

OBJECTIFS DU COURS :

L'objectif de ce module est d'approcher l'étudiant de l'opération « exploitation de la machine » par un ensemble de programmes constituant le système d'exploitation.

Etudier les principes, algorithmes et organisations ainsi fonctionnalités des systèmes d'exploitation. Le but est de dégager les concepts communs à la base des systèmes modernes tel que le temps partagé, l'ordonnancement, la gestion de la mémoire et des disques.

PLAN DE COURS :

Chapitre I : Introduction aux systèmes d'exploitation (Définition, fonctionnalités, historique, exemples)

Chapitre II : Développement d'un programme (Edition de texte, traduction, éditions de lien, chargement, ...)

Chapitre III : Rappels et les interruptions (Les unités dans le processeur, les registres, les modes d'adressage, les interruptions, ...)

Chapitre IV : Gestion de la mémoire centrale (objectifs, différents modes de partage, mémoire virtuelle, pagination, segmentation, ...)

Chapitre V : Gestion de processus et processeur (états de processus, dispatcher, scheduler, les mécanismes d'ordonnancement, ...)

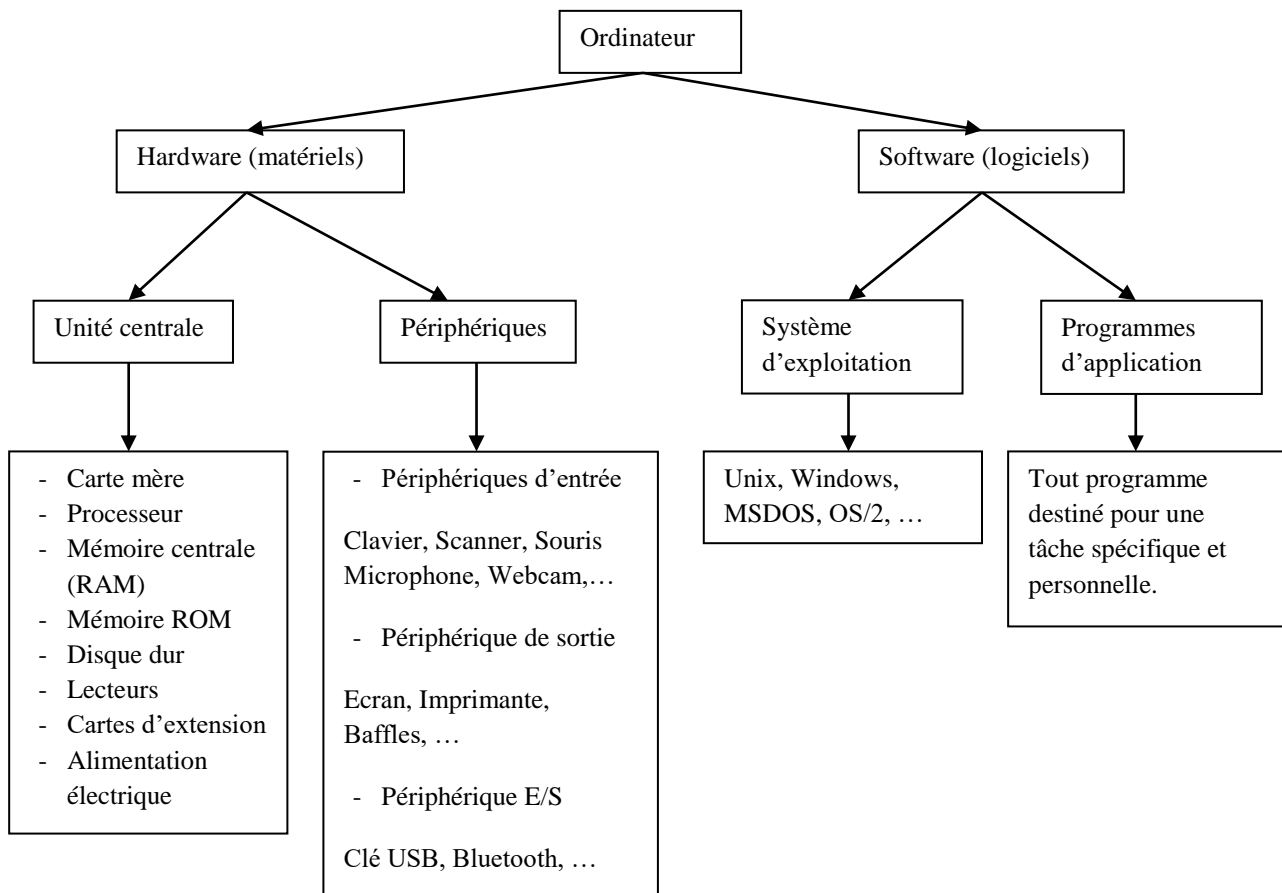
Chapitre VI : Gestion des entrées/sorties (Types d'E/S, Contrôleur de périphériques, modes de pilotage d'une E/S, ...)

CHAPITRE I : INTRODUCTION AUX SYSTÈMES D'EXPLOITATION

I/ Introduction :

Les concepteurs visaient par la création du premier ordinateur de faciliter la vie de l'être humain en automatisant tout type de tâche. Vers les années 40, un seul groupe de personnes concevait, construisait, programmait et utilisait la machine à cause de sa complexité (d'un autre terme, l'utilisation de l'informatique était limitée dans ce groupe), aux cours des années suivantes cette complexité est devenue réduite petit à petit jusqu'aux années 80 où l'ordinateur est évolué pour devenir à la portée de tout le monde tout en utilisant un élément intermédiaire qui doit s'acquitter de toutes les tâches. Cet intermédiaire entre l'utilisateur et l'ordinateur nommé « Système d'exploitation ».

II/ Rappel : Les composants d'un ordinateur



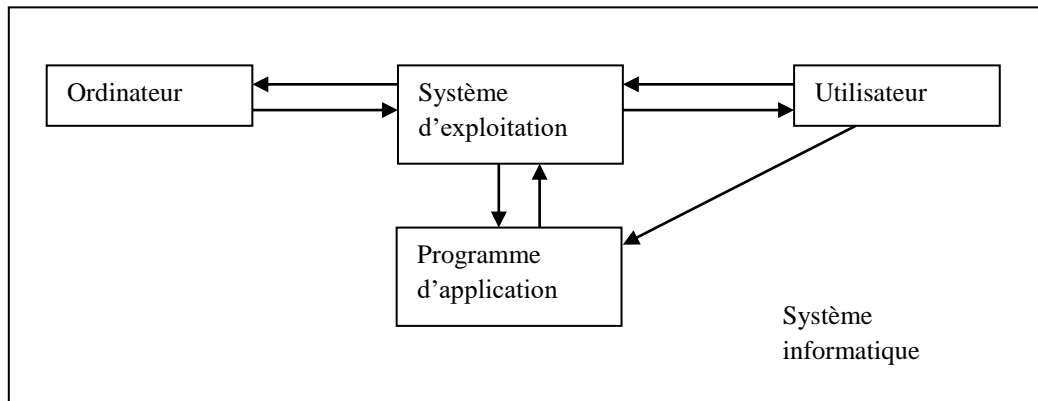
III/ Définition d'un système d'exploitation :

On peut considérer les définitions suivantes :

Définition1 : un système d'exploitation est l'intermédiaire entre l'ordinateur et l'utilisateur.

Définition2 : un système d'exploitation ou « Operating System » est un ensemble de programmes indispensables qui assurent la gestion, le contrôle et le bon fonctionnement des ressources de l'ordinateur.

Définition3 : un système d'exploitation joue le rôle d'un intermédiaire entre les programmes d'application et le matériel de la machine.



NB : un système informatique est constitué du hardware, SE, programmes d'application et utilisateurs

IV/ Les ressources gérées par le SE :

Le système d'exploitation gère deux catégories de ressources :

a/Ressources hardware : il gère le processeur, les mémoires et les périphériques

b/Ressources software : il gère :

- Les programmes d'application.
- Les données des utilisateurs

V/ Les fonctions d'un SE :

Le SE peut fournir plusieurs tâches pour assurer la commodité de l'utilisateur et l'efficacité de l'exécution :

a/ Contrôle d'exécution des programmes :

Le système se charge de contrôler les étapes d'exécution d'un programme (programme en disque → chargement en mémoire centrale → exécution sur le processeur).

Cette tâche devient plus difficile quand on sera dans un environnement multitâche où il faut décider :

- Quel programme exécuter en premier.
- Pour combien de temps.
- Donner la priorité à quel programme ?

b/ Gestion de la mémoire :

Le SE doit partager l'ensemble de l'espace mémoire entre les programmes et les données tout en sachant les espaces libres et occupés pour faciliter et faire rapidement l'exécution.

c/ Gestion de processeur :

Le système d'exploitation doit piloter l'exécution d'un processus au niveau du processeur. Il doit décider la politique d'ordonnancement de programmes au cours d'exécution, ainsi le temps d'exécution de chaque programme (surtout dans les systèmes à temps partagé).

Il doit gérer les états et les priorités d'exécution.

d/ Gestion des opérations des entrées/sorties et des périphériques :

Le SE fournit le service des demandes d'E/S indépendamment de la complexité du périphérique et support utilisé.

f/ Gestion des fichiers :

Le SE doit permettre l'accès des utilisateurs à leurs fichiers par des opérations spécifiques.

g/ Partage des ressources entre les utilisateurs :

Le système d'exploitation se charge de l'allocation de chaque ressource à l'utilisateur nécessaire en utilisant une politique prédéfinie. Il s'occupe à gérer l'accès à cette ressource ainsi que sa libération et sa récupération.

h/ Protection des utilisateurs contre les actions inattendues d'autre utilisateur :

Les actions inattendues sont des résultats de :

- L'utilisation de ressources par des utilisateurs différents.
- L'exécution simultanée de plusieurs programmes.
- Le partage d'un ordinateur par plusieurs utilisateurs.
-

i/ Détection des erreurs de matériels/logiciels et leurs recouvrements :

Le SE doit être capable de détecter les anomalies, les signaler et les corriger si possible.

j/ Virtualisation de la machine :

Le SE doit présenter à l'utilisateur une machine virtuelle (abstraite) à la place de la machine réelle (physique). Cette machine virtuelle fournit à l'utilisateur un langage de commande (ou une interface graphique) mieux adaptée que les instructions de la machine physique.

VI/ Classes de systèmes :

a/ Systèmes orientés processus industriels :

Ce type de système est destiné à piloter un processus industriel, et pour faire, le système doit être en mesure d'assurer les fonctions supplémentaires suivantes :

- Régulation : Le maintien des paramètres de fonctionnement (température, débit, pression, ...) à leurs valeurs nécessaires à la bonne marche de la production.
- Enregistrement : enregistrer périodiquement les valeurs des paramètres de fonctionnement, en vue de garder trace de l'historique, qui peut être utilisé dans des études ciblant l'amélioration de la production.
- Sécurité : pour éviter les catastrophes si certains paramètres dépassent leurs justes valeurs. Le système doit être doté d'une fonction de sécurité capable d'arrêter le processus complet en cas de défaillance.

b/ Systèmes orientés transactions :

Cette catégorie de système est destinée à conserver et gérer des volumes d'information importants (bases de données) comme le système de réservation d'Air Algérie ou le CCP. Les fonctions assurées par ce système sont :

- L'exécution interactive de transaction : ensemble d'opérations indivisibles exécutées sur des bases de données.
- Gestion de la concurrence : les informations dans les bases de données sont partagées entre différents utilisateurs qui peuvent y accéder simultanément. Le rôle de cette fonction est de gérer les conflits d'accès entre utilisateurs.

c/ Systèmes orientés création et exploitation de programmes :

Cette catégorie de système permet à l'utilisateur de créer ses propres programmes et les exploiter.

VII/ Historique et évolution des systèmes d'exploitation :

1/ Les systèmes purement séquentiels (1950-1960) :

Les systèmes d'exploitation dans cette période ont comme principe de travail d'exécuter un seul programme à la fois, on peut citer :

- Les programmes autonomes : Initialement, les périphériques des ordinateurs se résument en un lecteur de cartes, un perforateur de cartes, et une imprimante. Le mode de fonctionnement était assez simple, vu de l'utilisateur, puisque l'exécution d'un programme consistait à mettre dans le lecteur de cartes un paquet contenant la forme "binaire" du programme suivie des données. Après lecture du programme, l'ordinateur en lançait l'exécution, les résultats étant obtenus sous forme d'un paquet de cartes perforées, ou de lignes imprimées.
- Le moniteur d'enchaînement des travaux (1955): l'utilisateur prépare un paquet de cartes représentant son *travail* (*job* en anglais), et qui est constitué d'une ou plusieurs *étapes* (*steps*). Chaque étape correspond à un programme dont l'exécution est demandée par l'utilisateur sur un

jeu de données particulier. Ce programme peut être un des utilitaires conservés sur bande, tel qu'un compilateur, ou le résultat d'une étape précédente mémorisé temporairement sur bande.

2/ Les systèmes introduisant le parallélisme (1960-1965) :

Les plus répandus à l'époque c'est les systèmes à multiprogrammation qui ont comme principe de maintenir plusieurs travaux prêts à s'exécuter en mémoire et partager les ressources entre ces jobs. Le processeur est au départ alloué à un job et le reste sont en mémoire, dès que le job effectue une E/S alors le processeur est alloué au job suivant.

3/ Les systèmes à temps partagé (1965) :

Le processeur dans ces systèmes est commuté sur le processus suivant à chaque expiration d'un délai Q (Quantum de temps).

4/ Les systèmes temps réel :

Sont des systèmes spécialisés dédiés à des applications spécifiques en particulier les systèmes de contrôle, ce système est utilisé lorsqu'il y a des exigences de temps de réponse.

5/ Les systèmes parallèles :

Sont des systèmes fortement couplés possèdent plus d'un processeur qui partagent l'horloge et la mémoire.

6/ Les systèmes répartis :

Sont des systèmes qui exploitent plusieurs processeurs non forcément identiques et chacun possède sa mémoire.

7/ Les systèmes embarqués :

Sont des systèmes pour des terminaux mobiles.

VIII/ Exemples de systèmes d'exploitation :

Plusieurs variantes de systèmes d'exploitation ont apparu dès les années 50 jusqu'aujourd'hui, les fameux sont :

1/ MS-DOS: Microsoft Disk Operating system

C'est un système mono-utilisateur, il est connu au début des années 80, il est destiné au micro-ordinateur (les premiers IBM-PC 8088), conçu par Bill Gates, associé aujourd'hui avec Windows.

2/ Le système Windows :

Système conçu par Microsoft avec une interface graphique, son nom désigne l'utilisation d'une fenêtre pour chaque exécution. Il est destiné à plusieurs architectures (PC, Station de travail, Portable, Réseau client/serveur, ...).

Il est passé par plusieurs versions comme pour PC : 1.0, 2.0, 3.10, 3.11, 9x, 2000, Me, XP, Vista, Seven, 10, aussi : Windows Server, NT pour l'architecture multi-utilisateur et client/serveur.

3/ Le système UNIX : Uniplexed Information and Computer Service

Une famille de systèmes en temps partagé proposée pour la plupart des architectures, utilisée pour certains milieux d'industrie et de recherche.

- Historique :

1969 : Naissance d'UNICS aux laboratoires de Bell par Ken Thompson → une altération en UNIX.

1974 : Publication d'UNIX en langage C par Thompson et Ritchie.

1979 : Apparition des versions BSD (Berkeley Software Distribution) de l'université de Berkeley.

Jusqu'à 1980 : UNIX est offert aux maisons d'enseignements et aux entreprises.

Aux années 80 : La société AT & T commercialisa les versions System III, System IV et System V

1981-1994 : Apparition de Linux.

- Les raisons de succès d'UNIX :

- Ecrit à l'aide d'un langage de haut niveau par rapport aux autres systèmes à l'époque.
- Interface utilisateur adaptée.
- Open source → possibilité d'amélioration dans le code source par n'importe le qui ce qui résulte l'apparition de plusieurs variantes par plusieurs sociétés comme : System V de AT & T, XENIX par Microsoft, ULTRIX par DEC, pour Linux → Redhat, Coldera, ...
- Disponibilité de versions gratuites.

- Les services offerts par le noyau UNIX :

- La gestion des évènements matériels.
 - La gestion des ressources.
 - La gestion des structures de données.
 - La gestion de la mémoire et le processeur.
 - Fournit des services élémentaires.
- C'est un système multitâches, multi-utilisateurs, multissessions, et multipostes.

ANNEXE :

1/ Unité centrale :

Carte mère : c'est une carte électronique qui permet le branchement de tous les composants du PC et la circulation des informations.

Processeur : C'est le cerveau de la machine qui exécute toutes les opérations demandées et donne les réponses attendues.

Mémoire vive « RAM » : C'est le lieu de travail de processeur permet la sauvegarde des informations nécessaires pour le traitement, c'est une mémoire volatile qui s'efface à chaque coupure de courant électrique.

Disque dur :

C'est le lieu de stockage des informations, c'est une mémoire permanente.

Lecteur de CD/DVD et lecteur de disquette :

Ce sont des lecteurs qui permettent l'accès à des disques externes (disquette, DVD et CD-ROM).

Cartes d'extension :

Ce sont des cartes qu'on ajoute à la carte mère pour assurer d'autres fonctionnalités qui sont absentes dans la carte mère (Exemple : carte réseau, carte son, carte graphique, ...).

2/ Périphériques :

Périphériques d'entrée :

Ce sont les éléments du PC, tel que : Le clavier, la souris, le scanner, le modem...etc. permettant d'entrer des données.

Périphériques de sortie :

Ce sont les éléments du PC permettant de faire sortir des informations tel que : L'écran, l'imprimante, les baffles...

Périphériques d'entrée/sortie :

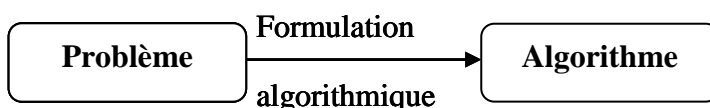
Ce sont les éléments du PC permettant d'entrer et de faire sortir des informations tel que les clés USB, les appareils à photos numériques, les imprimantes-scanner, ...

CHAPITRE II : DÉVELOPPEMENT D'UN PROGRAMME

I/ Introduction:

Pour qu'un problème puisse être interprété, exécuté et donne les résultats attendus par le processeur de la machine, il faut qu'il soit formulé dans un langage compréhensible par la machine et il faut que le système d'exploitation pilote son exécution.

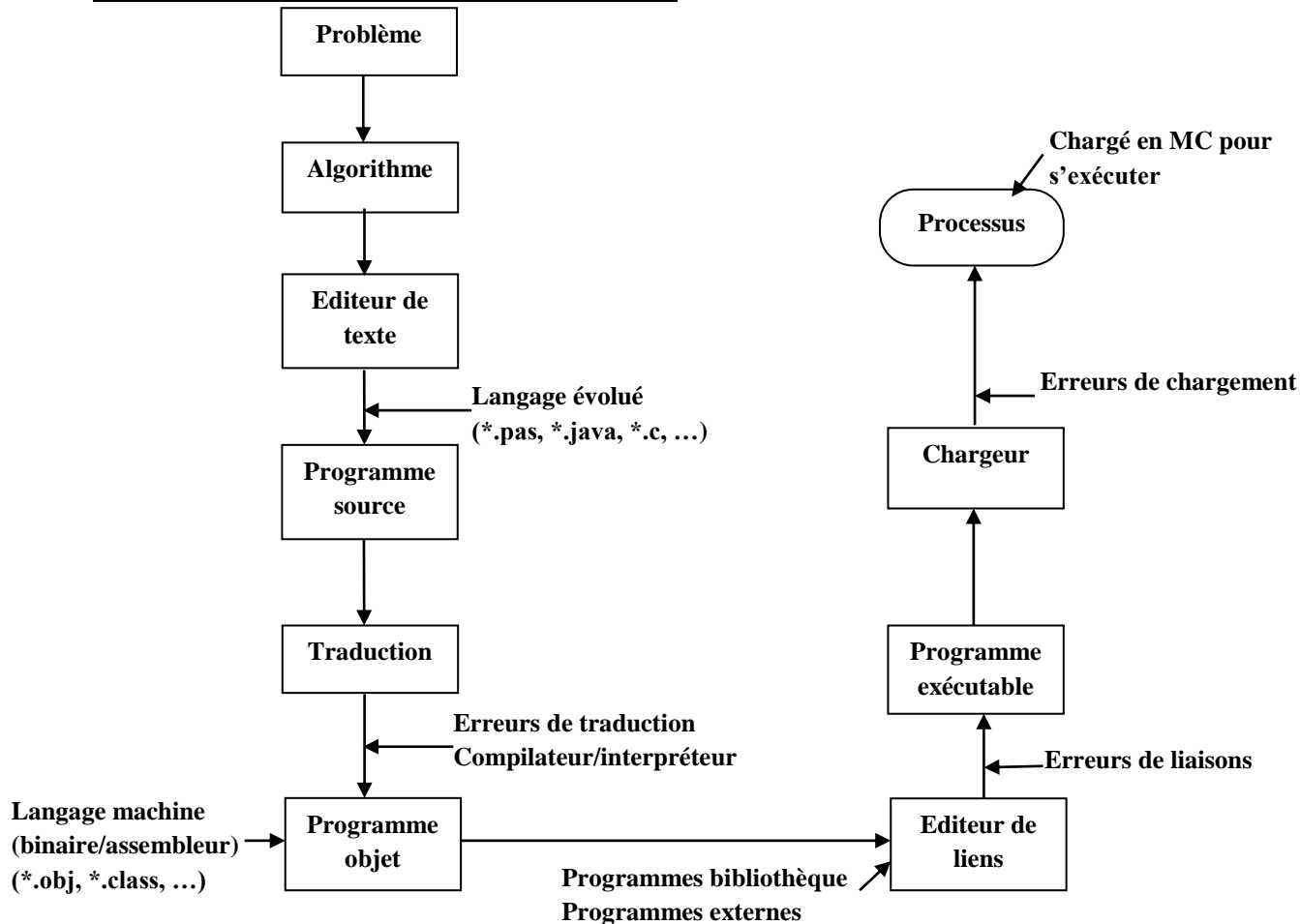
Pour le faire, le programmeur commence au début d'écrire son problème d'une manière structurée sous forme d'un algorithme.



Cet algorithme est écrit en langage humain et pour le rendre exécutable par l'ordinateur, le programmeur fait le passer par plusieurs étapes tout en utilisant des outils de langage de programmation et des outils système.



II/ Les étapes de cheminement d'un programme:



Ces étapes permettent de bien traduire le code source en code machine tout en permettant au programmeur de corriger les erreurs apparues dans chaque étape.

1/ L'édition de programme :

C'est l'étape dont on saisit l'algorithme établi pour résoudre le problème tout en le traduisant à un programme écrit dans un langage évolué (exemple : Java, Pascal, Delphi, C++, ...) ou dans le langage Assembleur. L'éditeur de texte est l'outil qui nous permet de réaliser cette étape, c'est un logiciel interactif soit associé au système, soit associé au langage de programmation. Il nous permet d'appliquer toutes les opérations nécessaires sur un fichier nommé « **Code source** »

Ce code source à une extension spécifique tout dépend du langage utilisé (*exemple* : *.pas, *.java, *.cpp, ...)

Exemple : Bloc-notes, Wordpad, Edit de MS-DOS, Editeur du compilateur C, JEDPlus pour java, ...

2/ La traduction en langage machine :

Le traducteur est un outil système qui permet de traduire les instructions écrites dans un langage de programmation vers des instructions écrites dans le langage machine (Binaire), on distingue deux types de traducteurs :

- **Le compilateur :**

Est un logiciel de complexité grandissante, permettant de traduire un programme source vers le langage machine en passant par :

- **L'analyse lexicale** : le rôle de cette phase est la reconnaissance des unités lexicales (tokens), en inspectant le code source caractère par caractère. Ces unités peuvent être des nombres, des identificateurs, des mots clés, des opérateurs, ... Chaque unité est décrite par un type (mot clé, identificateur, constante, opérateur, ...) et une valeur (le nom qui figure dans le code source). Alors le résultat de cette étape :
 - La création de la table de symbole.
 - Elimination de blancs et de commentaires.
 - Signalisation des erreurs lexicales.
- **L'analyse syntaxique** : vérifie que le code source à compiler respecte bien les règles syntaxiques du langage. Le résultat est la création de l'arbre syntaxique en se basant sur les unités lexicales et l'ensemble des règles syntaxiques du langage. Elle met à jour la table des symboles en ajoutant les informations manquantes comme les types des identificateurs, et elle signale les erreurs dues au non-respect de la syntaxe du langage.
- **L'analyse sémantique** : qui vérifie la sémantique du programme. Cet analyseur utilise l'arbre syntaxique pour identifier les opérandes et les opérations et il utilise la table des symboles pour identifier les types des opérandes. Il signale les erreurs sémantiques qui peuvent être faites dans le code.
- **L'optimisation de code.**
- **Transformation en code objet** : c'est la partie qui réalise effectivement la traduction du code source en code objet.

Lors de ces étapes le compilateur signale des erreurs à corriger qui peuvent apparaître dans chaque étape. Le résultat de cette phase est un fichier nommé code objet a une extension spécifique selon le langage de programmation (*exemple* : *.obj Pascal, *.class Java, ...)

- L'interpréteur :

L'interpréteur est un logiciel qui permet de traduire le programme vers le langage machine et de l'exécuter en même temps.

Ce type de traducteur ne résulte pas un fichier contenant le code objet ce qui impose l'interprétation à chaque fois d'exécution (*exemple* : l'interpréteur HTML).

N.B : le compilateur définit deux types de codes dans le code objet :

- Le code absolu qui a une adresse fixe dans la mémoire.
- Le code translatable qui peut se mettre dans n'importe quel emplacement dans la mémoire.

L'opération de translation consiste à ajouter à chaque emplacement qui contient une adresse, la valeur de l'adresse effective de l'implantation finale dans la mémoire.

3/ L'édition de liens :

Lors de la traduction d'un programme en code objet, le compilateur associe à chaque instruction son type, dont on peut avoir :

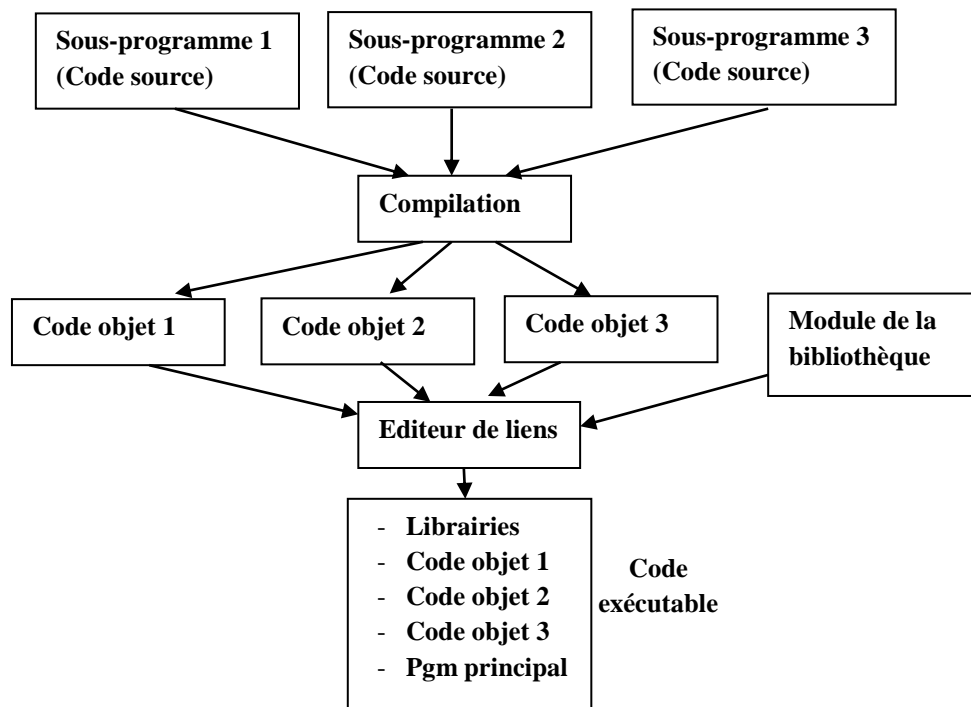
- Des données variables
- Des données constantes
- Des instructions
- Des appels à des procédures, des modules ou des variables externes.

Dans le dernier type, le programmeur peut référencer des structures ou des modules externes par rapport au module en question (il s'agit des références externes de la bibliothèque du langage ou personnelles). Alors le compilateur marque ces références sans les ramener et les ajouter au code objet.

Donc il associe à chaque référence soit :

- Un lien interne au programme, mais qui peut être accédé par d'autre modules → lien utilisable (définition externe).
- Un lien externe appartient à un autre module appelé dans ce programme → lien à satisfaire (référence externe).

L'éditeur de lien est un outil système qui a comme objectif d'accomplir la tâche du compilateur en ajoutant les codes objets de tous les liens à satisfaire dans le code objet principal. Alors l'éditeur de liens cherche l'origine de chaque référence externe et établit une table globale des modules contenant les informations : nom du module, taille et adresse d'implantation.



Le travail de l'éditeur de liens est basé sur la table de fonctions qui contient toutes les informations concernant la bibliothèque. On peut distinguer entre deux types des éditeurs de liens :

Editeur de lien statique : c'est l'édition de liens qui s'établit une fois pour toute et elle engendre un fichier résultat liant toutes les références externes avec le programme principal, nommé le fichier exécutable. Ce type des éditeurs de liens ne se refait pas à chaque exécution.

Exemple : l'éditeur de liens du langage Pascal → des fichiers *.exe.

Editeur de liens dynamique : c'est l'édition de liens qui s'établit à chaque demande d'exécution du programme, elle n'engendre pas un fichier exécutable.

Exemple : l'éditeur de liens du langage Java

4/ Le chargement :

Après l'édition de liens, le programme est prêt à s'exécuter, pour faire il faut le charger dans la mémoire centrale. Cette phase est faite par un outil système nommé le chargeur. Cette étape consiste à lire les instructions en mémoire secondaire, et les transférer en mémoire centrale tout en lisant au début l'adresse de chargement fournie par l'éditeur de liens. Dans ce cas le programme devient un processus.

On distingue deux types de chargement :

Le chargement absolu : le code chargé ne doit être pas mis dans un autre bloc de données différent de celui indiqué dans le code objet, (d'une autre façon recopier le code).

Le chargement relogable (translatable) : le chargement peut se réaliser à n'importe quelle adresse dans la mémoire centrale, la façon de traduire les adresses est basée sur les informations fournies par l'éditeur de liens.

5/ Le débogueur :

C'est un logiciel qui permet d'exécuter le programme pas à pas, d'afficher les valeurs des variables à tout moment et mettre en place des points d'arrêts sur des conditions ou des lignes du programme. Il offre au programmeur la possibilité de contrôler l'exécution et de détecter l'origine des erreurs non corrigées.

CHAPITRE III : RAPPELS ET LES INTERRUPTIONS

I/ Introduction:

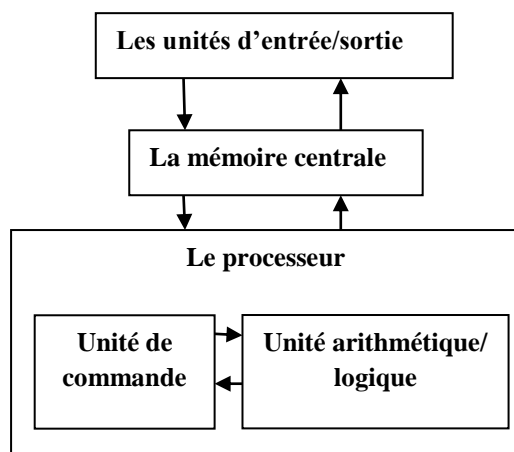
La mise en œuvre d'un système d'exploitation nécessite la connaissance des caractéristiques des éléments hardware, certains éléments participent dans l'exécution des programmes et autres interrompent leurs exécutions. Le système d'exploitation a comme fonctionnalité de les gérer pour conduire à une exécution fiable.

Dans ce chapitre, nous étudierons les composants du processeur et les concepts des interruptions.

II/ L'architecture Von Newmann:

La machine Von Newmann est constituée des éléments :

- La mémoire centrale pour sauvegarder les données et les programmes en cours d'exécution.
- Le processeur pour effectuer les calculs et les traitements nécessaires.
- Les périphériques pour l'échange de l'information.



III/ Le processeur:

Unité capable d'exécuter une instruction, elle se compose de :

1/ L'unité de contrôle et de commande :

Sert à contrôler le bon fonctionnement de traitement et commande le déroulement des instructions, elle se compose de :

- Compteur ordinal (CO) : registre contenant l'adresse du mot mémoire où se trouve l'instruction suivante à exécuter.
- Registre d'instruction (RI) : conserve l'instruction courante pendant son interprétation.
- Décodeur d'instruction (DI) : analyse le code opération de l'instruction pour distribuer les différentes commandes élémentaires.
- Séquenceur : il ordonne l'ensemble des microcommandes à l'ensemble des unités (mémoire, UAL, ...) pour réaliser le traitement de l'instruction.

- Registre d'état (Processor Status Word PSW): contient des informations sur l'état du CPU et les instructions qui viennent d'être exécutées, c'est un masque de n bits contenant des indicateurs nommés drapeaux (flags) comme : indicateur de débordement, indicateur de parité, indicateur du mode d'exécution, bit de protection, code d'interruptions, ...
- L'horloge: est un circuit qui transmet régulièrement selon une périodicité déterminée des impulsions électriques, elle définit le fonctionnement séquentiel du processeur de sorte que les cycles machines sont synchronisés avec l'horloge. (L'horloge est un signal carré avec une fréquence fixe comme 3Ghz).

2/ L'unité arithmétique et logique :

L'UAL est composée des circuits dont le but est d'effectuer un traitement (les calculs) sur les opérandes sous le contrôle de l'unité de commande.

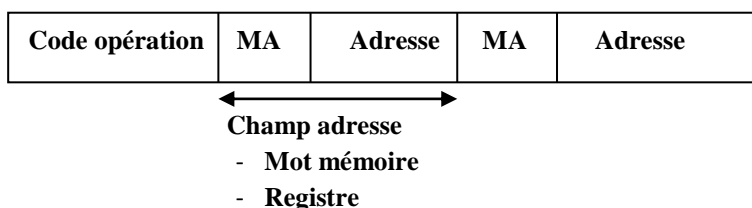
3/ Autres :

Le processeur contient d'autres registres dans la plupart des machines, citons :

- Les registres généraux : sauvegardent les instructions fréquemment utilisées et les résultats intermédiaires. Dans la plupart des machines, on dispose de 7 registres.
- Registre d'index : utilisé pour comptabiliser le nombre d'itération dans une boucle (par exemple les tableaux).
- Registre de base : utilisé pour calculer l'adresse effective dans la mémoire centrale pour un bloc d'instructions.
- Registre pointeur de pile : utilisé pour l'accès à la pile.
- Accumulateur : utilisé pour sauvegarder les résultats surtout dans la multiplication et l'addition.

V/ Les modes d'adressage:

Le mode d'adressage nous renseigne sur la manière dont l'instruction adresse ses opérandes, sachant que :



1/ Adressage immédiat :

L'opérande est une constante contenue dans le champ adresse

Exemple : MOV AX, #100 → charger dans le registre AX la valeur 100.

2/ Adressage direct :

L'opérande se trouve en mémoire centrale, le champ adresse contient son adresse effective.

Exemple : MOV R1, 100 → charger dans R1 le contenu du mot mémoire 100

3/ Adressage indirect :

Le champ adresse contient l'adresse d'un pointeur vers la mémoire.

Exemple : MOV R1, (R2) → charger dans R1 le contenu d'une adresse localisé par le pointeur dont son adresse existe dans R2

4/ Adressage indexé :

L'adresse de l'opérande est calculée en additionnant le contenu du champ adresse au contenu du registre d'index (utilisation dans les tableaux).

5/ Adressage basé :

Le contenu de registre de base + le contenu du champ adresse.

6/ Adressage relatif :

Le contenu du compteur ordinal + le contenu du champ adresse (utilisation dans les branchements).

VI/ Les interruptions:

Dans un ordinateur, existent deux types de programmes :

- Les programmes usagers, qui font un calcul utile.
- Les programmes du système, qui font un travail de supervision de tous les évènements arrivant à la machine.

Ces deux types de programmes se partagent le processeur de telle sorte de s'occuper des programmes utilisateurs et de prendre en considération les évènements produits.

Les évènements produits dans la machine sont deux types :

- Les évènements synchronisés liés aux programmes comme : la division par zéro, tentative d'accès à une zone interdite, ...
- Les évènements asynchrones liés aux matériels comme une fin d'opération d'E/S, signal d'horloge, ...

Ces évènements sont la manière dont le processeur contrôle continuellement l'état d'une ressource, sont nommés les interruptions.

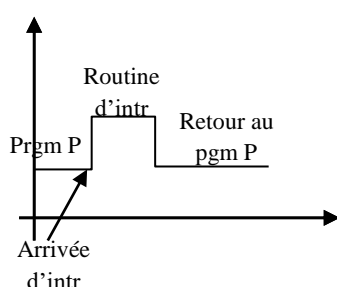
1/ Définition :

Une interruption constitue un mécanisme pour lequel les modules (E/S, mémoire, processus) peuvent interrompre le traitement normal du processeur.

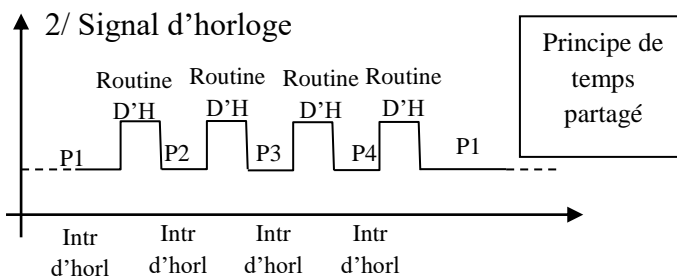
L'interruption est une réponse à un évènement qui interrompt l'exécution des programmes en cours à un point observable du processeur central, elle se traduit par un signal envoyé au processeur, elle permet de forcer le processeur à suspendre l'exécution du programme en cours et à déclencher l'exécution d'un programme prédéfini appelé « **routine d'interruption** ».

Exemple :

1/ Fin d'E/S



2/ Signal d'horloge



2/ Les types des interruptions :

On distingue deux types d'interruptions :

- Les interruptions externes liées aux matériels.
- Les interruptions internes liées aux programmes.

a/ Les interruptions externes causées par des organes externes au processeur comme les unités d'E/S, on peut citer :

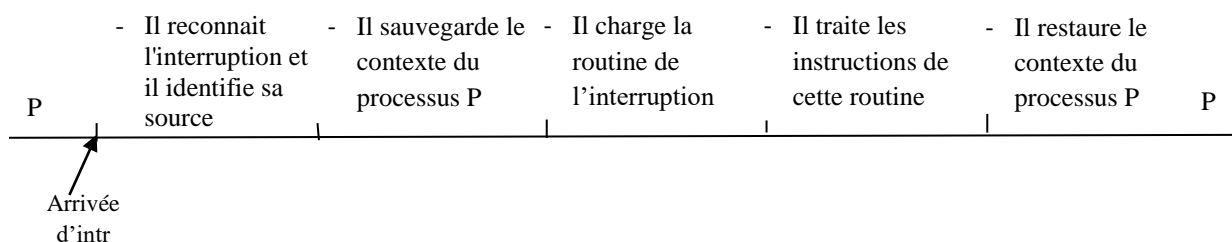
- Un périphérique devient prêt.
- Réinitialisation du système.
- Une erreur durant une E/S.
- Une fin d'E/S.

b/ Les interruptions internes dont on peut les diviser en deux types :

- Les interruptions de l'horloge : à travers ces interruptions, le SE gère le temps d'exécution entre l'ensemble des processus dans le système.
- Les déroutements : sont causées par des erreurs lors de l'exécution des programmes, un déroutement ne doit pas être masqué ou retardé, comme exemple :
 - Tentative d'exécution d'une opération interdite ou invalide.
 - Violation d'accès à une zone protégée ou inexistante.
 - Division par zéro.
 - Débordement arithmétique.
- Les appels au superviseur : sont causés par les instructions d'appel à des fonctions d'E/S système comme l'instruction de lecture à partir du clavier, l'instruction d'impression, ...

3/ Mécanisme de gestion des interruptions :

Quand un processus utilisateur **p** est en cours d'exécution, dans un temps **t**, et le processeur reçoit une interruption quelconque plus prioritaire, alors :



Plus précisément, la connaissance d'une interruption peut avoir lieu à des moments différents :

- *Pour une interruption externe* : le processeur possède d'un bit dit bit d'interruption qui prend la valeur 1 par le hardware quand une interruption est déclenchée par un périphérique. Le processeur teste la valeur de ce bit avant de commencer l'instruction suivante (alors les interruptions externes sont prises en compte au début de chaque instruction).
- *Pour une interruption interne* : elle aura lieu le moment de l'exécution d'une instruction (surtout un déroutement):
 - Dans l'étape de décodage (comme la tentative d'exécution d'une instruction non existante, ou une instruction privilégiée en mode esclave).
 - Dans l'étape de recherche de l'opérande (dans le cas de violation de mémoire).
 - Dans l'étape de déclenchement de l'opération (dans le cas de dépassement de capacité).

Pour identifier la source d'une interruption, la technique utilisée couramment est de consulter un vecteur de bits où chaque bit est associé à un périphérique.

N.B : sachant que :

- **Un contexte** du programme c'est l'ensemble des informations nécessaires pour la reprise d'un processus après son interruption, il contient : son compteur ordinal, son mot d'état, l'accumulateur et le contenu des registres généraux.
- L'opération de sauvegarde de contexte d'un processus et le chargement ou la restauration du contexte d'un autre processus est nommée : « **Commutation de contexte** ».

4/ Les conditions d'arrivée d'une interruption :

Pour que le processeur prenne en charge une interruption juste après son arrivée, il faut vérifier les conditions suivantes :

- Le processeur doit être dans un point observable, d'un autre terme, le code, qui est en train d'être exécuté, doit être un code divisible où on peut l'interrompre à tout moment et dans n'importe quelle instruction. (On dit un code indivisible dans le cas inverse)
- Le système d'interruption doit être actif, dont le processeur contient un mécanisme d'activation et de désactivation du système d'interruption. Il désactive ce système s'il est en train d'exécuter des instructions protégées contre les interruptions (exemple : les instructions de sauvegarde de contexte).
- L'interruption doit être plus prioritaire que le programme en cours.
- L'interruption doit être démasquée, où on peut masquer une interruption à cause de sa priorité. Quand une interruption masquée arrive, elle sera retardée.

5/ Les priorités entre les interruptions :

Durant un temps déterminé, on peut avoir plusieurs demandes d'interruptions simultanées, et pour les traiter d'une manière équitable le système définit un certain ordre de priorité entre les niveaux d'interruptions, ces priorités peuvent être fixes ou modifiables.

CHAPITRE IV : GESTION DE LA MÉMOIRE CENTRALE

I/ Généralités:

Une mémoire est un système capable d'acquérir, de conserver, et de restituer des informations binaires dans un ordinateur. La capacité d'une mémoire est la quantité d'information qu'elle peut stocker, elle est exprimée en bits ou en mots de 2^n bits.

Le temps d'accès et le temps qui s'écoule entre la demande de l'information et le moment où elle est disponible.

Quatre types de mémoires qui se diffèrent selon l'utilisation, la capacité, le temps d'accès et le coût :

- Les registres (sont les plus petites mémoires qui ont le plus court temps d'accès et les plus chers)
- La mémoire cache (elle a une faible capacité, très rapide, et le coût très élevé).
- La mémoire centrale (capacité de 128 MO \rightarrow 2 GO, avec un accès très rapide, moins cher).
- Les mémoires auxiliaires (elles ont une très grande capacité, avec un accès long et sont les moins chers).

II/ Définition de la mémoire centrale:

La mémoire centrale est un ensemble d'emplacements ou mots mémoires, chacun possédant sa propre adresse, elle répond aux opérations de lecture et d'écriture. La MC est utilisée pour le stockage des données et des programmes au cours d'exécution, c'est l'intermédiaire entre le processeur et le reste des organes de l'ordinateur.

L'ensemble de programmes permettant la gestion et le contrôle d'utilisation de la mémoire centrale est dit système de gestion de mémoire centrale.

III/ Objectifs du gestionnaire de mémoire:

Les objectifs du gestionnaire de MC sont :

- 1/ Le partage et le contrôle de l'espace mémoire entre les processus.
- 2/ La protection de l'espace alloué à chaque processus.
- 3/ La réallocation et le réarrangement de la mémoire pour optimiser son utilisation.
- 4/ L'organisation logique, c'est-à-dire, division logique de l'espace en partitions, pages ou segments.

Pour cela le gestionnaire de la MC :

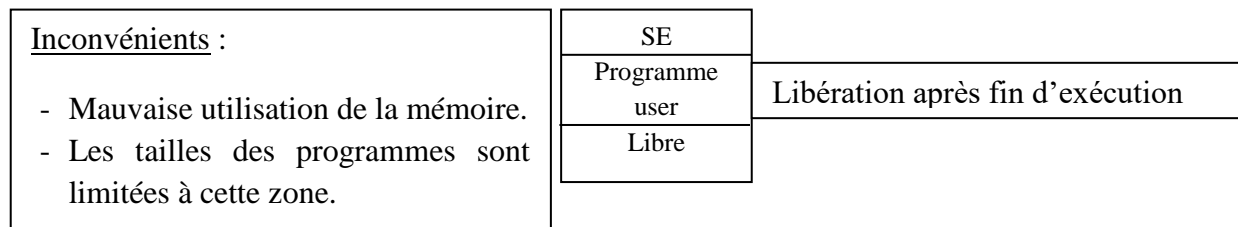
- Alloue l'espace mémoire aux processus.
- Libère l'espace occupé.
- Garde trace des parties mémoire occupées et libres.
- Gère les opérations de Swapping entre la MC et la mémoire secondaire.

IV/ Différents modes de partage de la MC :

1/ Modes n'utilisant pas la mémoire virtuelle :

Les programmes sont chargés seulement dans la MC, parmi ces modes on peut citer :

a/ Zone unique contigüe : utilisé dans les systèmes séquentiels, où il existe un seul programme à la fois d'une façon à partager la MC entre ce programme et le système d'exploitation.



N.B : Ce schéma est utilisé dans les premiers mini-ordinateurs (IBM. DS/360)

b/ Schéma à partitions multiples : appliqué dans l'exécution de plusieurs programmes (multiprogrammation) → c'est de diviser l'espace en **n** partitions, on distingue deux stratégies :

➤ Partitions multiples fixes :

La MC est divisée en **n** partitions de tailles fixées à la génération du SE, ce partitionnement ne se change pas jusqu'à l'arrêt de la machine. Un algorithme d'allocation est chargé de chercher une taille suffisante au programme en attente.

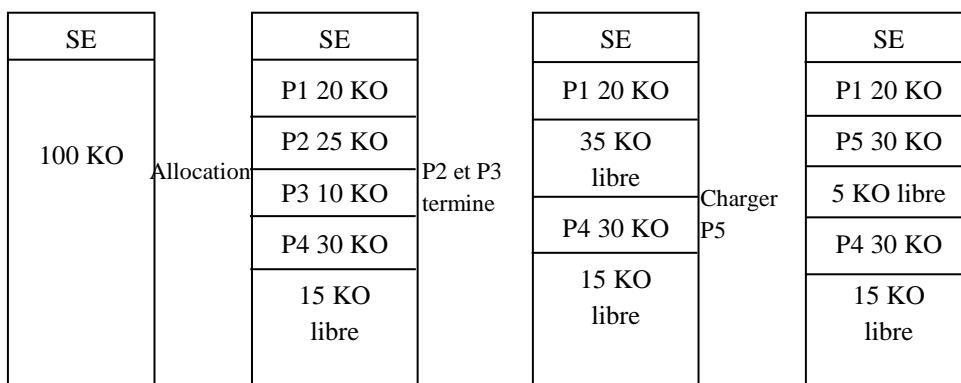
Inconvénients :

- Apparence de plusieurs fragments dans la mémoire inutilisable.
- Des partitions peuvent être libres alors que des programmes sont en attente → taille de programme > taille de la partition.
- Problème avec les programmes absolus qui nécessitent des adresses de chargement absolues → où on ne peut pas utiliser les partitions libres.

➤ Partitions multiples variables :

La mémoire est découpée dynamiquement selon la demande des programmes.

Exemple : partition 100KO pour programme utilisateur, P1 = 20, P2 = 25, P3 = 10, P4 = 30, P5 = 30



L'allocation dans cette stratégie est faite selon l'un des algorithmes suivants :

- **First Fit** : Le programme est placé dans la première partition trouvée \geq à celle du programme. Elle a comme inconvénient de laisser plusieurs fragments inutilisables.
- **Best Fit** : cette stratégie cherche de trouver la meilleure partition dont le résidu soit minimal (taille de partition – taille du programme \lll).
- **Worst Fit** : le programme est placé dans la partition la plus large, son objectif est de gagner une nouvelle partition pour un autre programme.

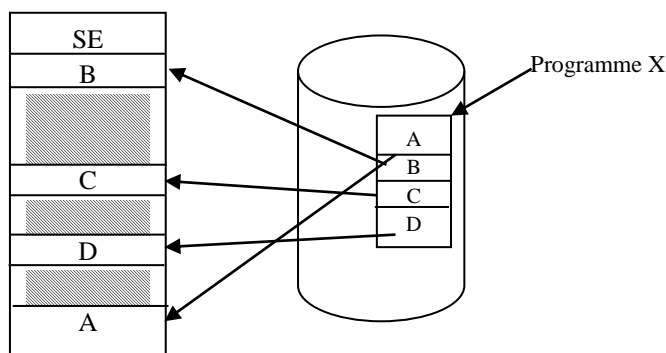
Compactage de MC : pour éviter les fragments inutilisables créés durant l'allocation des processus, un algorithme de compactage permet de rassembler les partitions libres en une seule partition avec une taille importante. Pour effectuer cette opération, il faut que les programmes à déplacer soient relogeables.

2/ Modes utilisant la mémoire virtuelle :

a/ La mémoire virtuelle :

Le principe de la mémoire virtuelle consiste à tendre l'espace physique de la MC à un espace mémoire plus grand, en utilisant une partie de la mémoire secondaire, exactement le disque dur. Les programmes sont dispersés entre la MC et la MS.

Les programmes sont chargés au début de lancement dans la MS, ensuite ils sont découpés en blocs et ils sont ramenés de la MS bloc par bloc vers la MC, le moment d'exécution de chaque bloc.



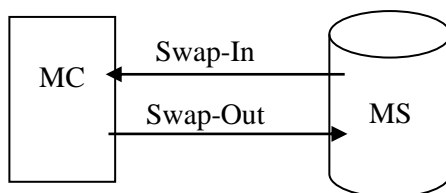
L'utilisation de la mémoire virtuelle a comme objectifs :

- Exécution des programmes lorsqu'ils sont partiellement en MC.
- Le chargement et l'exécution de programmes qui ont une plus grande taille par rapport à la MC.
- Une augmentation du degré de multiprogrammation.

b/ Le swapping (va-et-vient) :

C'est une opération utilisée dans le cas de la mémoire virtuelle, telque :

- Un programme est chargé dans sa totalité dans la MS ensuite il est transféré partie par partie dans la MC (Swap-In)
- Il s'exécute soit jusqu'à sa terminaison, ou son blocage.
- Dans le cas du blocage, le programme peut être transféré en MS (Swap-out)



Pour gérer l'espace de ces deux mémoires ainsi que pour faciliter les opérations de swapping, le système utilise l'une des techniques suivantes :

- La pagination
- La segmentation
- La segmentation paginée

c/ La pagination :

On parle de la pagination, lorsque :

- La mémoire centrale est organisée en blocs de tailles égales et fixes appelés **pages physiques** ou **cadres de pages**.
- Les processus sont divisés en blocs appelés **pages virtuelles** ayant la même taille que les cadres de pages.

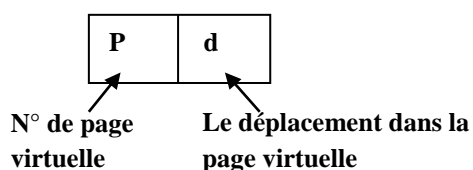
Alors les programmes sont mis dans la mémoire secondaire dans une zone swap dont on manipule un espace d'adressage virtuel. L'exécution d'un programme se fait en ramenant de la MS à la MC, page virtuelle par page virtuelle le moment de sa référence dans un cadre de page libre.

Où le processeur génère l'adresse de l'instruction suivante qui est une adresse virtuelle dont il faut référencer la page contenant cette adresse.

➤ Correspondance entre l'adresse virtuelle et l'adresse physique :

Lorsque la mémoire virtuelle est utilisée, l'adresse référencée soit logique, elle n'est pas mise directement dans le bus d'adresse, mais elle est envoyée à une unité physique appelée « Unité de gestion de mémoire MMU » qui a comme rôle de faire la correspondance entre cette adresse et son adresse réelle dans la MC, tel que :

- Le processeur génère une adresse virtuelle.
- Le MMU transforme cette adresse à un couple :



- Le gestionnaire de mémoire consulte **la table de pages** du programme en exécution pour savoir l'existence de cette page dans la MC.

- Si cette page n'existe pas dans la MC, il signale **un défaut de page**, et il la charge de la MS vers la MC.
- Il fait la correspondance entre le déplacement virtuelle et l'adresse physique de l'instruction dans la MC,
- Maintenant, cette adresse physique est transférée dans le bus d'adresses.

Table des pages : c'est une table contient toutes les informations nécessaires pour manipuler les pages, elle se trouve dans la MC incluse dans le **Bloc de Contrôle de Processus**, elle contient :

- Le numéro de page.
- L'adresse dans la MS \rightarrow S.
- Un bit indiquant son existence dans la MC \rightarrow R.
- L'adresse de cette page si elle est chargée en MC \rightarrow PR.
- Un bit indiquant si la page est modifiée ou non \rightarrow M.
- Un bit de protection.
- Un bit de référence.
- Gestion de la mémoire virtuelle dans la pagination :

1/ La recherche : le gestionnaire cherche l'existence de la page virtuelle pour la charger en MC soit dans la demande du processeur, soit par anticipation (prédiction d'utilisation).

2/ Le placement : à quel emplacement la page doit être placée en MC.

3/ Le déplacement : quelle page doit quitter la MC, si elle est saturée, pour charger une nouvelle page référencée, le gestionnaire cherche une page victime à remplacer selon une des stratégies :

- Stratégie FIFO \rightarrow la plus ancienne est remplacée.
- Stratégie LRU \rightarrow Least Recently Used (moins récemment utilisée).
- Algorithme optimal \rightarrow prédiction de la page qui ne sera pas référencée durant le plus grand temps).
- Stratégie NRU \rightarrow Not Recently Used dans un temps.
- Seconde Chance \rightarrow amélioration du FIFO, la première entrée + elle n'est jamais référencée.
- Avantages et inconvénients de la pagination :

Avantages :

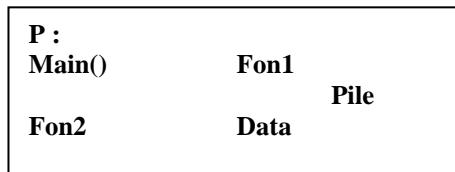
- Simplicité de chargement des pages virtuelles dans des pages physiques.
- Simplicité de correspondance entre l'adresse physique et l'adresse virtuelle.
- Le programmeur ne doit pas connaître la technique utilisée par le gestionnaire

Inconvénients :

- Augmentation de la taille de la table de page par rapport à la taille des programmes
- Le taux élevé des défauts de pages.
- Les données et les procédures ne sont pas séparées, ce qui complique leurs partages entre différents programmes.
- La protection des données et de procédures n'est pas faite d'une manière simple et séparée.

d/ La segmentation :

Une stratégie qui reproduit le découpage dans la mémoire tel qu'il est décrit par l'utilisateur et le compilateur (découpage logique)



Où :

- La mémoire centrale est organisée en blocs de tailles variables → **Segments physiques.**
- Chaque processus est découpé en unités logiques → Segments virtuelles (les variables globales, la pile d'appel des procédures, chaque procédure ou fonction, les variables locales, le programme principale).

N.B : les segments constituant un programme sont construits lors de la compilation.

e/ La segmentation paginée :

Cette technique a comme objectif de regrouper les avantages des deux techniques précédentes, tel que :

- Le programme est découpé, en premier lieu, en segments logiques avec des tailles variables, chaque segment est découpé en pages virtuelles de la même taille.
- La mémoire centrale est découpée en pages de la même taille.
- L'adresse virtuelle est un triple constitué du numéro de segment, numéro de page et le déplacement dans la page.

V/ La protection mémoire :

Tout système doit offrir un mécanisme de protection contre les accès non autorisés à des blocs de mémoire, suite à des erreurs de programmation ou autre. Les processus doivent être protégés entre eux contre leurs activités. Deux modes sont proposés :

1/ Protection du tout ou rien :

L'accès à une zone mémoire pour un processus donné est soit complètement autorisé, soit interdit, c'est-à-dire en sachant les limites début et fin de cette zone, on indique si ce qui existe entre ces limites soit protégé ou non.

2/ Protection spécifiée :

Permet de spécifier les opérations autorisées sur une zone mémoire, les opérations concernées sont :

- La lecture (Read).
- L'écriture (Write).
- L'exécution (Execute).

Où, on spécifie trois bits de protection RWE avec des valeurs différentes selon le niveau de protection (Exemple : 000 → accès interdit, 001 → exécution seule, ...)

CHAPITRE V : GESTION DE PROCESSUS ET DE PROCESSEUR

I/ Introduction :

Dans l'exécution des différents processus, il faut disposer de plusieurs ressources machine contribuant au long du cycle de vie d'un processus, la plus importante c'est la ressource qui effectue l'exécution elle-même, c'est le processeur. Alors, pour cette ressource, le système d'exploitation doit doter d'un ensemble de mécanismes gérant d'une manière efficace et optimale le travail du processeur et contrôlant l'exécution des processus.

II/ Le processus :

1/ Définition :

Un processus est un programme en cours d'exécution, il se trouve au niveau de la mémoire centrale. Le processus est créé le moment d'exécution et tué après sa fin.

2/ ce qui se passe après le lancement d'un programme :

Quand l'utilisateur lance l'exécution d'un programme, et quand la machine utilise le principe de la mémoire virtuelle et le multitâche, alors ce programme passe par les étapes suivantes :

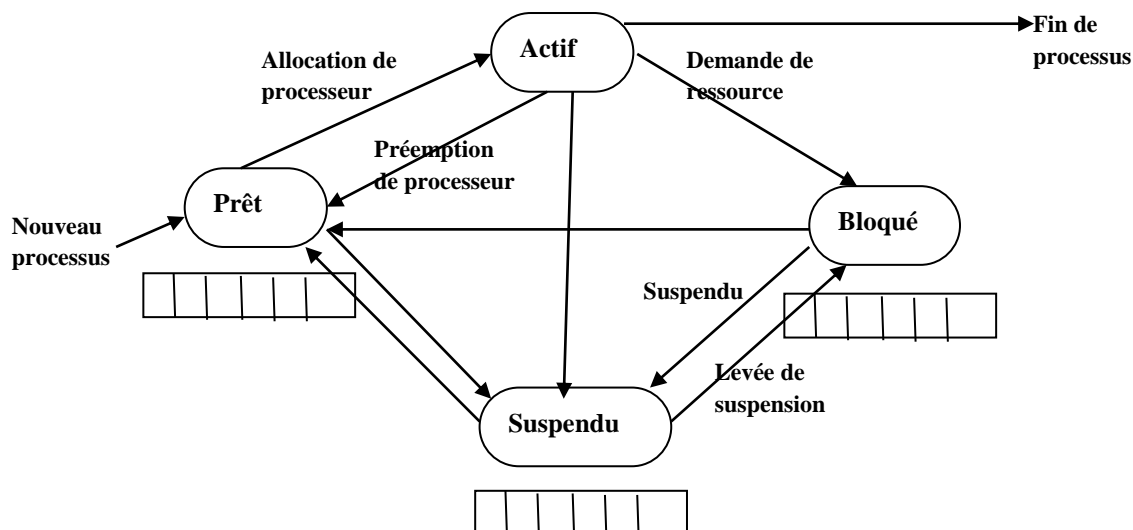
- Le système crée un **Job** au niveau de la mémoire virtuelle tout en chargeant le programme.
- Après, selon un critère ou autre ce job est admis dans le système tout en le chargeant vers la mémoire centrale dans sa totalité ou une partie, ce chargement crée ce qu'on appelle **un processus**.
- Le processus est dirigé, après, vers l'exécution dans le processeur selon des conditions spécifiques, où ce processus peut changer son état, durant cette exécution, plusieurs fois jusqu'à sa terminaison.

3/ Le changement d'état d'un processus :

Durant son exécution le processus peut être :

- Actif : (Elu) c'est-à-dire, le processus dispose de toutes les ressources nécessaires et il est en train de s'exécuter au niveau du processeur.
- Prêt : (Eligible) c'est-à-dire, le processus dispose de toutes les ressources nécessaires sauf le processeur → processus en attente du processeur.
- Bloqué : le processus est en attente d'une autre ressource que le processeur ou d'un évènement.
- Suspendu : le processus est arrêté temporairement par l'utilisateur ou le SE pour ne plus être en compétition pour le processeur.

Pour chacun des états prêt, bloqué et suspendu, on associe une file d'attente pour les processus dans cet état et elle est gérée par un outil du SE.



4/ Le descripteur d'un processus :

A la création d'un processus (idem pour le job), son image est créée dans la mémoire contenant toutes les informations nécessaires pour ce processus au long de son exécution, cette image représente aussi le processus au niveau des différentes files d'attente des différents états. Cette image est nommée descripteur ou bloc de contrôle de processus (job) PCB (JCB) qui contient :

Identité :	
Nom externe	nom interne
Priorité (dynamique, statique)	
Contexte : CO, PSW, RG, Acc, ...	
Etat	
Limites mémoires : Table des pages, les zones occupées, les adresses, ...	
Ressource : Types, quantité	
Informations de comptabilité	

Alors, les files d'attente utilisées dans chaque état contiennent les PCBs des processus, tel que :

- Le PCB d'un nouveau processus est mis dans la file d'attente des processus prêts.
- Attente jusqu'à sélection pour entrer dans le processeur → le processus devient actif, et on utilise un pointeur vers son PCB dans la mémoire.
- Si le processus attend une ressource ou un évènement, son PCB entre dans la file d'attente des processus bloqués.
- S'il est suspendu, son PCB se mis dans la file d'attente des processus suspendu.

5/ Les primitives de manipulation de processus :

La manipulation d'un processus se fait par des primitives nécessaires qui utilisent son PCB et se sont des procédures systèmes. Ces primitives d'exécutent d'une manière indivisible :

- La création d'un processus.
- Activation d'un processus prêt.
- Suspension d'un processus.
- Destruction d'un processus.

III/ Allocation de processeur à un processus :

Chaque fois, le processeur devient inactif, le système doit sélectionner un processus de la file d'attente des processus prêt et il le passe le contrôle, deux routines système sont concernées :

1/ Le dispatcher :

Il prend en charge l'allocation du processeur à un processus sélectionné au préalable, dont il assure :

- La commutation de contexte.
- Branchement vers la première instruction du processus pour l'exécution.

2/ Le scheduler :

Il s'occupe de la sélection du processus qui aura le prochain cycle d'exécution à partir de la file d'attente des processus prêts, c'est l'ordonnanceur des processus.

L'opération de scheduling est faite dans les situations suivantes :

- Lorsqu'un nouveau processus est créé, il faut décider s'il faut exécuter d'abord le processus parent ou le processus fils.
- Il faut prendre une décision d'ordonnancement lorsqu'un processus se termine → un autre processus doit être choisi dans le jeu de processus prêts.
- Lorsqu'un processus bloque sur des E/S, un sémaphore ou autre (à voir en 3LMD), un autre processus doit être sélectionné pour être exécuté.
- Lorsqu'une interruption d'E/S se produit, il faut également prendre une décision d'ordonnancement.
- Si l'horloge matérielle fournit des interruptions périodiques à une fréquence de 50 ou 60 Hz, par exemple, une décision d'ordonnancement peut être prise à chaque interruption d'horloge (système à temps partagé).

Il existe deux niveaux de scheduling :

- Scheduling des jobs pour décider quels processus à créer.
- Scheduling des processus pour décider quel processus prêt à exécuter.

➤ Le scheduling de job :

Il permet de :

- Suivre l'état du job.
- Allouer les ressources nécessaires.
- Récupérer les ressources après exécution.

➤ Le scheduling de processus :

Il permet de :

- Suivre l'état du processus en créant le PCB.
- Décide la politique d'allocation du processeur.
- Allouer le processeur.
- Reprendre le processeur du processus terminé ou interrompu.

3/ Les différentes politiques de scheduling :

➤ Le scheduling non préemptif :

Un algorithme d'ordonnancement non préemptif sélectionne un processus, puis le laisse s'exécuter jusqu'à ce qu'il bloque (soit sur une E/S, soit en attente d'un autre processus) ou qu'il libère volontairement le processeur. Même s'il s'exécute pendant des heures, il ne sera pas suspendu de force. En effet, aucune décision d'ordonnancement n'intervient pendant les interruptions d'horloge. Une fois le traitement de l'interruption est terminé, le processus qui était en cours d'exécution avant l'interruption est toujours relancé. On peut citer comme politiques :

- Politique FCFS : premier arrivé premier servi.
- Politique SJF : Short Job First → le job qui a le temps d'exécution le plus petit c'est le premier qui s'exécute.
- Politique par priorité : le processus avec la priorité la plus élevée soit le premier.

➤ Le scheduling préemptif :

Un algorithme de scheduling préemptif sélectionne un processus et le laisse s'exécuter pendant un délai déterminé. Si le processus est toujours en cours à l'issue de ce délai, il est suspendu (mis dans la file d'attente des processus prêts), et l'ordonnanceur sélectionne un autre processus à exécuter. L'ordonnancement préemptif nécessite une interruption à la fin du délai afin de redonner le contrôle de l'UC à l'ordonnanceur. On peut citer :

- Politique SRTF : Short Remaining Time First → le processus avec le temps restant le plus court d'abord.
- Politique tourniquet (Round Robin) : cette politique consiste à allouer le processeur suivant une durée limitée nommée **le quantum de temps**, où chaque processus entre pendant cette durée de temps et il sort et attend un autre tour d'exécution. Cette politique est réalisée à l'aide d'une file d'attente circulaire des processus prêts avec une gestion FIFO.
- Politique à plusieurs niveaux de queue : la file d'attente des processus prêts est subdivisée en plusieurs files suivant la classe des processus (Batch, interactifs, temps réel, ...), chacune est gérée indépendamment.

CHAPITRE VI : GESTION DES ENTRÉES/SORTIES

I/ Définition:

On appelle entrée/sortie (Input/Output), toute opération de transfert d'information entre le processeur, la mémoire et les organes externes.

- De l'extérieur → Entrée (Lecture)
- A l'extérieur → Sortie (Ecriture)

II/ Types d'E/S:

- E/S physique : gérée par le niveau le plus bas du système (matériels).
- E/S logique : est une instruction à l'intérieur d'un programme qui appelle une procédure d'E/S (Exemple : Scanf, Printf).
- E/S virtuelle : entre le processeur, la mémoire centrale et la mémoire virtuelle.
- E/S spoolée : liée à un périphérique lent, l'accès se fait par l'intermédiaire d'un fichier disque nommé « Spool »

III/ Pilote (Driver/ gestionnaire):

Le pilote est un programme spécifique, qui permet de lancer un transfert physique et contrôle le déroulement de l'opération tout en prenant en charge les erreurs.

Exemple : le pilote de l'imprimante permet de :

- S'assurer que l'imprimante est branchée.
- S'assurer qu'elle n'est pas occupée.
- S'assurer qu'il y a du papier.
- Engager la première feuille.
- Remplir le tampon par les caractères à imprimer.
- Lancer l'impression et passe à l'ensemble de caractères suivant.

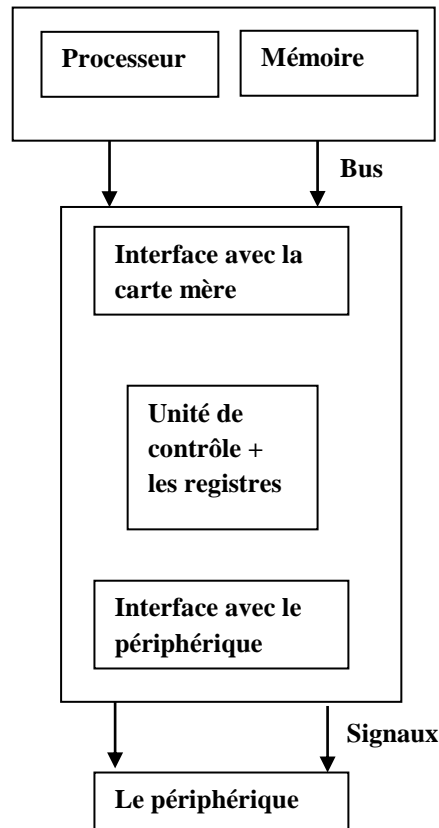
IV/ Les interfaces d'E/S:

L'interface est un dispositif qui permet de relier les périphériques à l'ordinateur, c'est-à-dire d'adapter les caractéristiques de chacune des deux parties carte mère et périphérique, ces caractéristiques sont :

- Type de liaison : série ou parallèle.
- Mode de transmission : synchrone ou asynchrone.
- Vitesse de transmission : rapide ou lent.

V/ Les contrôleurs de périphériques :

Servent à piloter les périphériques, chaque périphérique est lié à un bus par un contrôleur, le rôle du contrôleur est d'adapter la diversité des périphériques à une interface commune. C'est une carte électronique que l'on insère à la carte mère.



VI/ Mode de pilotage d'une entrée/sortie physique :

Le pilotage d'une unité périphérique par le processeur nécessite une synchronisation entre les actions processeurs et cette unité pilotée. Plusieurs techniques de pilotage des E/S sont utilisées :

1. E/S physique directe :

C'est de contrôler le déroulement des E/S physiques par le processeur lui-même, où on a :

- *Mode synchrone :*

Le processeur commande le périphérique depuis le début de l'E/S jusqu'à la fin. Pour démarrer une E/S le processeur commence par initialiser les registres appropriés du contrôleur par les informations nécessaires, puis il émet une commande au contrôleur pour qu'il lance l'opération d'E/S. celui-ci examine à son tour le contenu de ces registres pour déterminer l'action à effectuer et lance le transfert.

Durant le transfert physique du caractère, le processeur reste mobilisé à suivre l'opération d'E/S en consultant un indicateur spécifique au périphérique désignant son état dont il détecte la fin de transfert. (Appelé aussi attente active).

L'inconvénient majeur c'est la perte de temps CPU.

- *Mode scrutation par intervalle de temps :*

Dans ce mode, le processeur lance l'opération d'E/S puis il retourne à l'exécution des processus utilisateur, à des intervalles réguliers brefs, le processeur est interrompu pour vérifier si cette opération est achevée.

Même inconvénient de perte de temps due au traitement des interruptions fréquentes de l'horloge.

- *Mode asynchrone :*

Le processeur est libéré du contrôle de fin de transfert où le contrôleur est muni d'une ligne d'interruptions qui sera utilisée pour informer le processeur. Après la réalisation de transfert, le contrôleur génère une interruption qui informe de la fin de transfert dont il faut exécuter une routine d'interruption.

L'inconvénient de ce mode est la perte de temps pour exécuter ces routines d'interruption.

2. E/S physique indirecte :

Cette technique est proposée pour améliorer les performances du processeur, elle consiste à le libérer totalement du suivi des E/S jusqu'à ce que leur déroulement se termine. On utilise un dispositif indépendant dit canal.

- *Mode canal :*

Un canal est un processeur programmable qui peut exécuter une suite d'opérations d'E/S, il exécute un programme canal résidant en MC.

Le processeur initialise une opération d'E/S en envoyant une instruction spéciale au canal, et en lui indiquant où trouver le programme canal qui lance l'E/S. le canal exécute alors toutes les commandes de ce programme indépendamment du processeur. A la fin d'exécution le canal envoie une interruption au processeur pour l'avertir de la fin d'E/S.

- *L'accès direct mémoire (DMA) :*

Le DMA est un canal simplifié utilisé dans les ordinateurs de petite taille, il peut être connecté entre le contrôleur et le bus mémoire, permettant ainsi aux périphériques d'accéder à la mémoire sans passer par le CPU. Si le DMA et le CPU demandent simultanément un accès à la mémoire, le DMA passe en priorité.

Il se compose de :

- Un registre d'adresse mémoire (contient le début de bloc où se trouvent les informations dans la mémoire).
- Un registre de commande RC (indique le sens de transfert lecture ou écriture).
- Un registre compteur de caractères RCC (nombre de caractère à lire ou écrire).
- Le registre RAP (l'adresse du périphérique concerné).

Il fonctionne de la manière suivante :

- Le processeur initialise le DMA en envoyant dans le registre d'adresse l'adresse de début du bloc dans la mémoire, le nombre de caractères dans le RCC, fournit l'adresse du périphérique concerné dans le RAP et le code de l'opération à exécuter dans le registre RC.
- Le DMA remplace le processeur dans le transfert.
- Après la fin de l'opération, il transfère une interruption au processeur.

Exemple :

Expliquer en détails le principe de travail du contrôleur DMA pour transférer **500** caractères à un périphérique **écran** dont le numéro est **4**. Ces caractères se trouvent à partir de l'adresse **1400** dans la mémoire centrale.

- Le registre d'adresse = 1400.
- Le registre RCC = 500.
- Le registre RC = écriture.
- Le registre RAP = 4.

Le DMA lire le premier caractère de la mémoire centrale à partir de l'adresse 1400, le transfert dans le tampon de l'écran, décrémente le compteur et incrémente l'adresse pour passer au caractère suivant.