

IFT-2001

SYSTÈMES D'EXPLOITATION

Mondher Bouden

Chargé de cours

**Département d'informatique et de génie logiciel
Université Laval**

Automne 2020

Principaux sujets traités (Tanenbaum)

- Partie 1 {
- Processus et allocation du CPU (Chapitre 2)
 - processus, threads, ordonnanceur
 - Gestion de la mémoire (Chapitre 3)
 - architecture, pagination, segmentation
 - Programmation concurrente (Chapitre 6)
 - interblocages

- Partie 2 {
- Système de fichiers (Chapitre 4)
 - Gestion des entrées/sorties (Chapitre 5)
 - Multiprocesseurs (Chapitre 8)
 - Varia : gestion alimentation, hyperthreading

- Partie 1 & 2 • Études de cas Linux/Windows (Chapitre 10 & 11)

Pourquoi étudier les SE?

Pourquoi étudier les systèmes d'exploitation?

- Comprendre en profondeur systèmes informatiques / limites imposées
- Maximiser les performances de vos programmes
 - appels systèmes coûteux
 - multiprocesseurs
- Éviter les bugs difficiles à détecter
 - situation de compétition / race condition
 - exemple du *Therac-25*

Pourquoi étudier les systèmes d'exploitation?

- Design de système complet : choisir le S.E. approprié (pas de parfait)
 - embarqué (*embedded*)?
 - temps réel?
- Design de S.E. :
 - Google/Facebook/CISCO
- Toujours en contact avec les S.E. dans votre carrière
- Mieux comprendre évolution/amélioration

Tour d'horizon d'un système d'exploitation

Définition d'un SE

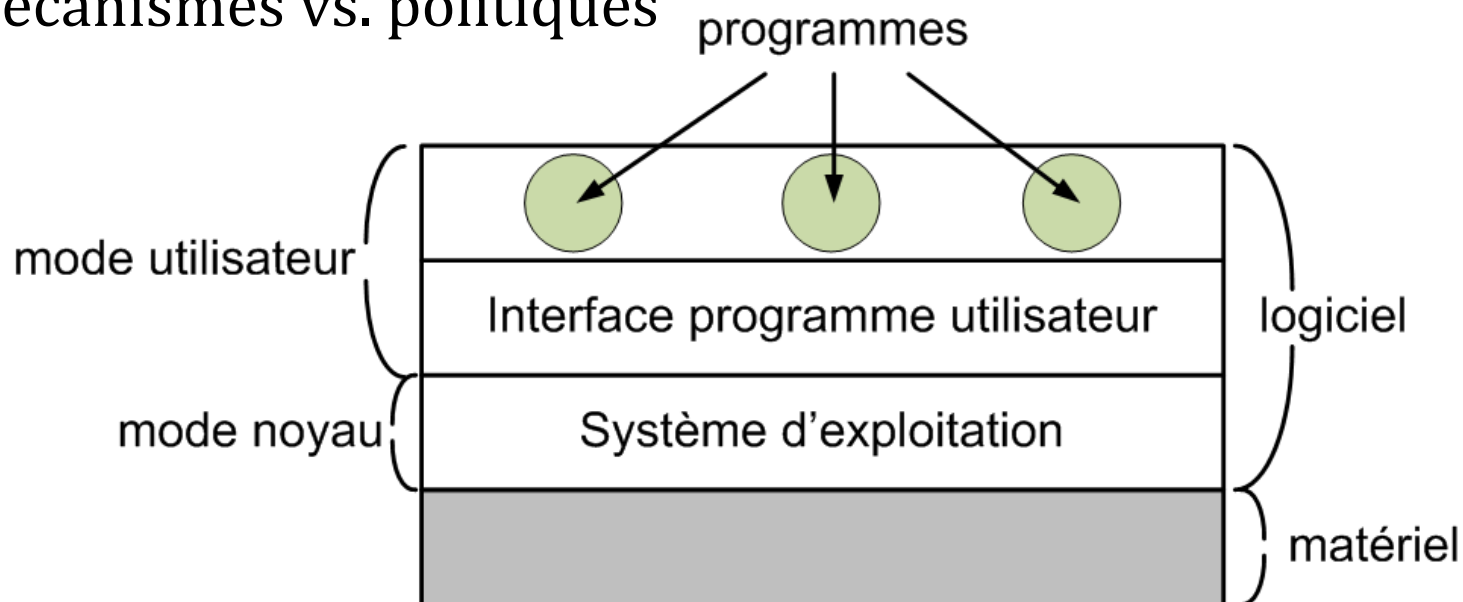
- Aucune définition universellement admise
- Une définition générale est :
“Un système d'exploitation est l'ensemble des programmes qui communiquent avec le matériel et contrôlent l'**allocation des ressources** aux autres programmes du système.”
- Le seul « exécutable » toujours chargé sur l'ordinateur est appelé **noyau** : cœur du S.E.

Analogies pour un SE

- Un illusionniste :
 - Fait disparaître certaines limites du matériels
 - Donne l'illusion que la machine a une mémoire infinie et une infinité de processeurs
- Un gouvernement :
 - Protège les utilisateurs les uns des autres
 - Partage des ressources de façon efficace et équitable

S.E. : 2 buts principaux

- Offrir une **abstraction** de haut-niveau des ressources
 - faciliter la tâche aux programmeurs/applications
- Gérer les ressources matérielles de l'ordinateur
 - performance et protection
 - mécanismes vs. politiques



Mécanismes vs. politiques

- **Mécanisme**

- éléments **matériels** ou **logiciels** pour **contrôler** l'utilisation d'une ressource
- exemple : protection de la mémoire

- **Politique**

- **choix** de comment utiliser ces ressources
- exemple : répartition de la mémoire entre processus

- Séparation mécanisme/politique augmente la flexibilité dans la conception de systèmes

S.E. n'est pas l'interface usager

- Pour beaucoup, S.E. est synonyme d'interface-usager



Dérivés d'UNIX

S.E. n'est pas l'interface usager

- GUI ne fait pas parti à proprement parlé du SE ...
 - ... même si c'est la partie la plus visible pour les utilisateurs.
 - KDE ou Gnome sur Linux, vs X-Window sur Unix
- *Vrais utilisateurs* des S.E. :
 - applications (entremise des appels systèmes)
 - programmeurs → **vous!**
- Dans ce cours : voir l'envers du décor (~~GUI~~)

Partage des ressources : multiplexage

- Dans l'**espace**

- disque dur
- mémoire vive (RAM)
- etc.

- Dans le **temps**

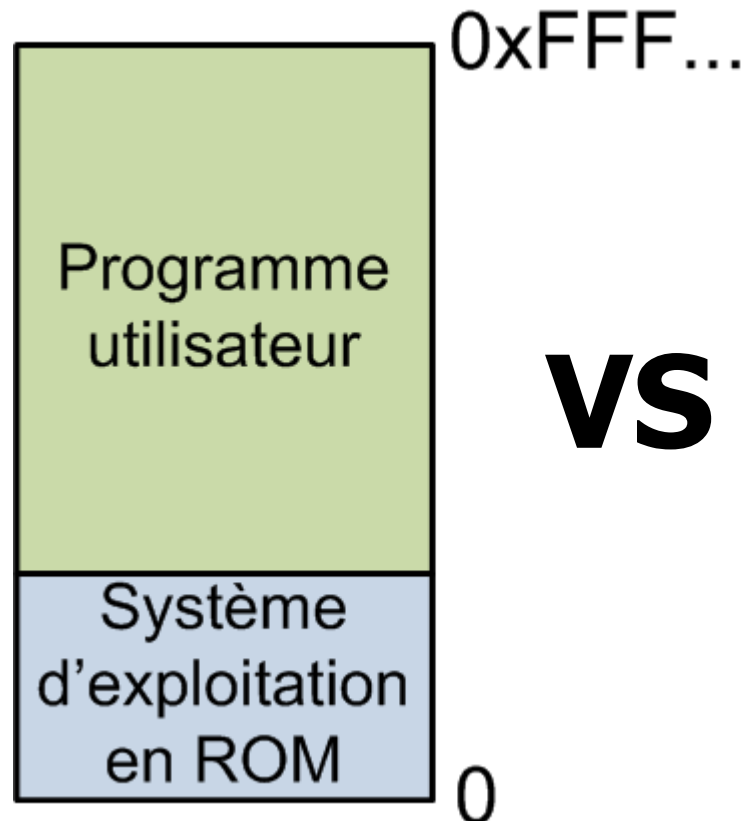
- CPU
- imprimante
- clavier
- etc.

Multiplexage dans l'espace

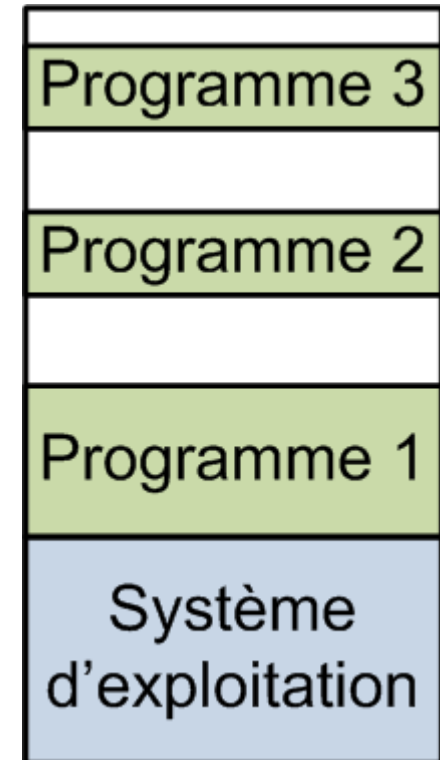
- Avoir plusieurs programmes chargés en mémoire en même temps



TRS-80



VS



aujourd'hui

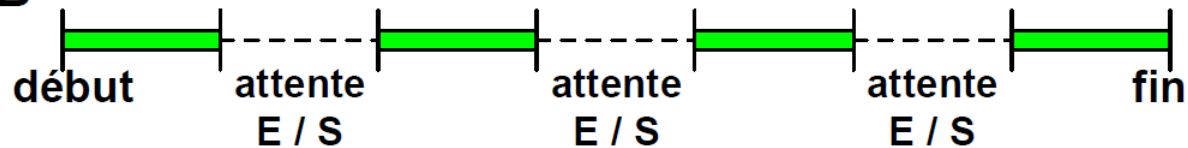
Multiplexage dans le temps : performance

Deux processus à exécuter

Processus A



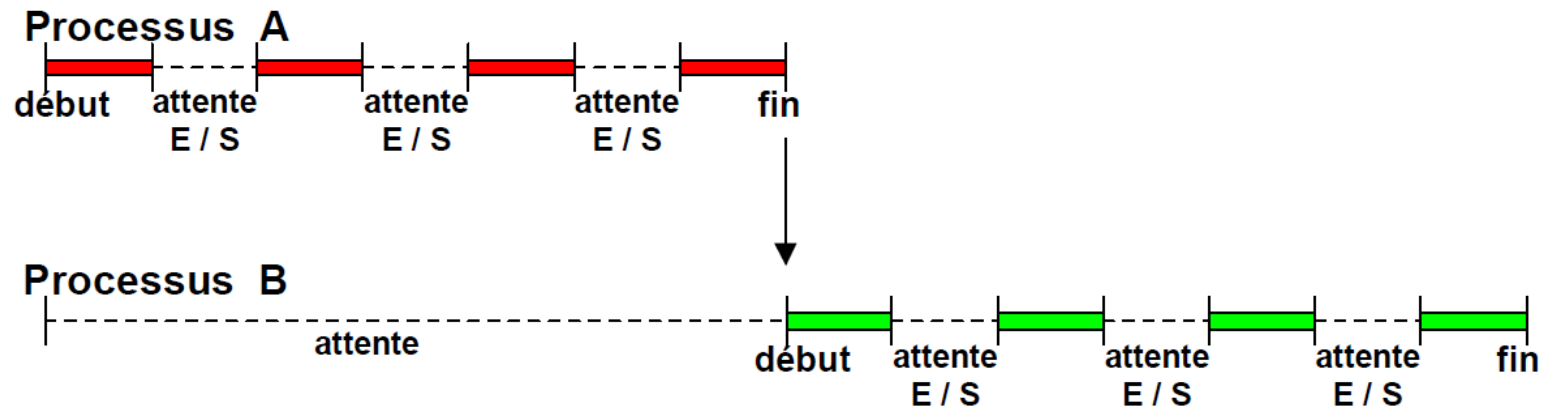
Processus B



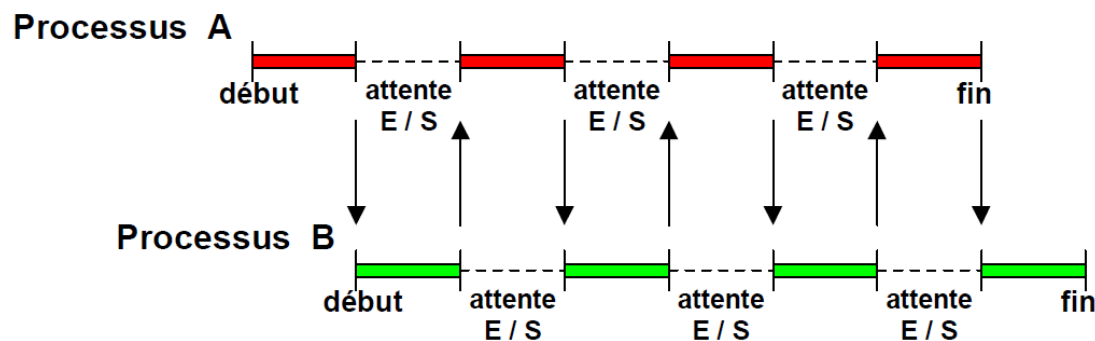
E / S = Entrée-sortie (*Input-Output*)
(par exemple lecture ou écriture sur disque)

Multiplexage dans le temps : performance

Sans multiplexage



Avec multiplexage + 2 processus en mémoire



Maximiser l'utilisation
du CPU

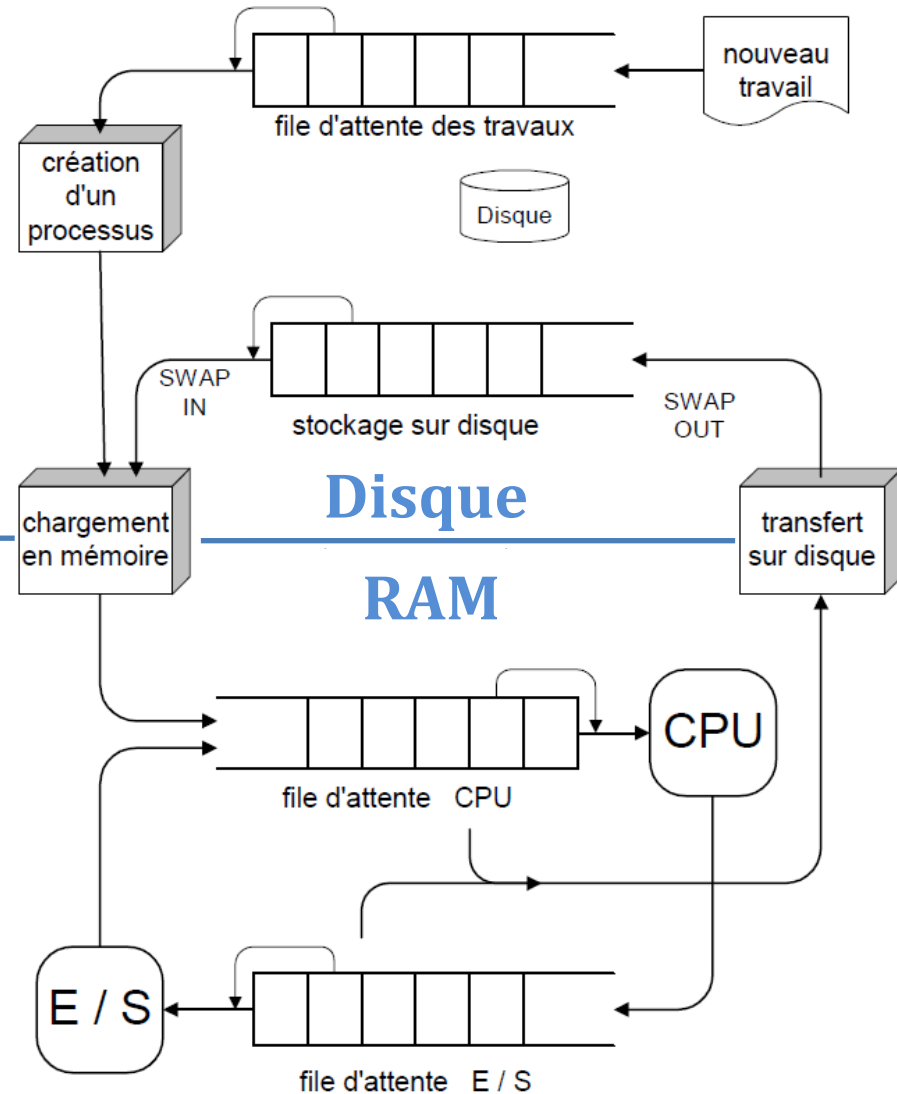
4 tâches principales par S.E.

- Gestion des processus/threads
 - Gestion de la mémoire
 - Gestion des fichiers
 - Gestion des entrées/sorties (périphériques)
- 1^{ère} moitié
- 2^{ème} moitié

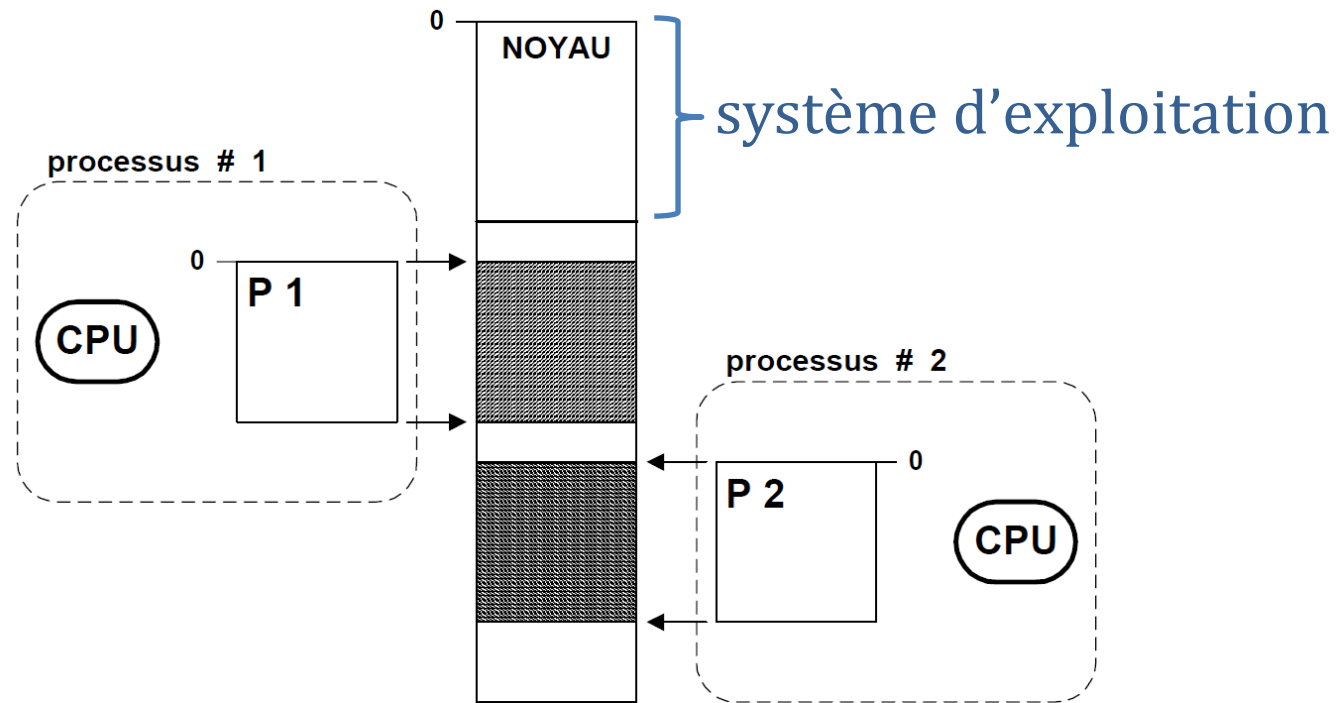
Gestion des processus

- Abstraction d'un programme en cours d'exécution

- Comment gérer quel processus Tourne ?



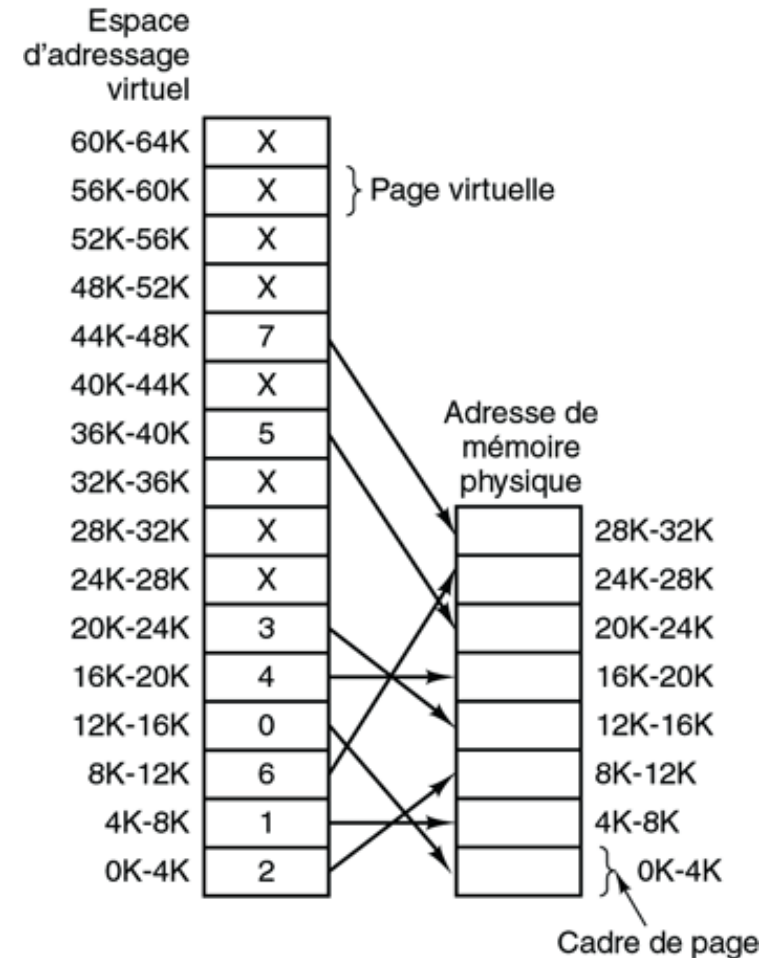
Partage de la mémoire : virtuelle



- Espaces d'adressages individuels
 - adresse logique → physique
- Protection (autres processus et S.E.)

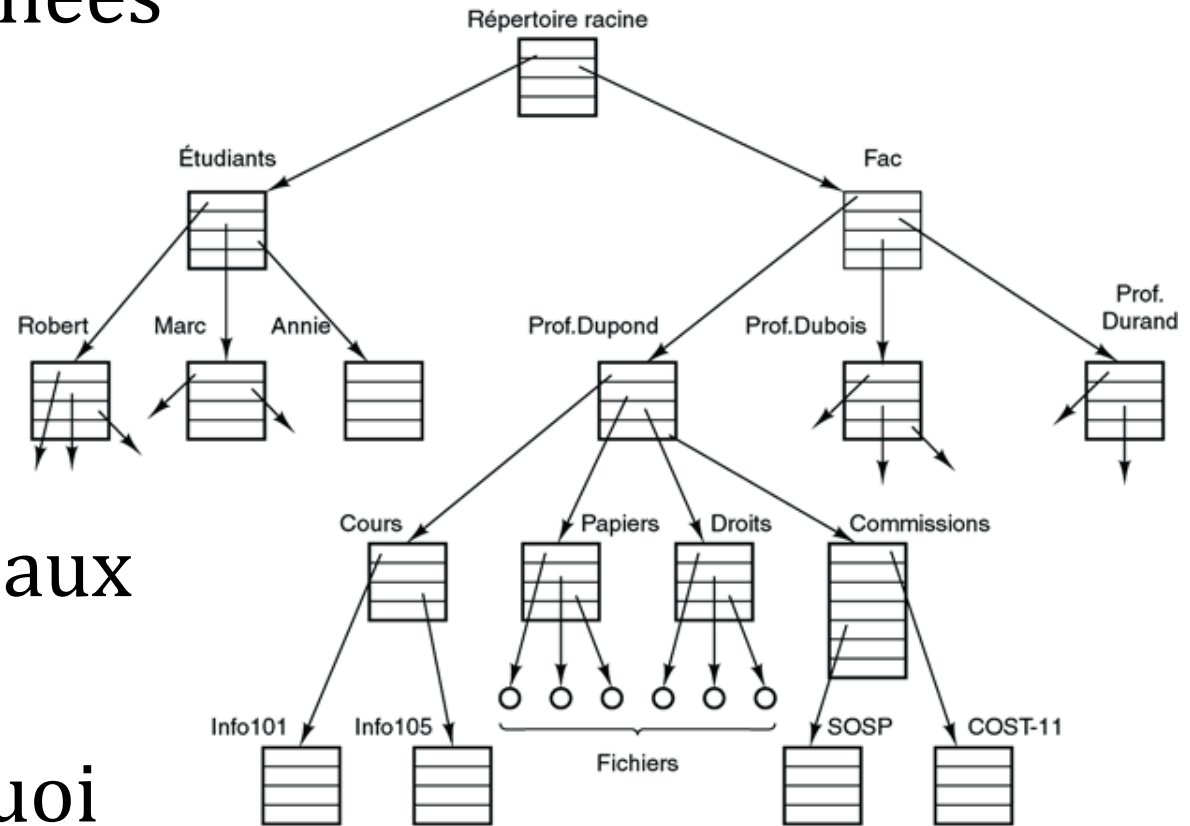
Pagination de la mémoire

- Toujours tout conserver à mémoire n'est pas optimal
- Début du programme :
 - initialisation
 - exécuté une seule fois
- Enlever le début du programme après exécution
 - par le programme lui-même ?
 - par le S.E. ?



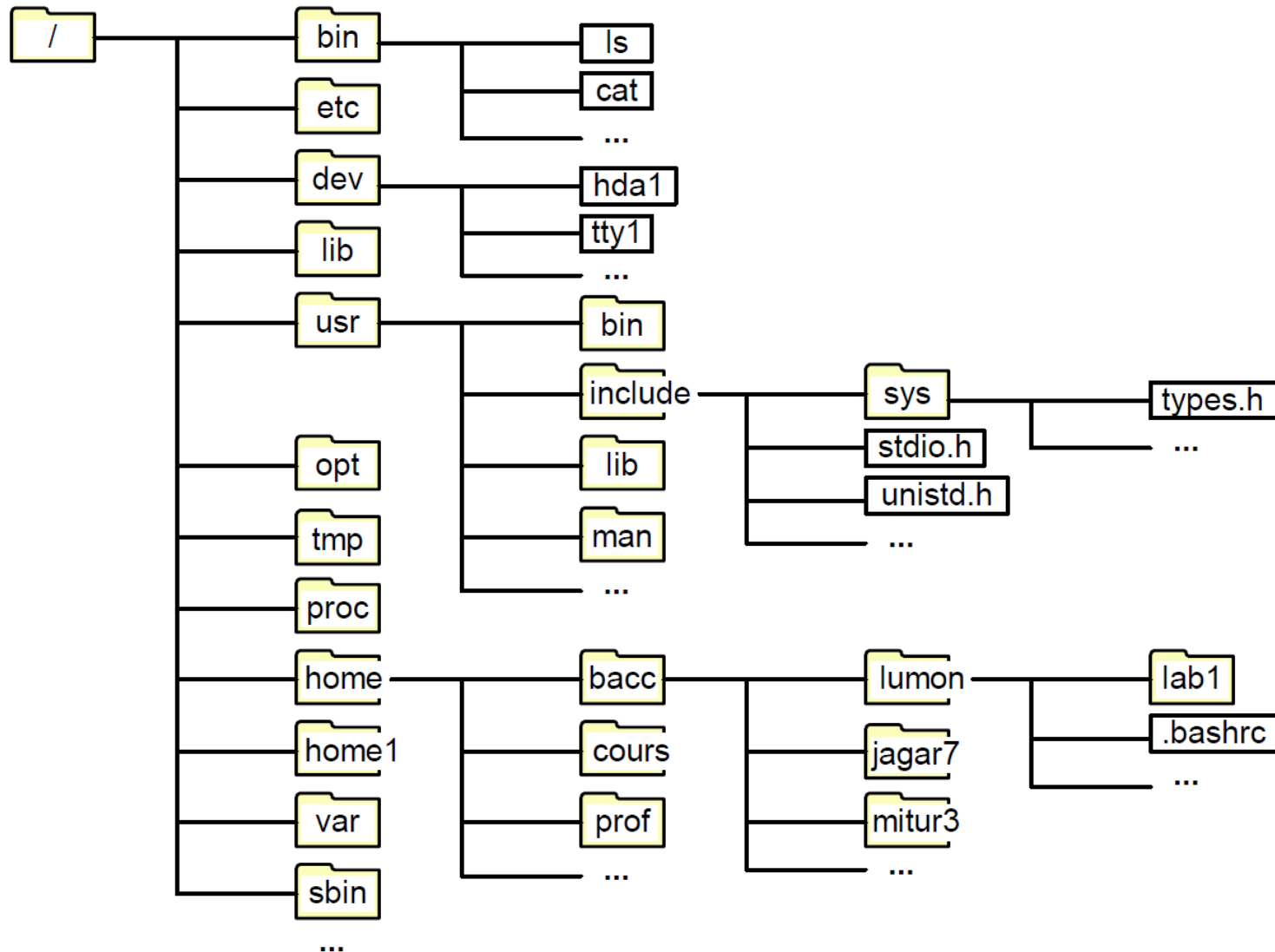
Gestion des fichiers

- Structure des données sur un disque / plusieurs disques
- Gestion des accès aux fichiers :
qui peut accéder quoi



© Pearson Education France

Fichiers : structure typique UNIX



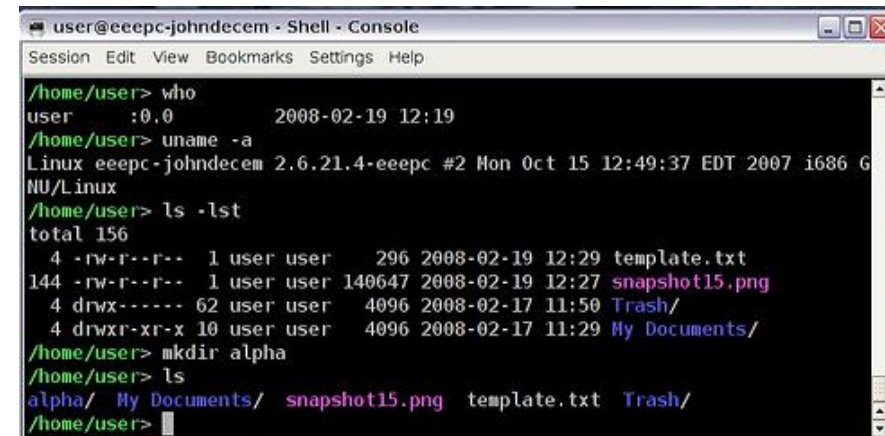
Gestion Entrée Sortie : lecteur disquette



- Beaucoup d'instructions bas-niveau:
 - Arrêter / démarrer le moteur
(pour éviter usure surface)
 - Déplacer la tête
 - Formatage
 - Calibration
 - etc.
- Programmeur est intéressé surtout
 - écrire
 - lire

3 Composantes du S.E.

- Noyau (partie centrale du système)
 - Accès à TOUTES les ressources
 - Analogie « *GodMode* »
- La librairie système (appels systèmes)
 - `exit()`, `read()`, `write()`
- Les programmes systèmes
 - `ls`, `rm`, `mv`, `who`,

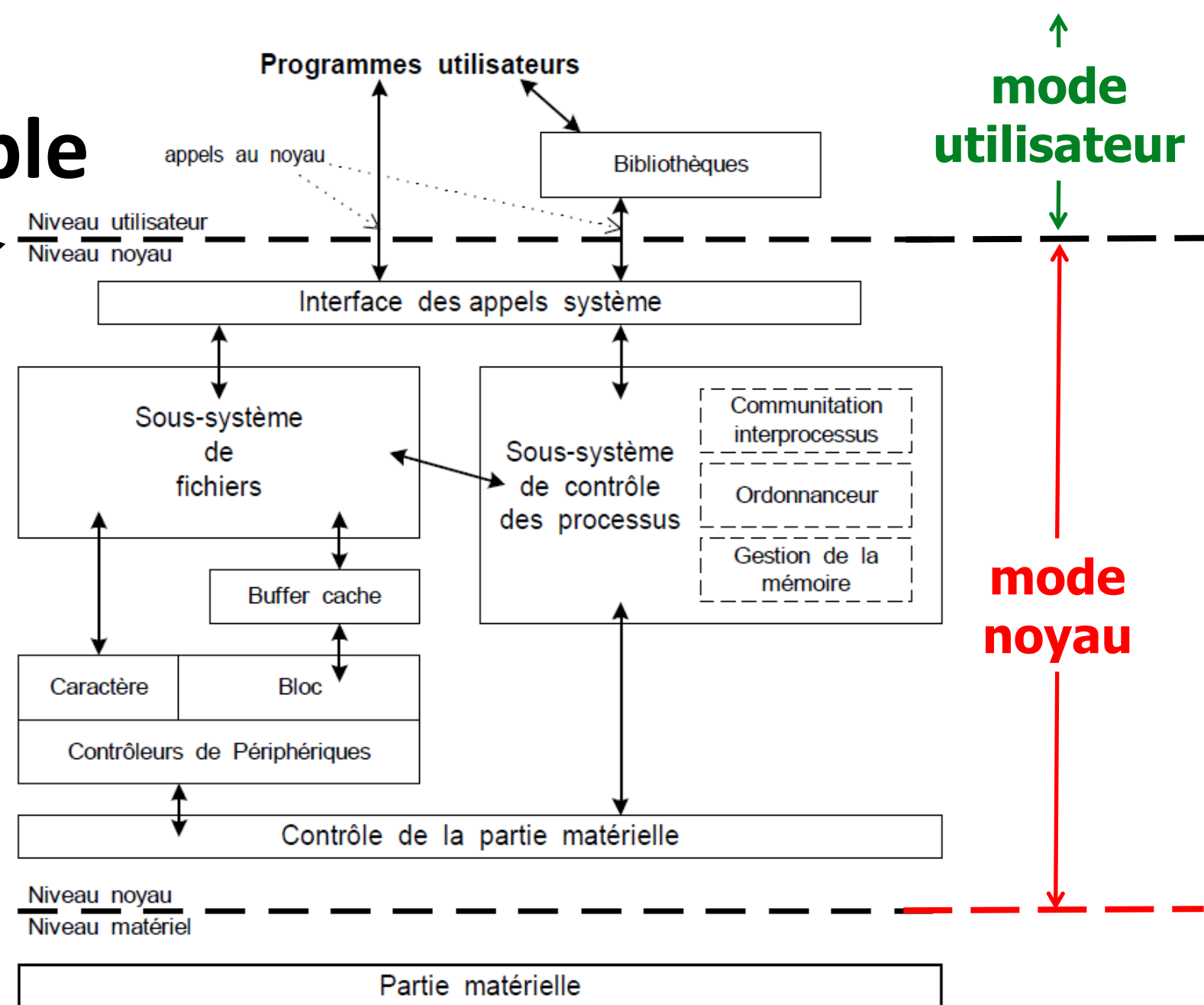


```
user@eeepc-johndecem - Shell - Console
Session Edit View Bookmarks Settings Help

/home/user> who
user      :0.0          2008-02-19 12:19
/home/user> uname -a
Linux eeepc-johndecem 2.6.21.4-eeepc #2 Mon Oct 15 12:49:37 EDT 2007 i686 GNU/Linux
/home/user> ls -l
total 156
 4 -rw-r--r--  1 user user    296 2008-02-19 12:29 template.txt
144 -rw-r--r--  1 user user 140647 2008-02-19 12:27 snapshot15.png
 4 drwx----- 62 user user   4096 2008-02-17 11:50 Trash/
 4 drwxr-xr-x 10 user user   4096 2008-02-17 11:29 My Documents/
/home/user> mkdir alpha
/home/user> ls
alpha/ My Documents/ snapshot15.png template.txt Trash/
/home/user>
```


Vue d'ensemble

« barrière »



Vue

d'ensemble

Programmes utilisateurs

appels au noyau

Bibliothèques

Niveau utilisateur
Niveau noyau

Interface des appels système

↑
**mode
utilisateur**
↓

« barrière »

CONCEPTS CLÉS

Communition
interprocessus

Ordonnanceur

Gestion de la
mémoire

Buffer cache

Caractère

Bloc

Contrôleurs de Périphériques

**mode
noyau**
↓

Contrôle de la partie matérielle

Niveau noyau
Niveau matériel

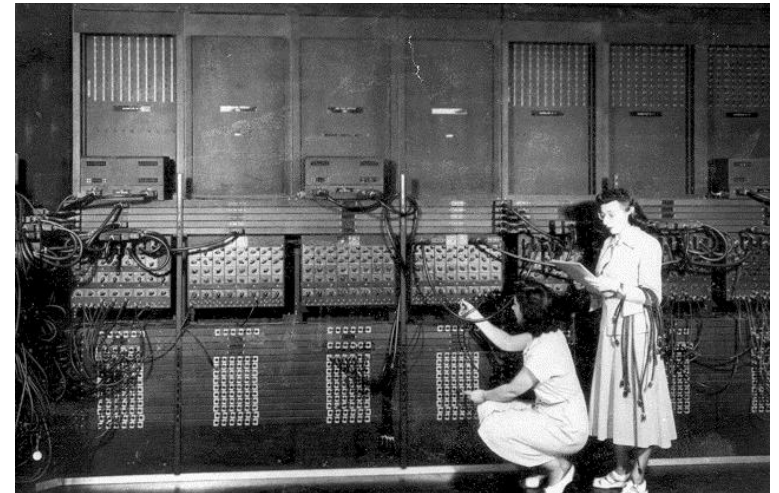
Partie matérielle

Historique

(Coévolution entre matériel et S.E.)

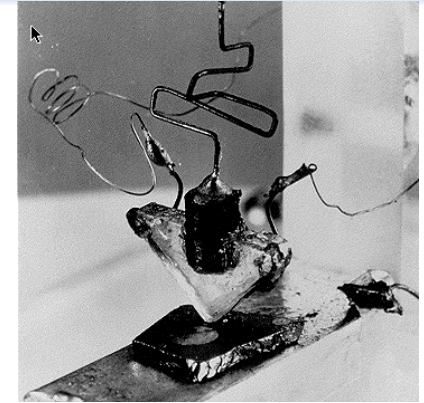
1^{ère} génération (1945-1955)

- **Tubes à vide**
- Eniac : 20,000 tubes
- horloge : 100 kHz
- durée record d'utilisation sans panne : 116 heures
- programmation par câblage
- aucun S.E., aucun langage
 - pas assez de mémoire
 - pas de périphérique



2^{ème} génération (1955-1965)

- **Transistors** : plus petits, fiable et économe en énergie
- Ère des *mainframes*
- Arrivée des S.E. / FORTRAN
- Traitement par lots

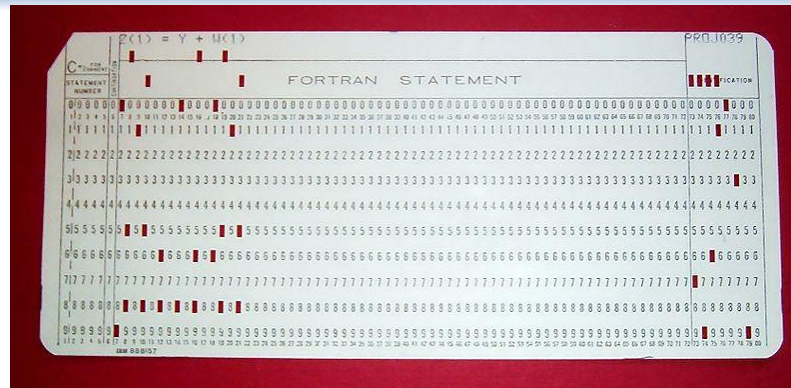


*premier prototype
de transistor*



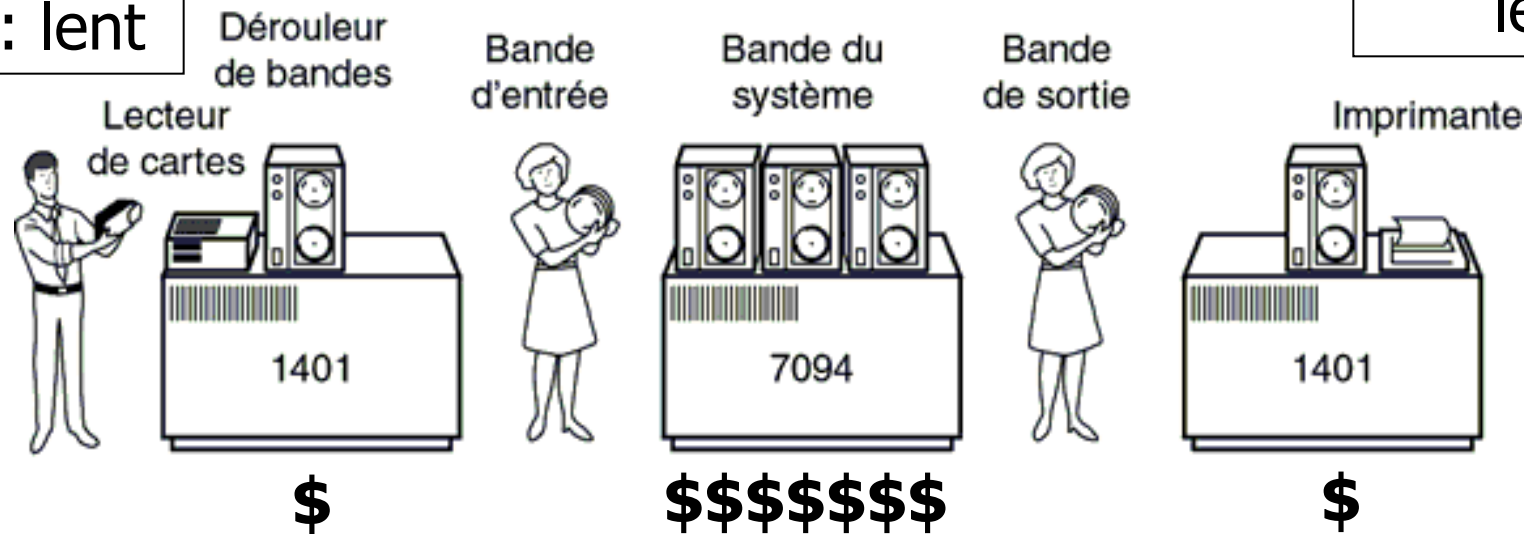
IBM 7094 : 32,768x36 mémoire

2^{ème} génération (1955-1965)



lecture
carte : lent

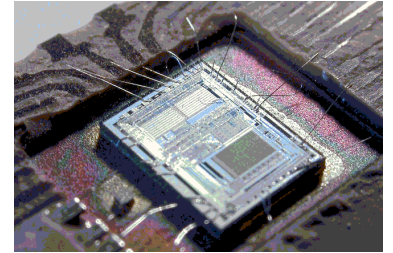
impression :
lent



Découpler entrées/sorties lentes de la machine principale

3^{ème} génération (1965-1980)

- **Circuits intégrés (CI)**
- Début 60, la NASA achetait 60% des CI.
- IBM lance famille d'ordinateur *System/360*
 - Multiprogrammation (+ d'un programme chargé)
 - Temps partagé (plusieurs utilisateurs en même temps)
- Concepteur du OS/360 : Fred Brooks
 - loi de Brooks : « *Ajouter des ressources humaines à un projet en retard sur les prévisions ne fait qu'accentuer ce retard.* »



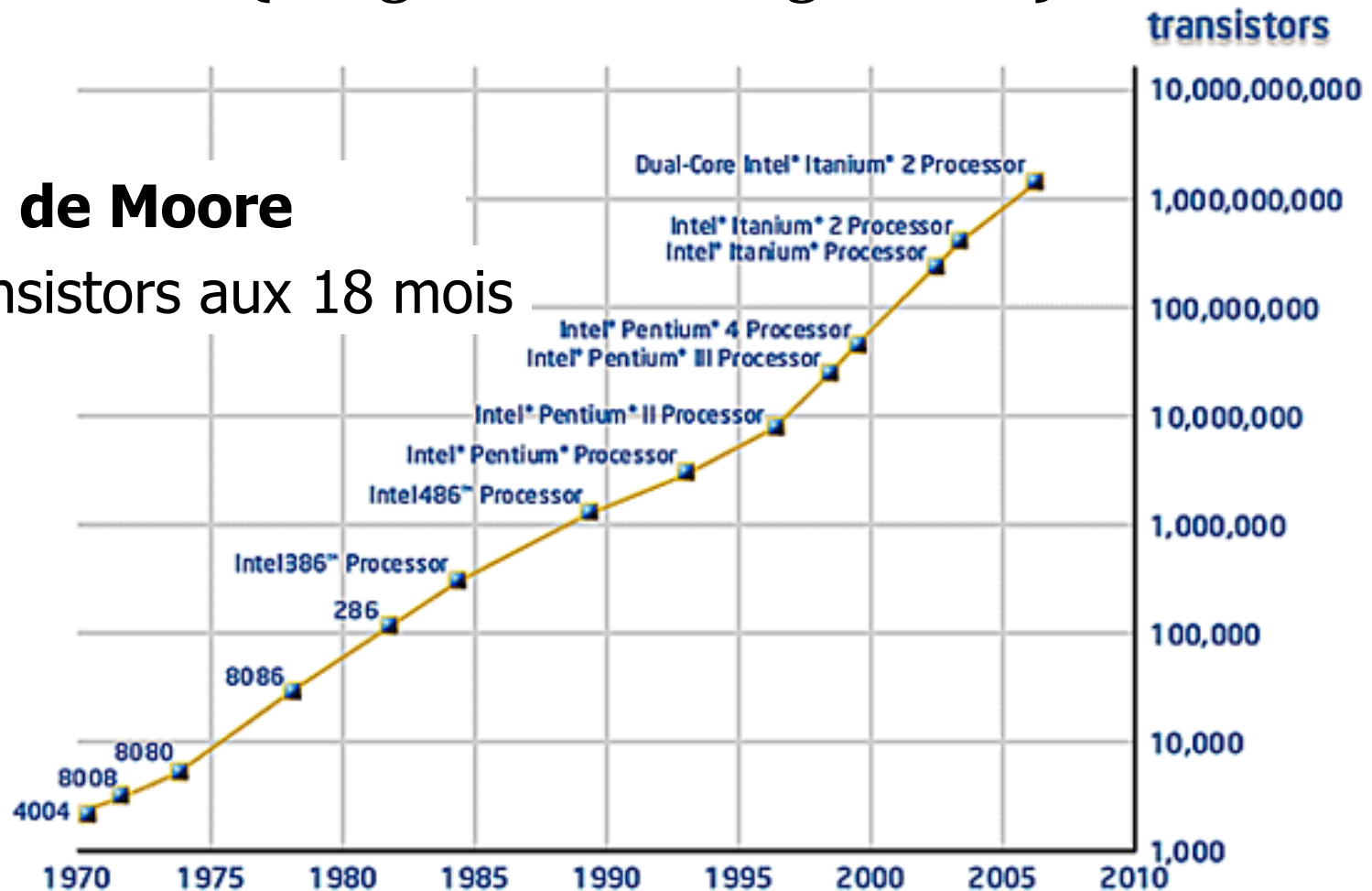
The mythical man-month

4^{ème} génération : ordinateurs personnels

- Circuits LSI (*Large Scale Integration*)

Loi de Moore

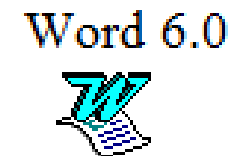
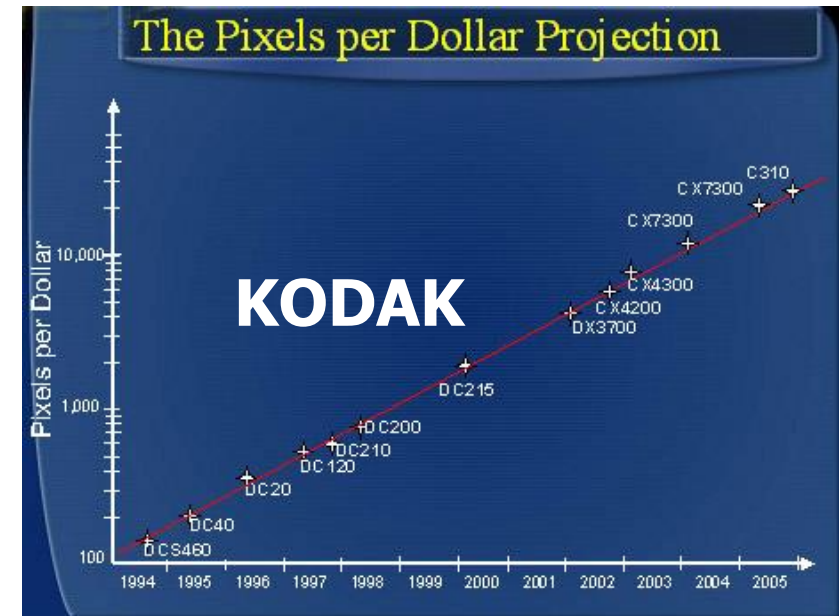
double transistors aux 18 mois



D'autres lois exponentielles

- Mémoire vive
- Disques durs
- Bande passante
- Pixels/dollars pour caméra
- À l'inverse :
 - loi de Wirth :

« le logiciel ralentit plus vite que le matériel n'accélère »



VS.



4^{ème} génération : ordinateurs personnels

- Prend sa source avec CPU 8080 d'*Intel*
- Apple, Commodore, Atari
- MS-DOS



IBM PC



Apple II

Lignées de Windows

- Windows 1.0 (non préemptible)
- Windows 2.0 (non préemptible)
- Windows 3.0/3.1/3.51 (non préemptible)
- Windows 95/98/Me
- Windows NT/2000/XP/Vista/7/8/10