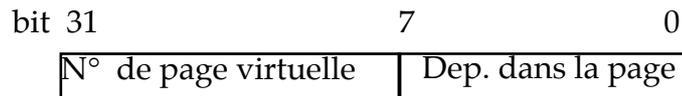


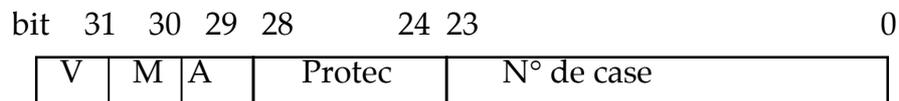
# ED 2 Systèmes

## Exercice 1 : 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).*

#### Application

La table des pages d'un processus est résidente en mémoire centrale.

Un processus référence l'adresse virtuelle 10/3000

l'entrée 10 de la table des pages indique V=1, N°de case=20

*Donner l'adresse physique correspondante.*

Le processus référence maintenant l'adresse virtuelle 15/2500

l'entrée 15 de la table des pages indique V=0, N°de case=12.

*Peut-on en déduire immédiatement l'adresse physique ? Sinon que doit faire le système ?*

### Question 3

Quelle est a priori la taille de la table des pages d'un processus ? Quelles les techniques peuvent être mises en oeuvre pour réduire cette taille ?

## Exercice 2 : Gestion de fichiers UNIX

Un processus Unix veut lire séquentiellement un fichier *Toto* de 8 Mo stocké sur disque, à raison de 256 octets à la fois. Pour localiser un fichier sur disque, le système dispose des tables suivantes :

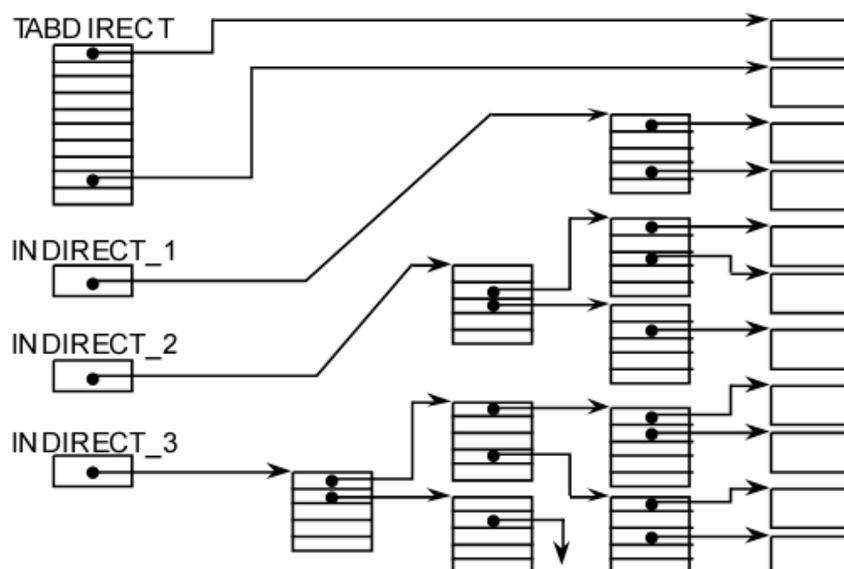
- une table, que nous appelons TABDIRECT, donnant les numéros de blocs pour les 10 premiers blocs de l'objet externe,
- un numéro de bloc, que nous appelons INDIRECT\_1, qui contient les numéros des  $p$  blocs suivants de l'objet externe,
- un numéro de bloc, que nous appelons INDIRECT\_2, qui contient les  $p$  numéros de blocs contenant les numéros des  $p^2$  blocs suivants de l'objet externe,
- un numéro de bloc, que nous appelons INDIRECT\_3, qui contient les  $p$  numéros de blocs contenant, au total,  $p^2$  numéros de blocs contenant les numéros des  $p^3$  blocs suivants de l'objet externe.

Avec des blocs de 1024 octets et un numéro de bloc de 4 octets, le paramètre  $p$  est égal à 256. Par ailleurs.

Le processus qui veut lire le fichier *Toto* fera donc 32768 demandes de lecture successives.

On suppose qu'il n'y a qu'un seul processus dans le système, que le temps d'accès moyen au disque est de 40 ms, et que le système n'utilise pas de tampons de bloc disque, ce qui implique que chaque fois qu'une information située dans un bloc disque est nécessaire, ce bloc doit être lu depuis le disque.

Évidemment le descripteur d'un fichier est ouvert, c'est-à-dire les informations TABDIRECT, INDIRECT\_1, INDIRECT\_2 et INDIRECT\_3, restent en mémoire centrale.



- 1- Décrire ce qui se passe lors des deux premières demandes de lecture de 256 octets, puis lors de la 5<sup>ième</sup> demande.*
- 2- Décrire ce qui se passe lors des 41<sup>ième</sup> et 45<sup>ième</sup> demandes de lecture de 256 octets.*
- 3- Décrire ce qui se passe lors des 1065<sup>ième</sup> et 1066<sup>ième</sup> demandes de lecture de 256 octets.*
- 4- Décrire ce qui se passe lors des 2089<sup>ième</sup> et 2090<sup>ième</sup> demandes de lecture de 256 octets.*
- 5- En déduire le nombre total d'accès disque nécessaires et le temps d'attente d'entrées-sorties.*