

Systemes d'Exploitation : Processus, interruptions, ordonnancement

DIU "Enseigner l'informatique au lycée"

Aix-Marseille Université

Faculté des Sciences

Sources :

- Cours L3 informatique "Systèmes d'exploitation" Leonardo Brenner - Jean-Luc Massat
- Cours L3 informatique Télé-enseignement "Systèmes d'exploitation" Jean-Marc Talbot
- Cours Bloc 3 DIU EIL Equipe de Lille

- 1 Introduction aux processus
 - Les 3 P...
 - Notion de programme
 - Notion de machine
 - Exécution d'un programme
 - Définition d'un processus
- 2 Interruptions
 - Interruption d'un processus
 - Interruption matérielle
 - Les appels systèmes
 - Déroutements
- 3 Ordonnancement
 - Problématique
 - Représentation d'un processus
 - Ordonnancement de processus
 - Algorithmes d'ordonnancement

Table de matière

- 1 Introduction aux processus
 - Les 3 P...
 - Notion de programme
 - Notion de machine
 - Exécution d'un programme
 - Définition d'un processus
- 2 Interruptions
 - Interruption d'un processus
 - Interruption matérielle
 - Les appels systèmes
 - Déroutements
- 3 Ordonnancement
 - Problématique
 - Représentation d'un processus
 - Ordonnancement de processus
 - Algorithmes d'ordonnancement

Les 3 P.....

- Programme
- Processus
- Processeur

Un programme

Définition : Programme

Un programme est une suite d'instructions.

- C'est une description statique
- Les instructions doivent être exécutées dans l'ordre par un processeur.

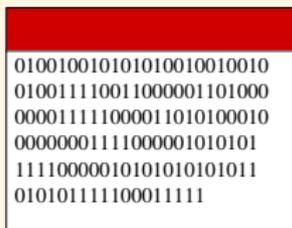
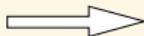
Création d'un programme



```
int main () {  
  ...  
  ...  
}
```

Code source

Compilation



```
010010010101010010010010  
010011110011000001101000  
000011111000011010100010  
00000001111000001010101  
11110000010101010101011  
010101111100011111
```

Code exécutable
(programme)

Exemples de programmes

- ls, mkdir, dir, cp, mv . . .
- Navigateur Internet, gestionnaire de mails . . .

Type d'instructions

Privilégiées

- Lancement des E/S
- Contrôle des interruptions
- Contrôle du matériel

Ordinaires

- Mouvement de données
- Structure de contrôle
- Calcul
- Appel du système d'exploitation

Un processus

Définition : Processus

Un processus est un programme en cours d'exécution.

- C'est une activité dynamique, temporelle
- La vie d'un processus : création d'un processus, exécution, fin d'un processus

Un processus

Définition : Processus

Un processus est un programme en cours d'exécution.

- C'est une activité dynamique, temporelle
- La vie d'un processus : création d'un processus, exécution, fin d'un processus

Un processus est une instance d'exécution d'un programme

- plusieurs exécutions de programmes
- plusieurs exécutions d'un même programme
- plusieurs exécutions « simultanées » de programmes différents
- plusieurs exécutions « simultanées » d'un même programme

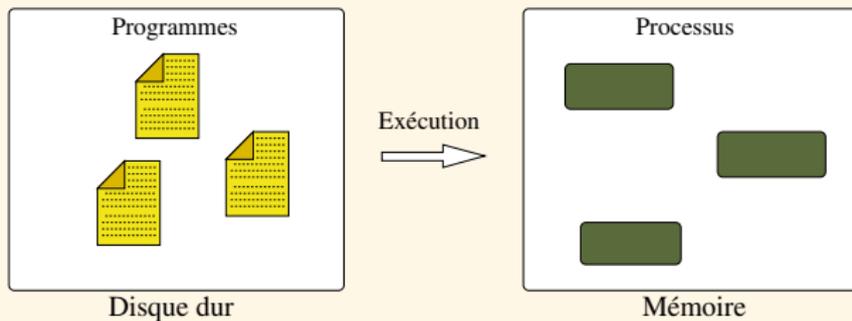
Multiprogrammation

Du programme au processus

Programmes

- Sous Linux : fichiers avec des droits en exécution
- Sous Windows : fichiers d'extensions .com, .exe

Exécution d'un programme



Un processeur

Définition : Processeur

Un processeur est un dispositif matériel.

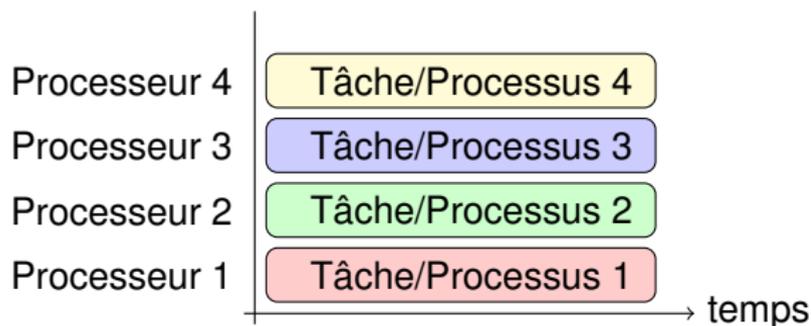
- C'est une ressource de calcul
- Le processeur fait progresser le processus qui lui est affecté

Un processeur

Définition : Processeur

Un processeur est un dispositif matériel.

- C'est une ressource de calcul
 - Le processeur fait progresser le processus qui lui est affecté
-
- Plusieurs processeurs \Rightarrow plusieurs processus exécutés simultanément (**Parallélisme**)



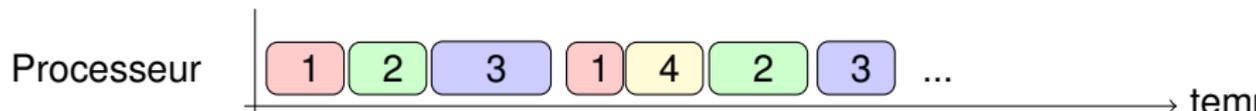
Un processeur

Définition : Processeur

Un processeur est un dispositif matériel.

- C'est une ressource de calcul
- Le processeur fait progresser le processus qui lui est affecté

- Plusieurs processeurs \Rightarrow plusieurs processus exécutés simultanément (**Parallélisme**)
- un seul processeur et plusieurs processus : partage de la ressource processeur entre les processus (rôle du SE qui virtualise la ressource processeur en donnant l'illusion à chaque processus qu'il dispose en plein du processeur)



Un processeur

Définition : Processeur

Un processeur est un dispositif matériel.

- C'est une ressource de calcul
 - Le processeur fait progresser le processus qui lui est affecté
-
- Plusieurs processeurs \Rightarrow plusieurs processus exécutés simultanément (**Parallélisme**)
 - un seul processeur et plusieurs processus : partage de la ressource processeur entre les processus (rôle du SE qui virtualise la ressource processeur en donnant l'illusion à chaque processus qu'il dispose en plein du processeur)

Choix du processus à affecter et du processeur ou le mettre

=

ordonnancement

Notion de machine

Une machine

- Processeur
- Mémoire
- Organes d'E/S :
 - ▶ Contrôleurs
 - ▶ Canaux

État d'une machine

L'état d'une machine c'est l'état de ses composants.

Description du processeur

Un processeur

- M.E.P. ou P.S.W. (Mot d'état du processeur ou *Processor Status Word*)
- Unité de calcul

M.E.P.

- Registres spécialisés :
 - ▶ compteur ordinal (CO)
 - ▶ mode d'exécution (MODE)
 - ▶ pointeur de pile (SP)
 - ▶ masque d'interruptions
 - ▶ codes de conditions
 - ▶ ...
- Registres généraux

Exécution d'un programme (1/2)

Mode de exécution

- esclave ou utilisateur - les instructions privilégiées sont interdites ainsi que les E/S directes
- maître ou superviseur (ou système) - les restrictions disparaissent

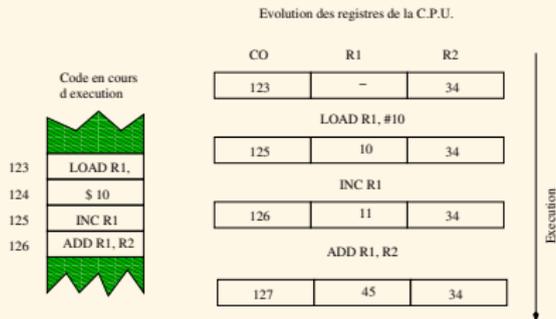
Passage entre les deux modes

- appel système demandé directement par le processus
- interruption matérielle

Exécution d'un programme (2/2)

Exécution

Une exécution d'un programme via un processus, c'est une évolution **discrète** de l'état de la machine.



Points observables

On peut observer l'évolution d'une machine sur certains points. On les appelle **points observables**. Ils sont situés dans le temps à la fin de l'exécution d'une instruction.

Processus

Définition : processus

Un processus est une abstraction de l'exécution d'un programme.

Processus

Définition : processus

Un processus est une abstraction de l'exécution d'un programme.

Processeur partagé \Rightarrow nécessaire pour chaque processus de conserver son état courant lorsqu'il lâche le processeur, pour conserver l'état dans lequel il repartira quand il le possédera à nouveau.

Processus

Définition : processus

Un processus est une abstraction de l'exécution d'un programme.

Processeur partagé \Rightarrow nécessaire pour chaque processus de conserver son état courant lorsqu'il lâche le processeur, pour conserver l'état dans lequel il repartira quand il le possédera à nouveau.

Composition d'un processus

Un processus comporte :

- un espace d'adressage ;
- le bloc du contrôle du processus, décomposé, en autre :
 - ▶ entrée dans la table des processus (identité du processus) ;
 - ▶ une `zone u`, allouée dynamiquement à la création du processus, données du processus.
 - ▶ de l'ensemble des ressources utilisées (fichiers, données de communication, ...)

Le tout forme le **contexte d'exécution** du processus.

Attributs d'un processus

- Identification univoque
 - ▶ PID : process ID
 - ▶ numéro entier `pid_t`
- Propriétaire
 - ▶ utilisateur qui a lancé le processus, son groupe
 - ▶ détermine les droits du processus
- Répertoire de travail
 - ▶ origine de l'interprétation des chemins relatifs
- Hiérarchie des processus
 - ▶ création de processus. . . par un processus
 - ▶ chaque processus a donc un processus père
 - ▶ processus init ancêtre de tous les processus
 - ▶ héritage — répertoire de travail, etc.

Création et mort d'un processus

- Mort d'un processus : lorsqu'un processus se termine ou est terminé (`kill`), le SE se charge de libérer l'espace utilisé par ce processus et les ressources qu'il utilise
- Création d'un processus : le SE crée les processus sur demande par un appel système. Le processus ayant réalisé cet appel est le **le père** du créé (qui en est le **le fils**).

Création et mort d'un processus

- Mort d'un processus : lorsqu'un processus se termine ou est terminé (`kill`), le SE se charge de libérer l'espace utilisé par ce processus et les ressources qu'il utilise
- Création d'un processus : le SE crée les processus sur demande par un appel système. Le processus ayant réalisé cet appel est le **le père** du créé (qui en est le **le fils**).

2 types de processus :

- les processus utilisateurs (droits limités - création uniquement de processus utilisateurs)
- les processus système/noyau (droits étendus - création de processus système et utilisateur)

Création et mort d'un processus

Initialement, un seul processus est créé au démarrage (le processus `init`).
C'est un processus système/noyau

2 types de création :

- par clonage : `fork` (unix)
- par lancement : `spawn` (windows)

Processus et shell (bash)

% commande

- création d'un processus qui va exécuter le programme `commande`

% `ps ax`

- liste les processus

% `top`

- affichage en continu des informations relatives aux processus

% `kill -9 pid`

- « tue » le processus désigné : envoi d'un signal 9

% `killall nom`

- « tue » les processus désignés par leur nom

Table de matière

- 1 Introduction aux processus
 - Les 3 P...
 - Notion de programme
 - Notion de machine
 - Exécution d'un programme
 - Définition d'un processus
- 2 Interruptions
 - Interruption d'un processus
 - Interruption matérielle
 - Les appels systèmes
 - Déroutements
- 3 Ordonnancement
 - Problématique
 - Représentation d'un processus
 - Ordonnancement de processus
 - Algorithmes d'ordonnancement

Interruption

Interruption = interruption de l'exécution normale d'un processus.

Exemple : prise en compte rapide d'un événement tout en effectuant un travail.

Solution

L'*attente active*. Soit F un indicateur qui signale l'événement

```
...  
si (F = 1) alors  
  F := 0  
  <traiter l'événement>  
fin si  
...
```

Problèmes de la solution

- le test **explicite** est gênant et peu sécurisé,
- perte de temps C.P.U.,
- pas de réaction immédiate et risque de collision.

Interruption d'un processus

Interruption

Opération **indivisible** effectuée par la C.P.U. qui change le CO et le MODE. Les interruptions sont déclenchées uniquement sur les points observables (points **interruptibles**).

Vecteur d'interruptions

Chaque interruption a une cause identifiée par un entier. Le **vecteur d'interruptions** (VI) est une table logée dans les adresses basses de la mémoire.

Interruption d'un processus

Prise en compte d'un interruption k

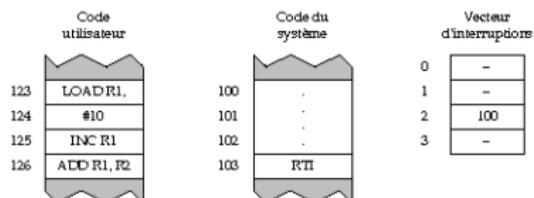
interruption de cause k

```
PUSH CO, MODE;  
MODE := maître;  
CO := vi[k];
```

reprise après interruption

```
POP MODE;  
POP CO;
```

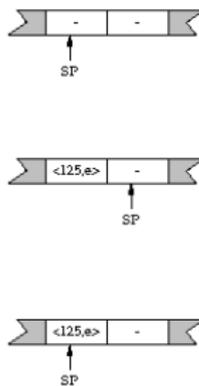
Exemple de traitement d'une interruption



Evolution des registres de la C.P.U.

CO	MODE	SP	R1	R2
123	esclave	200	-	34
LOADR1, #10				
125	esclave	200	10	34
< interruption de cause n° 2 >				
100	maître	201	10	34
.
103	maître	201	10	34
RTI				
125	esclave	200	10	34
INCR1				
126	esclave	200	11	34

Evolution de la pile



Structure générale d'un traitant

Programme **traitant**

Le programme exécuté après une interruption est appelé le **traitant** de cette interruption.

Structure d'un traitant

Les traitants ont la structure suivante :

1. sauvegarde du contexte ;
2. traitement de la cause ;
3. restauration du contexte ;
4. retour au processus interrompu (instruction `RTI`).

Utilisation des interruptions

Types d'interruptions

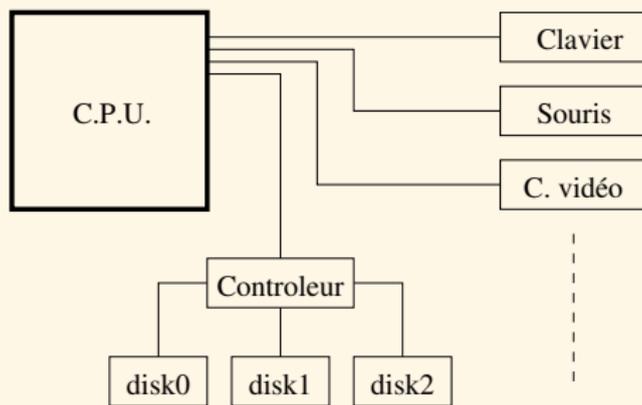
Il existe trois types d'interruptions :

- **Interruption matérielle** : réaction aux événements extérieurs ;
- **Appel au superviseur** : appel explicite d'une routine système ;
- **Déroutement** : traitement des erreurs et des situations anormales.

Interruption matérielle

Principe

Un signal (électrique) est émis vers le processeur qui provoque une interruption.



Il y a donc un branchement vers le programme de traitement des interruptions (« le traitant ») avec un **changement de mode**.

Les appels systèmes

Principe

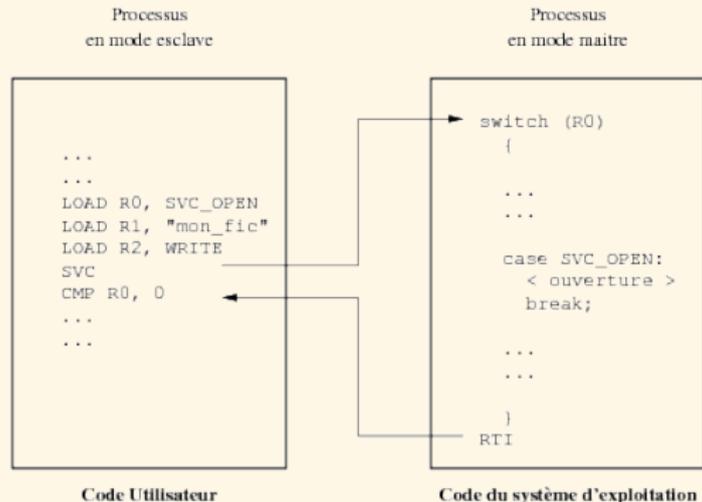
Si les processus s'exécutent en mode « **esclave / utilisateur** », les E/S directes leurs sont interdites.

- Pour réaliser des E/S, les processus doivent envoyer des requêtes au système d'exploitation.
- L'appel de fonction classique pose des problèmes pour réaliser ces requêtes :
 - ▶ le nombre de points d'entrée est important,
 - ▶ le mode d'exécution reste « esclave »,
 - ▶ le code du S.E. est accessible en lecture.

Les appels systèmes

Moyens

Les trappes (« Trap ») : Appel explicite du système par le biais d'une instruction qui déclenche une interruption.



Les appels systèmes

Avantages

Avantages des appels par interruption :

- Changement de mode,
- Le nombre de points d'entrée est limité à un,
- On obtient un sas qui isole le S.E.,
- Une librairie standard offre :
 - ▶ une interface système agréable et simplifiée,
 - ▶ une interface indépendante du système d'exploitation,
 - ▶ l'implantation réelle des appels systèmes en assembleur.

Les appels systèmes

Structure du traitant

- 1 sauvegarde du contexte,
- 2 vérifier la conformité de la requête,
 - ▶ vérifier la nature de la requête,
 - ▶ vérifier la nature des arguments,
 - ▶ vérifier les droits du processus demandeur.
- 3 exécuter la requête,
- 4 choisir le prochain processus p à exécuter,
- 5 restaurer le contexte de p ,
- 6 relancer le processus p .

Déroutements

Principe

L'objectif des déroutements est double :

- traitement **systematique** des erreurs ou des situations anormales (défaut de page),
- **protection** et bonne utilisation de la machine (les processus sont surveillés).

En cas d'erreur

Si l'exécution d'une instruction produit une erreur alors, il y a interruption et branchement vers la routine du système qui traite les erreurs.

Les causes possibles sont :

- données incorrectes (division par zéro),
- opération interdite (instruction privilégiée),
- instruction inconnue,
- accès à une zone mémoire interdite,
- erreur de bus mémoire.

Déroutements

Exemples

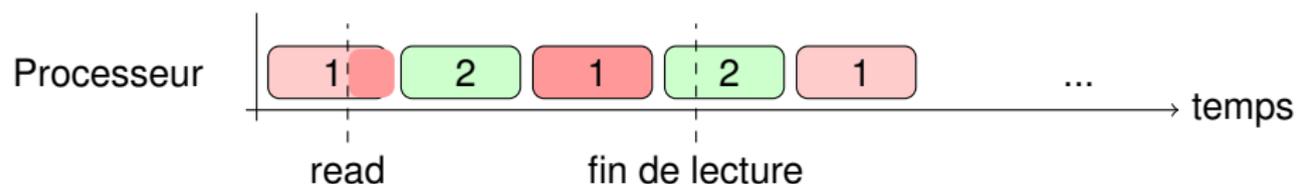
- Implantation d'une gestion utilisateur des erreurs,
- Gestion de la mémoire virtuelle,
- Ajout de nouvelles instruction par simulation logicielle,
- Le mode TRACE sur le processeur 68000 :
 - ▶ à chaque point interruptible, la C.P.U. génère un déroutement,
 - ▶ le S.E. récupère donc le contrôle entre chaque instructions,
 - ▶ il peut visualiser l'évolution des registres.

Interruptions pour les entrées/sorties

Pour une lecture sur le disque,

Première idée :

- le processus utilisateur P réalise un **appel système**
- P attend (de manière active) la fin de la lecture
- P reprend ses traitements

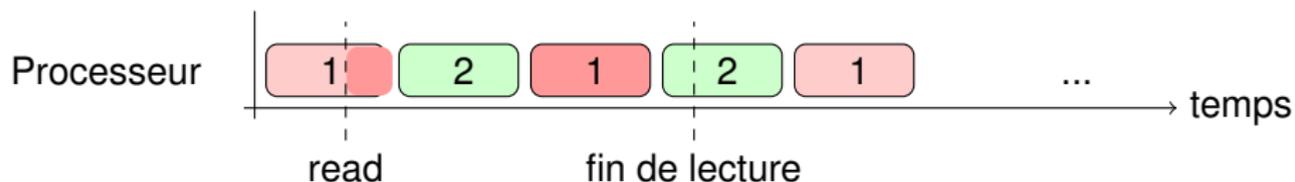


Interruptions pour les entrées/sorties

Pour une lecture sur le disque,

Première idée :

- le processus utilisateur P réalise un **appel système**
- P attend (de manière active) la fin de la lecture
- P reprend ses traitements



Mauvaise solution : le processus utilise du temps processeur pour attendre

Interruptions pour les entrées/sorties

Pour une lecture sur le disque,

Deuxième idée :

- le processus utilisateur P réalise un **appel système**
- P est mis en attente sur l'évènement de fin de lecture
- la fin de la lecture déclenche une interruption d'entrée/sortie
- le traitement de l'interruption place P parmi les processus "prêt"
- P reprendra ses traitements quand le processeur lui sera alloué

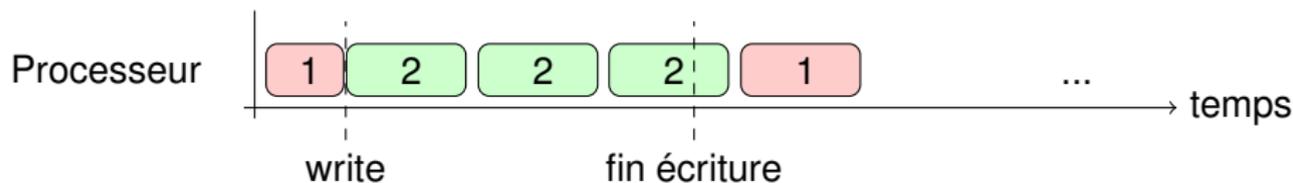
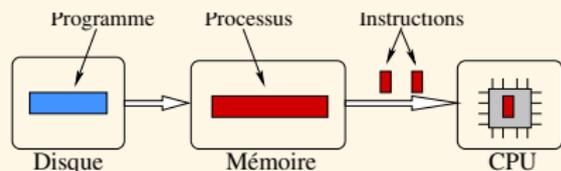


Table de matière

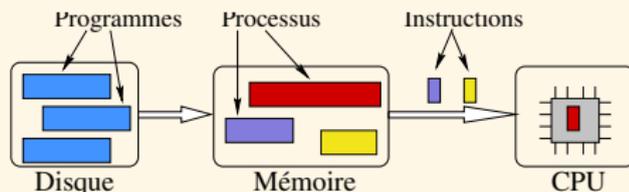
- 1 Introduction aux processus
 - Les 3 P...
 - Notion de programme
 - Notion de machine
 - Exécution d'un programme
 - Définition d'un processus
- 2 Interruptions
 - Interruption d'un processus
 - Interruption matérielle
 - Les appels systèmes
 - Déroutements
- 3 Ordonnancement
 - Problématique
 - Représentation d'un processus
 - Ordonnancement de processus
 - Algorithmes d'ordonnancement

Problématique

Exécution de processus



Monoprogrammation



Multiprogrammation

Cas d'un système multiprogrammé

- Plusieurs processus en mémoire, un seul à la fois accède au CPU ;
- Comment choisir le processus qui a accès au CPU ? Comment changer cet accès au cours de l'exécution ?

Représentation d'un processus

Données sur un processus

Pour chaque processus le système maintient les données suivantes :

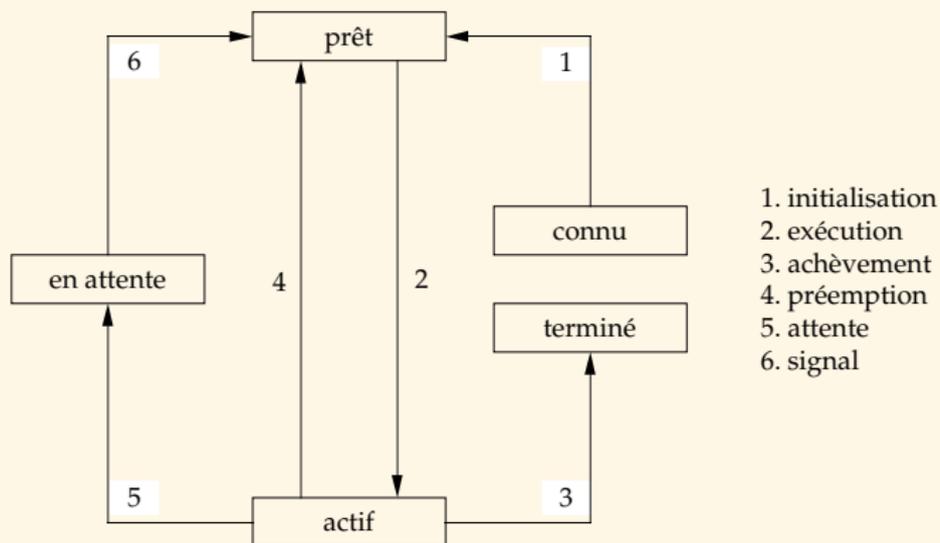
- un identifiant (PID),
- des informations diverses (priorités, filiations, propriétaires, ...),
- un état opérationnel,
- un contexte d'exécution,
- des statistiques (temps de CPU, # d'E/S, # de défauts de pages, ...).

Ces informations sont rangées dans un PCB (*Process Control Block*).

État d'un processus

États opérationnels

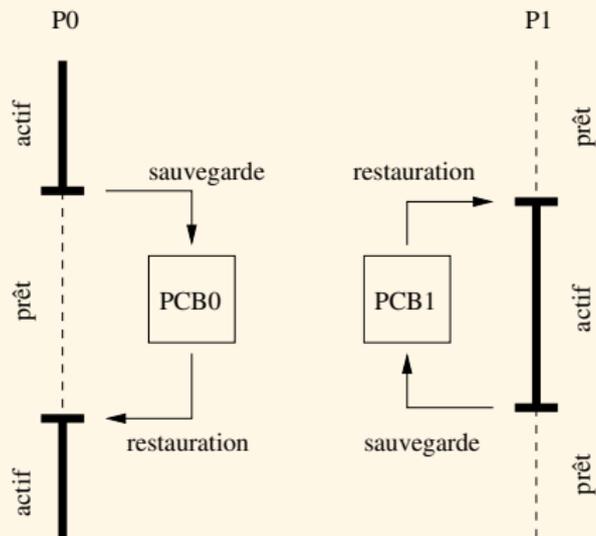
Chaque processus est dans l'un des **états opérationnels** suivants :



Rôle de PCB

Commutation de contexte

Le rôle du PCB dans la commutation de contexte entre deux processus :



Contexte d'un processus

Quelques données nécessaires à la commutation de contexte

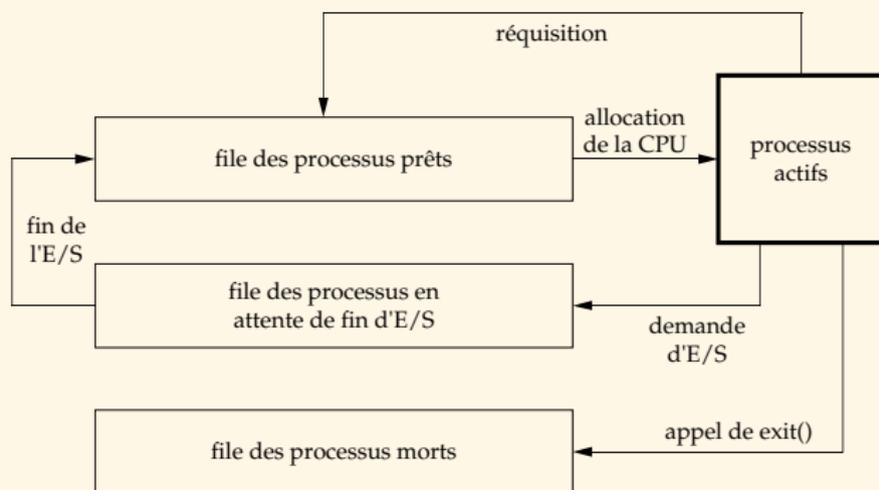
- mot d'état, contexte de l'unité centrale :
 - ▶ accumulateur,
 - ▶ registre d'instruction et compteur d'instruction,
 - ▶ registres d'état du processeur,
 - ▶ registres d'états du processus.
- état du processus,
- variables globales statiques dynamiques,
- entrée dans la table du processus,
- zone u,
- piles utilisateur et système,
- zones de codes et de données.

Files et états des processus

Files des processus

Le système maintient un ensemble de files dans lesquelles il range les PCB. On trouve par exemple un file

- des processus prêts,
- des processus en attente (d'une ressource, de la fin d'une E/S, ...),
- etc.



Ordonnancement de processus

Définition : Ordonnanceur

L'ordonnanceur est un algorithme qui va élire le processus qui a accès à la CPU. Ce processus a le « privilège » d'accéder à la CPU pendant un intervalle de temps.

Perte de la CPU

Il y a perte de la CPU dans trois cas :

- interruption extérieur,
- déroutements,
- appels système.

Commutation de contexte

Définition : Commutation de contexte

Une commutation de contexte consiste à sauvegarder l'état d'un processus et à restaurer l'état d'un autre processus.

Ordonnanceur non préemptif

On sélectionne un processus qui s'exécute jusqu'à ce qu'il libère volontairement le processeur.

Ordonnanceur préemptif (avec réquisition)

- On sélectionne un processus qui s'exécute pendant un intervalle de temps (quantum) ;
- Si le processus est toujours en cours d'exécution après cet intervalle de temps, il est suspendu et un autre processus est choisi

First In First Out

Description de l'algorithme *First In First Out*

Les processus sont exécutés dans l'ordre de soumission.

Propriétés de l'algorithme

- Pas de notion de priorité sur les processus ;
- Non préemptif (pas de réquisition) ;
- Pas de connaissance sur la durée des processus ;

Round Robin (Tourniquet)

Description de l'algorithme *Round Robin*

Les processus sont rangés dans une file. Un intervalle de temps, appelé *quantum*, est assigné à chaque job. Si l'exécution du processus n'est pas terminée à la fin du quantum, il est replacé en fin de file.

Propriétés de l'algorithme

- Pas de notion de priorité sur les processus ;
- Préemptif ;
- Pas de connaissance sur la durée des processus.
- Equitable

Files de priorité

Description de l'algorithme *Files de priorité*

Il existe une file par ordre de priorité. La gestion de chaque file est faite par l'algorithme Round Robin. Les files sont traitées dans l'ordre décroissant.

Propriétés de l'algorithme

- Notion de priorité sur les processus ;
- Préemptif ;
- Pas de connaissance sur la durée des processus.

Files de priorité dynamique

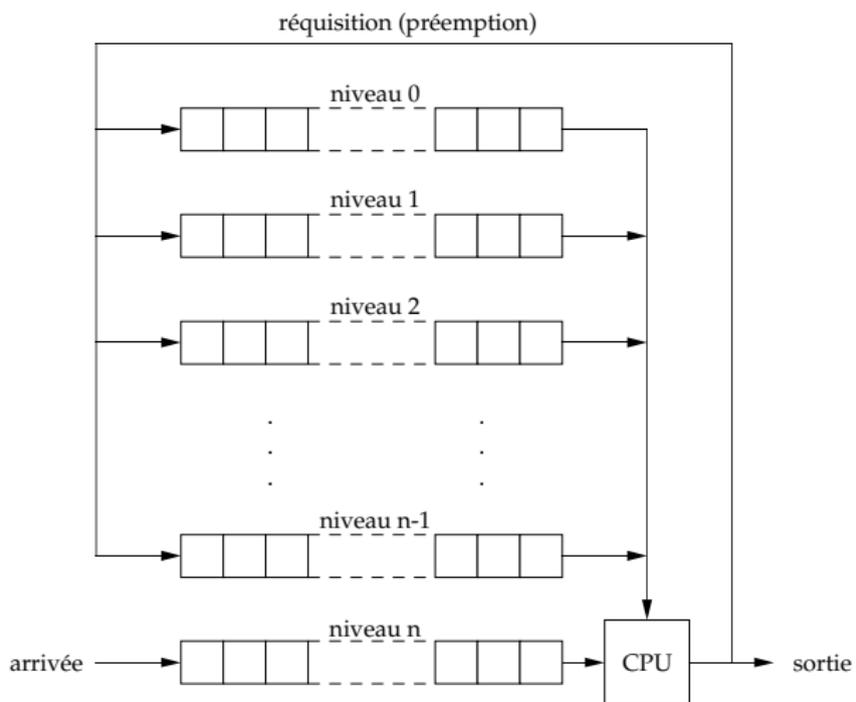
Description de l'algorithme *Files de priorité dynamique*

Il existe une file par ordre de priorité plus une file pour le nouveau processus. Tout les nouveaux processus sont placés sur ce file plus prioritaire. Après la première exécution, le processus est placé dans une autre file de priorité.

Propriétés de l'algorithme

- Notion de priorité sur les processus ;
- La priorité est calculé dynamiquement ;
- Différentes équations peuvent être utilisées ;
- Préemptif ;
- Pas de connaissance sur la durée des processus.

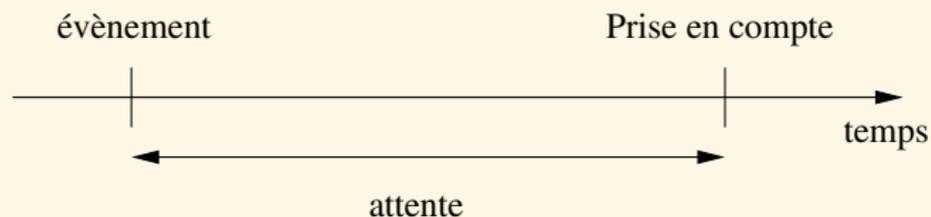
Files de priorité dynamique



Processus temps réel

Le cas des processus temps réel

Prise en compte d'un événement :



si

n est le nombre de processus temps réel,

s le temps pris par l'ordonnanceur,

t le temps moyen pris par les processus T.R.

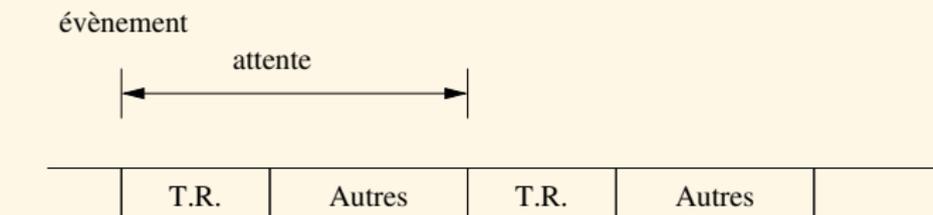
alors

$$(n - 1) \times (s + t) < \text{attente}$$

Mélange temps réel et temps machine

Division du temps CPU

Le temps CPU peut être divisé en une partie fixe réservée au processus Temps Réel, et une partie réservée aux autres processus.



Les algorithmes précédents s'appliquent dans la partie T.R.,
Les algorithmes classiques sont utilisés par la partie « Autres ».