

## EXERCICES- Gestion Mémoire

### 1. Gestion mémoire partagée

Comment implanter le partage de code (code réentrant) dans un système qui gère la mémoire par pagination ?

*On fait pointer les tables des pages de chaque processus vers les mêmes pages contenant le code réentrant. Ca permet d'éviter de faire des copies de pages.*

### 2. Pagination

#### 2.1

Soit la table de pages suivante :

0	4
1	6
2	8
3	9
4	12
5	1

Sachant que les pages virtuelles et physiques font 1K octets, quelle est l'adresse mémoire correspondant à chacune des adresses virtuelles suivantes codées en hexadécimal : 142A et 0AF1

0	1	2	3	4	5	6	7	8
0	1024	2048	3072	4096	5120	6144	7168	8192

$142A_{hex} = 5162_{dec}$ . 5162 correspond à la page virtuelle 5 donc à la page physique 1 donc à l'adresse 1024.

$0AF1_{hex} = 2801_{dec}$ . 2801 correspond à la page virtuelle 2 donc à la page physique 8 donc à l'adresse 8192.

## 2.2

Un programme a besoin de 5 pages virtuelles numérotées 0 à 4. Au cours de son déroulement, il utilise les pages dans l'ordre suivant : 012301401234

Questions :

a. S'il reste 3 pages libre en mémoire, indiquer la séquence des défauts de page, sachant que l'algorithme de remplacement est FIFO,

*Voici la suite des pages contenues dans le cache à chaque utilisation d'une nouvelle page :*

```
- 0 0 0 1 2 3 0 0 0 1 4 4
- - 1 1 2 3 0 1 1 1 4 2 2
- - - 2 3 0 1 4 4 4 2 3 3
  d d d d d d d - - d d d
```

b. Même question avec 4 pages disponibles en mémoire. Observation?

*Voici la suite des pages contenues dans le cache à chaque utilisation d'une nouvelle page :*

```
- 0 0 0 0 0 0 1 2 3 4 0 1
- - 1 1 1 1 1 2 3 4 0 1 2
- - - 2 2 2 2 3 4 0 1 2 3
- - - - 3 3 3 4 0 1 2 3 4
  d d d d - - d d d d d d
```

*Sur cette séquence, il y a autant de défaut de pages avec 3 pages disponibles en mémoire qu'avec 4 pages disponibles en mémoire. Rajouter une page en mémoire n'a donc pas apporté d'optimisation. Il faudrait peut-être rajouter plus de pages en mémoire mais pas trop non plus sinon le cache n'est plus une petite mémoire rapide.*