

Introduction aux système d'exploitation

- Définition et rôle d'un système d'exploitation
- Les types de système d'exploitation
- Les fonctions du système d'exploitation
- Notions de base

Définition et rôle d'un système d'exploitation

Définition d'un système d'exploitation

Le **système d'exploitation** est un ensemble de programmes qui réalisent l'**interface** entre le **matériel** de l'ordinateur et les **utilisateurs**.

Il a deux objectifs principaux :

- construction au dessus du matériel d'une machine virtuelle plus facile d'emploi et plus conviviale
- prise en charge de la gestion de plus en plus complexe des ressources et partage de celle-ci

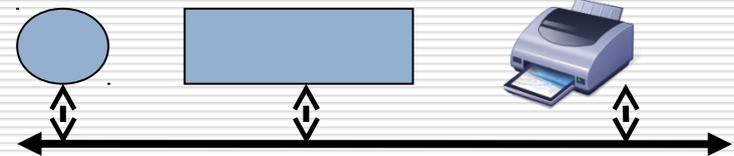
Comme son nom le suggère, le SE a en charge l'exploitation de la machine pour en faciliter l'**accès**, le **partage** et pour l'**optimiser**



Logiciels utilisateurs
-Navigateur (IE, Firefox)
-Traitement de texte, tableur (packoffice, openoffice)
- Messagerie (webmail, outlook, thunderbird)
- Jeux

Système d'exploitation

Cpu mémoire centrale périphériques



Machine Matérielle

Rôle : faciliter la machine physique

Construction au dessus du matériel d'une machine virtuelle

ECRIRE(données)

Exécuter (Programme)

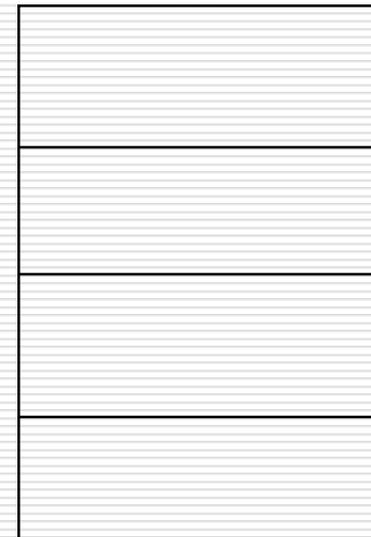
Initialiser cpu E/S

Gestion par IRQ

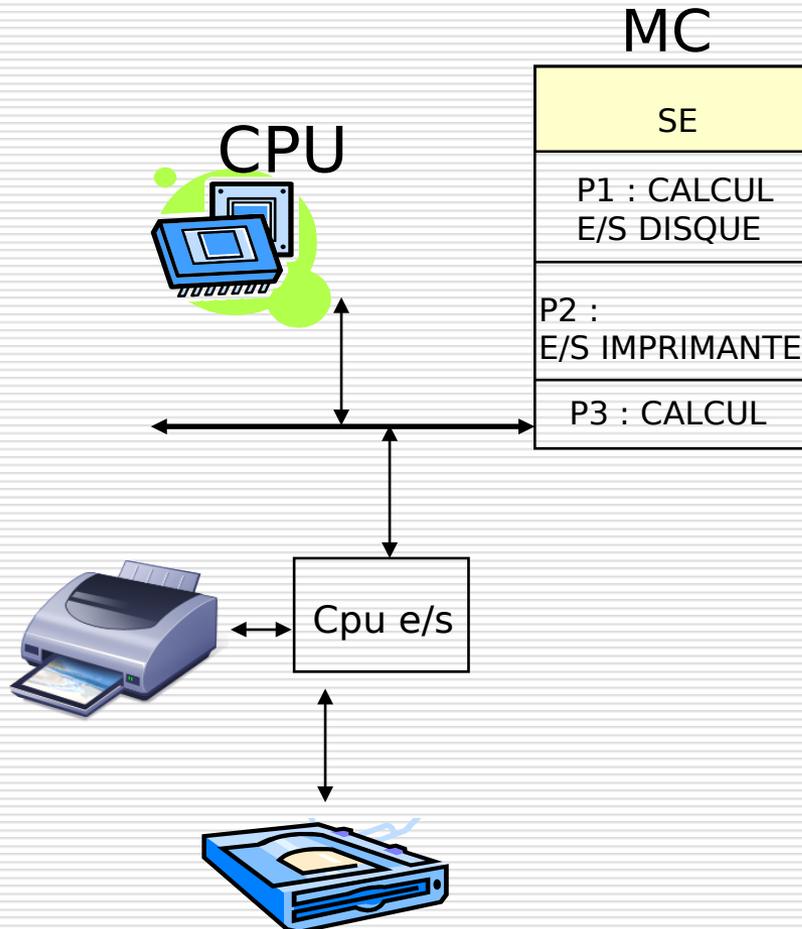


Charger le programme en MC
(pagination / segmentation...)

MC



Prise en charge de la gestion et du partage des ressources



- Plusieurs programmes en MC
 - le DMA gère les entrées/sorties
 - parallèlement, le CPU effectue les "calculs"
- Partager la machine physique
 - le processeur : qui s'exécute ?
 - la mémoire centrale :
 - protection SE/ Programmes Utilisateurs
 - protection entre programmes Utilisateurs
 - les périphériques

Rôle d'un système d'exploitation



ECRIRE(données)

MACHINE VIRTUELLE

Exécuter (Programme)

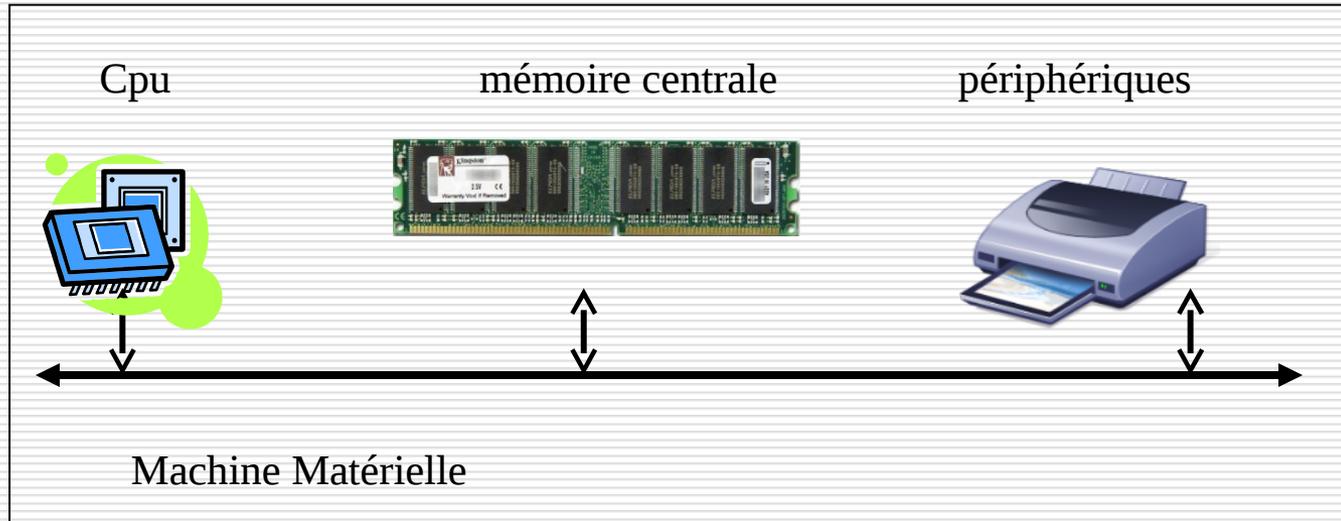
SE

Interface pour masquer les caractéristiques matérielles

Appels Système

Commandes

Matériel
MACHINE
PHYSIQUE



Les types de système d'exploitation

Les programmes sont exécutés **en différés les uns à la suite des autres.**

- Pas de concurrence, la mise en œuvre de l'utilisation du processeur, de la mémoire et des périphériques est simple
- Puisque le processeur est nettement plus rapide que les périphérique , il est inutilisé 90 % du temps
- Pour des applications nécessitant de très gros calculs mais peu d'implication de l'utilisateur

Exemples d'applications :

- Météo
- Statistiques
- Impôts

Plusieurs programmes sont **simultanément en cours d'exécution**.

- L'utilisation du processeur est partagée par planification
- Le système d'exploitation comporte un mécanisme de protection
- Les programmes sont exécutés dans un mode non-privilegié, dans lequel l'exécution de certaines instructions est interdite

Ils sont souvent **multi-utilisateurs** : plusieurs utilisateurs sont simultanément en dialogue avec l'ordinateur. Il faut donner à chaque utilisateur l'impression qu'il est le seul à utiliser l'ordinateur

Exemples d'applications :

- Linux
- Unix

Systemes temps réel (réactifs)

Le **temps d'exécution** des programmes est contraint par une **date butoir** appelée échéance.

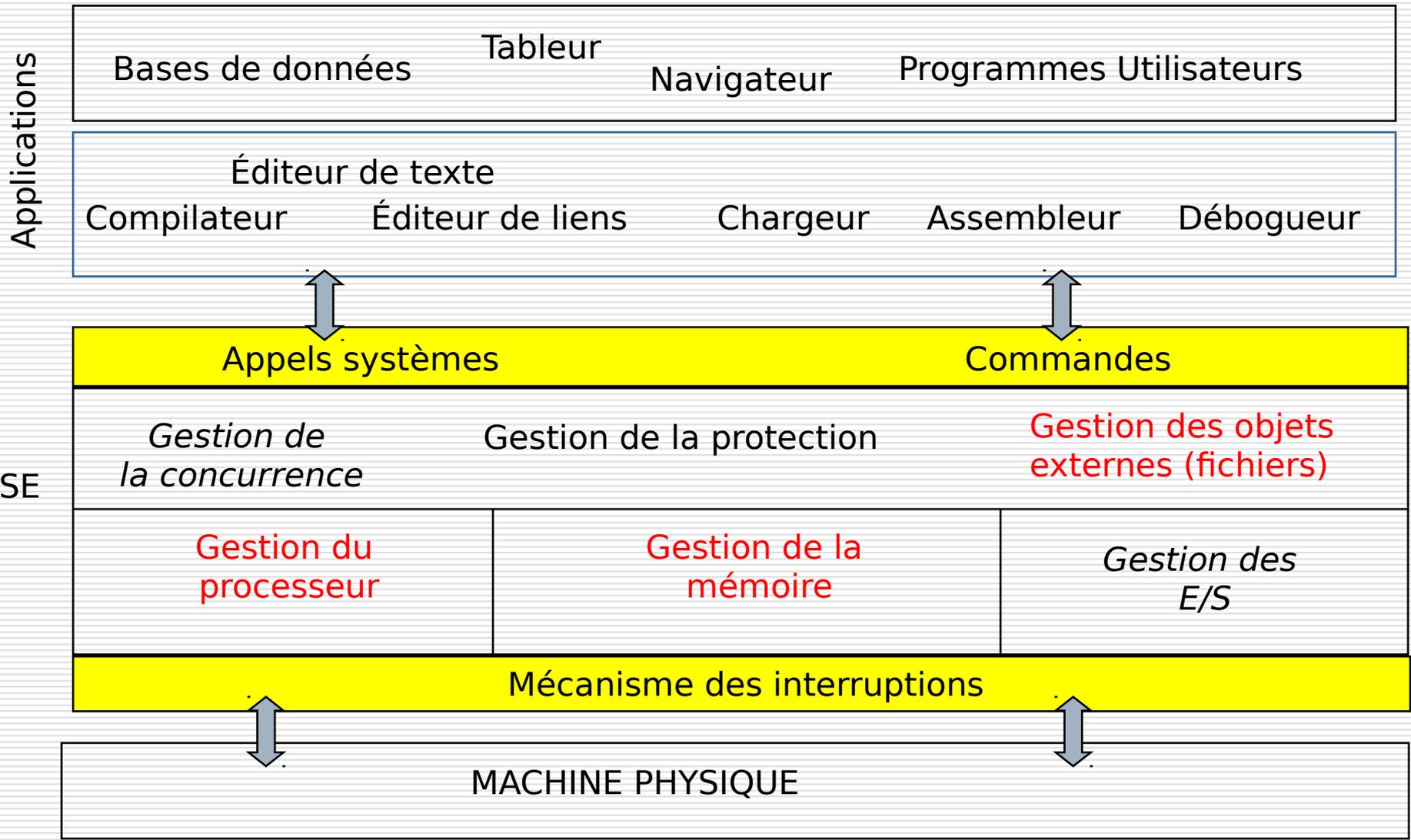
Comme ces systèmes sont souvent interfacés à un environnement dynamique (procédé) délivrant des événements synchrones ou asynchrones auxquels ils doivent réagir, on parle aussi de **systemes réactifs**

Exemples d'applications :

- L'industrie de production (systemes de contrôle de procédé : usines, centrales nucléaires) ;
- Les salles de marché (traitement des données boursières en « temps réel ») ;
- L'aéronautique (systemes de pilotage embarqués : avions, satellites) ;

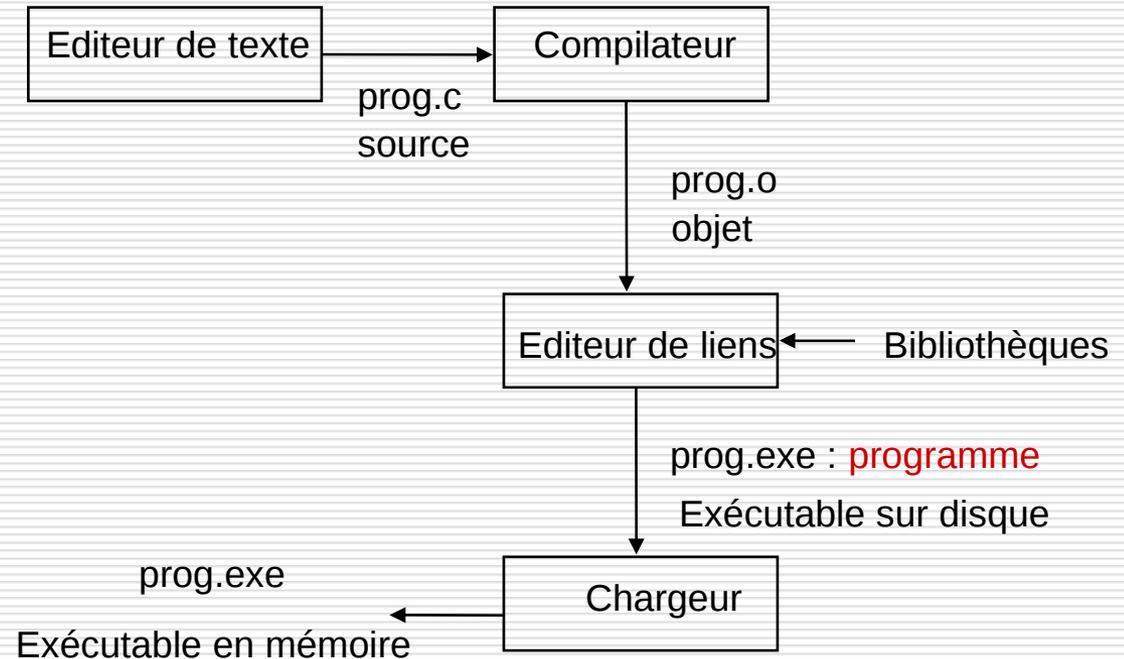
Les fonctions du système d'exploitation

Fonctions d'un système d'exploitation



Cette chaîne est l'ensemble des étapes nécessaires à la construction d'un programme exécutable à partir d'un fichier source :

- ❑ La compilation
- ❑ L'édition des liens
- ❑ Le chargement



La gestion du processeur

Le système doit gérer l'allocation du processeur aux différents programmes pouvant s'exécuter (**processus**)

Un processus est un programme en cours d'exécution

Cette allocation se fait par le biais d'un **algorithme d'ordonnancement** qui planifie l'exécution des programmes

Un algorithme d'ordonnancement est un programme du système qui détermine l'ordre d'exécution des processus

Ordonnancement : un exemple



1. A, calcul 10 ms

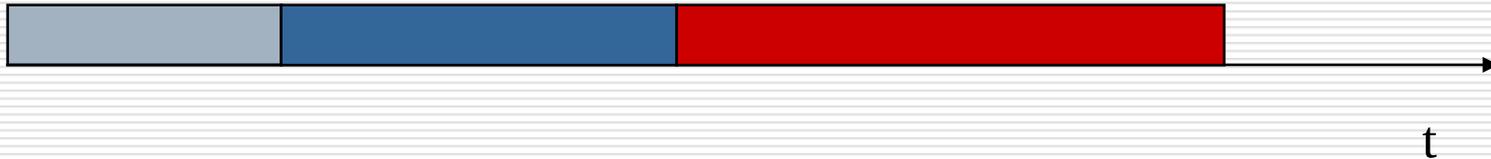


2. B, calcul 20 ms

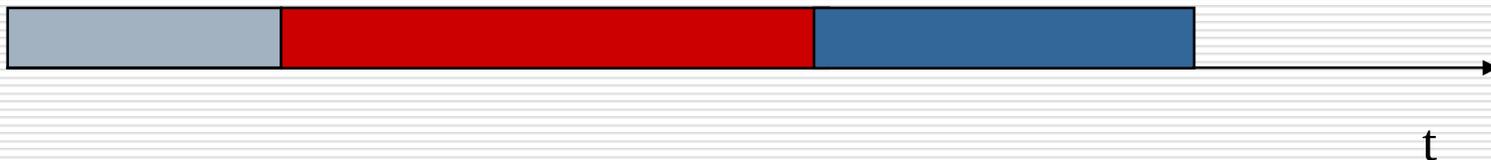


3. C, calcul 15 ms

A, B et C sont trois programmes à exécuter (processus).



Planification/ordonnancement selon le temps d'exécution



Planification/ordonnancement selon l'ordre d'arrivée

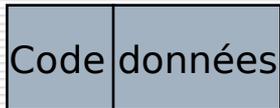
Le système doit gérer **l'allocation de la mémoire centrale** entre les différents programmes pouvant s'exécuter

Comme la mémoire physique est souvent trop petite pour contenir la totalité des programmes, la gestion de la mémoire se fait selon le principe de la **mémoire virtuelle**.

Principe de la mémoire virtuelle :

A un instant donné, seules sont chargées en MC, les parties de code et données utiles à l'exécution

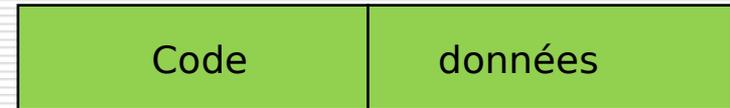
Gestion de la MC : un exemple



A, 30 Ko

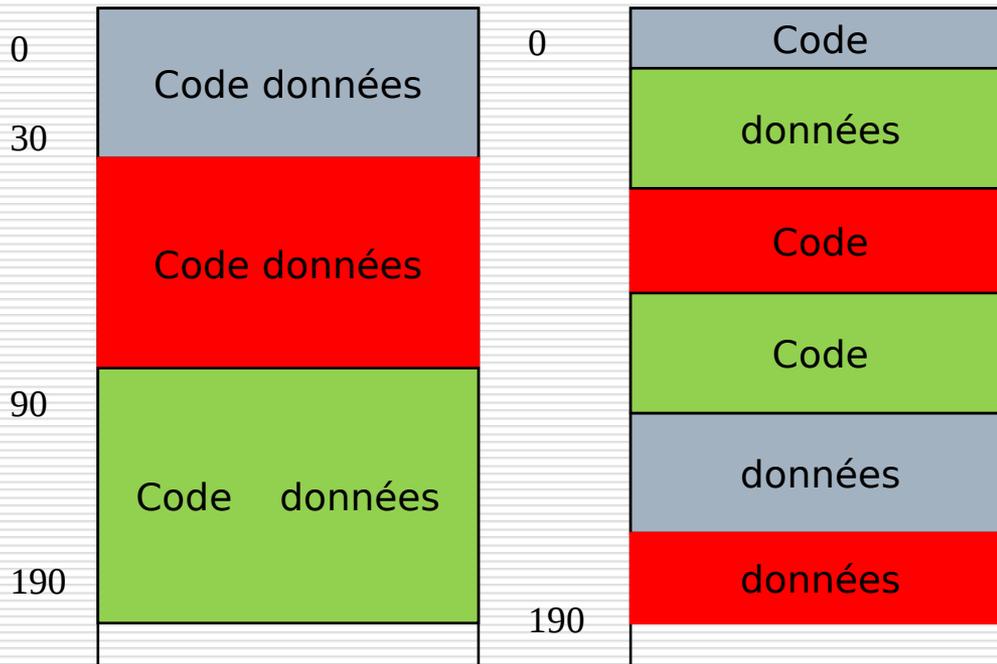


B, 60 Ko



C, 100 Ko

A, B et C sont trois programmes à exécuter (processus).

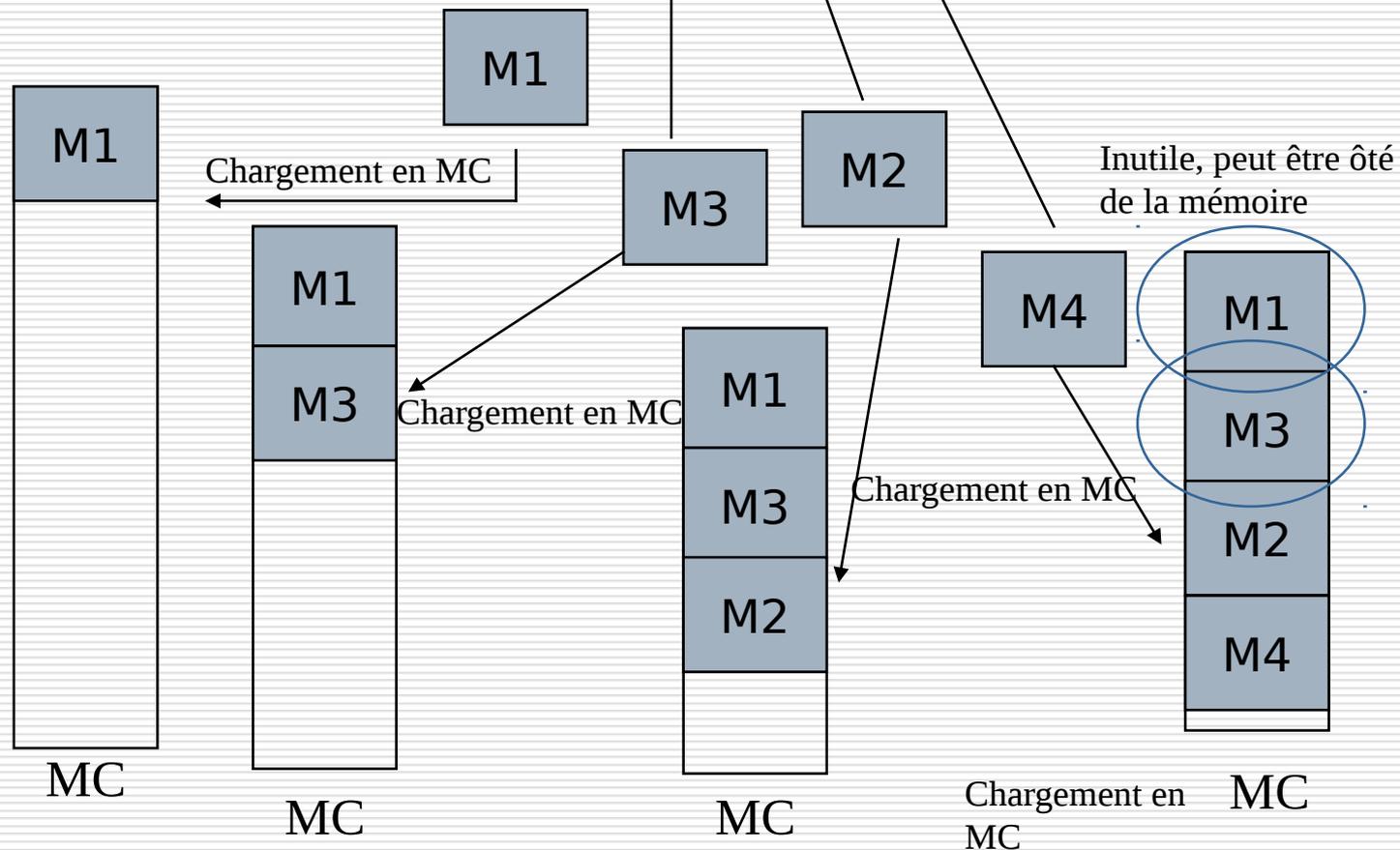


La **méthode d'allocation** de la MC détermine comment l'espace d'adressage de chaque processus est placé en MC.

Gestion de la MC : un exemple

Seules les parties utiles au temps t sont en mémoire centrale

Références aux « morceaux » : $11(11311)^{10}22(22422)^{10}$



- La mémoire centrale est une mémoire volatile
- Les données devant être conservées au delà de l'arrêt de la machine, doivent être stockées sur une mémoire de masse (disque dur, clé USB, disque SSD...)
- La gestion de l'allocation des mémoires de masse ainsi que l'accès aux données stockées s'appuient sur la notion de **fichiers** et de **Système de Gestion de Fichiers (SGF)**.

Systeme de gestion de fichiers

Le **fichier** est l'unité de conservation sur le support de masse
Le répertoire du disque permet de connaître les fichiers présents sur le support de masse

Programme :

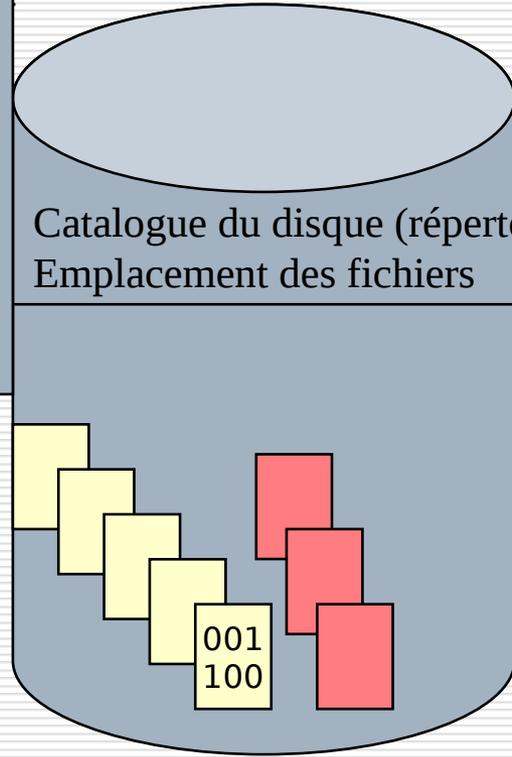
Pour les élèves
Lire (nom_élève);
Lire (note_examen);
Écrire (table);
fait
Sauvegarder
(mon_fichier)

totot	12
dupont	20
cesar	09
cléopâtre	15

Programme
instructions

SGF
Créer fichier
Lire fichier
Ecrire fichier
...

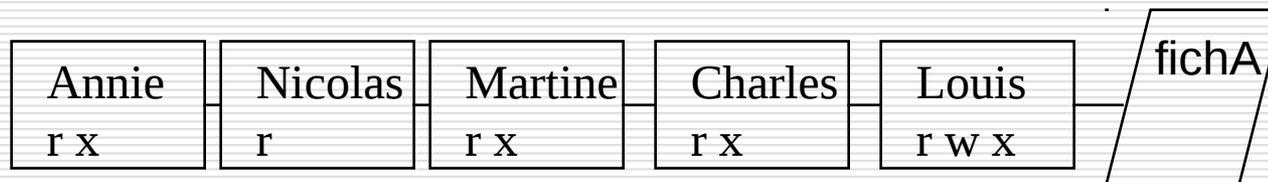
Programme
donnée



Fichier
physique
Ensemble de
blocs
physiques
(formatage)

Gestion de la protection

- Le SE doit fournir des mécanismes garantissant que ses ressources (CPU, mémoire, fichiers) ne peuvent être utilisées que par les programmes auxquels les **droits nécessaires** ont été accordés.
- **Exemple** : droits en lecture (r), écriture (w), exécution (x) pour un fichier

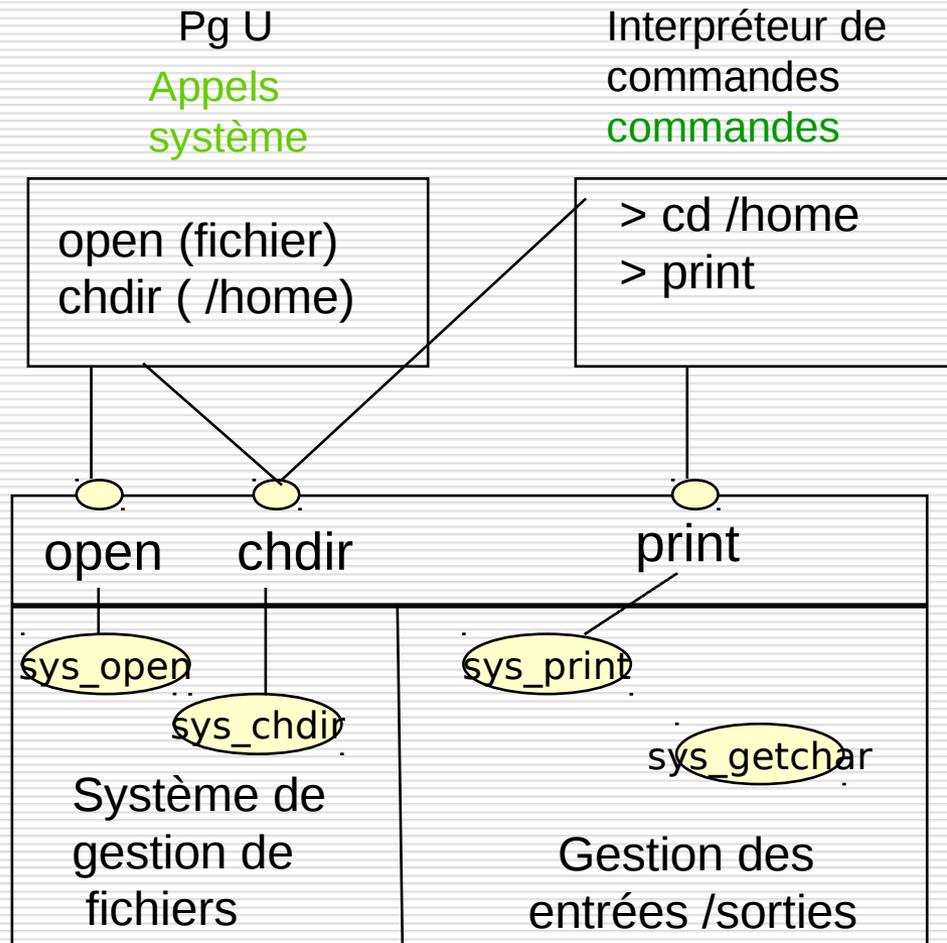


Notions de base

- Modes d'exécution
- Interruptions logicielles et matérielles

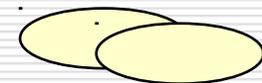
Appels système et commandes

Les fonctionnalités du système d'exploitation sont accessibles par le biais des **commandes** ou des **appels système**



Interface d'appel

SE : ensemble de fonctions



L'interpréteur de commandes

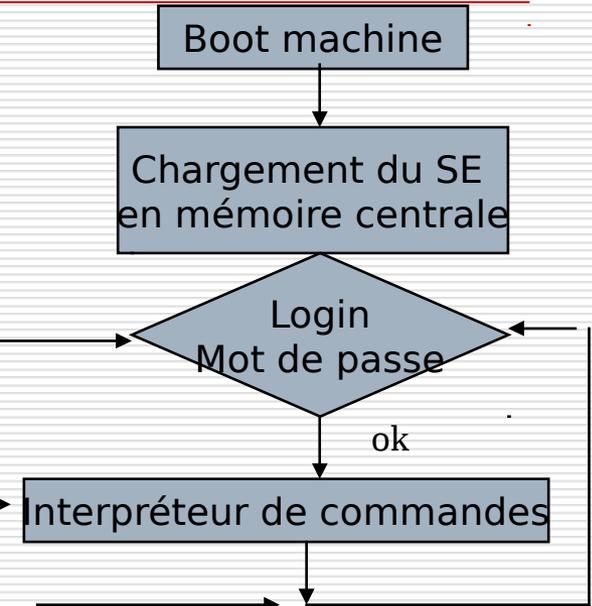
Un **interpréteur de commandes** est un programme du système d'exploitation. Son rôle est de traiter les commandes tapées au clavier par l'utilisateur.



Un utilisateur ouvre une session de travail

L'utilisateur tape des commandes

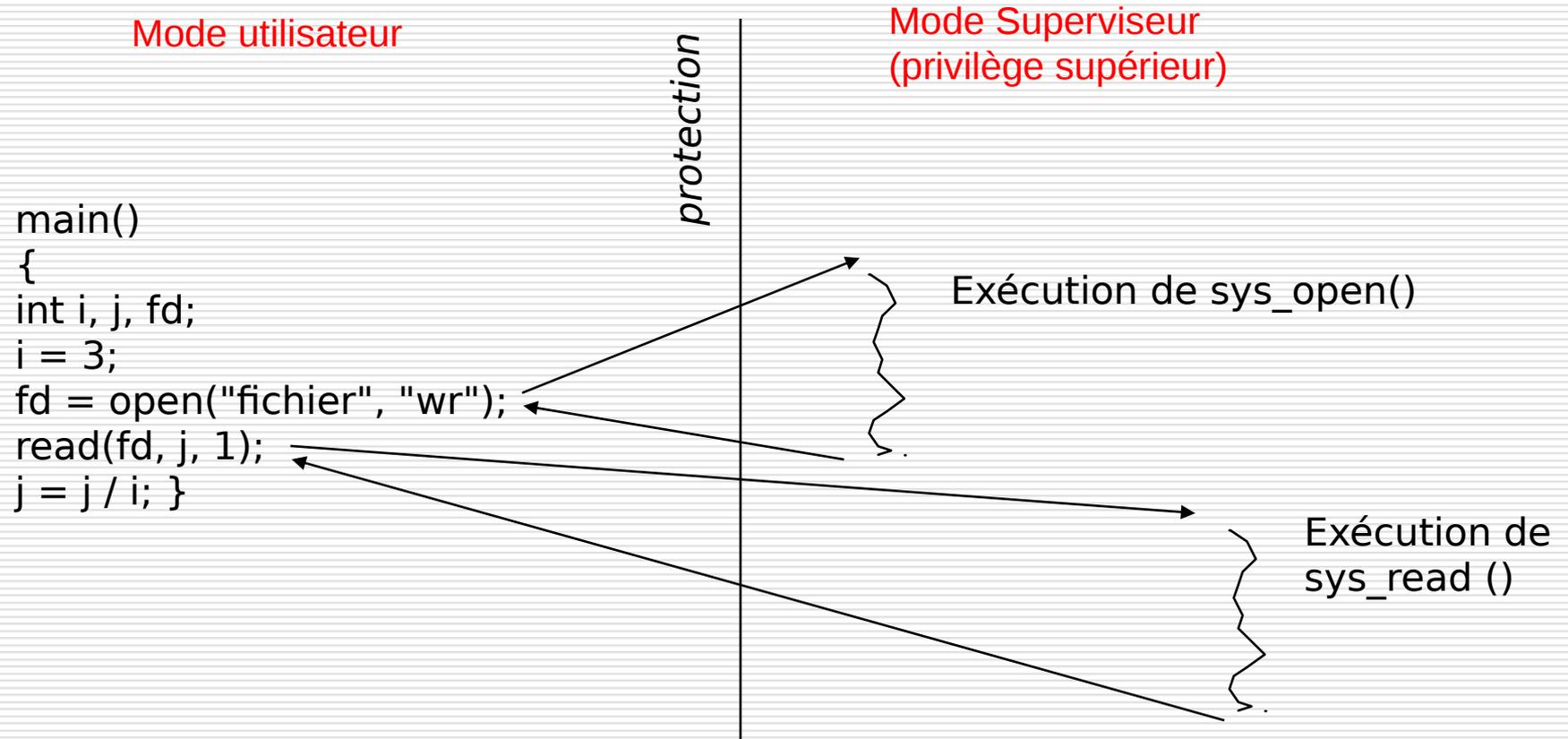
L'utilisateur clôt la session de travail



```
TANT QUE l'utilisateur ne ferme pas la session FAIRE
  Emettre un signe d'invite ;
  Lire la ligne courante
  Exécuter la commande indiquée sur cette ligne
FIN
```

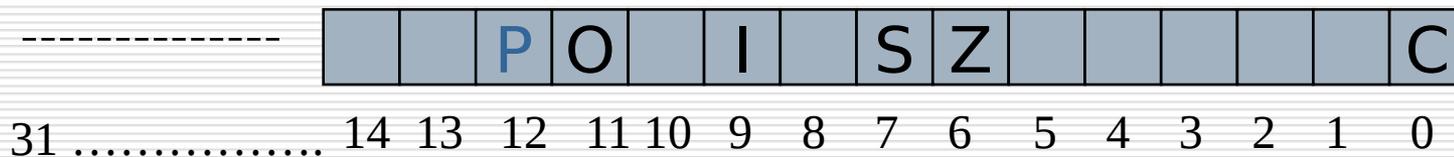
Modes d'exécutions

Lors de l'appel a une fonction du système, le programme utilisateur passe d'un **mode d'exécution dit utilisateur** à un **mode d'exécution dit superviseur**.



Modes d'exécutions

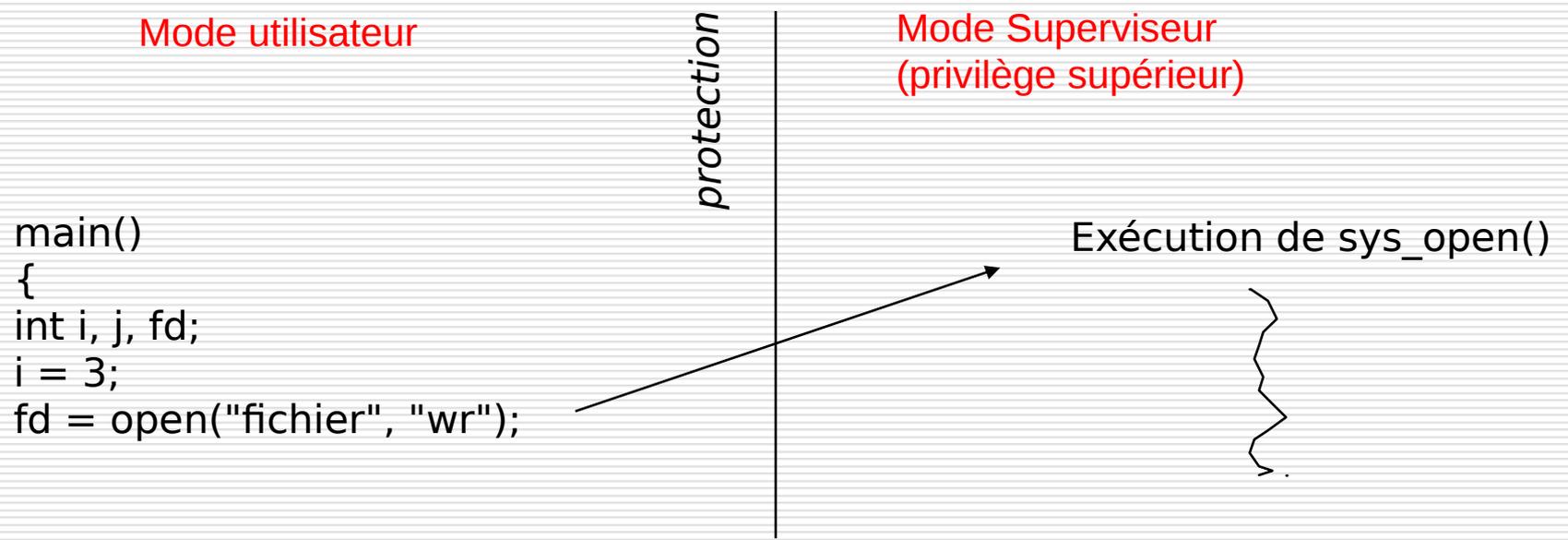
- Le **mode superviseur** est le mode d'exécution du système d'exploitation. C'est un mode privilégié.
- Le **mode utilisateur** est le mode d'exécution de tout programme utilisateur. Ce mode est restrictif; des instructions machine sont interdites (manipulation des Interruptions notamment).
- Le passage entre les deux modes est appelé **changement de contexte**.
- Les modes d' exécution sont codés dans le registre d'état du processeur : le registre PSW



Exemple processeur 486

Modes d'exécutions

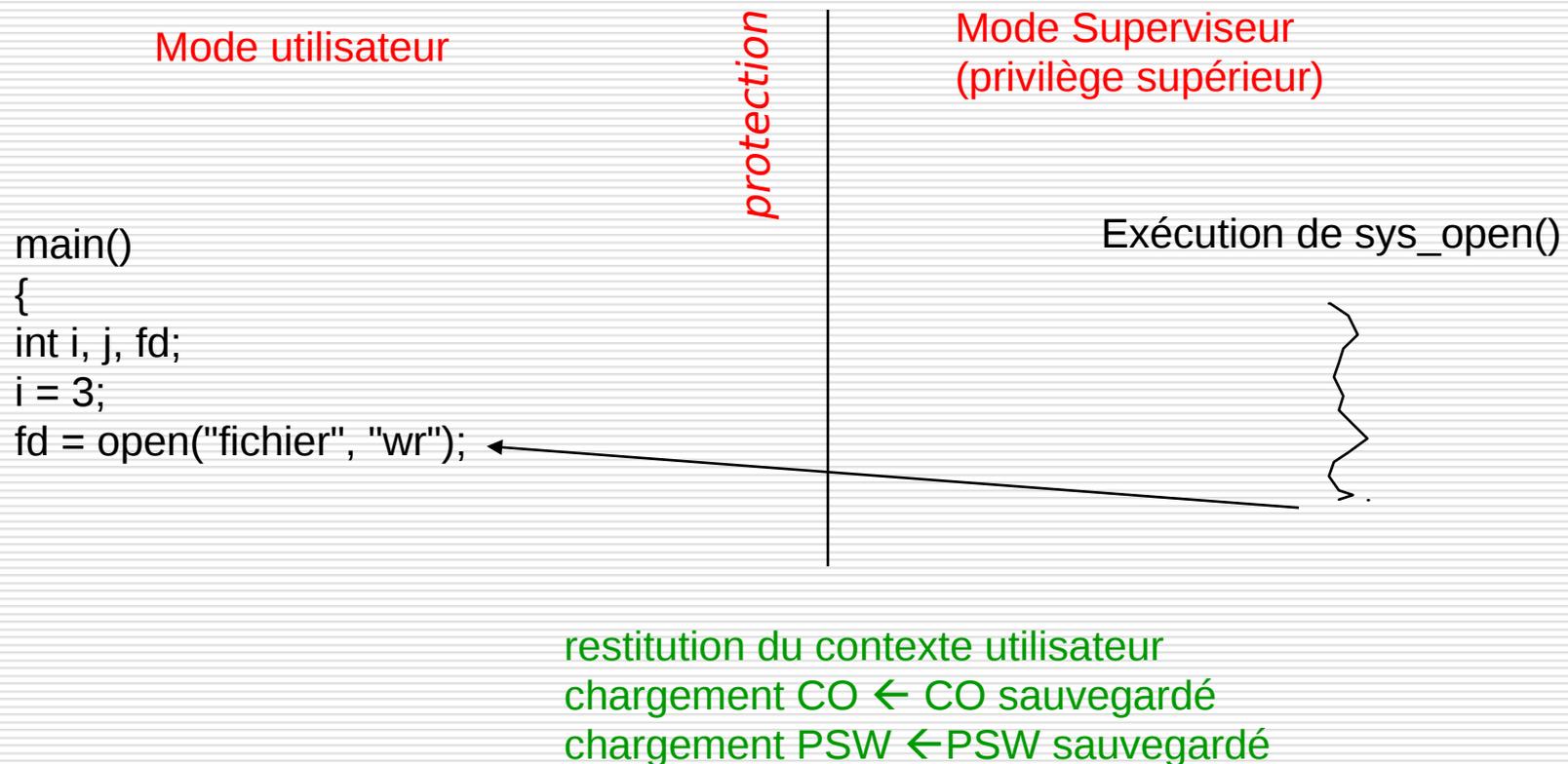
Le passage du mode utilisateur au mode superviseur s'accompagne d'opérations de **commutation de contexte : sauvegarde de contexte utilisateur**



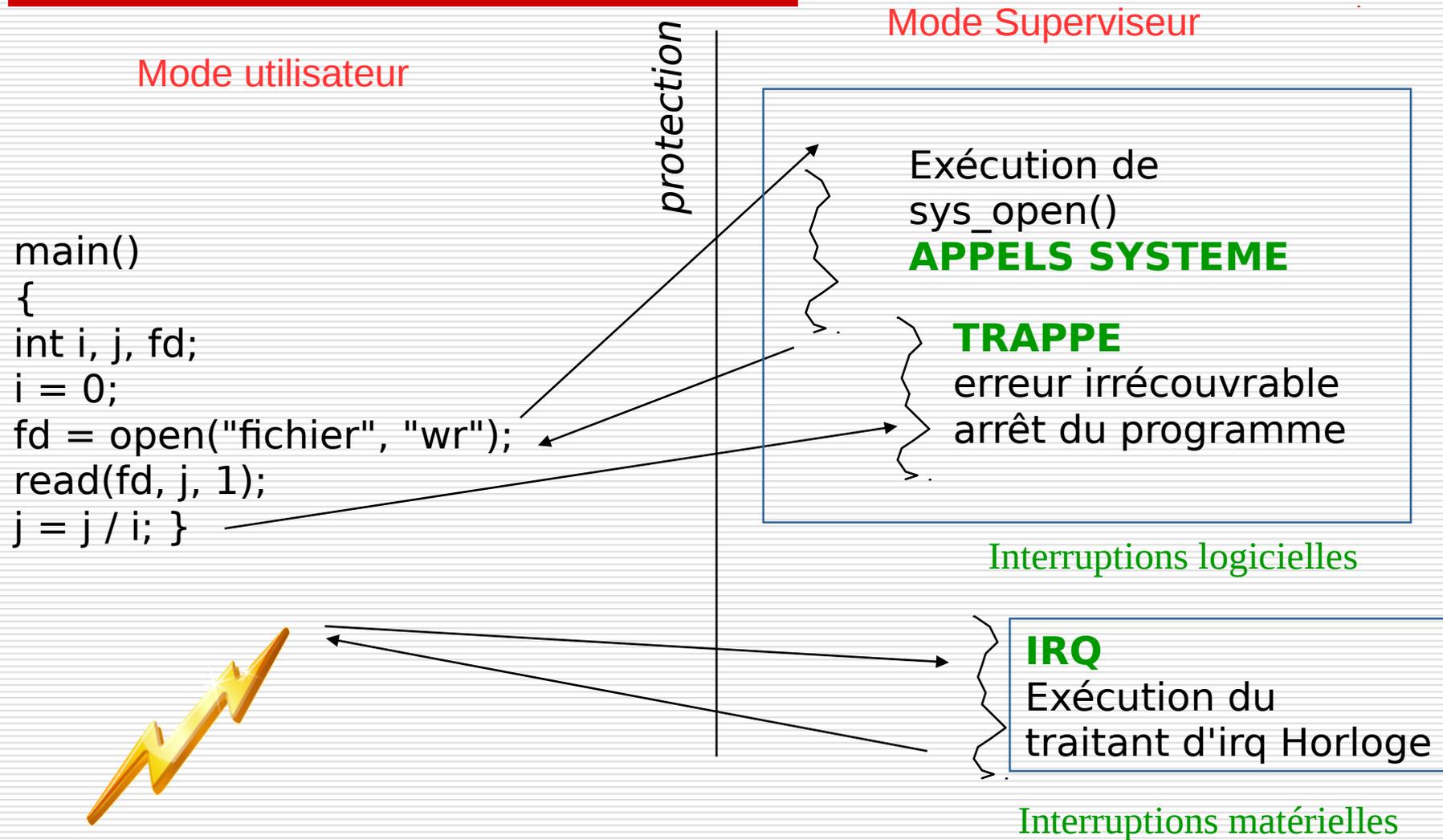
sauvegarde CO, PSW utilisateur
 chargement CO ← adresse de la fonction open
 chargement PSW ← mode superviseur

Modes d'exécutions

Le passage du mode superviseur au mode utilisateur s'accompagne d'opérations de **commutation de contexte : restitution de contexte utilisateur**



Quand passer en mode superviseur depuis un programme utilisateur ?



L'exécution d'un programme s'effectue instruction après instruction.

Une **interruption** est un mécanisme permettant de stopper l'exécution du programme en cours afin d'aller exécuter une tâche jugée plus prioritaire.

Une **interruption** permet de signaler un événement survenu sur la machine et d'exécuter un traitement spécifique (programme de service ou **routine d'interruption**) lié à cet événement.

Elle évite au processeur de **scruter** les périphériques

On distingue principalement deux types d'événements :

- **Les interruptions externes ou matérielles** sont émises par les périphériques du processeur (fin d'écriture disques, plus de papier imprimante...). Ce sont les **IRQs**.
- **Les interruptions internes ou logicielles** sont émises par le processeur lui-même lorsqu'il rencontre une erreur dans l'exécution du programme (division par zéro, accès mémoire illégal). Ce sont les **trappes**.

Exemple : les interruptions sous Intel x386

Chaque interruption est identifiée par un numéro sur 8 bits, il y a 256 interruptions différentes.

- 32 trappes par défaut du processeur
 - 0 : division par 0;
 - 6 : instruction inconnue ;
 - 13 : accès mémoire illégal.
- 16 interruptions matérielles;
- interruptions logicielles supplémentaires
 - ↳ appel système

Le numéro indexe une entrée de la *table des vecteurs d'interruptions* qui associe au numéro l'adresse du gestionnaire de traitement.

Les interruptions matérielles : asynchrones

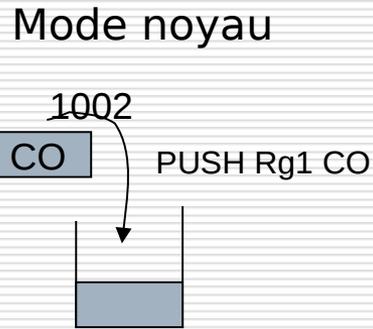
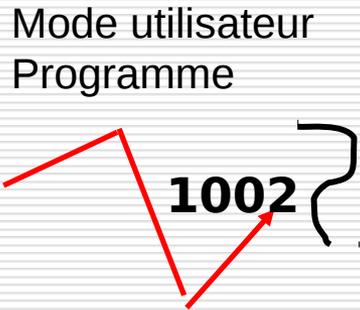
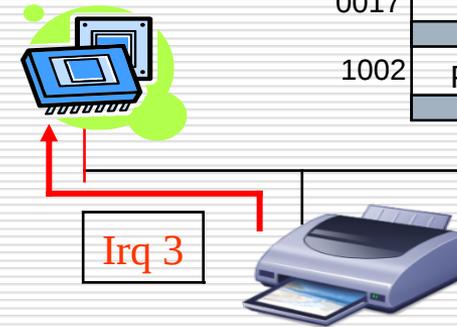


Table des vecteurs d'interruptions

0000	IRQ 3	0017
0017	traitant IRQ3	
1002	Programme	



Interruption n°3

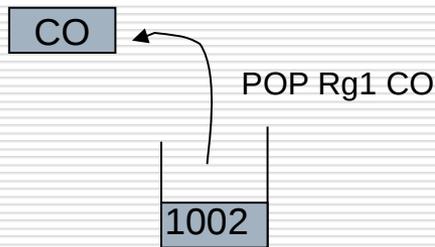
CO ← 0017

Traitant d'IRQ n°3

0017

Traitement IRQ

1002



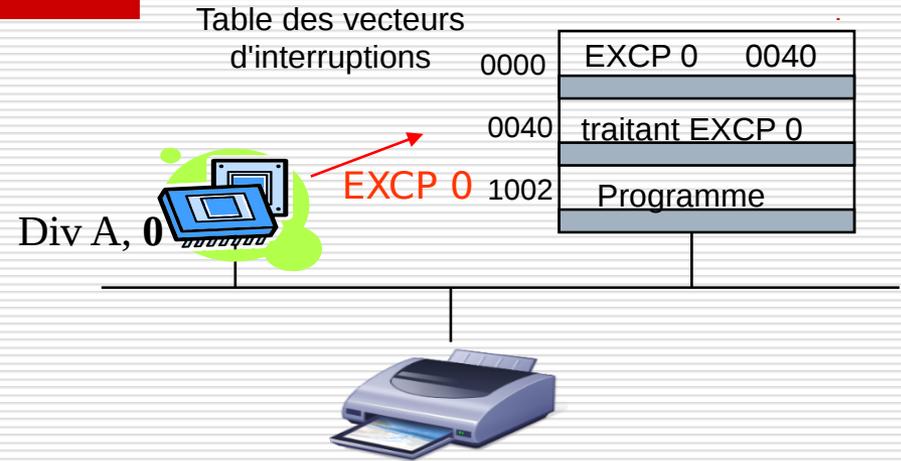
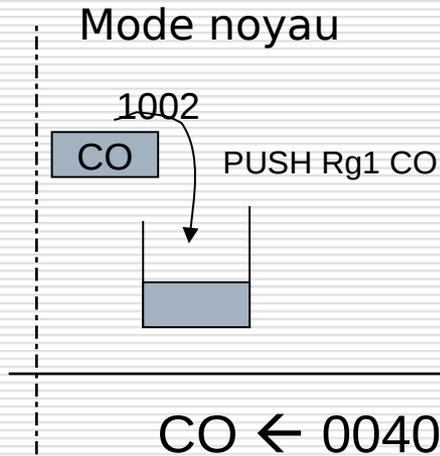
Les interruptions logicielles : synchrones

Mode utilisateur
Programme

I = 0;
A = 12,

A = A / I; **1002**

Levée exception 0



Traitant Exception 0

0040

Traitement Exception 0
Arrêt du programme

Création d'une image du programme fautif (Core) analysable par un outil débogueur

- ❑ Les fonctions du système sont exécutées en **mode superviseur**.
- ❑ Le basculement du mode utilisateur au mode noyau s'effectue par la levée de l'**exception particulière** (Exception 2Eh sous Intel) réalisée au sein d'une **routine d'enveloppe** stockée dans une bibliothèque des appels système.
- ❑ Chaque fonction du système est connue au niveau du système d'exploitation, par un numéro (exemple open, 5). **Une table des appels système** associe à chaque numéro de fonction, son adresse en mémoire centrale

Réalisation d'un appel système

Mode utilisateur
Programme
f = open (nom_fichier)

1002

Routine d'enveloppe
(bibliothèque)
open

Passage des paramètres
(nom_fichier, numéro de
l'appel système)

Appel exception 2E

Passage des paramètres
(descripteur f)
Code erreur éventuel

Table des vecteurs
d'interruptions

0000

EXCP 2E 0050

Table des appels
système

5

0070

0050

traitant EXCP 2E

0070

Fct sys 5 (sys_open)

1002

Programme

Bibliothèque : open



Mode noyau

0050

**Traitement Exception 2E
Consultation de la table
des appels système et
récupération de l'adresse de
la fonction système
sys_open**

0070

Exécution de la fonction sys_open