

Systemes d'Exploitation Architecture pour les systemes

Didier Verna

didier@lrde.epita.fr

<http://www.lrde.epita.fr/~didier>

Version 2@1.7 – 30 novembre 2004



Table des matières

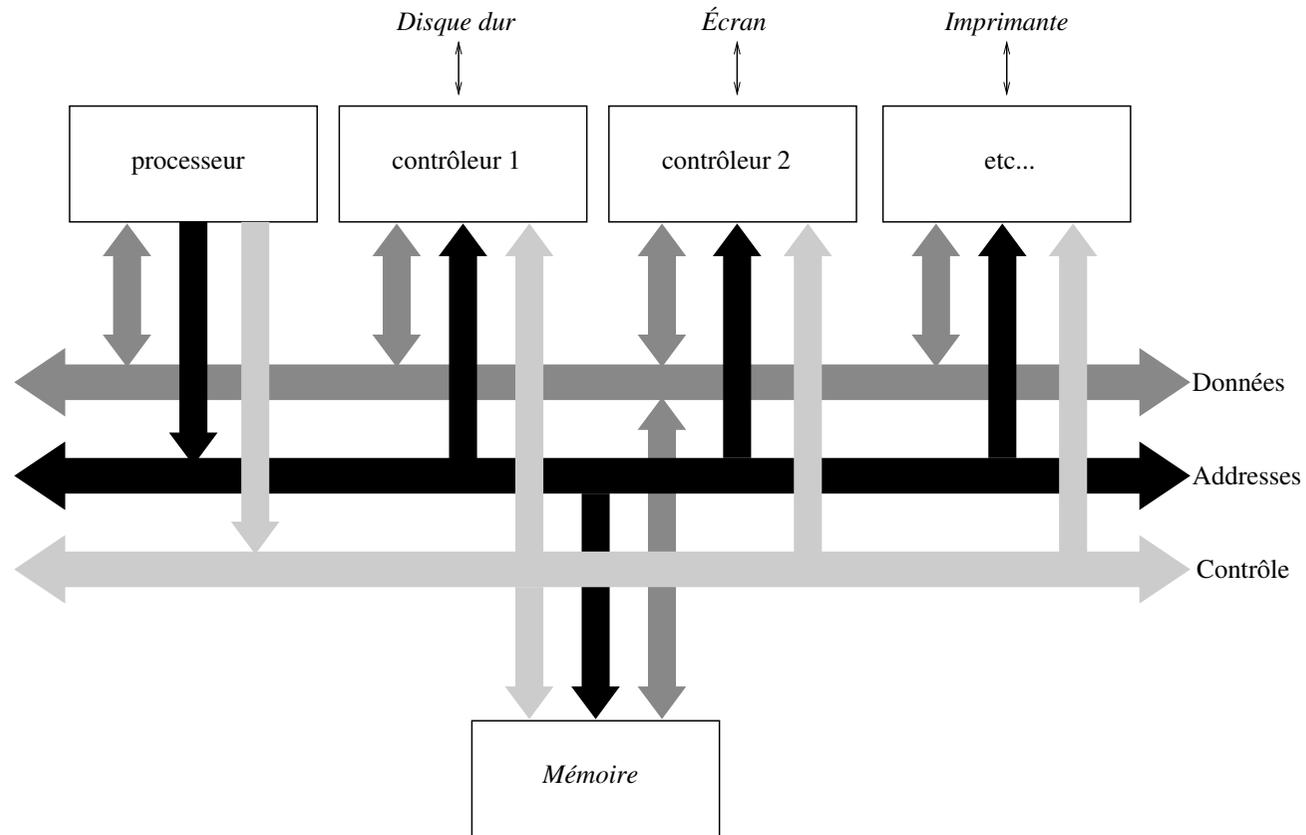
Fonctionnement global	2
Initialisation	5
Interruptions	6
Fonctionnement CPU / Mémoire	9
Mémoires	10
Fonctionnement des entrées / sorties	11
Techniques d'E/S	13
Protection matérielle	15
Protection de la mémoire	16
Protection du CPU	17

Fonctionnement global

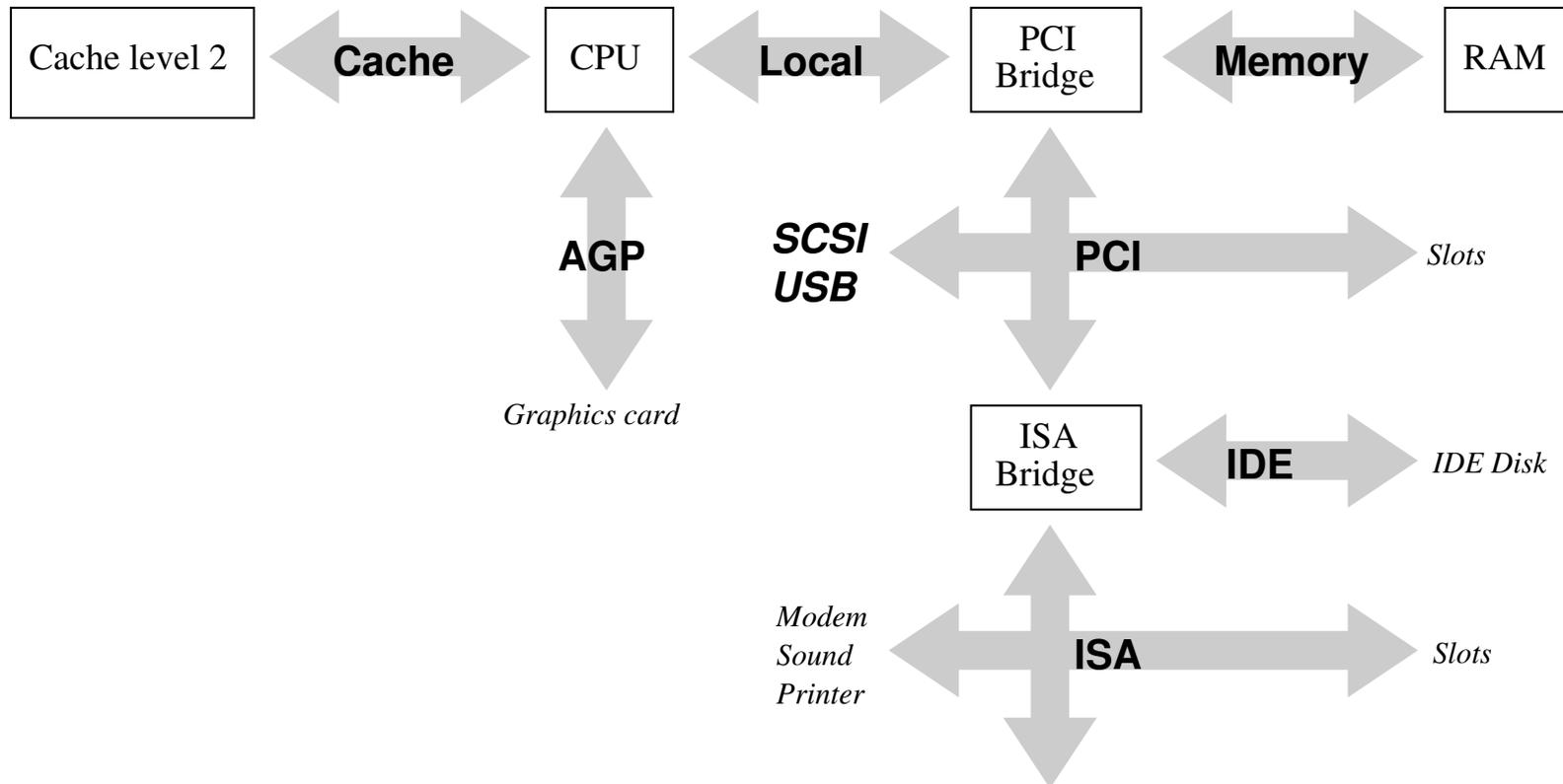
Un système informatique est composé de :

- un (ou plusieurs) processeur(s),
- de la mémoire,
- des contrôleurs de périphériques,
- les périphériques associés,
- des bus pour lier le tout.

Bus



Exemple : architecture Pentium



Et aussi IEEE 1394 (FireWire)...

Initialisation

BIOS (Basic Input Output System) :

- initialise le matériel (registres processeur, mémoire...)
- scanne les bus (ISA et PCI d'abord) pour trouver un périphérique amorçable (bootable)
Premier secteur \implies partition active \implies deuxième « boot loader » ou système
- charge le système d'exploitation en mémoire

Remarque : Conflit d'IRQ, périphériques legacy... \implies « Plug and Play » (Plug and Pray).

Le système d'exploitation :

- lance le premier processus (« init »)
- attend un événement

Les événements sont produits par des « interruptions ».

Interruptions

Seul moyen d'interrompre le traitement en cours d'exécution

Deux types d'interruption :

- **Interruptions matérielles** : déclenchées par le processeur, les périphériques. . .
(ex.. fin d'une opération d'entrée / sortie)
- **Interruptions logicielles** : déclenchées par des « appels système »
(ex.. accès mémoire)

Remarque : « Exception » = interruption logicielle suite à une erreur (division par 0, accès mémoire illégal. . .)

Traitement des interruptions

- Chaque interruption est associée à une routine fixe de traitement.
- L'arrivée d'une interruption suspend le traitement en cours, exécute la routine associée puis reprend le traitement précédent l'interruption.

⇒ **Vecteur d'interruptions** : tableau situé en mémoire basse et contenant les adresses de début de chaque routine.

⇒ L'adresse de l'instruction interrompue est stockée dans la pile du système. Les routines de traitement des interruptions doivent elle-même sauvegarder l'état du processeur si besoin est.

Interruption des interruptions

- **Masquage simple :**

Aucune interruption ne peut intervenir pendant le traitement d'une interruption : leur traitement est différé jusqu'à la fin du traitement de l'interruption en cours.

- **Masquage sélectif :**

Chaque interruption est associée à un niveau de priorité. Une interruption peut interrompre une autre interruption de priorité inférieure. Les interruptions de même niveau sont gérées par masquage simple.

Fonctionnement CPU / Mémoire

Cycle typique d'exécution « Von Neumann »

- Extraction d'une instruction
 - Stockage dans le registre d'instruction
 - Décodage
 - Extraction de données éventuelles (opérandes)
 - Exécution
-
- Jeu d'instructions spécifique à chaque CPU
 - Utilisation de registres CPU. Registres spéciaux : PC (Program Counter), SP (Stack Pointer), PSW (Program Status Word)
 - Architectures modernes : pipelines, CPU super-scalaires...RISC

Mémoires

	Type	Temps d'accès	Capacité
Accès direct :	Registres	1ns	Ko
	Cache	2ns	1 – 10 Mo
	RAM / ROM / CMOS	10ns	10 Mo – 1 Go
	EEPROM	1 μ s	
Accès indirect :	Disques magnétiques	10ms	500 Mo – 100 Go
	Bandes magnétiques	100s	100 Go – 1 To

⇒ Le CPU n'accède qu'à la mémoire directe.

Fontionnement des entrées / sorties

Contrôleur de périphériques :

- Gérer un ou plusieurs périphériques de même type (IDE, SCSI...)
- Harmoniser leur comportement (vitesse d'accès, structure...)
- Assurer le transfert de données entre ces périphériques et la mémoire locale du contrôleur

⇒ Offrir une interface plus simple au système d'exploitation

Pilotes (drivers) et noyau :

- Édition de lien du noyau (UNIX)
- Décrits dans un fichier centralisé (Windows)
- installés « au vol » / « à chaud » / HotPlug (USB, modules...)

Fonctionnement des pilotes

- Requête d'E/S par positionnement des registres du contrôleur.
- Signalement de la fin d'une E/S par une interruption.

E/S configurées en mémoire :

- Adresses mémoire correspondant aux registres des contrôleurs
- Pas besoin d'instructions d'E/S spécifiques
- Évite le contact direct entre le matériel et les programmes utilisateurs

⇒ Pratique pour les périphériques à temps de réponse rapide (adaptateurs vidéos) ou utilisés fréquemment (modem).

Remarque : Il existe également des périphériques permettant l'acquisition de données avant que celles-ci soient requises.

Techniques d'E/S

processus utilisateur \Rightarrow Appel système \Rightarrow Pilote \Rightarrow Contrôleur

- **E/S synchrones :**

Le pilote attend la fin de la requête : scrutation du périphérique (polling), attente active.
Une seule requête d'E/S à la fois.

- **E/S asynchrones :**

Demande d'interruption pour signaler la fin de la requête, retour immédiat du pilote.
Possibilité d'E/S simultanées.

\implies Nécessité de maintenir une « table d'état des périphériques ».

Direct Memory Access (DMA)

Problème : les périphériques rapides (disques, réseau. . .) sont susceptibles de générer énormément d'interruptions (à une fréquence proche du temps de traitement des interruptions).

Solution : transférer des blocs entiers de données entre le contrôleur et la RAM, sans intervention du processeur.

⇒ Un interruption par bloc et non plus par mot.

Protection matérielle

Partage des ressources \implies problèmes de cohérence et de sécurité. Une solution : le fonctionnement en **mode double**.

- **Mode utilisateur :**

Mode d'exécution des processus clients : le système commute en mode utilisateur dès qu'il donne la main à un processus.

- **Mode superviseur :**

Mode d'exécution du système d'exploitation : le matériel commute en mode superviseur dès qu'une interruption est reçue.

- **Instruction privilégiée :**

Instruction potentiellement nuisible ne pouvant s'effectuer qu'en mode superviseur.

Au minimum : toutes les instructions d'E/S.

Protection de la mémoire

- Empêcher la modification du vecteur d'interruptions et des routines de traitement de ces interruptions.
- Protéger le système lui-même, et les utilisateurs entre eux.

Cas le plus simple :

- À chaque processus, on associe un registre *base* et un registre *limite* définissant ses capacités d'accès à la mémoire
- Des instructions privilégiées permettent au système d'exploitation de charger ces registres
- Le matériel effectue le contrôle d'accès

Protection du CPU

But : s'assurer que tout processus client finit par rendre la main au système.

⇒ Utilisation d'une horloge.

- **Temps fixe** : l'horloge génère une interruption à intervalle régulier.
- **Temps variable** : temps fixe + compteur. Le système d'exploitation fixe la valeur du compteur. Le compteur est décrémenté à temps fixe. Une interruption est générée quand le compteur atteint la valeur 0.

⇒ Les horloges sont utilisées pour implémenter le temps partagé. Il existe en général une horloge distincte pour le calcul de date.