

## Adresse virtuelle

1

Un ordinateur possède un système de mémoire virtuelle avec pagination qui utilise des pages de 4 KB. Son espace de mémoire virtuelle est de  $2^{32}$  bytes et la mémoire physique a une taille de  $2^{18}$  bytes (256 KB). Répondre aux questions suivantes :

- Quel est le format de l'adresse virtuelle? Indiquer chaque champ et le nombre de bits de chacun.
- Combien de cadres de page (frames) y-a-t-il dans la mémoire physique?
- Quel est le nombre maximum d'entrées dans la table de pages?

2

Supposons une machine avec 2 GBytes de mémoire physique et une plage d'adressage virtuelle de 32 bits avec 1024 pages. Répondre aux questions suivantes :

- Quelle est la taille d'une page?
- Combien de cadres de page y a-t-il dans la mémoire physique?
- Supposons deux pages A et B d'un processus qui sont affectées par la table de pages vers le même cadre de page en mémoire physique. Si la séquence d'accès par le processus est A, B, A, quelle sera la valeur du valid-bit de la page B à la fin de cette séquence? Justifiez.

3

Supposons une machine avec 32 KBytes de mémoire, une plage d'adressage virtuelle de 16 bits et 4KBytes de taille de page.

- Dans l'adresse virtuelle, combien de bits faut-il utiliser pour le numéro de page virtuelle? Quelle est la taille maximale de mémoire qui peut être gérée? Justifiez.
- Traduisez l'adresse virtuelle suivante en adresse réelle :

1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1

en utilisant la table de pages suivante :

15	000	0
14	000	0
13	000	0
12	000	0
11	111	0
10	000	0
9	101	1
8	000	0
7	000	0
6	000	0
5	011	1
4	100	1
3	000	1
2	110	1
1	001	1
0	010	1

- Une fois l'adresse trouvée en mémoire physique, faudra-t-il remplacer la page? Pourquoi?

## Structure d'un processus dans la mémoire<sup>1</sup>

La mémoire d'un processus est structurée en termes de segments de code, de donnée, de tas et de pile. À tout segment est associé un ensemble de droits du type rwx (comme pour les fichiers) et une indication du caractère privé (p) ou partagé (s) du segment au processus. Un segment peut contenir tout ou partie d'un fichier : celui-ci est déterminé par le décalage par rapport au début du fichier (offset), un type de périphérique (dev), un numéro de fichier (inode) et le chemin d'accès au fichier. Un tel segment est appelé une projection du fichier.

Un segment n'est pas nécessairement associé à un fichier, comme par exemple le tas ou la pile qui ne contiennent que des données. La cartographie d'un processus sous Linux est disponible

1. Tiré d'un exercice proposé par Stefan Bornhofen

dans le fichier /proc/<numero du processus>/maps.

4

Ecrivez un programme C/C++ contenant :

- Une constante gérée par le pre-processeur (avec #define).
- Une constante (avec const).
- Une variable globale.
- Une variable à l'intérieur du main().
- Un tableau alloué dynamiquement (en utilisant malloc() ou calloc()).

5

Ajoutez les instructions suivantes dans votre programme :

- Affichez le PID du processus à l'aide de la fonction getpid.
- Pour chacune des variables/constantes ci-dessus, affichez leur adresse :
  - En C : printf ("%p\n", variable);
  - En C++ : cout << "adresse = " << static\_cast<void\*>(variable) << endl;
- Insérez une pause juste derrière l'affichage (par exemple demandez à lire un caractère).

6

Exécutez et lorsque le programme est en pause, regardez dans le fichier maps associé au processus dans quels segments sont positionnées chaque variable/constante. Expliquez.