SMP sur x86

Xavier Deguillard – xavier@lse.epita.fr

LSE

26 octobre 2011

Plan

Xavier Deguillard
–
xavier@lse.epita.fr

Introduction

Xavier Deguillard
–
xavier@lse.epita.fr

- SMP : Symmetric Multiprocessing
- Permet d'utiliser pleinement les processeurs et machines actuelles en permettant l'exécution simultanée de N processus, sur N processeurs.
- Nécessité d'utiliser des locks pour protéger les accès concurrents à la mémoire.

Pré requis

Xavier Deguillard
–
xavier@lse.epita.fr

Boot x86 et son initialisation :

- Segmentation et GDT
- Interruptions et IDT
- Pagination
- IRQ et PIC (Programmable Interrupt Controller)
- etc.

Plan

2 Pré-init

APICs

Xavier Deguillard
–
xavier@lse.epita.fr

APIC : Advanced Programmable Interrupt Controller, inventé pour pallier aux défauts du PIC. Il est découpé en 2 parties :

LAPIC

Chaque processeur possède un LAPIC, qui l'identifie, il est connecté au bus ICC (Interrupt Controller Communications) afin de pouvoir communiquer avec les autres LAPIC à l'aide d'IPI (Inter Process Interrupt).

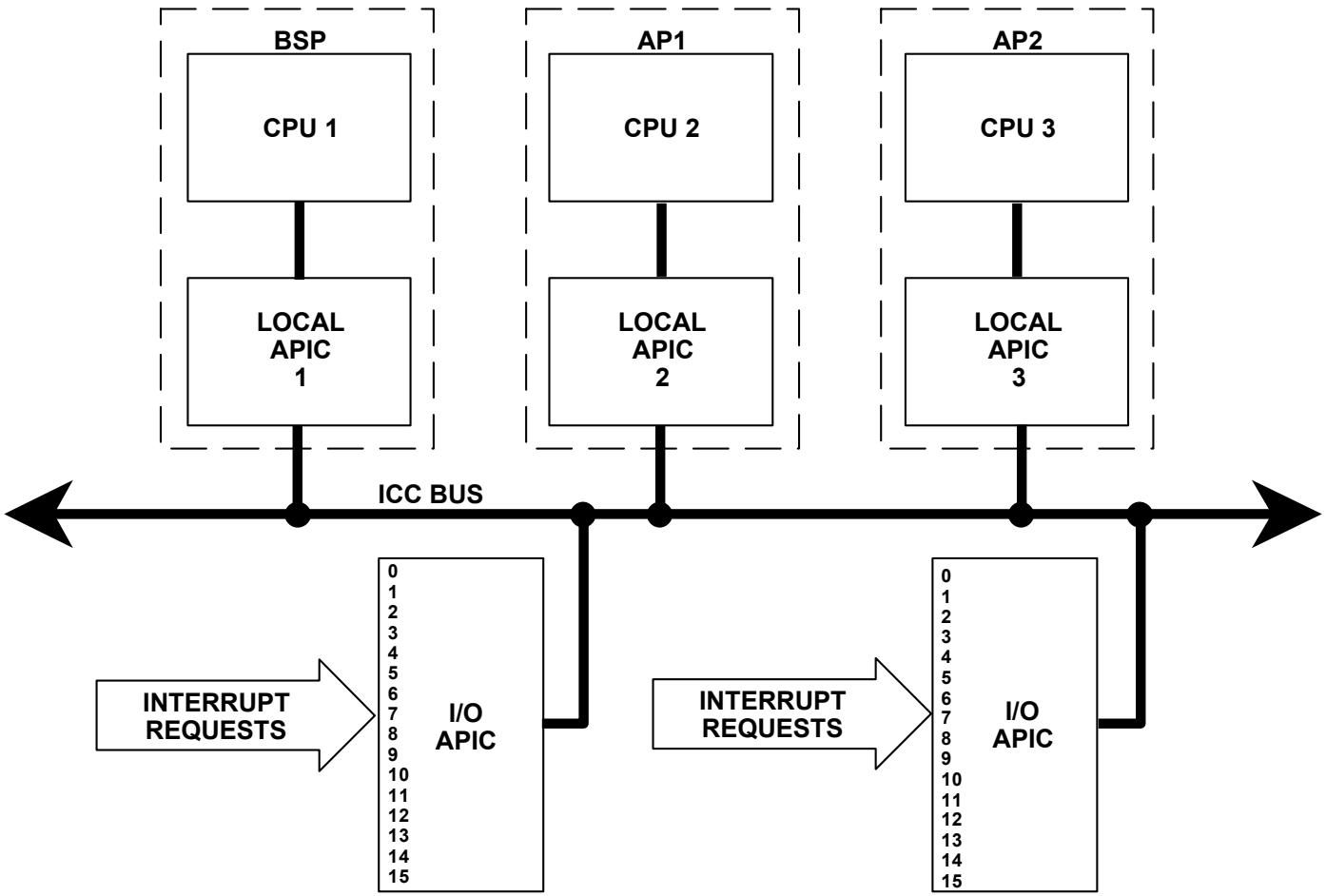
IOAPIC

Les périphériques y sont directement connecté. Son but est de notifier les CPUs d'une IRQ venant d'un périphérique. Il peut envoyer l'interruption directement à un LAPIC à l'aide de son identifiant, ou à un groupe de processeur.

Les APICs sont nécessaire à l'initialisation du SMP.

APICs

Xavier Deguillard
–
xavier@lse.epita.fr



Topologie old-school : MP-table

Créée en 1993 pour faciliter la configuration du SMP. Elle décrit à la fois le nombre de CPU, leurs identifiants, mais également les bus ainsi que la configuration des différents APICs.

Son adresse est non standard, il faut chercher la chaine _MP_ en mémoire !

Où ?

1. In the first kilobyte of Extended BIOS Data Area (EBDA), or
2. Within the last kilobyte of system base memory (e.g., 639K-640K for systems with 640 KB of base memory or 511K-512K for systems with 512 KB of base memory) if the EBDA segment is undefined, or
3. In the BIOS ROM address space between 0F0000h and 0FFFFFh.

Xavier Deguillard
–
xavier@lse.epita.fr

Topologie old-school : MP-table

Xavier Deguillard
–
xavier@lse.epita.fr

Créée en 1993 pour faciliter la configuration du SMP. Elle décrit à la fois le nombre de CPU, leurs identifiants, mais également les bus ainsi que la configuration des différents APICs.
Son adresse est non standard, il faut chercher la chaine _MP_ en mémoire !

Où ?

1. In the first kilobyte of Extended BIOS Data Area (EBDA), or
2. Within the last kilobyte of system base memory (e.g., 639K-640K for systems with 640 KB of base memory or 511K-512K for systems with 512 KB of base memory) if the EBDA segment is undefined, or
3. In the BIOS ROM address space between 0F0000h and 0FFFFFh.

Using ACPI (MADT) for SMP configuration information

```
% dmesg | grep ACPI | wc -l
141
```


Xavier Deguillard
–
xavier@lse.epita.fr

Plan

3 Init

Xavier Deguillard
–
xavier@lse.epita.fr

Maintenant que les APICs sont configurés et que nous avons le nombre de CPU de la machine ainsi que leur APICs id, nous pouvons les lancer.
Par convention on appelle AP (Application Processor) un processeur auxiliaire, par opposition au BSP (Bootstrap Processor)

Du coté BSP

Xavier Deguillard
–
xavier@lse.epita.fr

Le BSP envoie juste deux IPI à l'AP : un INIT, et un STARTUP. Le premier reset le processeur, le deuxième permet d'indiquer quelle est l'adresse d'initialisation du processeur.

Du coté AP

Xavier Deguillard
–
xavier@lse.epita.fr

Un AP, tout comme le BSP boote en mode 16 bits, il faut donc avant toute chose initialiser le mode 32 bits, puis la pagination, et si nécessaire le mode 64 bits, pour enfin pouvoir exécuter du code kernel.

Plan

SMP sur x86

Xavier Deguillard
–
xavier@lse.epita.fr

4 Post-Init

Xavier Deguillard
–
xavier@lse.epita.fr

Si tout s'est bien déroulé, le kernel dispose désormais de N fils d'exécution : un par processeur. Il faut maintenant penser à verrouiller toutes les globales, afin de ne pas avoir de surprises !