

10. Amélioration des performances - organisation de la mémoire, mémoire virtuelle. Cache - principes et algorithmes de gestion de l'accès

10.1. Amélioration des performances - organisation de la mémoire, mémoire virtuelle.

10.1.1. Amélioration des performances - organisation de la mémoire

➤ Organisation de la mémoire

La cellule (la plus petite partie adressable d'une mémoire) peut être:

- le bit, on a alors une machine à bits;
- le caractère (octet, byte), on a alors une machine à caractères;
- le mot, on a alors une machine à mots.

Il existe plusieurs façons différentes d'organiser une mémoire.

A titre d'exemple, une mémoire formée de 96 bits peut avoir les trois organisations suivantes (en termes d'adressage):

A - 6 mots de 16 bits

B - 8 mots de 12 bits

C - 12 mots de 8 bits,

impliquant respectivement 6, 8 et 12 adresses.

A : Mémoire de 96 bits organisée en 6 mots de 16 bits

<u>mots</u>	<u>bits</u>
0	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
1	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
2	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
3	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
4	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
5	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

B : Mémoire de 96 bits organisée en 8 mots de 12 bits

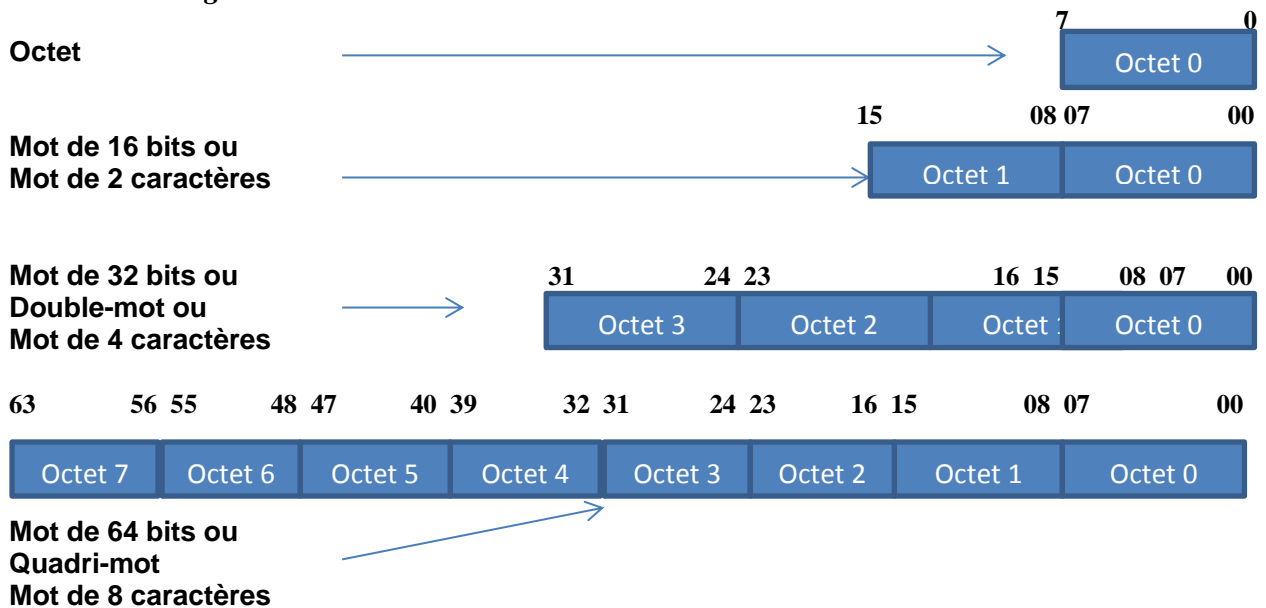
<u>mots</u>	<u>bits</u>
0	11 10 9 8 7 6 5 4 3 2 1 0
1	11 10 9 8 7 6 5 4 3 2 1 0
2	11 10 9 8 7 6 5 4 3 2 1 0
3	11 10 9 8 7 6 5 4 3 2 1 0
4	11 10 9 8 7 6 5 4 3 2 1 0
5	11 10 9 8 7 6 5 4 3 2 1 0
6	11 10 9 8 7 6 5 4 3 2 1 0
7	11 10 9 8 7 6 5 4 3 2 1 0

C : Mémoire de 96 bits organisée en 12 mots de 8 bits

<u>mots</u>	<u>bits</u>
0	7 6 5 4 3 2 1 0
1	7 6 5 4 3 2 1 0
2	7 6 5 4 3 2 1 0
3	7 6 5 4 3 2 1 0
4	7 6 5 4 3 2 1 0
5	7 6 5 4 3 2 1 0
6	7 6 5 4 3 2 1 0
7	7 6 5 4 3 2 1 0
8	7 6 5 4 3 2 1 0
9	7 6 5 4 3 2 1 0
10	7 6 5 4 3 2 1 0
11	7 6 5 4 3 2 1 0

Lorsque la cellule mémoire comporte plus d'un octet, elle est toujours adressée par rapport à l'octet de plus faible poids. Les octets sont généralement numérotés comme les bits, de droite à gauche.

➤ Adressage des octets



Par contre, le choix d'une organisation de la mémoire a des répercussions sur la performance. Voyons, par exemple, comment une chaîne de caractères, disons:

IF X > 0 THEN GO TO 123;

peut être emmagasinée dans la mémoire.

Si la machine possède une mémoire à 4 octets par mot, alors on trouvera en général 4 caractères par cellule, soit de gauche à droite ou de droite à gauche selon les machines.

Pour tout changement dans la chaîne de caractères demandé par l'utilisateur, le programme éditeur devra faire les déplacements nécessaires à l'insertion de nouveaux caractères, ce qui demande un certain temps.

On peut diminuer ce temps de manipulation si on utilise une autre organisation de la mémoire. Dans cette nouvelle organisation, chaque cellule contient un seul caractère (au lieu de 4), et on y insère un pointeur qui indique où se trouve le prochain caractère.

On remarque cependant que la performance ainsi gagnée se paie par le fait qu'un espace plus grand de mémoire est requis pour mémoriser le même nombre de caractères (dû à la redondance des pointeurs).

Lorsqu'on modifie la chaîne originale pour
READ B,
à la chaîne modifiée par l'éditeur, on reçoit:
READ A, B.

10.1.2. La mémoire virtuelle

La mémoire primaire de l'ordinateur représente la mémoire physique de tous les processus actifs, mais elle est insuffisante pour leur chargement total. Ainsi cette mémoire physique s'élargit avec la mémoire libre des dispositifs à disques. Pendant ce processus, seulement la partie active des programmes, exécutés dans un moment donné par le processeur central, est laissée dans la mémoire physique. La coordination se réalise par le système opérationnel de l'ordinateur. Ce mécanisme est appelé *mémoire virtuelle* et on l'applique toujours en multiprogrammation avec la division de temps. Cette notion - *mémoire virtuelle* – représente une telle organisation de la mémoire d'ordinateur (de système), qui permet l'indépendance des applications d'utilisateur par rapport à l'adressage physique des instructions et des données dans la mémoire de l'ordinateur. Cela se réalise en utilisant deux méthodes pour la gestion d'accès dans cet énorme espace adressable: gestion de l'accès en segments de la mémoire et gestion de l'accès en pages de la mémoire.

10.2. Cache - principes et algorithmes de gestion de l'accès

10.2.1. Cache – principes

La mémoire de cache est dédiée de sauvegarder transitoirement les données, les instructions, les adresses dans le processeur central. Elle est d'une très haute rapidité - c'est la mémoire avec la plus haute rapidité dans toute la mémoire de système de l'ordinateur. Technologiquement elle se réalise comme mémoire de registre avec adressage direct du type DRAM. Cette mémoire de cache stocke l'information nécessaire pour le fonctionnement du processeur central de plusieurs pas successifs de l'algorithme de calcul dans tout moment de fonctionnement de l'ordinateur. Logiquement la place de la mémoire de cache est entre le processeur central et la plus lente d'elle mémoire opérationnelle. La capacitance de la mémoire de cache ne passe pas quelques pourcentages de celle de la mémoire opérationnelle. Lorsque le processeur central cherche l'instruction (la commande) ou l'opérande, il vérifie premièrement la mémoire de cache (si elle existe). En présence de l'information cherchée dans la mémoire de cache, cette information s'extrait maximum vite pour l'ordinateur donné – avec la fréquence de mesure du processeur central. Si cette information est absente dans la mémoire de cache, elle s'extrait de la mémoire opérationnelle et parallèlement est enregistrée (copiée) dans la mémoire de cache.

L'effectivité de la conception d'une telle mémoire de cache est assurée par deux principes, établis statistiquement :

- ✓ *Proximité d'espace*, permettant l'usage des données, situées très proche pour les moments successifs du calcul ;
- ✓ *Proximité de temps*, permettant dans ces moments successifs du calcul de traiter les mêmes données - par exemple pour des cycles, pour les algorithmes récursifs, etc.

10.2.2. Algorithmes de gestion de l'accès

Dans l'organisation de la mémoire de cache on applique deux méthodes – deux algorithmes de gestion de l'accès:

➤ **Reflét direct (Direct Mapped)**: Chez lui toute zone de la mémoire opérationnelle possède une place correspondante dans la mémoire de cache, qui permet aux données cherchées d'une zone d'être vite décrites. Le défaut, lorsqu'il y a un grand nombre d'opérations de lire/écrire dans l'espace limité de la mémoire de cache, est un possible comblement de cette mémoire.

➤ **Organisation associative de la mémoire de cache**: Chez elle la recherche des données se réalise d'une partie de ces données, sur la base de la comparaison parallèle entre un indice associatif donné et les champs synonymes de mots dans le milieu de stockage. Cette méthode permet d'éviter le comblement, mais il est plus difficile de découvrir l'information dans plus d'une zone de la mémoire opérationnelle. Cette méthode est plus effective.