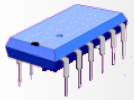
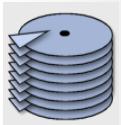
Sommaire



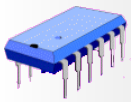


Qu'est-ce qu'un système d'exploitation ?

Gestion des processus

Gestion de la mémoire

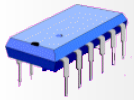
Fichiers et systèmes de gestion de fichiers

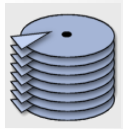


Qu'est-ce qu'un système d'exploitation ?

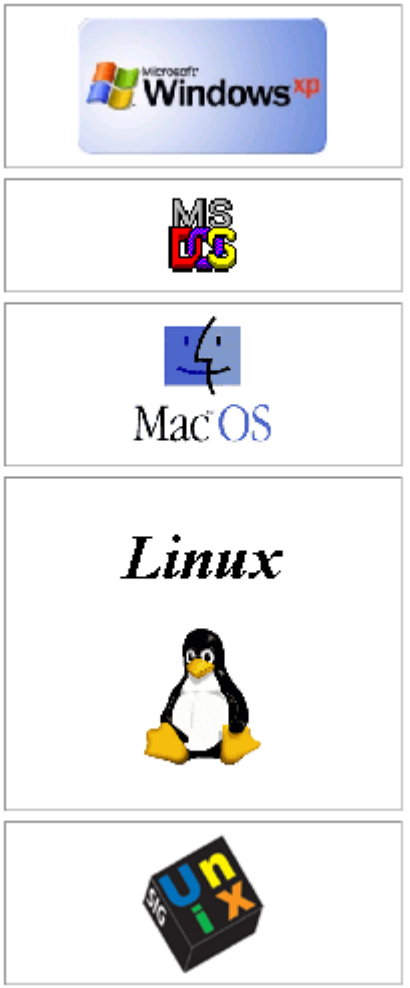
*OS: Operating System

Quelques exemples





Les principaux

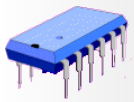


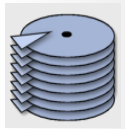
L'OS n'est pas forcément lié au type de machine !

On peut faire tourner Linux, Unix, Windows ou BeOs sur un PC !

Introduction

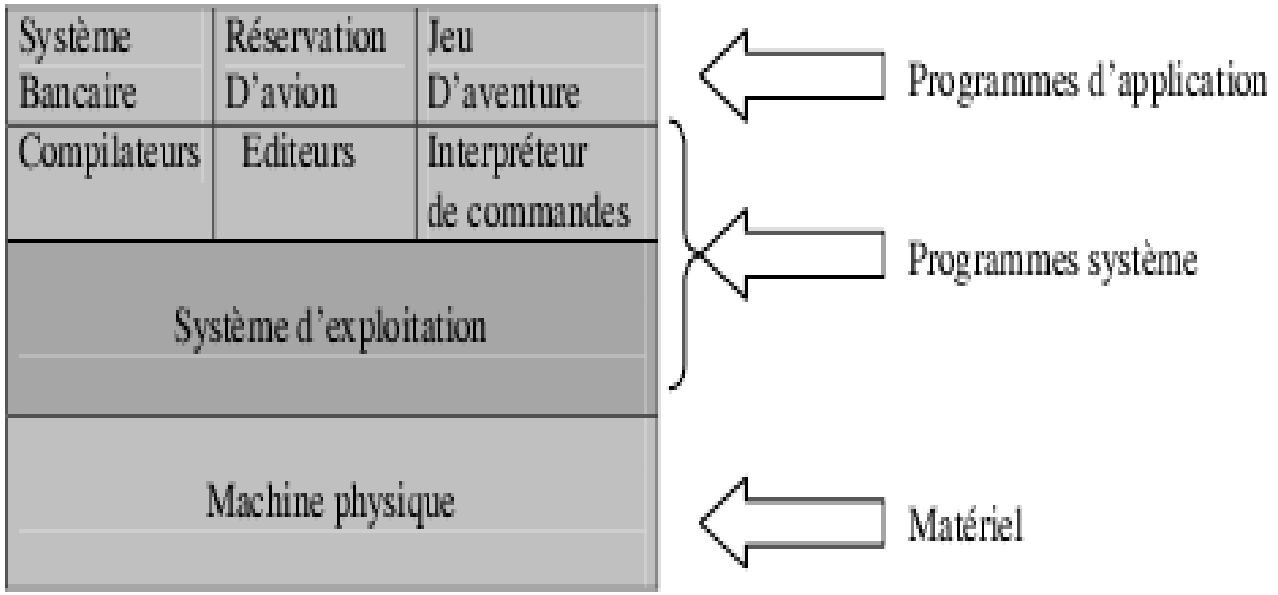
- Problèmes de départ
  - Matériels variés (processeurs, quantité de mémoire, taille des disques durs, affichage ...)
  - Plusieurs programmes doivent pouvoir être exécutés en même temps
  - Plusieurs personnes peuvent utiliser la machine





- L'écriture d'un programme doit être libérée de ces contraintes.

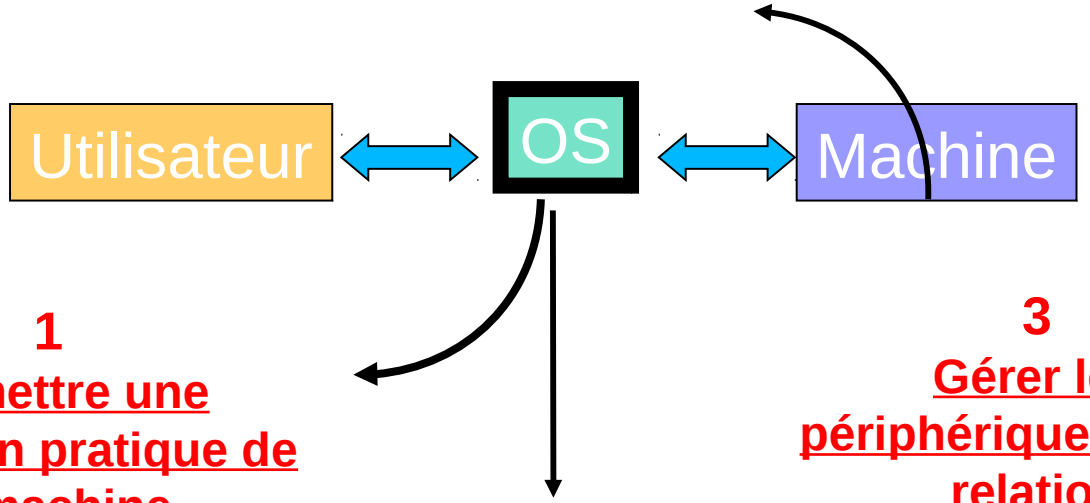
Rôles du système d'exploitation



Réalise l'interface entre le matériel et le programmeur / utilisateur

- contrôle les ressources de l'ordinateur
- fournit la base sur laquelle seront construits les programmes d'application

Les différents mode d'action



1
Permettre une utilisation pratique de la machine

Assurer un ensemble de services en présentant aux utilisateurs une interface mieux adaptée à leurs besoins que celle de la machine physique

2
Gérer et partager les ressources

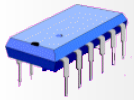
3
Gérer les périphériques et leur relation

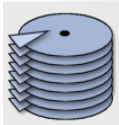
Gérer les ressources de l'installation matérielle en assurant leurs partages entre un ensemble plus ou moins grand d'utilisateurs

Au démarrage : le BIOS

BIOS (Basic Input Output System)

- détecte les périphériques
- initialise le matériel (registres processeur, mémoire etc.)
- scanne les bus pour trouver un périphérique amorçable (bootable)
- charge le système d'exploitation en mémoire





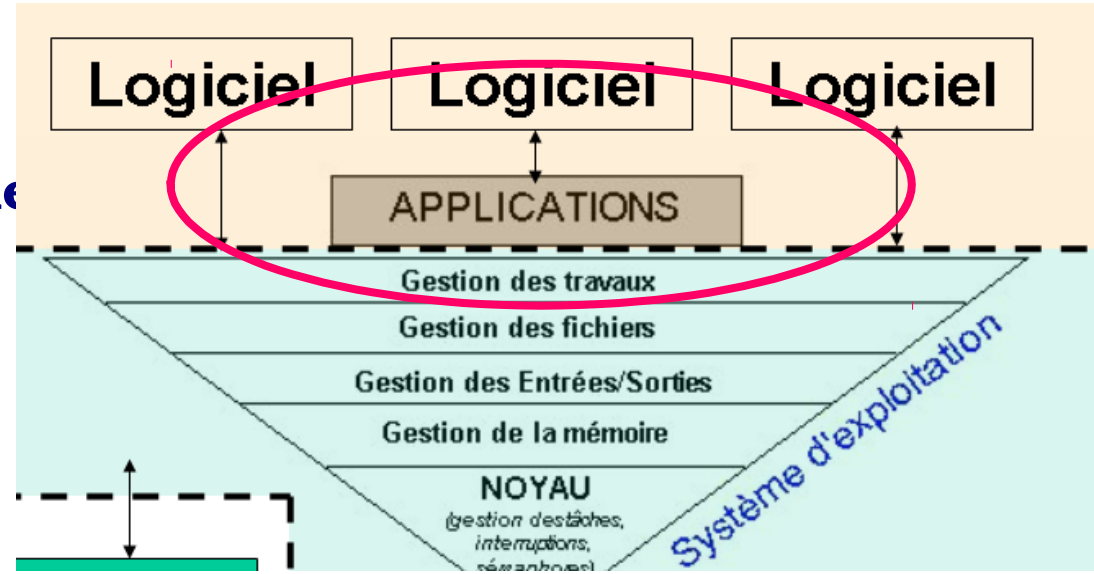
Système d'Exploitation

- re-détecte les périphériques, les ré-initialise
- lance le premier processus
- attend un événement

Au plus haut niveau, les applications

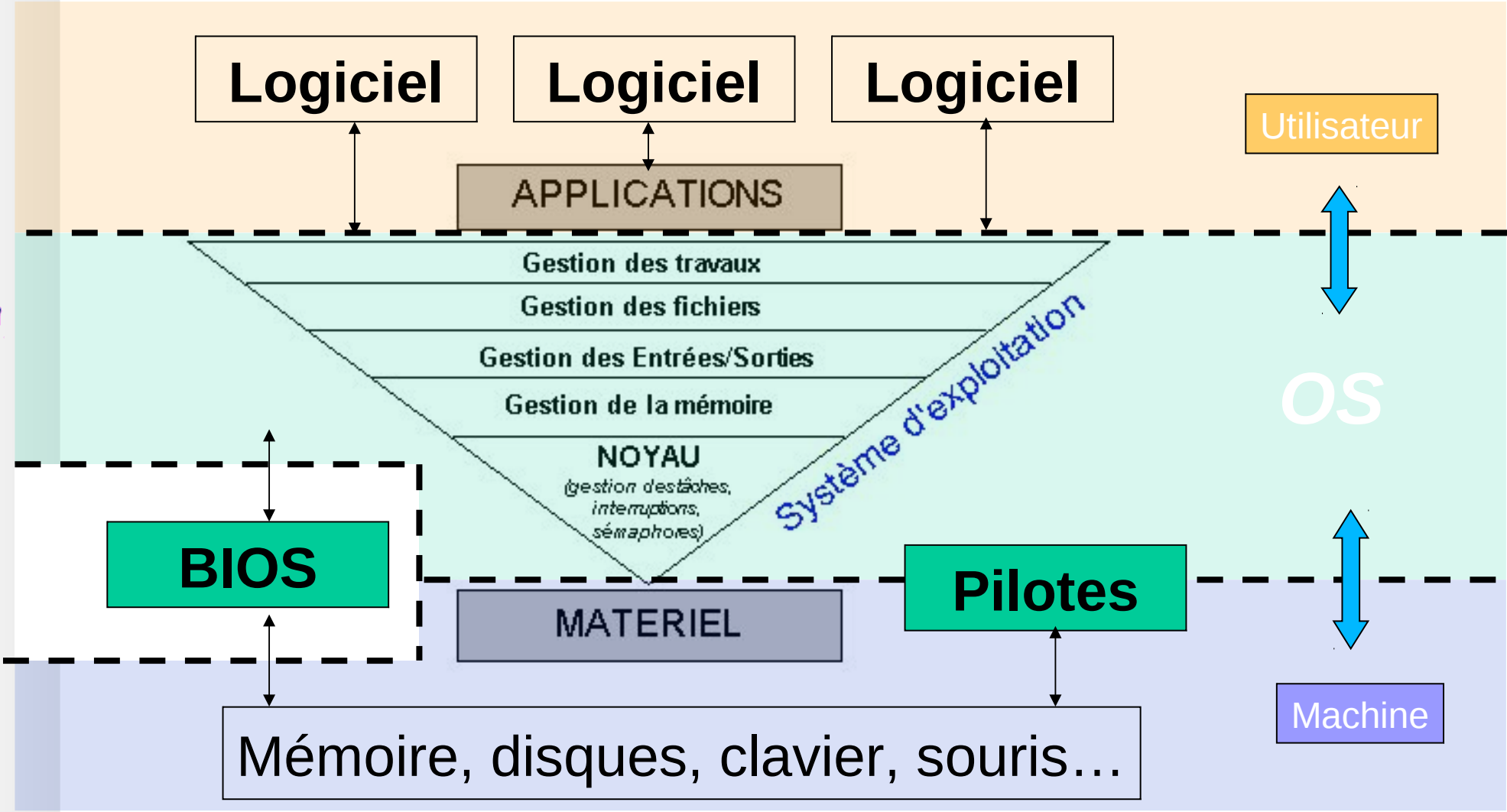
Elles s'appuient sur l'OS (utilisent des fonctions spécifiques de l'OS) pour **faciliter l'exécution de tâches complexes**

Ex :
OpenOffice
Firefox, Word…

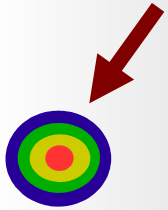


*Une application est écrite **spécifiquement** pour un OS (« compilée » pour cet OS) et **ne fonctionne pas sur un autre** (cf Word, Internet Explorer…) Elle peut être éventuellement « **portée** » sur un autre OS (OpenOffice, Mozilla…)*

Vue générale

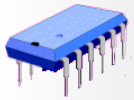


Les composants de l'OS

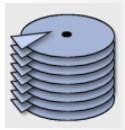


Les systèmes d'exploitation comportent plusieurs composants qui permettent de les différencier

- **Le noyau** (processus, périphériques, mémoire, fichiers)

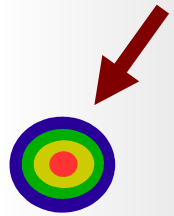


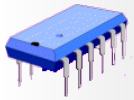
- **L'interpréteur de commandes** et **l'environnement graphique**



- **Le système de fichiers** (et des arborescences)

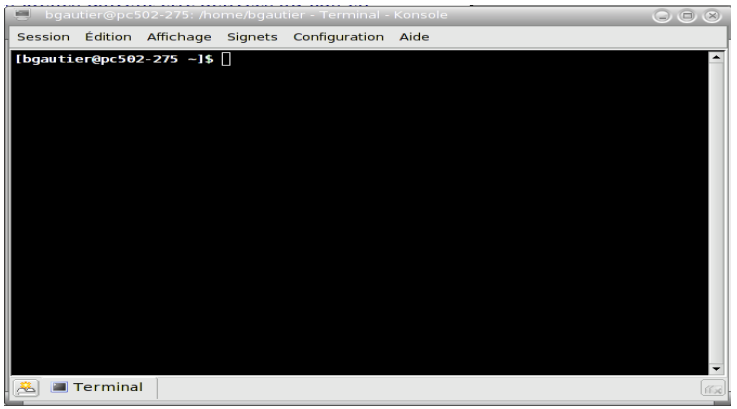
Le noyau

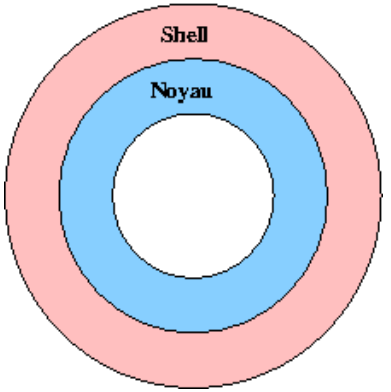




- Gestion des Processus
- Communication avec les Périphériques
- Gestion de la Mémoire
- Gestion des accès fichiers

L'interpréteur de commandes





- **Interface de base permettant de commander le système = shell**

  Sous Windows : DOS (Disk Operating System)

  Sous Unix/Linux : bash, c-shell, ksh...

Ex : lister le contenu d'un répertoire :
DOS : commande **dir**
Linux : commande **ls**

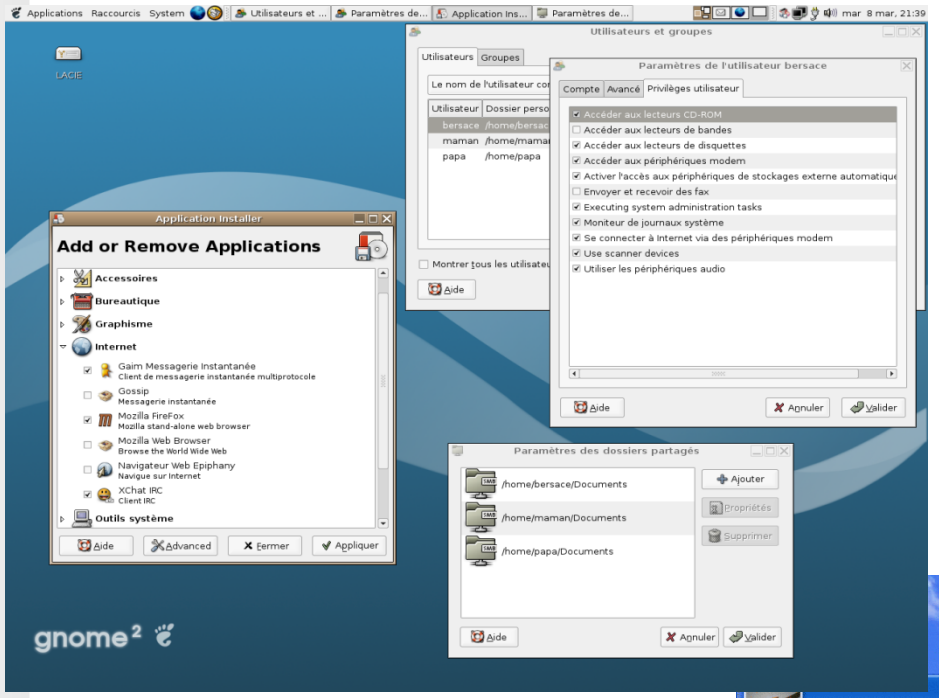
Interface graphique

Systèmes à fenêtres et menus

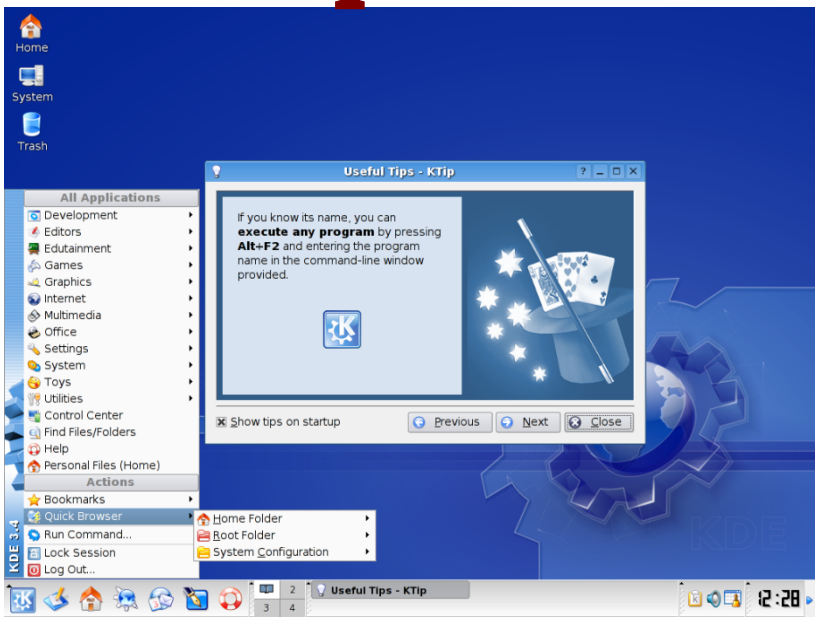
Sous Windows : Une interface unique

Sous Unix/Linux : KDE (utilisé à l'INSA), Gnome, WindowMaker, Enlightment, BlackBox...

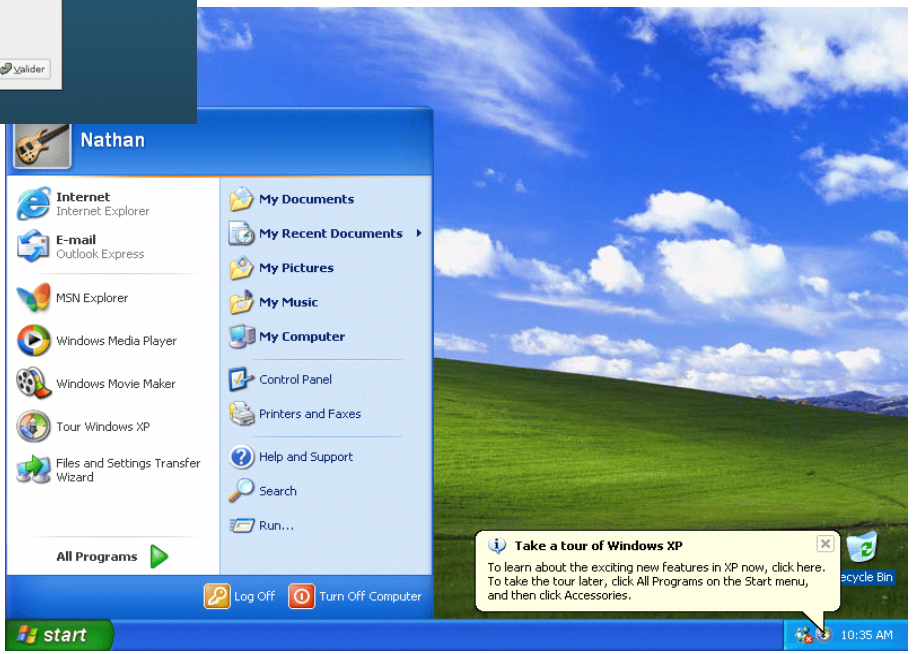
Interface graphique : exemples



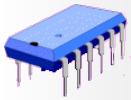
Gnome

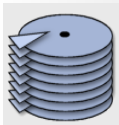


KDE

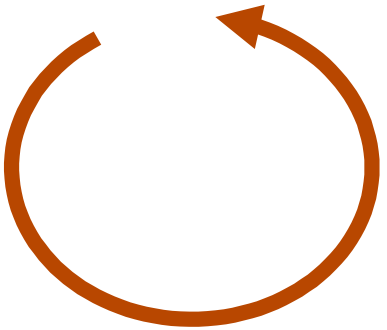


Windows XP





Processus et gestion de processus

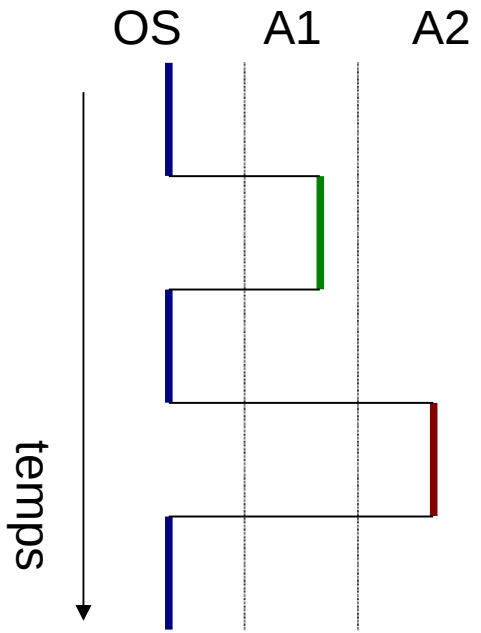


La notion de processus

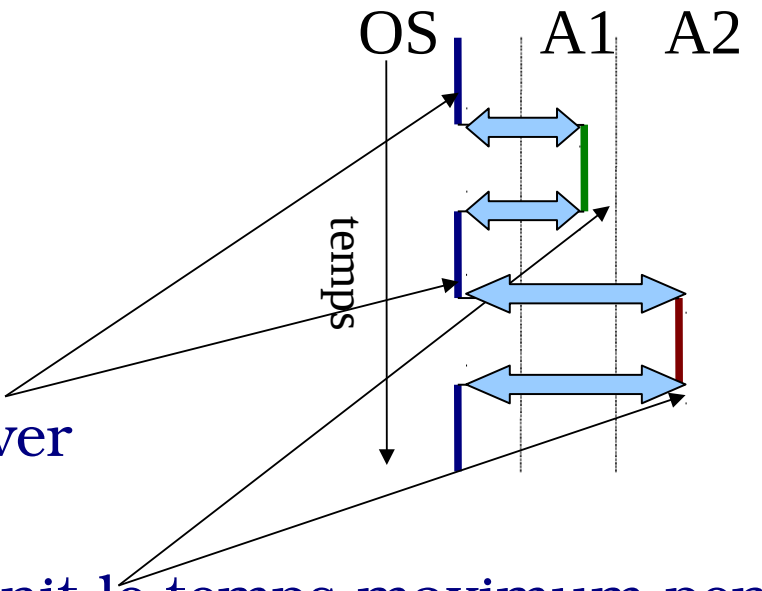
- **Programme qui est en train d'être exécuté**
- **Environnement multi-tâches sur une machine mono-processeur:**

Le système arbitre l'exécution des tâches

Il reprend la main entre les tâches, ou à leur demande, pour mettre en place l'illusion du multi-tâches.



Organisation des processus



Scheduler (ordonnanceur) :

liste les processus prêts

choisit le processus à activer

Dispatcher (allocateur) : Définit le temps maximum pendant lequel un processus pourra rester actif. Passe le contrôle au scheduler lorsque ce temps est écoulé

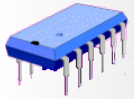
Passages OS ⟺ Application appelés **changement de contexte**.

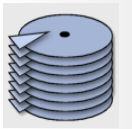
Priorité d'accès au processeur : appelée **gentillesse** sous Unix

Liste des processus

En pratique :

- **Sous Windows 2000, accès au gestionnaire de tâches par Alt/Ctrl/Suppr**
- **Sous Unix/Linux, commande *ps* ou *top*** (liste actualisée classée de la plus gourmande en CPU à la moins gourmande)





USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
...										
veglin	18639	0.0	2.8	27500	14636	?	S	21:22	0:00	konsole
veglin	18644	0.0	0.3	3320	1972	pts/1	S	21:22	0:00	/bin/bash
veglin	19635	0.2	3.5	54508	18472	?	S	21:25	0:01	kpdf
veglin	22951	0.0	0.1	2576	824	pts/1	R	21:37	0:00	ps aux

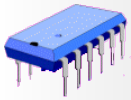
commande

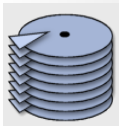
Temps processeur utilisé

Etat : sleeping or running

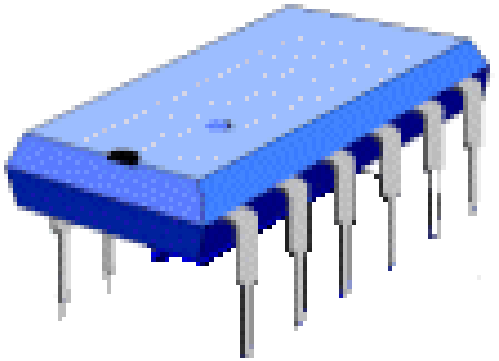
Port auquel est relié le processus, indique si un terminal est attaché au processus

Chaque processus est repéré par son numéro : PID (process identifier)





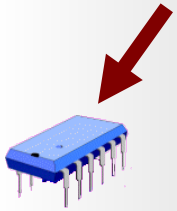
Gestion de la mémoire
et
Mémoire virtuelle



Gestion de la mémoire

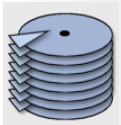
Le système gère la mémoire et en attribut des morceaux aux applications (programme et données).

Il doit également disposer d'un espace pour stocker ses propres données (tables, contextes des applications, ...)



Le système doit présenter une image de la mémoire aux applications indépendante de son mode de gestion:

- des applications ne doivent pas pouvoir manipuler les données d'autres applications

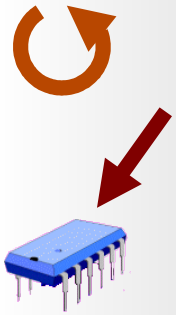


- une application est écrite avec des variables et l'OS doit leur associer des adresses mémoire

- il est parfois nécessaire de faire croire à une application qu'elle dispose de plus de mémoire qu'il n'existe réellement.

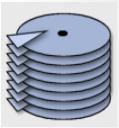
La pagination et les adresses virtuelles

La mémoire physique est découpée en morceaux, appelés *pages*.

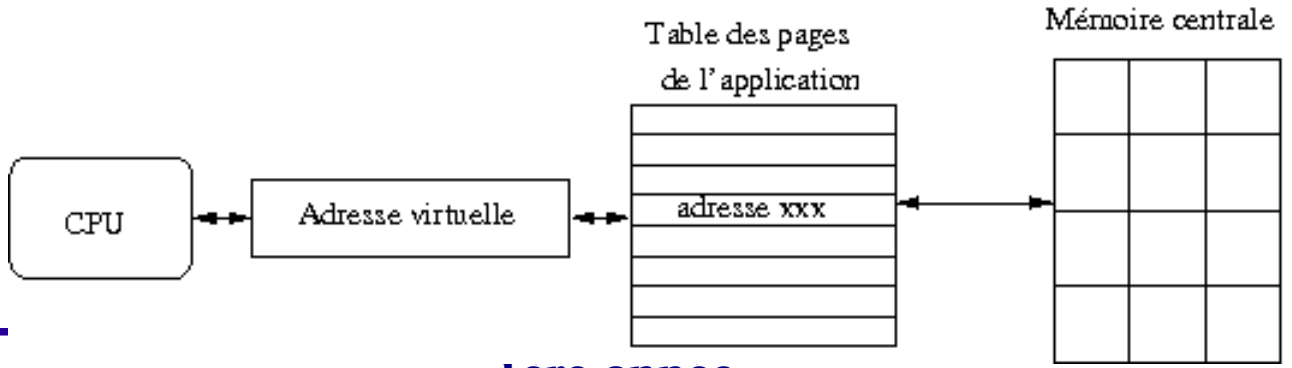


Le système gère les tables de pages attribuées aux applications.

Un mécanisme de traduction est associé a cette table : les adresses vues par l'application ne sont pas les adresses réelles. Ce mécanisme est appelé *adressage virtuel*.

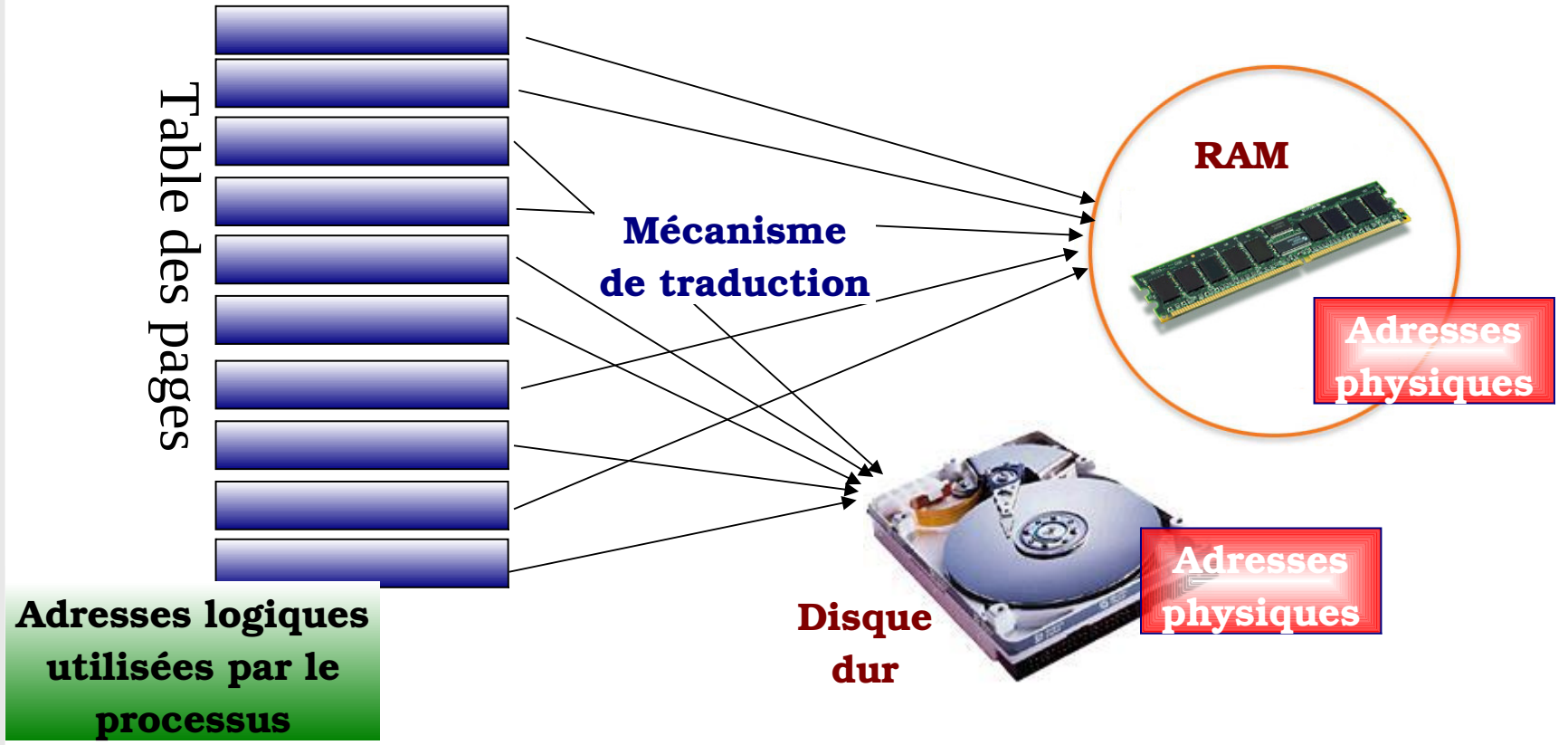


Chaque application a sa propre table des pages (c'est une partie de son contexte). Plusieurs applications peuvent manipuler les mêmes adresses virtuelles.

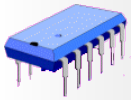


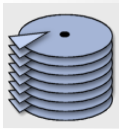
La mémoire virtuelle

Le système peut, temporairement, copier des pages de la mémoire vers le disque dur. Ce mécanisme est le *swap*.

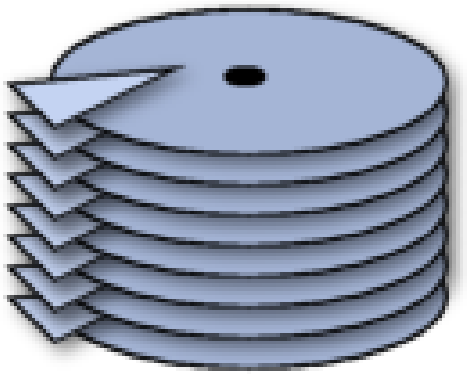


L'utilisation du swap permet d'étendre la mémoire mais se paye en rapidité d'exécution ! (accès au disque plus lent qu'à la RAM, gestion du swap)





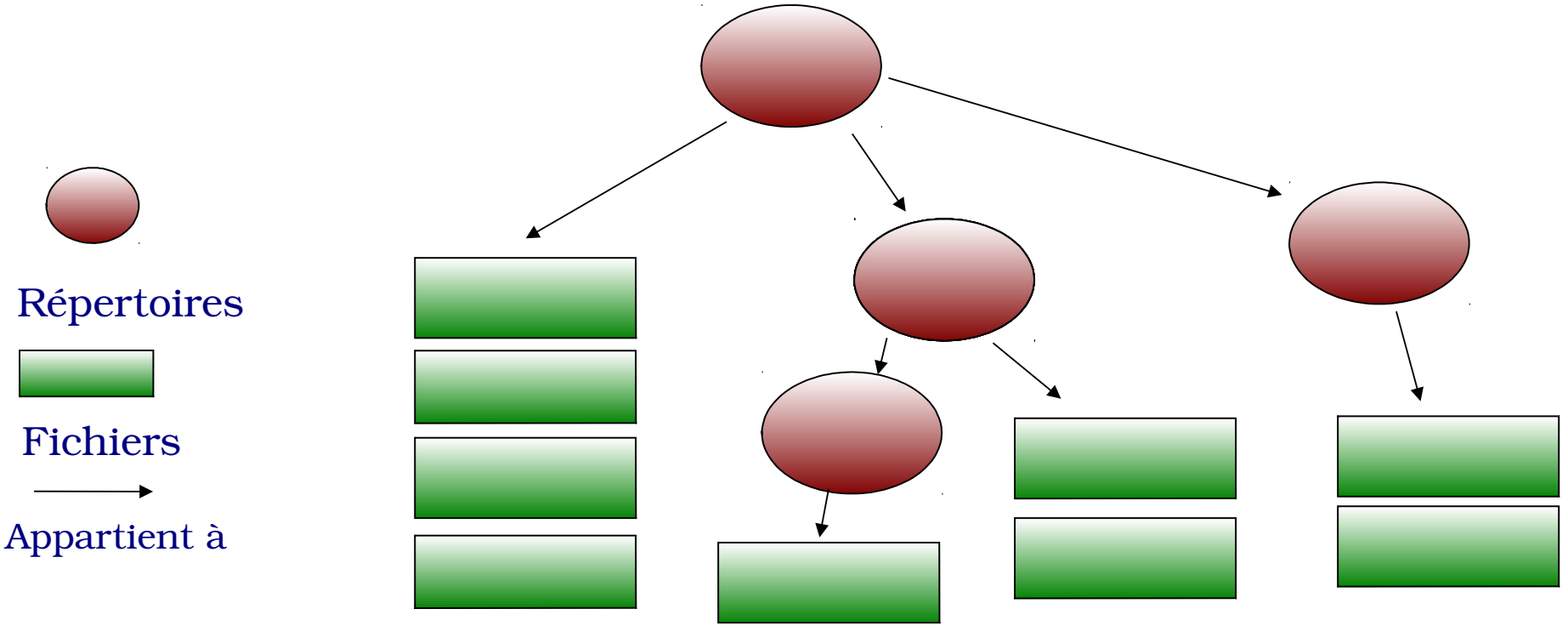
Fichiers et arborescences



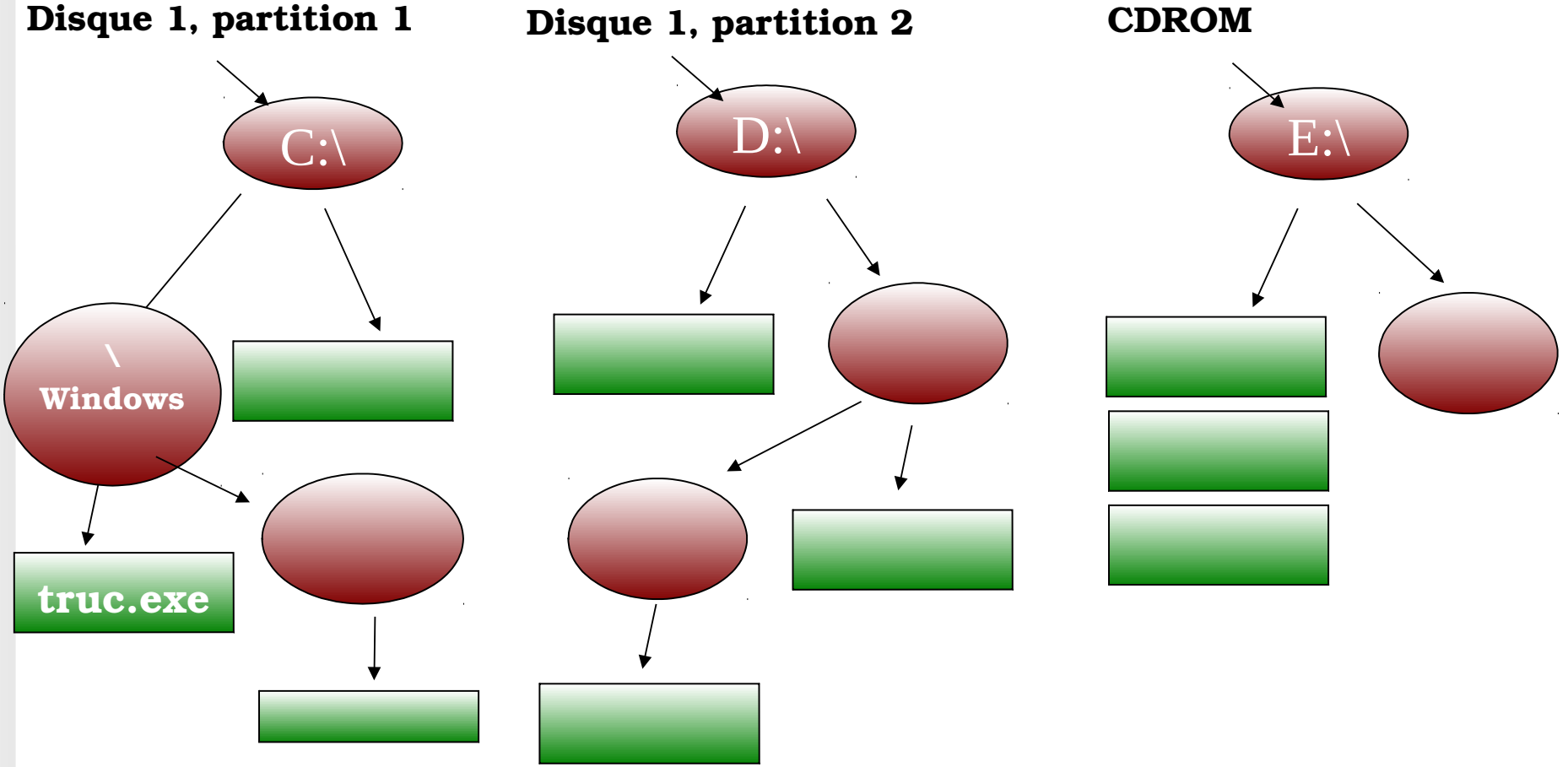
L'arborescence

- **Un répertoire est une collection de fichiers ou de répertoires (appelés alors sous-répertoires)**

Fichiers et répertoires sont regroupés dans une structure **arborescente**: chaque fichier connaît son père (le répertoire auquel il appartient). L'arborescence peut être parcourue dans les deux sens.



L'arborescence Windows

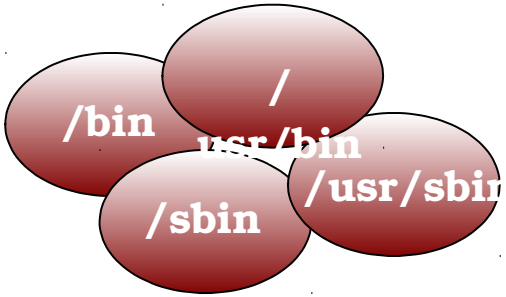


Une arborescence par unité de disque physique ou logique (par partition).

Ex : C:\Windows\truc.exe (Attention au : « \ » !)

L'arborescence Linux

- **L'arborescence Unix est plus ou moins standardisée**



Commandes et utilitaires du système

/etc — **Fichiers de configuration et adminstration du système**

/usr — **Programmes additionnels aux systèmes + installés par les utilisateurs**

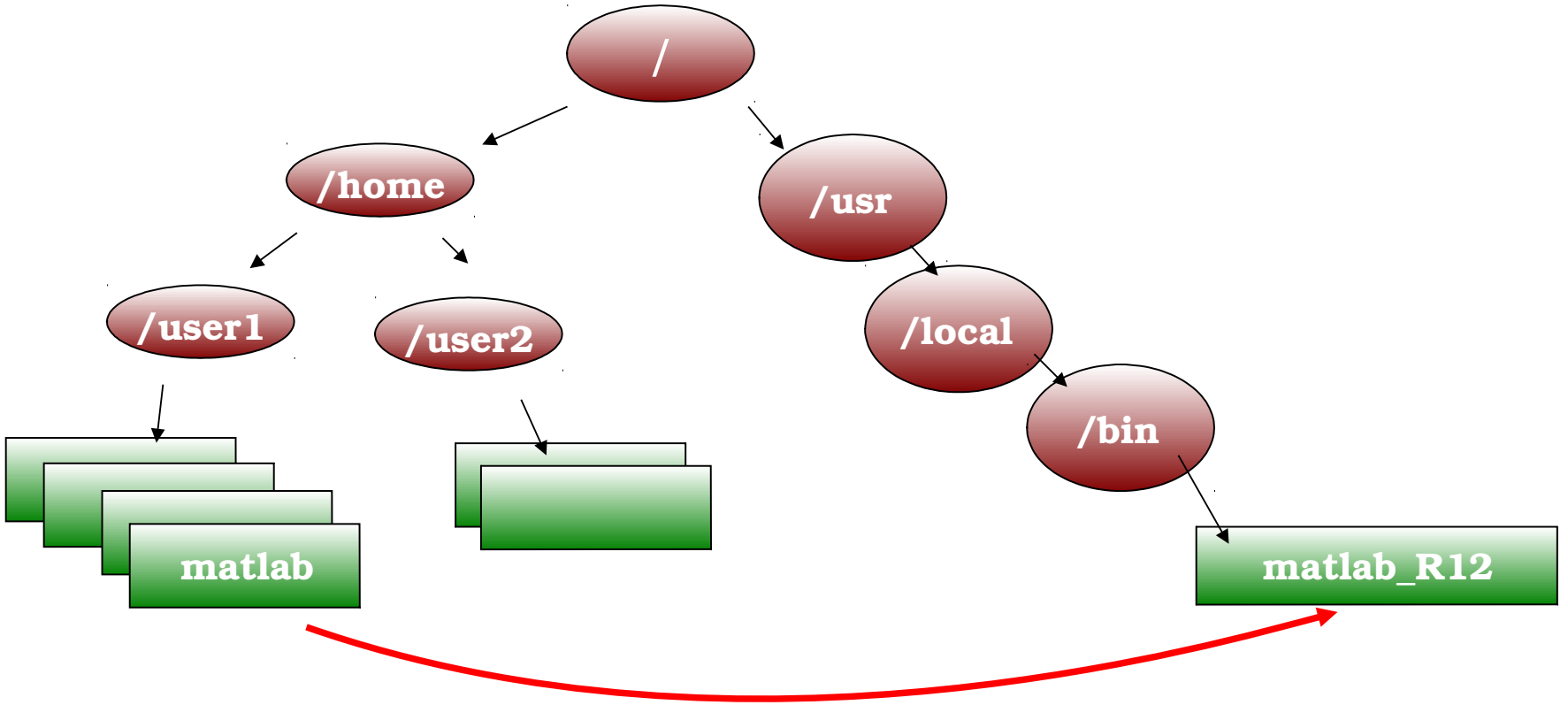
/dev — **Périphériques système**

/tmp — **Fichiers temporaires du système**

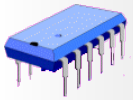
/home — **Répertoires personnels :**
1 par compte, porte le nom du login
exemple : /home/aeinstein pour l'utilisateur Albert Einstein

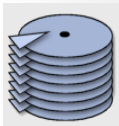
Les liens symboliques

- Ils permettent d'assouplir le parcours de l'arborescence (pas besoin de remonter au répertoire parent)

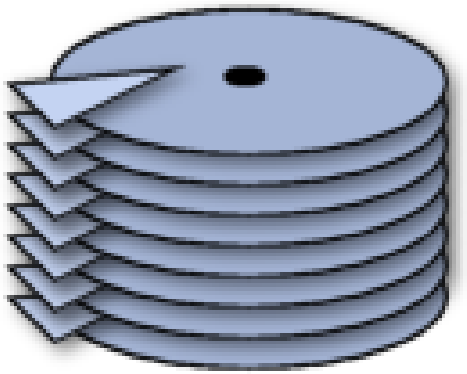


On les nomme « raccourcis » sous Windows.





Gestion des fichiers et droits d'accès



Permissions sur un fichier : exemple d'Unix/Linux

Certains systèmes de fichiers gèrent des permissions sur les fichiers :

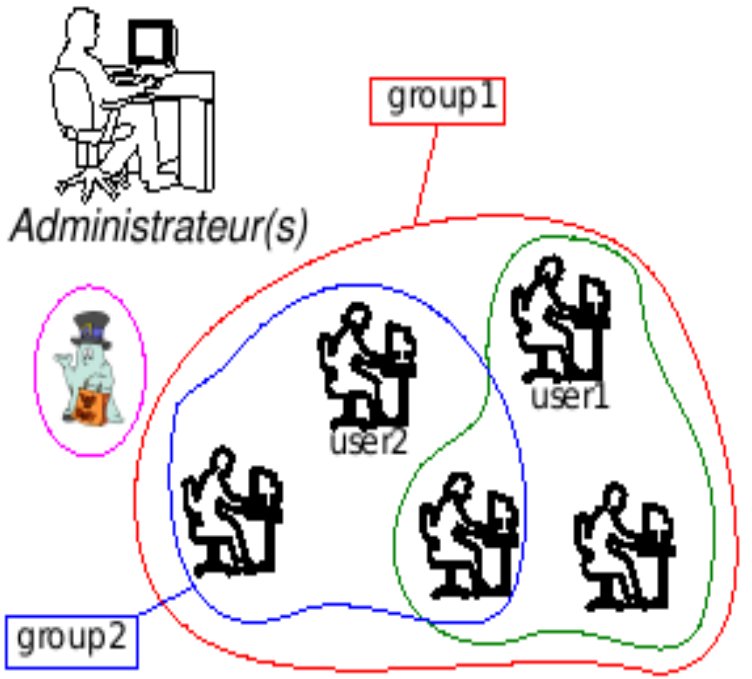
Lecture (read : r), **Ecriture** (write : w), **Exécution** (execute : x)

Un fichier peut accorder des permissions à 3 types d'utilisateurs :

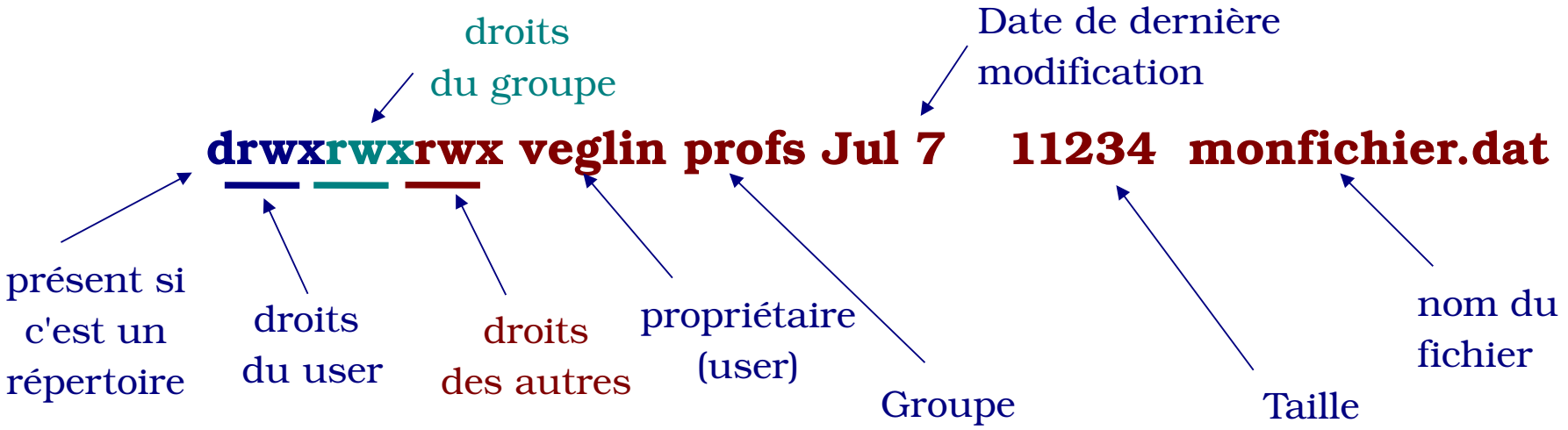
Le propriétaire du fichier (**user** : par défaut, celui qui l'a créé) : identifié par son login (compte)

Le groupe (group) auquel le propriétaire appartient

Les autres (others)



Permissions sur un fichier : exemple d'Unix/Linux



Exemple :

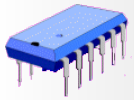
_ rwx r _ x r_ _ veglin profs Jul 7 11234 monfichier.dat

→ Commande **ls -l** sous invite de commande pour voir les permissions sur un fichier

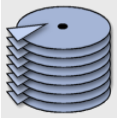
En résumé

- Le système d'exploitation est le premier programme démarré par la machine

- Il permet de s'affranchir de la gestion fine du matériel et propose des services de haut niveau tels que des processus, de la mémoire et des périphériques.



- Ces mécanismes sont souvent très compliqués comme illustré sur la gestion des processus ou de la mémoire



- Chaque système propose sa propre organisation et ses propres services. Les applications doivent être écrites pour un système d'exploitation donné.

- Les performances d'un système d'exploitation sont essentielles pour le bon fonctionnement d'une machine.