Institut Supérieur de l'Education et de la Formation Continue

Systèmes d'exploitation
3ème Année Informatique

Latifa RABAI

Juin 2006

Sommaire

Chapitre 1 Rappels sur la structure des systèmes informatiques

I. Les Composants matériels

II. Les Composants Logiciels

Chapitre 2 Les Types de systèmes d'exploitation

I. Définition

II. Les Types de systèmes d'explotation

II. Les Types de systèmes d'explotation

III. Le noyau d'un systèmes d'explotation

IV. Les Taches d'un systèmes d'explotation

Chapitre 3 La gestion des processus

I. Définition d'un processus

III. Bloc de contrôle de processus

IV. La suspension et l'activation

Chapitre 4 La gestion du processeur

I. Introduction

II. Les Algorithmes d'ordonnancement

Chapitre 5 La gestion de fichiers

I. Motivation

II. Caractéristiques des fichiers

III. Types de fichiers

IV. Attributs de fichiers

V. Opérations sur les fichiers

VI. Stockage des fichiers

VII. Les catalogues

VIII. Fiabilité d'un système de fichiers

IX. Sécurité et fichiers

Chapitre 6 La gestion de la mémoire

I. Motivation

II. Méthodes de gestion de la mémoire

III. Techniques de swapping

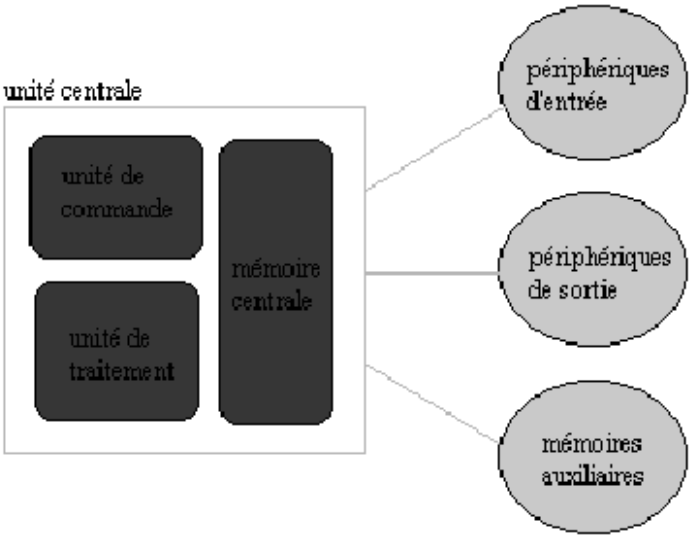
IV. Mémoire virtuelle

V. Traduction d'adresses Adresse Virtuelle en Adresse Physique

VI. Le défaut de page

Chapitre 1 Rappels sur la structure des systèmes informatiques

I. Les Composants matériels



I.1 L'unité centrale des traitements

Appelée aussi processeur, elle correspond au moteur de la machine : elle exécute les instructions des programmes, elle est composée de trois composants :

- L'unité de commande qui commande et ordonne les transferts d'information entre mémoire centrale (MC) unité d'échange (UE), registres, etc.

- L'unité arithmétique et logique qui exécute les instructions d'un programme, elle effectue les opérations arithmétique et logique.
- Les registres qui mémorisent temporairement des informations nécessaires à l'exécution d'une instruction.

I.2 La Mémoire Centrale (MC)

Contient les programmes et les données à traiter, elle peut être accédée en lcture ou en écriture. Ces accès nécessitent les informations suivantes :

- Type d'accès (Lecture ou écriture)
- Adresse du mot à lire ou à écrire.
- Le mot à écrire ou à lire.

Remarque : *Les informations dans la mémoire centrale sont linéaires et contiguës.*

On distingue deux types de mémoires, les mémoires vives et les mémoires mortes.

- Les mémoires vives (RAM : Random Access Memory) :

Ces mémoires sont caractérisées par :

  - Accès en Lecture/Ecriture .
  - Accès aléatoire
  - Volatilité.

- Les mémoires mortes (ROM : Read Only Memory) :

Ces mémoires sont caractérisées par

  - Accès uniquement en lecture
  - Conservation de l'information même en cas de coupure de courant.

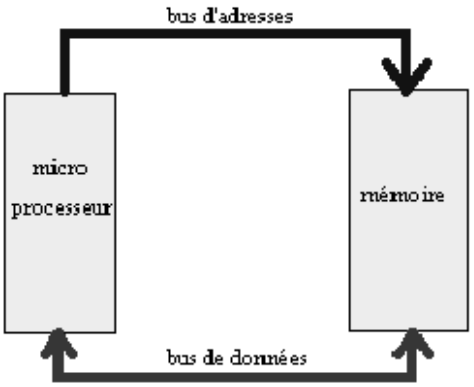
I.3 Les Périphériques

Se sont des dispositifs matériels qui servent au transfert d'information de la mémoire vers l'extérieur ou de l'extérieur vers la mémoire.
Exemple : L'écran/Clavier assure la communication entre l'utilisateur et la machine.

I.4 Les bus

Assurent un lien physique entre les différents composants

Exemple de bus :



II. Les Composants Logiciels

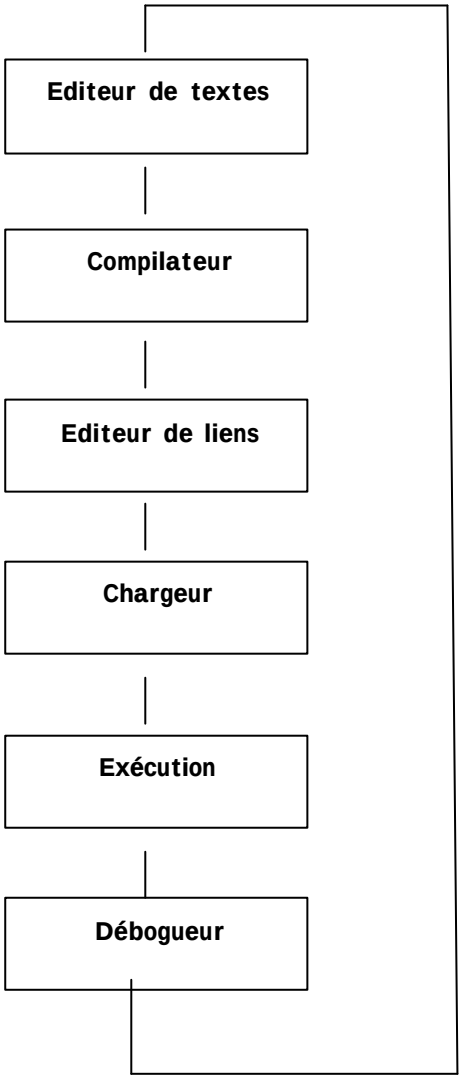
L'utilisation d'un ordinateur sous entend l'exécution d'un programme par ce dernier c'est pourquoi toute application nécessite du matériel et du logiciel.
Les logiciels se composent de deux familles, les logiciels de base (ou programmes systèmes) qui permettent le fonctionnement de l'ordinateur et les logiciels d'application qui résolvent les problèmes des utilisateurs.

II.1 Les logiciels de base

Ce sont des programmes qui gèrent l'ensemble du système et permettent son fonctionnement. Le programme système fondamental est le système d'exploitation.

- **Le système d'application** : c'est un ensemble de programmes qui se chargent de tous les problèmes relatifs à l'exploitation de l'ordinateur. Il effectue fondamentalement deux fonctions distinctes :
    - Fournir une machine virtuelle (composants matériels + logiciels).
    - Assurer la gestion des ressources matérielles et logicielles.

- **Les logiciels utilitaires** : ils aident à développer les applications, le compilateur, l'assembleur, les éditeurs de liens…



- **Editeur de textes**

C'est un logiciel interactif qui permet de saisir du texte à partir d'un clavier, et de le stocker dans un fichier. Les principales fonctions d'un éditeur sont :

- La visualisation du texte sur écran.

- Le déplacement et le positionnement dans le texte.
- La modification du texte.
- La recherche de chaînes de caractères, etc.

On peut distinguer plusieurs types d'éditeurs :

1) Editeurs syntaxiques : ce sont des éditeurs adaptés au traitement des programmes sources. Ils vérifient la syntaxe du programme au fur et à mesure qu'on le tape. Ils permettent aussi de générer automatiquement les structures syntaxiques propres au langage de programmation utilisé.

2) Traitements de textes : ces types d'éditeurs offre des fonctionnalités beaucoup plus poussées pour la manipulation de caractères, pour la justification de textes, pour l'insertion d'objets (Graphiques, tableaux, images, etc.), pour mettre en page le texte, etc.

- **Compilateur**

Un compilateur est un programme qui transforme un module en code source en un module en code objet.

- **Editeur de liens**

C'est un logiciel qui permet de combiner plusieurs programmes objets en un seul. En effet, pour pouvoir développer de gros programmes, on structure ces derniers en modules, qui sont traduits indépendamment. L'éditeur de liens prend ces différents modules et les regroupe, pour former un programme complet qui peut s'exécuter. Ce programme est stocké sur mémoire secondaire.

- **Le chargeur**

C'est un programme qui s'occupe d'amener en mémoire centrale, depuis une mémoire secondaire, un programme prêt à être exécuté.

Remarque : *Un programme ne peut s'exécuter que s'il est chargé en mémoire centrale.*

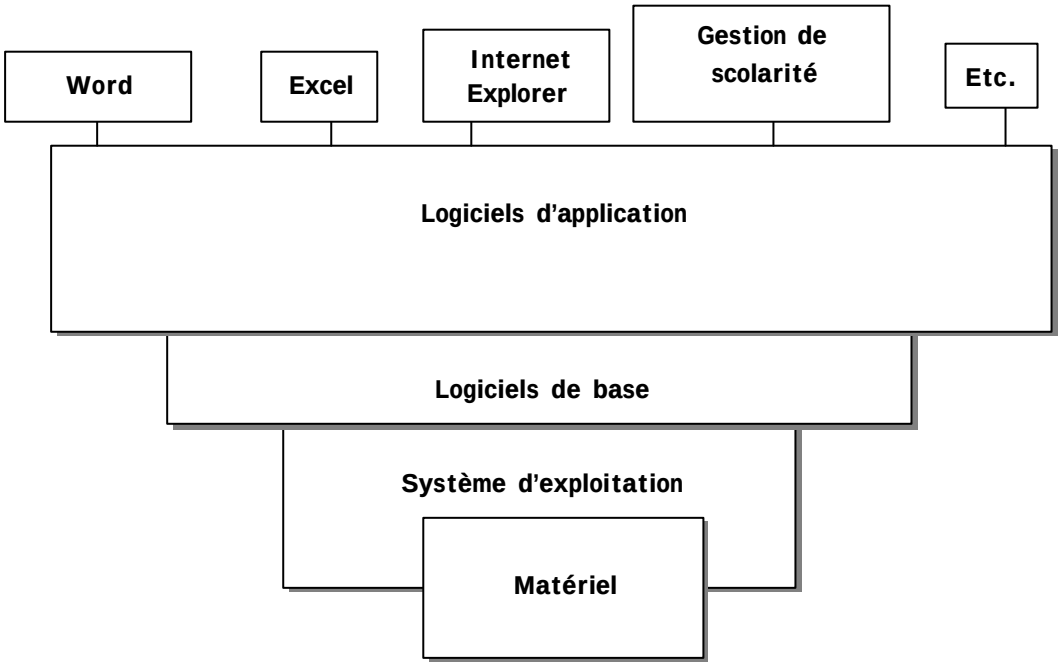
- **Débogueur**

C'est un logiciel qui facilite la mise au point de programmes (détection et correction d'erreurs). Il permet d'examiner le contenu de la mémoire centrale, ainsi que le contenu des différents registres, pour pouvoir suivre l'exécution du programme instruction par instruction.

II.2 Les logiciels d'application

Ce sont des programmes développés pour satisfaire un besoin spécifique (logiciels de comptabilité, bureautique, navigation Internet…). Ces logiciels sont développés par :

1) Des entreprises de production de logiciels (Microsoft).
2) Les utilisateurs eux-mêmes, pour résoudre un problème spécifique.



Chapitre 2 Les Types de systèmes d'exploitation

I. Définition

Un système d'exploitation est un logiciel permettant d'exploiter les ressources matérielles pour l'utilisateur. Il est considéré comme étant une partie importante d'un système informatique. En effet, ce système d'exploitation va constituer la machine virtuelle.

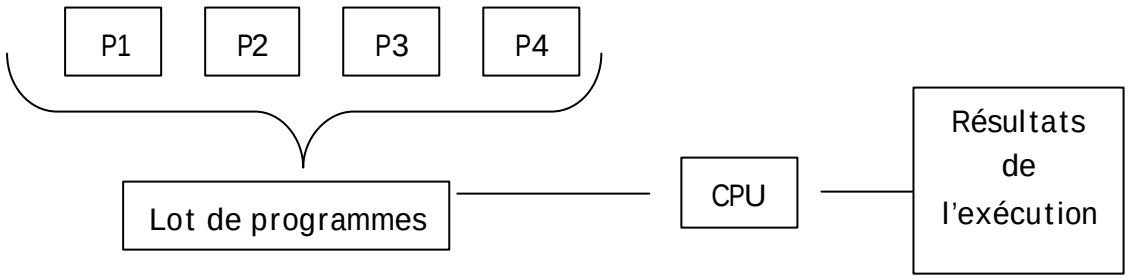
- Un système d'exploitation est un ensemble de programmes qui exploite les ressources matérielles pour l'utilisateur

- Un système d'exploitation est une interface entre l'utilisateur qui exploite des programmes d'application et les ressources matérielles (MC,Disques,…)

II. Les types de systèmes d'exploitation

II.1 Les systèmes en mode batch

Dans ces systèmes d'exploitation, on préparait à l'avance les programmes et un moniteur d'enchaînement s'occupe de l'exécution en séquence de ces programmes, l'utilisateur ne possédait pas le contrôle de la machine



On dit que le système exécute ***un lot de programmes*** d'où le nom batch.
Les systèmes les plus communs étaient FMS (Fortran Monitor System) et IBSYS (système d' IBM).
Ces systèmes sont caractérisés par une très mauvaise gestion du processeur et par une lenteur due à la séquentialité d'exécution.

II.2 Les systèmes de multiprogrammation

Ces systèmes sont apparus avec le développement des circuits intégrés d'une part et deux autres avancés technologiques d'autre part :

- Introduction du support magnétique à accès direct (disques) et la réduction du temps d'accès.

- Introduction des canaux d'E/S qui fonctionnent comme des processeurs d'E/S indépendamment du CPU qui devient libre pour exécuter autre chose.

En fait pour ces systèmes on peut dire qu'il y a deux processeurs :

- Un processeur de calcul
- Un processeur d'E/S (Canal d'E/S)

La principale caractéristique de ces systèmes est la simultanéité des traitements (calculs et E/S).

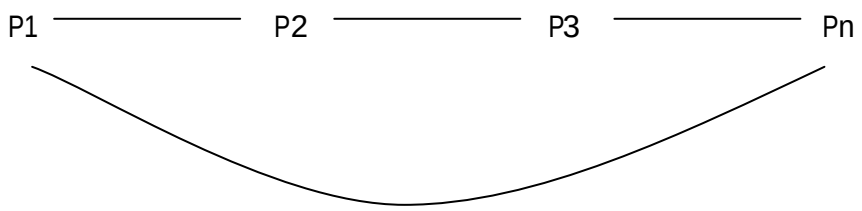
II.3 Les systèmes Temps Partagé (Time Sharing)

C'est une variante de la multiprogrammation, donc le principe est le suivant :

- Plusieurs programmes en MC.
- Simultanéité en calcul et en E/S grâce aux canaux.
- *Partage de temps qui soit équitable entre les différents utilisateurs.*

Le principe de fonctionnement de ces systèmes se résument en :

1/ Les programmes qui demandent le CPU sont structurés ou organisés d'une certaine manière dans la MC. Les programmes sont en effet organisés sous forme de *liste circulaire* basée sur les *dates d'arrivée*.



Chaque programme a un successeur, même Pn a un successeur qui est P1.

Remarques

a) *Le nombre de programmes peut augmenter, Insertion dans la liste.*
b) *Le nombre de programme peut diminuer, un programme quitte la MC.*

2/ Le système utilise la notion de ***quantum*** pour répartir le temps entre les programmes.

Un quantum est une unité de temps indivisible durant laquelle un programme dispose de manière exclusive du processeur. Un quantum est choisi de telle sorte qu'il ne soit pas perceptible par l'utilisateur.

3/ Le principe d'allocation du processeur est le suivant étant donnée la liste circulaire précédente en MC:

- Le système alloue le CPU pendant un quantum à P1
  - P1 se termine et il est retiré de la liste

Ou bien

  - P1 ne se termine pas, le système lui retire le CPU (*Préemption*) et l'alloue au programme suivant P2.

Ce système a l'avantage d'être équitable : d'une part, il y a l'ordre d'arrivée des programmes et d'autre part les quantums sont indépendants des programmes.

II.4 Les systèmes pour ordinateurs personnels

Venus avec le développement des circuits LSI (Large Scale Integration), le coût du matériel a baissé donc il est devenu possible d'avoir un système informatique dédié à un seul utilisateur. L'objectif primordial était la commodité de l'utilisateur et la rapidité de réaction.
Deux systèmes d'exploitation ont dominé le marché :

- MS Dos pour les compatibles IBM PC étendu ensuite par Microsoft pour y inclure un système de fenêtres à savoir ***Windows***. IBM a rendu plus puissant DOS avec le système multitâche OS/2
- UNIX pour toutes les plateformes.

II.5 Les systèmes Parallèles

Il s'agit d'une architecture composée de plusieurs processeurs (CPU) et d'une seule mémoire centrale (MC).

Ces systèmes sont caractérisés par une augmentation de la puissance de calcul (parallélisme réel) et une fiabilité intéressante, mais l'inconvénient majeur de ces types de systèmes est la présence d'une seule mémoire centrale.
De point de vue système d'exploitation, il existe deux architectures parallèles :

- Symétrique : Chaque processeur travaille avec une copie du même système
- Asymétrique : Chaque processeur a une tache spécifique et un processeur maître qui ordonnance et alloue du travail aux processeurs esclaves.

II.6 Les systèmes Distribués/Réparties

Il s'agit d'un ensemble d'unités composées d'éléments autonomes appelés sites ou nœuds reliés entre eux par un réseau de communication.
Dans ces systèmes, les utilisateurs peuvent se connecter sur une machine distante et transférer des fichiers d'une machine à une autre ou partager des ressources (Imprimante, Base de données répartie, etc.). Chaque ordinateur fonctionne sous son propre système d'exploitation et a un utilisateur particulier.
Un système d'exploitation distribué requiert une interface réseau ainsi que les programmes qui permettent une connexion distante.

III. Le noyau d'un système d'exploitation

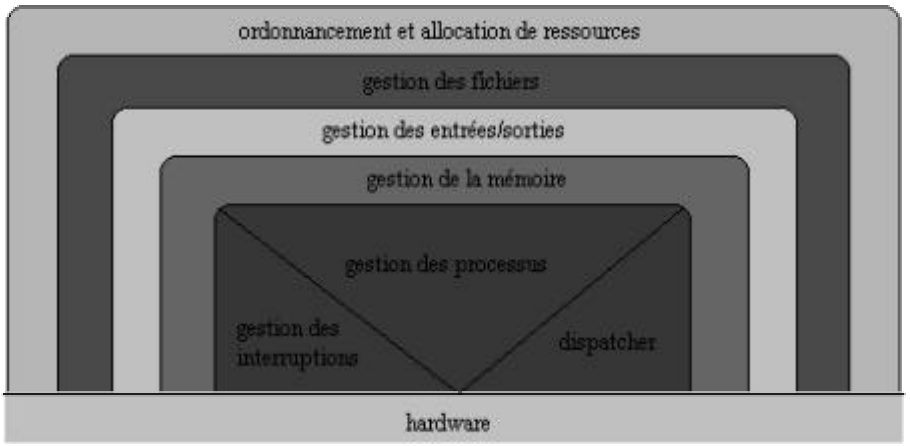
Le noyau d'un système d'exploitation est constitué d'un ensemble d'opérations de base ou primitives à partir desquels le reste du système peut être construit. La notion de noyau varie d'un système à un autre.

Les primitives se répartissent en 4 classes :

- primitives pour la gestion de processus.
- primitives pour la gestion des ressources
- primitives d'entrée/sorties : lecture, écriture, initialisation et supervision du transfert de données entre la mémoire centrale et les périphériques.
- primitives de gestion des interruptions.

Un exemple de modèle de noyau est constitué de couches dont la complexité croît quand on s'éloigne du noyau : couche "gestion de la mémoire", couche "gestion des entrées/sorties", couche "gestion des fichiers", couche "ordonnancement et allocation de ressources". Le noyau, quant à lui, est composé de trois fonctionnalités (dans ce modèle) :

- gestion des processus : création, synchronisation, communication
- dispatcher : ordonnancement de bas niveau : allocation de processeur
- gestion des interruptions : FLIH (First Level Interrupt Handling)



IV. Les taches d'un système d'exploitation

Le système d'exploitation doit s'occuper de :

- l'exécution des commandes d'entrée/sortie
- la gestion de la mémoire
- la gestion des fichiers
- la multiprogrammation (« multitâche ») :
    - Comment exécuter plusieurs programmes à la fois ?
    - Et si ces programmes veulent tous accéder à l'imprimante en même temps ?
    - Et s'ils doivent échanger des informations ?…
    - Comment exécuter un seul programme avec plusieurs processeurs ?

- la sécurité (gestion des accès,…)

Le SE doit aussi fournir :

- un langage de commande
- divers utilitaires (compilateurs, éditeurs, outils, …)
- (une interface graphique pour l'utilisateur)

Chapitre 3 La gestion des processus

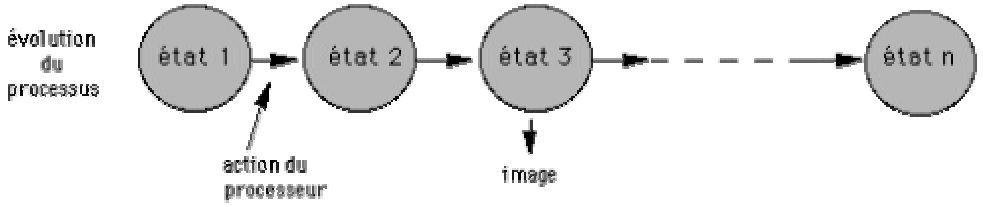
I. Définition d'un processus

On peut donner au processus plusieurs définitions, comme :

- Un programme en exécution
- unité d'exécution
- Une activité asynchrone
- L'entité assignée par un processeur
- Une unité d'envoi

Remarque *: Un programme est une entité passive, par contre un processus est une entité active avec un pointeur d'instructions spécifiant l'instruction suivante à exécuter et un ensemble de ressources associées*

On définira ici un **processus** comme une activité séquentielle gérée par le système d'exploitation et qui évolue sous l'action d'un processeur.



Pour son exécution, un processus nécessite, en effet de consommer ou de produire des ressources.

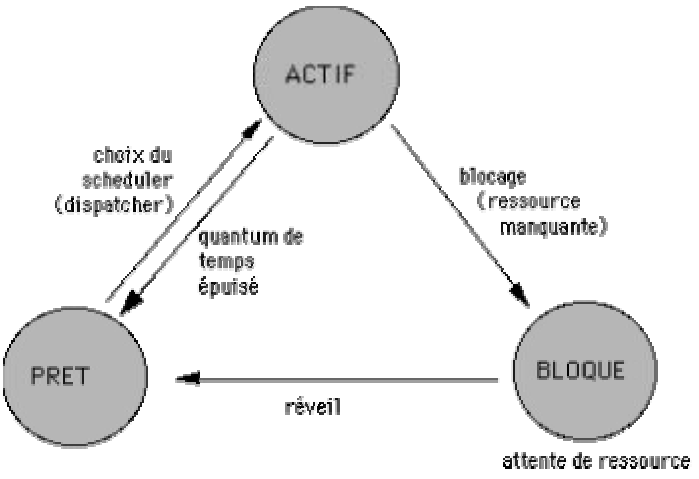
Le contexte d'un processus est un ensemble d'informations composé de

- données sur le processeur (mot d'état, registres)
- données sur l'emploi de la mémoire (zones utilisées)
- données caractéristiques du processus (nom, priorité, droits,...)

II. Etats d'un processus

D'une manière plus formelle, un processus est un programme à exécuter et cette exécution doit progresser séquentiellement c'est-à-dire à n'importe quel moment une instruction au plus est exécutée au nom du processus.
L'état d'un processus est définit en partie par l'activité courante exercée par ce dernier.

- ACTIF ou Running ou en exécution quand ses instructions sont exécutées par le processeur.
- PRET ou Ready ou exécutable quand le processus attend d'être affecté à un processeur.
- BLOQUE ou Blocked quand le processus attend qu'un évènement se produise avant d'être exécuter



Un processeur est affecté ou retiré à un processus par un module du système d'exploitation appelé "dispatcher" (ou Ordonnanceur de bas niveau). Cette situation fait que les transitions entre les trois états ACTIF, PRET, BLOQUE des processus s'opèrent dans les conditions suivantes :

- Transition ACTIF --> BLOQUE :

Le processus nécessite une ressource manquante : le processeur lui est retiré et il se met en attente

- Transition BLOQUE --> PRET :

La ressource devient disponible et est allouée au processus qui se "réveille" et passe à l'état exécutable

- Transition PRET --> ACTIF :

L'Ordonnanceur de bas niveau (dispatcher) choisit le processus pour lui donner le processeur

- Transition ACTIF --> PRET :

Le processeur est retiré au processus pour une raison de gestion du système (un événement est survenu et un nouveau choix doit être fait ou encore le quantum de temps du processus sur le processeur s'est écoulé)

III. Bloc de contrôle de processus

Chaque processus est représenté dans le système d'exploitation par un bloc de contrôle de processus (PCB). Ce bloc contient plusieurs éléments relatifs à un processus ou permettent de les atteindre :

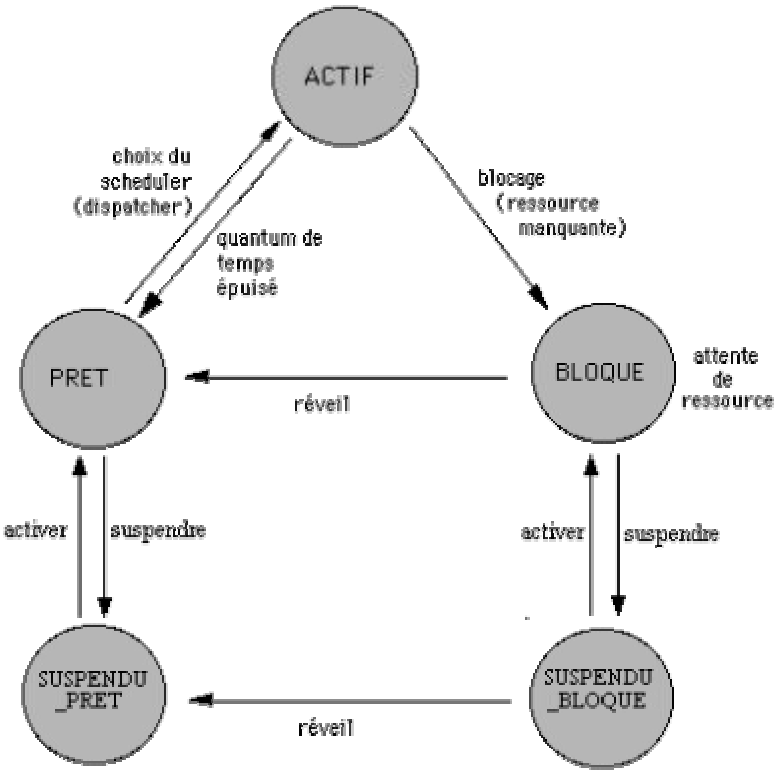
- état du processus
- compteur de programme
- pointeur de pile
- allocation mémoire
- état des fichiers ouverts par le processus
- informations sur le temps d'utilisation des ressources
- informations d'ordonnancement
- zones mémoire et registres de sauvegarde
- etc .....

IV. La suspension et l'activation

Deux autres états sont ajoutés aux états ACTIF, PRET, BLOQUE:

- SUSPENDU_PRET
- SUSPENDU_BLOQUE.

Un processus est suspendu lorsqu 'il ne réside plus en mémoire principale et est stocké en mémoire secondaire.



Les nouvelles transitions induites par les deux nouveaux états sont :

- Transition BLOQUE --> SUSPENDU_BLOQUE

Cette transition a généralement lieu lorsqu'il n'y a pas de processus dans l'état PRET. On suspend alors un processus BLOQUE pour permettre l'accès à la mémoire d'un nouveau processus PRET.

- Transition SUSPENDU_BLOQUE --> SUSPENDU_PRET

Le processus dans l'état SUSPENDU_BLOQUE reçoit les ressources nécessaires ; il passe alors dans l'état SUSPENDU_PRET mais reste en mémoire secondaire.

- Transition SUSPENDU_BLOQUE --> BLOQUE

Cette circonstance est généralement exceptionnelle mais est permise sur certains systèmes.

- Transition SUSPENDU_PRET --> PRET

S'il n'y a plus de processus PRET en mémoire principale, on utilise cette transition ; elle peut également s'opérer pour des processus dans l'état SUSPENDU_PRET qui possède une priorité plus importante que les processus dans l'état PRET.

- Transition PRET --> SUSPENDU_PRET

Cette transition s'effectue de manière volontaire pour des raisons de gestion de la mémoire principale (allègement).

Chapitre 4 La gestion du processeur

I. Introduction

Chaque fois que le processeur devient inactif, le système d'exploitation doit sélectionner l'un des processus de la file d'attente des processus prêts. L'ordonnancement est une fonction principale du système d'exploitation réalisée par un ordonnanceur.

Un Ordonnanceur (scheduler) comporte deux parties :

- Ordonnanceur de haut niveau (Long Term Scheduling): politique de décision, affectation de priorité
- Ordonnanceur de bas niveau ou distributeur (Short Term Scheduling ou dispatcher) : affectation d'un processeur à un processus

Pour l'ordonnanceur de haut niveau, trois concepts sont à retenir :

- Mode de décision

Il s'agit de spécifier des instants où les priorités des processus sont évaluées et comparées et où des processus sont sélectionnés pour exécution. Entre deux instants consécutifs, l'allocation de processeurs aux processus ne change pas.

- mode **non pré-emptif** : un processus en exécution continue jusqu'à ce qu'il se termine ou se bloque (non adéquat pour les systèmes temps réel et temps partagé)

- mode **pré-emptif** : un processus en exécution peut être interrompu par diverses causes : un nouveau processus arrive, un processus existant est réveillé, un temps q

s'est écoulé, la priorité d'un processus prêt est devenue plus grande que celle du processus actif, ...

- Fonction de priorité

A tout moment une priorité est affectée à un processus du fait du résultat de l'application d'une fonction de priorité : ses variables sont divers paramètres (liés au processus ou au système) et la valeur retour est la priorité. Les paramètres utilisés sont usuellement :

- besoins mémoire
- temps d'utilisation CPU
- temps réel dans le système (temps d'utilisation CPU + temps d'attente)
- temps total de service
- priorités externes

La priorité affectée est d'autant plus grande que le temps total de service est court :

-Un processus "court" devrait être terminé rapidement
- Traiter d'abord les travaux courts réduit le temps total d'exécution de tous les processus.

Remarque :

Les priorités externes sont utilisées pour différencier les classes d'utilisateurs et de processus.

II. Les algorithmes d'ordonnancement

Les décisions de l'ordonnancement du processeur peuvent avoir lieu lorsqu'un processus commute (transite) d'un état à un autre.

II.1 Algorithme du premier venu premier servi (First Come First Served : FCFS) :

c'est l'algorithme le plus simple d'ordonnancement, géré facilement par une file d'attente FIFO (first in first out). Quand l'unité centrale est libre, elle est allouée au processus en tête de la file d'attente des processus « prêts ».

Considérons l'exemple suivant :

Processus	Temps CPU
P1	25
P2	3
P3	11

On suppose que l'ordre d'arrivée des processus est : P1 ; P2 ; P3.

Le Diagramme de Gantt, relatif à l'exécution appelé aussi diagramme d'exécution est alors :

P1	P2	P3

0 25 28 39

-Temps d'attente P1 = 0.
-Temps d'attente P2 = 25.
-Temps d'attente P3 = 28.
☞ Temps d'attente moyen = (25 + 28) / 3 = 17,66.

Si l'ordre d'arrivée est : P2 ; P3 ; P1.

-Temps d'attente P1 =14.
-Temps d'attente P2 = 0.
-Temps d'attente P3 = 3.
☞ Temps d'attente moyen = (14 + 3) / 3 = 5,66.

Conclusion : l'algorithme FCFS n'est généralement pas optimal. Il va dépendre de l'ordre de l'arrivée des processus, ainsi que du temps d'exécution.

Remarque :
Cet algorithme est particulièrement incommode pour les systèmes en temps partagé.

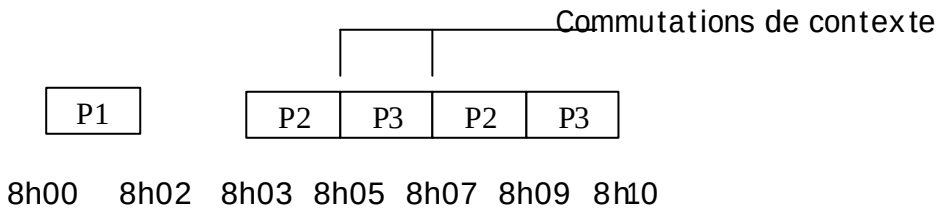
II.2 Algorithme du Tourniquet (Round Robin : RR) :

Cet algorithme a été conçu spécialement pour les systèmes en temps partagé. Il est similaire au FCFS, mais on peut commuter d'un processus à un autre. On peut donc définir une petite unité de temps appelée quantum. La file d'attente des processus prêts est traitée comme une file circulaire. L'Ordonnanceur du processeur parcourt la file d'attente des processus prêts, en allouant le processeur à chaque processus, pendant un intervalle de temps allant jusqu'à une tranche de temps.

Exemple : soit la table du processus suivant :

Processus	Temps d'arrivée	Temps CPU
P1	8h00	2
P2	8h03	4
P3	8h03	3

On suppose ici que la durée d'un quantum q est de 2 unités.

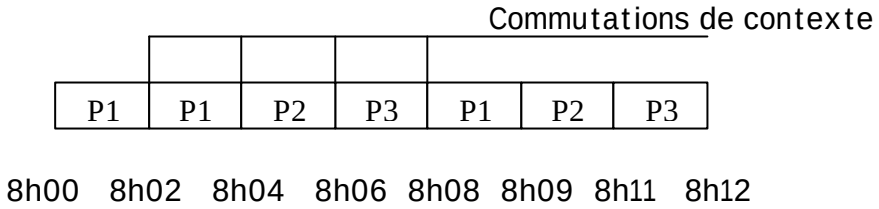


Le processus effectue une commutation de contexte uniquement lorsqu'il atteint la fin du quantum, et qu'il ne s'est pas terminé.

Temps d'attente P1 = 0
Temps d'attente P2 = 2 (lorsque P3 est en exécution, P2 est en attente).
Temps d'attente P3 = 4.

Temps d'attente moyen : 6/3 = 2 unités.

◆Si temps CPU de P1 = 5 :



Temps d'attente P1 = 4
Temps d'attente P2 = 4
Temps d'attente P3 = 6
Temps d'attente moyen = 14/3 unités.

II.3 Algorithme avec priorité :

Chaque processus est muni d'une priorité qui va lui permettre d'être exécuté. Le processus le plus prioritaire va s'exécuter en premier lieu.
La préemption est le fait d'interrompre le processus pour exécuter une autre tâche.

Priorité avec préemption :

Exemple1 : soit la table du processus suivant :

Date d'arrivée	Processus	Temps CPU	Priorité
9h00	P1	6	3
9h03	P2	5	4
9h05	P3	3	2
9h05	P4	4	5

On suppose que la priorité est croissante.

Le diagramme de Gantt d'exécution est alors

P1	P2	P4	P2	P1	P3

9h00 9h03 9h05 9h09 9h12 9h15 9h18

Temps d'attente P1 = 9
Temps d'attente P2 = 4
Temps d'attente P3 = 10
Temps d'attente P4 = 0
Temps d'attente moyen = 23/4 unités.

Priorité sans préemption :

P1	P4	P2	P3

9h00 9h06 9h10 9h15 9h18

Temps d'attente P1 = 0
Temps d'attente P2 = 7
Temps d'attente P3 = 10
Temps d'attente P4 = 1
Temps d'attente moyen : 18/4 unités.

Remarque :
Par défaut, on considère toujours la priorité avec préemption !

Chapitre 5 La gestion des fichiers

I. Motivation

La mémoire principale est de taille restreinte, volatile, on a donc besoin de stocker des informations à plus long terme, de grande taille et partageable.
La gestion de fichiers est une tache du système d'exploitation (file system) qui s'occupe de ces questions :

- de quelle manière nommer les fichiers ?
- comment les structurer ?
- comment les utiliser, les protéger … ?

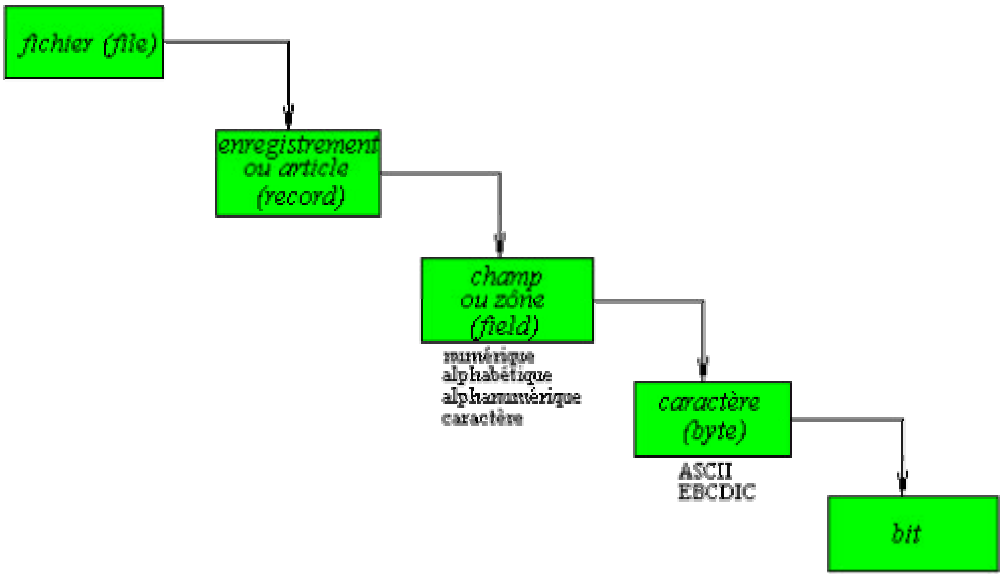
II. Caractéristiques des fichiers

Un fichier est caractérisé par plusieurs attributs :

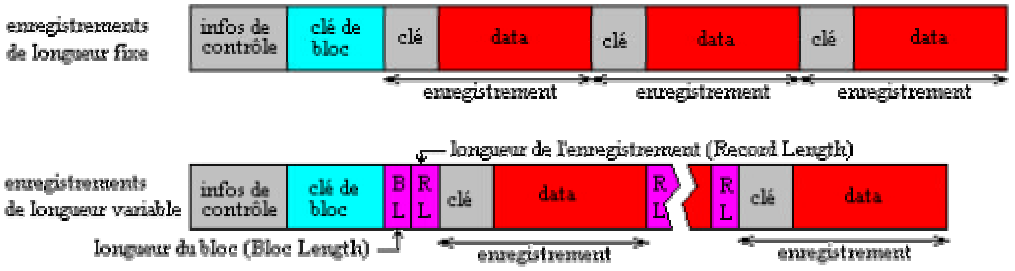
Nom : variable selon les SE composé de lettres de l'alphabet, de chiffres et caractères spéciaux, possédant une longueur et peut avoir deux parties (nom.extension) obligatoire / non.

Structure :

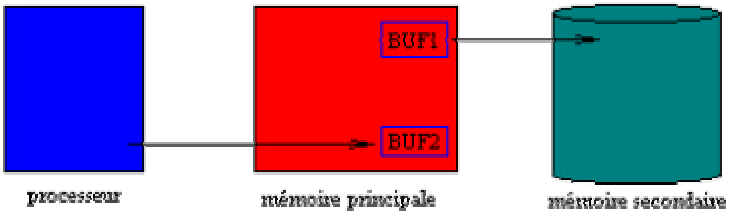
En général, pour les fichiers structurés, il y a une hiérarchie de structure :



Sur un disque, l'unité d'information (de transfert entre la mémoire secondaire et la mémoire principale) est le bloc. Un bloc correspond en général à plusieurs enregistrements :



Les transferts entre mémoire secondaire et mémoire principale nécessitent l'utilisation de tampons (buffers). Une méthode classique est celle du "double buffering". Pendant que le processeur écrit dans BUF2, le contenu de BUF1 est copié en mémoire secondaire :



Accès :

Le mode d'accès aux informations dans un fichier dépend de la technologie du support. Plusieurs modes d'accès sont présentés :

- organisation séquentielle : placement des enregistrements en séquence physique ; utilisation imposée dans les bandes magnétiques ;
- organisation séquentielle indexée : placement des enregistrements en séquence logique suivant une clé contenue dans chaque enregistrement ; utilisation d'une table d'index pour faciliter les recherches ; il y a donc ici un double accès : quasi-direct avec la table d'index et séquentiel dans l'ordre des clés.
- organisation directe : accès direct par adresse physique (en anglais ce mode est appelé "random"). Le temps d'accès à une information est le même quelque soit l'emplacement de l'information

III. Types de fichiers

On note quatre types de fichiers :

- **Fichiers ordinaires** : Utilisés pour stocker les informations des utilisateurs (fichiers classiques), deux exemples peuvent être présentés :
  - Fichier ASCII : C'est un ensemble de lignes de texte dont le caractère « fin de ligne » est variable selon les système d'exploitation. L' avantage principale est qu'il est facilement lisible et modifiable.
  - Fichier binaire : Structuré pour que l'on puisse retrouver ses informations.
- **Catalogues (directories)** : Utilisés essentiellement pour les fichiers systèmes contenant la structure du système de fichiers.
- **Fichiers spéciaux caractères** : Utilisés pour la modélisation d 'E/S (terminaux, imprimante,…).
- **Fichiers spéciaux bloc** : Utilisés pour la modélisation des disques.

IV. Attributs de fichiers

La description de base d'un fichier comporte le nom du fichier et ses données. A cette description s'ajoute un ensemble d' informations complémentaires appelés attributs variables selon le Système d'Exploitation.
On peut citer des exemples d'attributs qui sont :

- Protection
- Mot de passe du fichier
- Créateur
- Propriétaire
- Indicateur lecture seule
- Indicateur fichier caché
- Indicateur fichier système
- Indicateur d'archivage
- Indicateur ASCII/Binaire
- Indicateur fichier temporaire
- Indicateur de verrouillage
- Longueur d'enregistrement
- Date de création
- Date du dernier accès
- Date de modification
- Taille courante
- Taille maximale
- ...

V. Opérations sur les fichiers

Les opérations courantes à effectuer sur un fichier sont :

- CREATE (création d'un fichier vide)
- DELETE
- OPEN
- CLOSE
- READ (lecture de n octets à partir de la position courante)
- WRITE
- APPEND (écriture en fin de fichier)

- SEEK (indication de la position des données à lire/écrire)
- GET ATTRIBUTES
- SET ATTRIBUTES
- RENAME

Ces opérations se font grâce à des appels systèmes :

VI. Stockage des fichiers

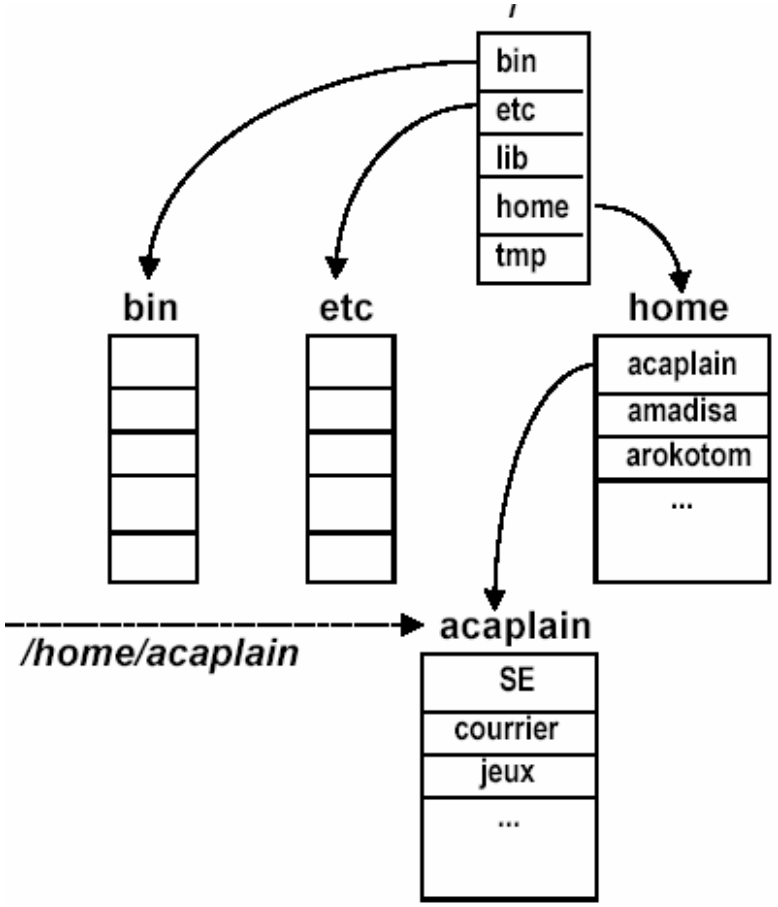
On va s'intéresser essentiellement à la manière qu'utilise le Système d'Exploitation pour sauvegarder les fichiers sur le disque. On peut considérer quatre catégories d'allocation.

- **Allocation contiguë** : tous les blocs se suivent, c'est la méthode la plus simple mais coûteuse en espace disque.
- **Allocation «liste chaînée»** : chaque bloc contient l'adresse du bloc suivant donc pas d'espace perdu mais accès aléatoire lent.
- **Allocation «liste chaînée indexée»** : regroupement des «adresses de bloc» dans une table (index) en mémoire [MS-DOS] l'inconvénient est que la table est entièrement en mémoire.
- **«Nœuds d'information» (inode):** Il s'agit dans ce cas d'utiliser une table associée à chaque fichier, contenant les attributs et les adresses des blocs sur le disque [UNIX], l'avantage est que la table est de petite dimension et que l'i-node contient aussi l'adresse de la liste des blocs vides (utile en cas d'ajout).

VII. Les Catalogues

Utilisés pour la mémorisation des noms, des attributs et des adresses des fichiers. On doit connaître :

- le chemin d'accès c'est-à-dire comment arriver à un fichier situé dans cet arbre ?
- Le chemin absolu à partir de la racine de l'arbre
- Le chemin relatif à partir de la position courante dans l'arbre



Ex. d'arborescence de catalogues sous Unix

VIII. Fiabilité d'un système de fichiers

Qui n'a jamais perdu un fichier lors d'un plantage, panne de courant ou autre ?
Dans ce cas le système d'exploitation pourra effectuer trois actions :

- **Gestion des blocs endommagés** : mémoriser la liste des blocs endommagés pour éviter de les réutiliser directement sur une piste du disque ou dans un fichier.
- **Sauvegarde :** Utiliser soit les méthodes « double disque » ou bien les sauvegardes incrémentales.
- **Cohérence du système de fichiers** : si une panne survient au moment ou un bloc est écrit, le système de fichiers devient incohérent, le SE doit posséder un utilitaire de vérification (et réparation) de la cohérence du système de fichiers

IX. Sécurité et fichiers

Les fichiers doivent être sécurisés contre :

- Perte de données :
  - feu, inondations, …
  - erreurs matérielles / logicielles
  - erreurs humaines
- Intrusions humaines :
  - indiscrétion occasionnelle, jeu
  - chantage économique, piratage commercial, militaire
- Utilisation des informations :
  - protection de l'utilisateur contre un mauvais usage des informations le concernant
  - interdiction de « croiser » des fichiers

Sécurité et fichiers : comment ?

- **Contre les pertes d'informations** :

  – sauvegarde des données dans un lieu différent
  – logiciels de récupération de données sur les disques
  – «commandes simples» : en évitant les erreurs de frappe classique

- **Contre les intrusions humaines** :

  – logiciels anti-virus sur des systèmes peu protégés (Windows)
  – identification de l'utilisateur + accès restreint à ses fichiers et commandes
  – contre-mesures = un compte est bloqué lorsque 3 essais de connexion ont échoué, …
  – tests de sécurité par des experts

Chapitre 6 La gestion de la mémoire

I. Motivation

Le premier ordinateur commercialisé était (PDP8, 1963) qui avait uniquement 4 Ko de mémoire. Le PC domestique actuel est doté d'au moins de 256 Mo de RAM.

Même si la mémoire augmente, la taille des applications et du SE aussi !

d'où l'une des taches du système d'exploitation est la gestion de la mémoire. En effet le système d'exploitation doit :

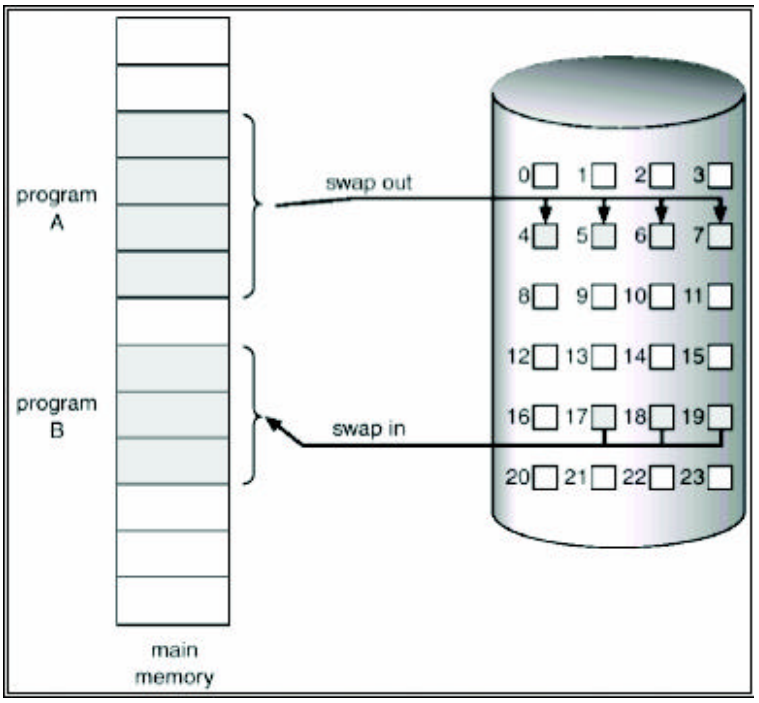
- Connaître les espaces libres
- Allouer de la mémoire à un processus qui en a besoin
- Récupérer la mémoire d'un processus qui est terminé
- Traiter les échanges mémoire - disque **...**

II. Méthodes de gestion de la mémoire

Il existe deux méthodes de gestion de la mémoire :

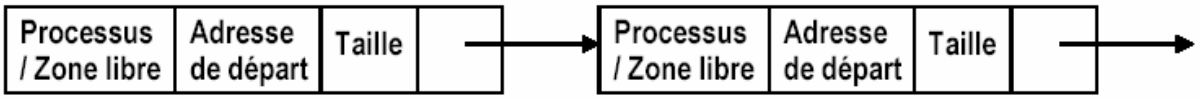
- **Un seul processus en mémoire** : Méthode la plus simple, mais interdit la multiprogrammation, exemple de l'IBM PC.
- **Plusieurs processus en mémoire :** Dans ce cas on aura besoin de "stocker" momentanément des processus sur le disque en utilisant la technique de swapping.

III. Techniques de Swapping



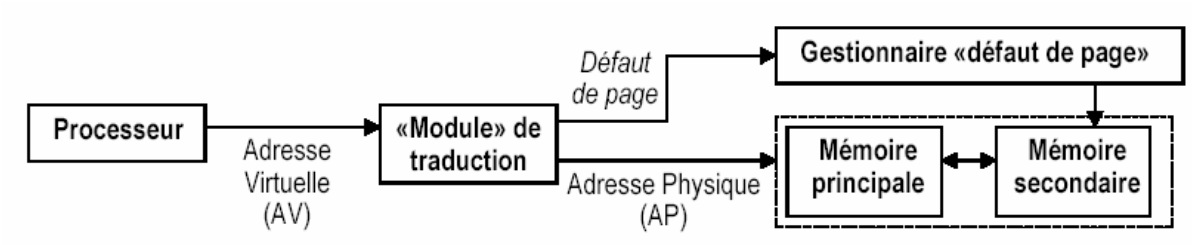
C'est le mouvement des processus entre la mémoire principale et le disque. Chaque processus actif occupe une partie variable de la mémoire et lorsqu'un processus n'est plus actif, il faut le sauvegarder sur le disque.
La zone de swap peut être prévue d'avance par le SE ou allouée à la création du processus et lorsque un processus se réveille, il faut lui trouver une place en mémoire. Le système aura besoin de mémoriser l'occupation de la mémoire principale, il pourra utiliser :

- Un tableau d'occupation par blocs (quelques Ko)
- Une liste chaînée :



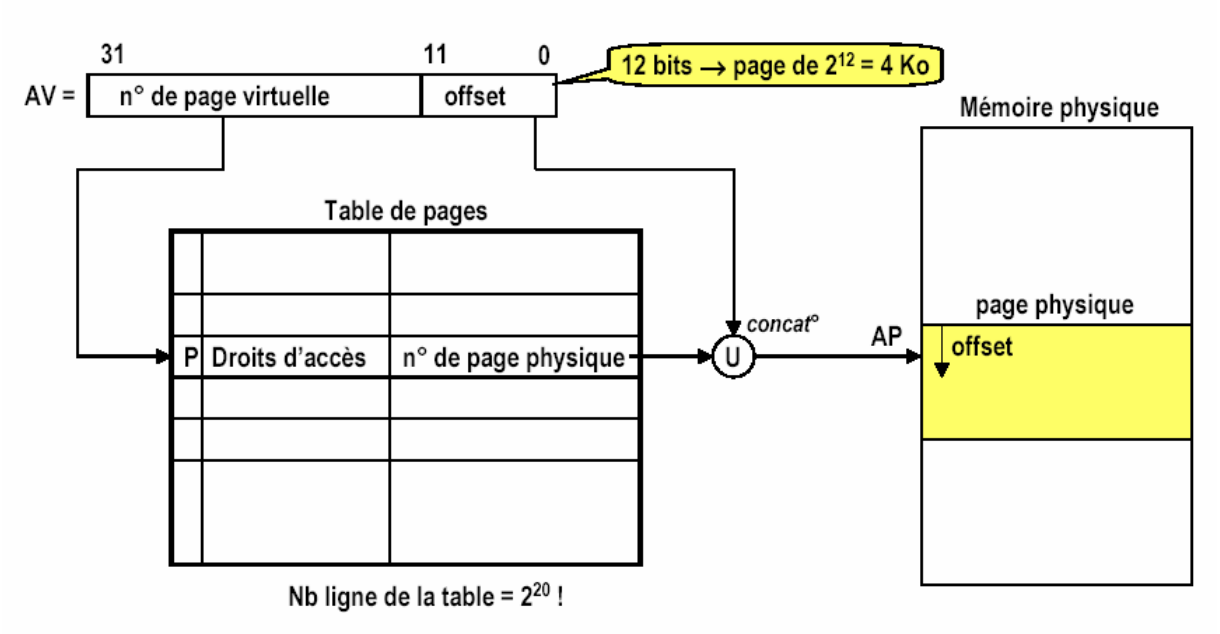
IV. Mémoire virtuelle

Chaque programme considère qu'il s'exécute tout seul à partir de l 'adresse 0 (IP=0) et qu'il a toute la mémoire (mémoire virtuelle).
En réalité, plusieurs programmes s'exécutent à la fois, chacun « possède » une partie de la mémoire physique (pagination/segmentation) et l'accès aux pages mémoire est protégé par le SE (protection).
Le SE doit donc traduire les adresses virtuelles utilisées par les programmes en adresses physiques

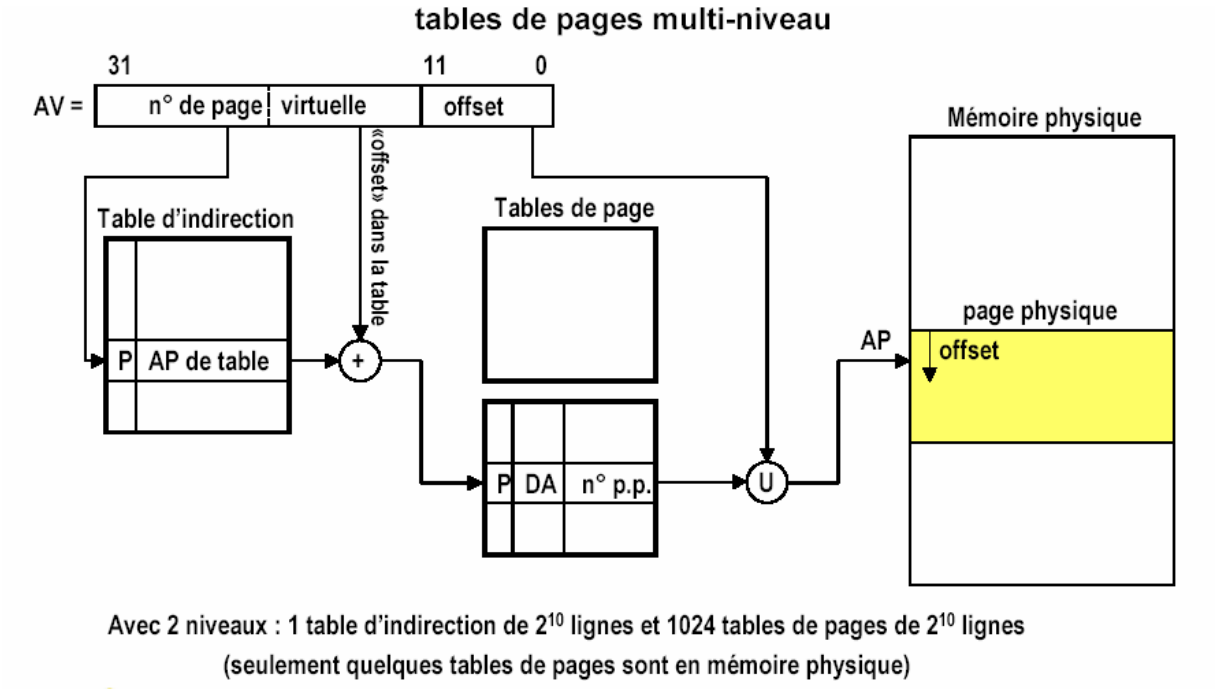


V. Traduction d'adresses Adresse Virtuelle en Adresse Physique :

Le SE utilisera une table de pages (située en mémoire physique)



Pour réduire la présence de la table de pages en mémoire physique, le SE utilisera des tables de pages multi niveaux.



En conclusion on peut dire que la mémoire virtuelle permet l'exécution de programmes plus grands que la mémoire principale, elle favorise aussi :

• La protection : les différents programmes n'interfèrent pas

• La flexibilité : les programmes peuvent se placer n'importe où en mémoire.

Le SE pourra utiliser des mécanismes avancés tels que:

- tables de page multi-niveaux
- accélération de la traduction AV-AP en gardant un «cache» des traductions récentes (TLB : Translation Look-aside Buffer).

VI. Le défaut de page

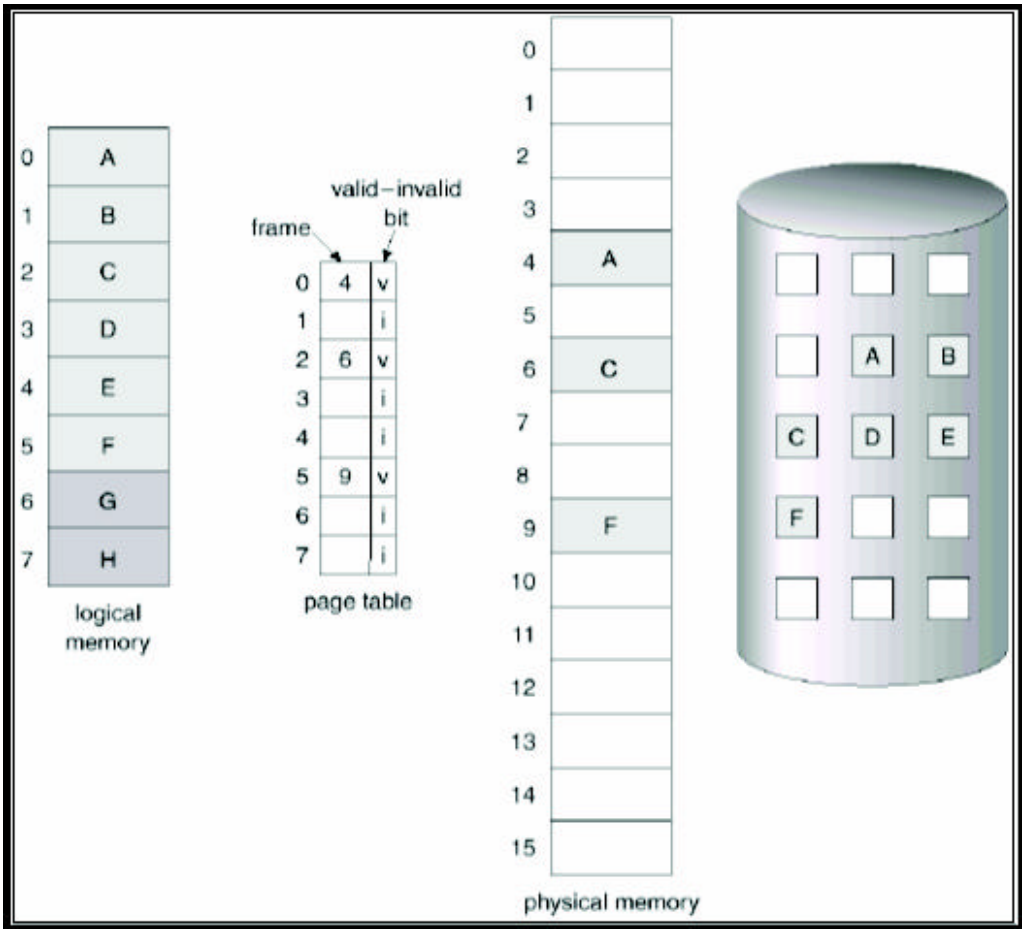
L'adresse virtuelle référence une page qui n'est pas présente en mémoire physique. Le mécanisme d'adressage génère un défaut de page.

Si la mémoire physique est pleine :

- virer de la mémoire physique une page (remplacement) :
  - choisir une page "victime",
  - si elle a été modifiée, la réécrire sur disque,
  - modifier les indicateurs de présence en TPV ;

Puis, dans tous les cas :

- charger la page référencée en mémoire physique (placement) ;
- modifier les indicateurs de présence en TPV.



Différence entre pages sur disque et pages en mémoire

Le bit de présence dans la table des pages est :

- Valide : adresse légale et page en mémoire
- Invalide : adresse illégale ou page sur disque

Pour le choix d'une page victime (remplacement)
De nombreux algorithmes :

- FIFO - First In First Out : ordre chronologique de chargement ;
- LRU - Least Recently Used : ordre chronologique d'utilisation ;
- LFU - Least Frequently Used ;
- Random : au hasard ;
- Algorithme optimal.

VI.1 Algorithme de remplacement : FIFO

L'algorithme le plus simple est Premier Entré Premier Sorti (First-In-First-Out).
Quand une victime doit être sélectionnée c'est la page la plus ancienne qui est sélectionnée.

Exemple

```
Pages adressées :
   7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

Mémoire de 3 blocs :
b1 7 7 7 2 2 2 2 4 4 4 0 0 0 0 0 0 0 7 7 7
b2   0 0 0 0 3 3 3 2 2 2 2 2 1 1 1 1 1 0 0
b3     1 1 1 1 0 0 0 3 3 3 3 3 2 2 2 2 2 1
```

VI.2 Algorithme de remplacement Optimal

Utiliser comme victime la page qui ne sera pas utilisée pendant le plus longtemps.
Cet algorithme n'existe pas mais il est utile pour comparaison.

```
Pages adressées :
   7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

Mémoire de 3 blocs :
b1 7 7 7 2   2   2     2     2       7
b2   0 0 0   0   4     0     0       0
b3     1 1   3   3     3     1       1
```

VI.2 Algorithme de remplacement LRU

LRU (Least Recently Used page).
Nous utilisons ici le vieillissement d'une page et non plus l'ordre de création de la page. On fait le pari que les pages qui ont été récemment utilisées le seront dans un proche avenir, alors que les pages qui n'ont pas été utilisées depuis longtemps ne sont plus utiles.

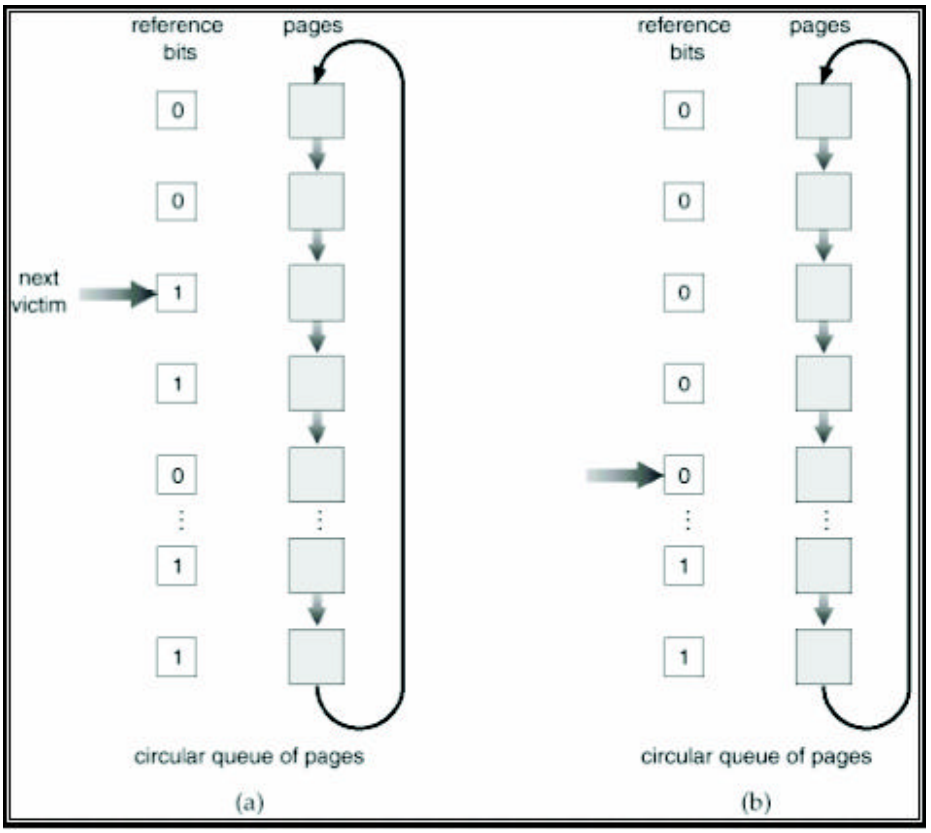
```
Pages adressées :
   7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

Mémoire de 3 blocs :
b1 7 7 7 2   2   4 4 4 0     1   1   1
b2   0 0 0   0   0 0 3 3     3   0   0
b3     1 1   3   3 2 2 2     2   2   7
```

Utilisé, considéré performant

VI.3 Algorithme de remplacement deuxième chance (horloge)

Un bit associé `a chaque page est positionné à 1 à chaque fois qu'une page est utilisée par un processus. Avant de retirer une page de la mémoire, on va essayer de lui donner une deuxième chance. On utilise un algorithme FIFO plus la deuxième chance :

- Si le bit d'utilisation est à 0, la page est swappée hors mémoire (elle n'a pas été utilisée depuis la dernière demande de page).
- Si le bit est à 1, il est positionné a zéro et l'on cherche une autre victime. Ainsi cette page ne sera swappée hors mémoire que si toutes les autres pages ont été utilisées, et utilisent aussi leur deuxième chance.

On peut voir ceci comme une queue circulaire, o`u l'on avance sur les pages qui ont le bit à 1 (en le positionnant à zéro) jusqu'à ce que l'on trouve une page avec le bit d'utilisation à zéro.

Deuxième chance : exemple