

# Programmation Système

---

## Cours 1

### Matériel

Brice Goglin

## Copyright

---

- Copyright © 2004 Brice Goglin – all rights reserved
- Ce support de cours est soumis aux droits d'auteur et n'est donc pas dans le domaine public. Sa reproduction est cependant autorisée sous réserve de respecter les conditions suivantes :
  - Si ce document est reproduit pour les besoins personnels du reproducteur, toute forme de reproduction (totale ou partielle) est autorisée à condition de citer l'auteur.
  - Si le document est reproduit dans le but d'être distribué à des tierces personnes, il devra être reproduit dans son intégralité sans aucune modification. Cette notice de copyright devra donc être présente. De plus, il ne devra pas être vendu.
  - Cependant, dans le seul cas d'un enseignement gratuit, une participation aux frais de reproduction pourra être demandée, mais elle ne pourra pas être supérieure au prix du papier et de l'encre composant le document.
  - Toute reproduction sortant du cadre précisé ci-dessus est interdite sans l'accord écrit préalable de l'auteur.

## Plan

---

- Architecture générale des ordinateurs
- Exécution
- Hiérarchie mémoire
- Interruptions et exceptions
- Communications avec périphériques
- Architectures complexes

---

## Architecture générale des ordinateurs

## Description générale

- Processeur : traite les instructions
- Mémoire : stocke données et programmes
- Bus mémoire : gère les accès mémoire
- Bus I/O gère les communications avec les périphériques
- Périphériques : entrées/sorties (stockage, interaction, ...)
- BIOS : système minimal pour booter

## Généralités sur les processeurs

- Idées des années 40
- Révolution du transistor et des semi-conducteurs
- Loi de Moore
- 4 Ghz, 0,06  $\mu\text{m}$
- Dissipation de chaleur devient trop contraignante
  - Multiplication des files d'exécution
- Limite quantique ?

## Exécution

## Registres

- Registres utilisateurs
  - Données
  - Adresses
    - Segment, pile, ...
- Registres de status et contrôle
  - *Program Counter* (PC)
  - Status arithmétique

## Appel de procédure et Pile

- Empiler données locales
- Conserver adresse retour et frame précédente
- Registres de pile pour procédure courante
  - *Stack Pointer*
  - *Frame Pointer*
- Stocké de haut en bas sur IA32

## Exécution

- Code opération + Localisation des données
- Opération de type entier, flottant ou mémoire
- CISC ou RISC
- Contrôle par horloge
- Traitement multiple SISD, SIMD, MIMD, MISD
- Différents modes d'exécution
  - Protégé, réel, ...

## Pipeline

- Augmentation de la fréquence par découpage du traitement des instructions
  - Chargement, décodage, exécution
- Pipeline de plus en plus long
- Branches cassent le pipeline
  - Optimisation par prédiction, exploration, ...

## Hiérarchie mémoire

## Hiérarchie mémoire

- Registres du processeur
- Mémoire centrale
  - Stockage volatile (disparaît au reboot)
  - Adressage 32 ou 64 bits
  - Relativement lent
- Disque, bande, ...
  - Stockage persistant
  - Très lent

## Cache

- Zone de stockage intermédiaire
  - plus petite mais plus rapide
- Conserve zones récemment accédées
- Précharge zones proches qui pourraient être accédées peu après
- Algorithme de remplacement LRU

## Exemple de caches

- Mémoire cache, jusqu'à trois niveaux
  - Placé entre processeur et mémoire
  - Très rapide
  - Utilisé de manière transparente
  - Invalidation logicielle possible
- Cache dans les contrôleurs disques
- Espace swap géré par OS
- Cache disque géré par OS

## Interruption et exécution

## Interruptions

- Requêtes d'un périphérique vers le processeur
  - Un numéro permet de réperer le périphérique
- Suspension du programme en cours
- Déroutement vers un code de traitement
  - *handler* fixé par le système d'exploitation
- Retour au code initial

## Exceptions

- En cas de problème
  - Erreur arithmétique, mauvais accès à la mémoire
- Déroutement vers code de traitement
- Exception spéciale pour appels système
  - Changement de mode d'exécution

## Communication avec les périphériques

## Entrées/Sorties

- Mapping de registre des périphériques
- Ecriture de commandes
- Traitement dans le périphérique
- Lecture de résultats
- *Programmed I/O*
  - Attente active du bon statut

## Entrées/Sorties (2/2)

- *Interrupt-driven I/O* (modèle asynchrone)
  - Recouvrement traitement périphérique par autre chose dans le processeur
  - Périphérique envoie IRQ quand terminé
  - Le *handler* du processeur récupère les informations du périphérique puis retourne à l'exécution normale
- *Direct Memory Access*
  - Transfert de données sans le processeur

## Ordres de grandeur

- Processeur : Ghz
- Mémoire cache : Mo, ns
- Mémoire : Go, 10 ns
- Bus Mémoire : Go/s
- Bus PCI : 500 Mo/s, dizaines de cycles
- Interruption : 10  $\mu$ s
- Disque : 100 Go, ms, 50 Mo/s
- Réseau : variable

## Architecture complexes

## SMP, NUMA et systèmes distribués

- *Symmetric Multi-Processing*
  - Plusieurs processeurs, une seule mémoire
  - Maintenir cohérence entre différents caches
- *Non-Uniform Memory Access*
  - Plusieurs noeuds avec plusieurs processeurs
  - Un banc mémoire par noeud, accessible par tous
- *Distributed Systems*
  - Un banc mémoire par noeud, le seul accessible