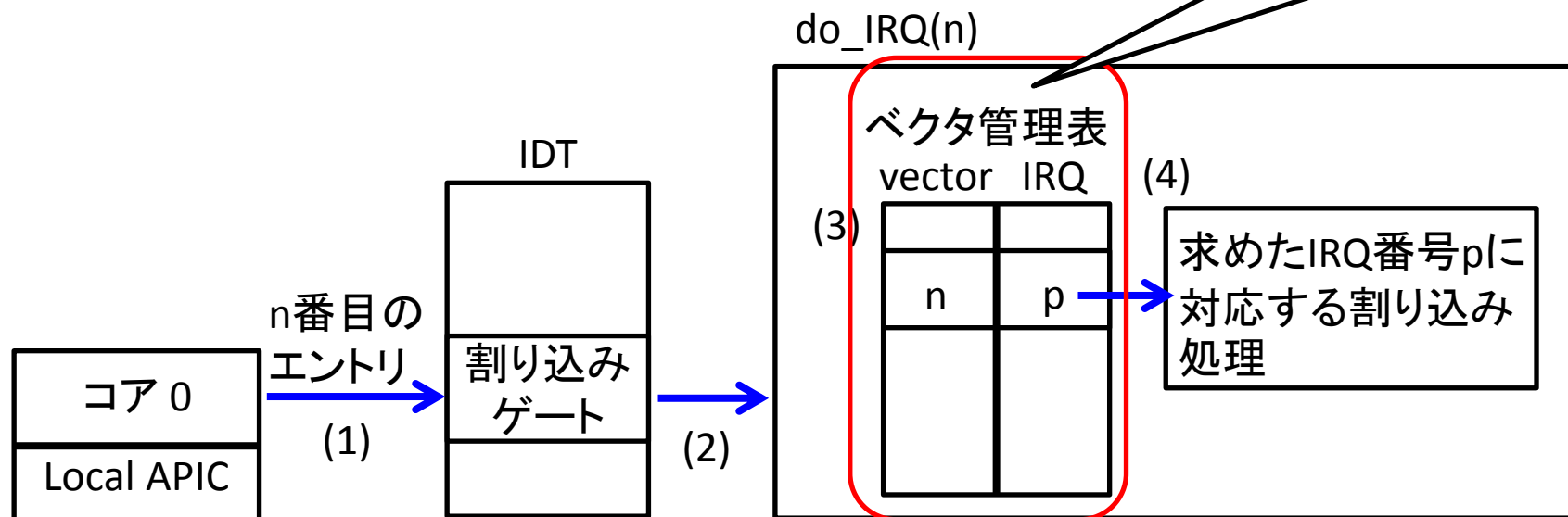


割り込み処理を呼び出すまでの処理

<ベクタ番号からIRQ番号への変換>

- (1) 割り込みゲートを呼び出す
- (2) do_IRQ()を呼び出す
- (3) ベクタ番号nからIRQ番号pを求める
- (4) IRQ番号pに対応する割り込み処理を行う

OSノード間でベクタ
管理表に不整合が
発生



ベクタ番号割り当てにおける問題点

<問題点>

ベクタ管理表の不整合による上書きが発生

➡ **上書きされたIRQ番号の割り込みが利用できない**

OSノード0 (Linux)の
ベクタ管理表

vector IRQ

48	0
49	16
50	2
51	3
52	4

(1) OSノード0が
ベクタ番号を割
り当て、ベクタ
管理表とリダイ
レクションテー
ブルに書き込む

OSノード1 (Android)の
ベクタ管理表

vector IRQ

48	0
49	16
50	2
51	3
52	4

(2) OSノード1がベク
タ番号を割り当てる

I/O APICのリダイレクション
テーブル

0	...	16	...	23
		49		

(3) OSノード1が
IRQ16番にベクタ番
号49番を割り当て、
ベクタ管理表を上書
きる

対処

<対処方針>

ベクタ番号を割り当てる際、以下の2つを守る必要

- (1) 共有するIRQ番号には、同じベクタ番号を割り当てる
- (2) すでにI/O APICでIRQ番号に対応付けられたベクタ番号を別のIRQ番号に割り当てない

<対処>

OSノード0: I/O APIC (IRQ0～IRQ15番)に割り当てるベクタ番号
にはIRQ16番以降を割り当てない

OSノード1以降: 割り当てようとしているベクタ番号が共有する
IRQ番号に割り当てられているか否か確認する

これらの対処により, **LinuxとAndroidの共存制御が可能**

評価

<評価環境>

OS	Android 4.0
CPU	Intel Core i7-870 @ 2.93GHz
メモリ	608MB

<ベンチマーク>

CPU, メモリ, およびI/Oの性能を測定し, スコアを算出

	未改変のAndroid	混載したAndroid
Antutu	19396.2	19392.0
Geekbench 2	3494.6	3494.4
Quadrant	6591.4	6589.8

※測定は5回ずつ行い平均値を算出

未改変のAndroidと混載したAndroidの測定結果に差はない

本発表のまとめ

<実績>

- (1) LinuxとAndroidの混載を実現
- (2) 割り込みベクタ調停方式を提案
- (3) 評価
 - (A) ベンチマークを測定し、未改変のAndroidと混載したAndroidで比較

<今後の課題>

- (1) AndroidからLinuxの起動の実現