

# ED 2

## Gestion de la mémoire

### Exercice 1 : Compréhension du mécanisme de mémoire virtuelle

L'espace d'adressage logique (virtuel) d'un programme en cours d'exécution est une suite de pages. Soit le programme suivant :

Adresse de début 1020

Début :

```
I1 : Charger le mot d'adresse 6144 dans R0
I2 : Empiler R0
I3 : Appeler une procedure située à l'adresse 5120 et
    placer l'adresse de retour dans la pile
I4 : Décrémenter le pointeur de pile 16
I5 : Comparer la valeur en sommet de pile à la constante 4
I6 : Effectuer un branchement à l'adresse 5652 si égalité
```

Fin

La taille des pages est de 512 octets, chaque instruction est codée sur 4 octets.

La pile est située à l'adresse 8192 et progresse vers les adresses décroissantes.

#### Question

*Donner la liste ordonnée des pages utilisées par ce programme.*

### Exercice 2 : Adressage dans une mémoire virtuelle paginée

Soit le programme suivant:

```
Program Machin;
chaine : array[0..1999] of Byte;
i : integer; -- i sur 2 octets
begin
  For i:= 0 to 1999 Do chaine[i]:= '*';
end ;
```

Les instructions de ce programme tiennent sur 20 octets. Ce programme est exécuté sur une machine à mémoire paginée (taille de la mémoire réelle = 1 MégaOctets et taille d'une page = 512 octets). Les instructions à référence mémoire ont un champ adresse de 24 bits.

#### Question

*a) Donner la taille de l'espace adressable, le nombre de bits du déplacement, le nombre de bits d'une adresse virtuelle, le nombre de bits d'une adresse réelle, le nombre de bits du numéro de page virtuelle, le nombre de bits du numéro de page réelle, le nombre d'entrées de la table des pages.*

b) *Le chargement de ce programme en mémoire engendre-t-il une fragmentation interne? Si non, pourquoi ? Si oui, la chiffrer.*

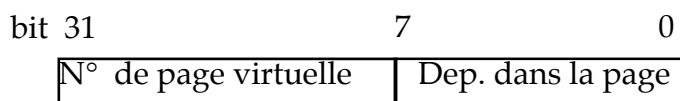
Les pages de ce programme (numérotées 0,1,2 et 3) sont respectivement implantées en mémoire dans les pages réelles 9, 17, 65 et 33.

c) *Donner l'adresse virtuelle et l'adresse réelle correspondante [en décimal, sous la forme (numéro de page, déplacement)] générées lors de l'exécution de chacune des instructions suivantes:*

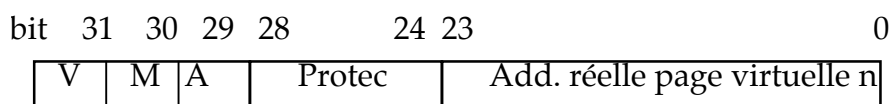
chaine[10]:= '\*'; chaine[515]:= '\*'; chaine[1026]:= '\*'; chaine[1999]:= '\*'

### Exercice 3 : Mécanisme de gestion d'une mémoire virtuelle paginée

Dans un ordinateur, dont la mémoire est gérée selon le principe d'une mémoire virtuelle paginée, une adresse a la structure suivante :



A chaque processus est associée une table des pages dont la n-ième entrée a la structure suivante :



avec :

V : bit de validité = 1 si la page est résidente ; = 0 sinon

M : bit de modification = 1 si la page a été modifiée depuis son chargement ; = 0 sinon

A : bit de dernier accès ; mis à 1 à chaque accès à une page. Remis à 0 périodiquement par un processus système

Protec : mode de protection d'accès à la page.

#### Question 1

*Expliquer l'utilité de ces différents champs.*

## Question 2

Décrire sous forme algorithmique les opérations réalisées par le matériel lors du décodage d'une adresse. (Faire abstraction de la gestion de la protection).

## Question 3

Quelle est a priori la taille de la table des pages d'un processus ? Rappeler les techniques qui sont mises en oeuvre pour réduire cette taille.

## Question 4

Rappeler le fonctionnement des algorithmes de remplacement de pages FIFO ("First In-First Out"), c'est à dire à l'ancienneté de chargement, et LRU ("Least Recently Used"), c'est à dire à l'ancienneté de référence.

Soit la liste des pages virtuelles référencées aux instants  $t = 1, 2, \dots, 11$  :

3, 5, 6, 8, 3, 9, 6, 12, 3, 6, 10

Sachant qu'il n'y a que 4 places en mémoire centrale, compléter les deux tableaux suivants représentant les pages présentes en mémoire centrale pour chaque politique.

Places en mémoire centrale	Pages virtuelles résidentes										
	1	2	3	4	5	6	7	8	9	10	11
1	3	3	3	3	3						
2		5	5	5	5						
3			6	6	6						
4				8	8						

### Politique FIFO

Places en mémoire centrale	Pages virtuelles résidentes										
	1	2	3	4	5	6	7	8	9	10	11
1	3	3	3	3	3						
2		5	5	5	5						
3			6	6	6						
4				8	8						

### Politique LRU

*Comparer le nombre de défauts de pages provoqués , dans l'exemple, par les deux politiques.*

*Commenter le résultat trouvé.*

## **Exercice 4 : Influence de la mémoire virtuelle sur la performance des programmes**

On considère une mémoire virtuelle paginée dont les pages ont une taille de 2Ko et les mots sont de 32 bits.

A ; array(1..512,1..512) of integer ;

--un tableau est stocké ligne par ligne en mémoire

Soient les deux programmes suivants :

Cas 1

For j=1..512

    For i=1..512

        A(i, j)=0

Cas2

For i=1..512

    For j=1..512

        A(i, j)=0

### **Question**

*On suppose qu'aucune page n'est préalablement chargée en mémoire (allocation à la demande), comparer le nombre de défauts de pages dans les deux cas.*

## **Exercice 5**

On souhaiterait rajouter une instruction vectorielle à un processeur classique muni de registres 32 bits, d'un chemin de données de 32 bits, d'une mémoire adressable par 32 bits d'adresse, d'un gestionnaire de mémoire virtuelle fournissant des pages à la demande pour chaque défaut rencontré.

On considère des pages de 4Ko et des vecteurs de nombres flottants simple précision(4 octets) inférieurs à 1024 éléments.

### **Question**

*On considère l'instruction d'addition vectorielle*

$V1 \leftarrow V1 + V2$   $V1, V2$  vecteur de taille n

*Identifier le problème crucial à résoudre pour implanter cette instruction et proposer une solution pour contourner cette difficulté.*