

## 8 – PAGINATION MÉMOIRE

### 1. PAGINATION

Le principe de la pagination réside dans la division de la mémoire en zones de tailles fixes appelées "pages". L'espace de travail d'un processus est divisé en pages. Quand le processus est exécuté, seules les pages dont il a besoin sont chargées en mémoire centrale.

On dispose d'une machine monoprocesseur, ayant une mémoire centrale de 32K mots, une page faisant 512 mots (64 pages en mémoire centrale), un bloc (unité de stockage sur le disque) faisant 512 mots. On considère le programme suivant :

Adresse	Instruction	Signification
E	Loadx N	reg d'index $X = N$
	Load S, X	reg accumulateur $A = S[X]$
	Add V	$A = A + V$
	Store T, X	$T[X] = A$
	Subx K	$X = X - K$
	Brxpz E + 1	Si $X \geq 0$ alors aller en E + 1
	Stop	

Où:

- X est un registre d'index,
- A est un registre accumulateur,
- E est l'adresse du début du programme :  $512 * 1 + 508$ ,
- S et T sont des tableaux de taille [0 .. N], S [0] étant à l'adresse  $512 * 11 + 168$  et T[0] se trouvant à l'adresse  $512 * 12 + 456$ ,
- N est une constante égale à 799 se trouvant à l'adresse  $512 * 6 + 500$ ,
- V est une constante égale à 2 se trouvant à l'adresse  $512 * 8 + 100$ ,
- K est une constante égale à 2 se trouvant à l'adresse  $512 * 8 + 10$ .

Ce programme effectue la chose suivante :

```
x = N;
do {
    T[X] = S[X] + V;
    x -= K;
} while (x >= 0);
```

En fin d'exécution, les pages modifiées sont recopiées en mémoire secondaire.

#### 1.1.

Sachant que chaque instruction, constante ou variable simple occupe un mot mémoire, représenter l'espace d'adressage du processus (numéro de page / déplacement dans la page de chaque donnée). Quels droits doit-on affecter aux différentes pages du processus ?

Les actions de gestion de la mémoire peuvent être représentées à l'aide des opérations suivantes :

- Trap (p) : Défaut de page pour la page p.  
 Charg (p,c) : Chargement de la page p dans la case c en mémoire centrale.  
 Dech (p,c) : Déchargement de la page p se trouvant à la place c en mémoire centrale.  
 Mod (p,c,d) : Modification de la table des pages, p : numéro de page du processus, c place en mémoire centrale, d droits d'accès de la page.

Le processus précédent dispose pour s'exécuter des pages de mémoire centrale 17, 21, 22, 23, 37 et 42.

---

**1.2.**

En utilisant un algorithme premier chargé/premier déchargé et en supposant qu'au départ, aucune page n'est chargée, décrire les actions de gestion de mémoire sous la forme d'une suite composée des opérations précédentes. Donner la table des pages finale.

---

**2. SEGMENTATION PAGINÉE**

On considère une mémoire *segmentée paginée*. La taille des pages est de 512 mots.

Le processus P possède 3 segments : le segment 0 pour le code, le segment 1 pour la pile et le segment 2 pour les données. Le segment 0 a 1500 mots, le segment 1 en a 2000 et le segment 2 en a 3000. On suppose que la table des segments et les tables de pages sont déjà chargées en mémoire.

---

**2.1.**

Quels sont les droits associés à chaque segment ? Quel est l'avantage de découper ainsi l'espace d'adressage du processus ?

Seules les pages suivantes ont été chargées (on donne le doublet <numéro de segment, numéro de page> ) :

- <0, 0> à l'adresse physique 2560
- <1, 2> à l'adresse physique 4096
- <2, 3> à l'adresse physique 1024.

---

**2.2.**

Dans quelle case mémoire est chargée la page <0, 0> ?

---

**2.3.**

Décrire brièvement ce qui se passe sur une tentative de lecture en <0 : 1678>, <1 : 567> et <2 : 1600> (Ces adresses sont au format segmenté).

---

**2.4.**

Connaît-on l'adresse physique de la variable d'adresse virtuelle <1 : 1060> ? Si oui, quelle est-elle ? Sinon, justifiez.

---

**3. CACHE ET MÉMOIRE VIRTUELLE**

Avec une mémoire paginée, l'accès à une donnée nécessite deux accès à la mémoire : un pour lire la table des pages, l'autre pour lire la donnée. Pour ne pas diviser par deux les performances du système, on utilise un cache spécial, appelé TLB (Translation Look-aside

Buffers), qui permet de stocker le couple  $\langle n^\circ \text{ de page, case mémoire} \rangle$ . Si le couple correspondant à la page accédée est présent dans la TLB, alors l'accès à la table des pages devient inutile. Typiquement, la TLB contient entre 8 et 2048 entrées.

---

**3.1.**

On suppose qu'un accès à la TLB demande 20 ns et qu'un accès mémoire demande 100 ns. Pour une TLB avec un taux de hit de 90%, quel est le temps moyen d'accès à une donnée ?

En plus de la TLB qui permet une translation rapide de l'adresse logique à l'adresse physique, le système dispose d'un cache pour améliorer la vitesse d'accès aux données. Le cache est une mémoire à accès rapide, structurée en lignes. Chaque ligne stocke un bloc de données d'une part, tout ou partie de l'adresse du bloc permettant d'identifier le bloc de façon unique, ainsi que diverses informations relatives à la validité, l'ancienneté, ... dudit bloc.

La combinaison de la mémoire cache et de la mémoire virtuelle est délicate : ou bien le cache contient les adresses physiques des blocs, et dans ces conditions la traduction adresse virtuelle - adresse physique doit être faite entre le processeur et le cache; ou bien le cache contient les adresses virtuelles des blocs, et la traduction adresse virtuelle-adresse physique a lieu entre le cache et la mémoire.

---

**3.2.**

Représentez schématiquement l'organisation de la hiérarchie mémoire dans les deux cas. Quels sont les avantages et inconvénients de chaque organisation ?