

20.- Mémoire : du bit à l'abstraction vue par le processus

Niveau : L2/L3

Prérequis : Bases en programmation

Motivation : Comprendre la notion de "mémoire" d'un ordinateur et son utilisation pour un processus

I] Notions de mémoire

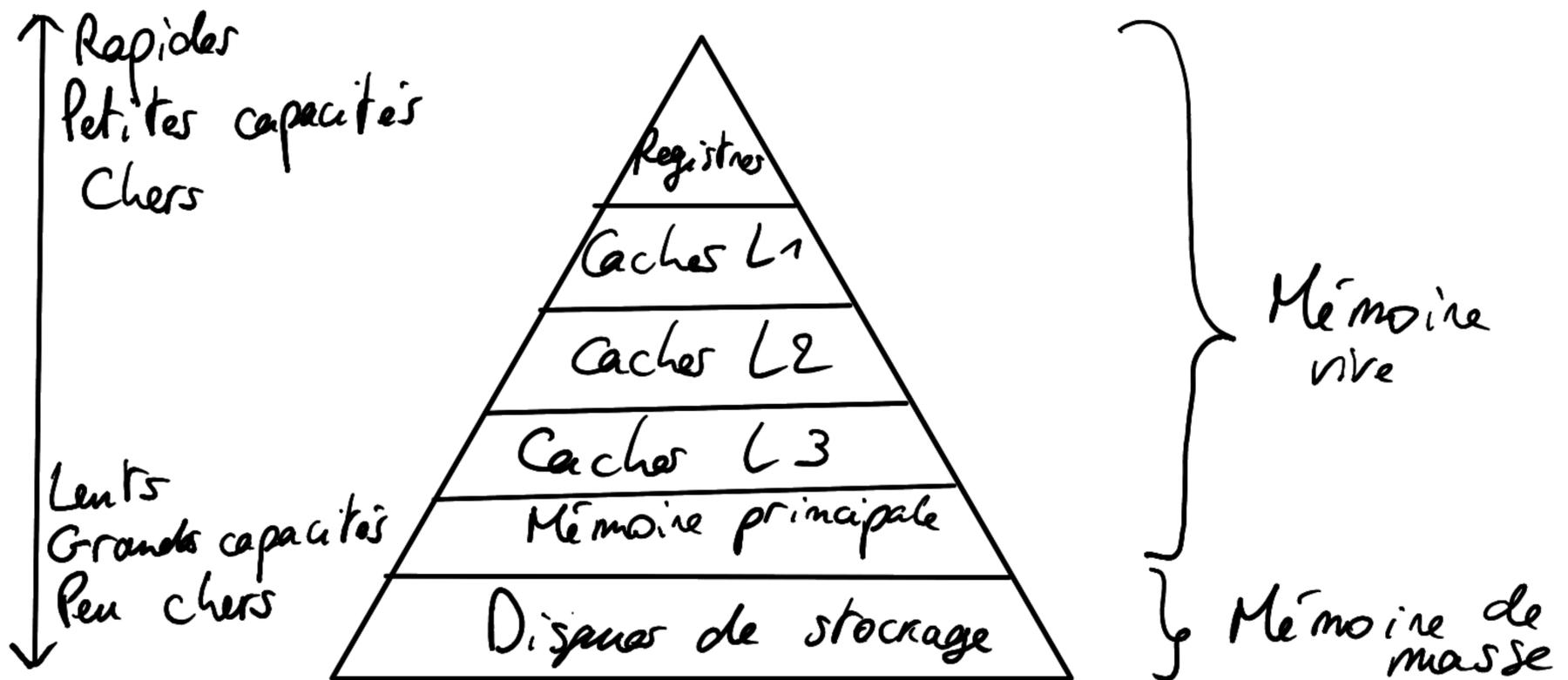
1.) Introduction:

Définition 1: Une **mémoire** (informatique) est un composant électronique de stockage d'informations. On distingue trois types de mémoire : la **mémoire morte**, la **mémoire vive** et la **mémoire de masse**.

Définition 2: L'unité de stockage est le **bit**, pouvant prendre les valeurs 0 et 1. Un groupe de 8 bits est appelé un **octet** (**byte** en anglais).

Remarque 3: Pour la mémoire, on utilise les préfixes binaires et non les préfixes S.I : ceux-ci sont définis par puissances de 2^{10} et non 10^3 .

Proposition 4: Dans la plupart des appareils numériques, la mémoire est organisée sous forme de hiérarchie, appelée la **hiérarchie mémoire**. Celle-ci est basée sur différents composants faisant un compromis entre capacité, vitesse d'accès et coût.



Exemple d'une hiérarchie mémoire d'un ordinateur

2.) Registres

Definition 5 : Les **registres** sont des mémoires vives de la taille d'un **mot** (usuellement 32 ou 64 bits) immédiatement accessibles par le CPU.

Remarque 6 : Tous les registres sont nommés, et certains peuvent contenir des informations spéciales.

Exemple 7 :

- PC contient l'adresse mémoire de la prochaine instruction exécutée par le CPU.
- CR3 contient l'adresse de la table des pages (voir II).

DEV : Principe et fonctionnement de la bascule D.

3.) Mémoire principale

Definition 8 : la **mémoire principale** (ou **mémoire vive**, ou **RAM**) est la mémoire stockant toute les informations et les états des différents programmes exécutés sur une machine.

Propriété 9 : Les mémoires vives sont à **accès direct**, c'est-à-dire que le temps d'accès à une information stockée est la même quelle que soit son adresse.

4.) Caches

Définition 10 : Les **caches (mémoires)** sont des composants électroniques ou logiciels recopiant une partie de la mémoire à laquelle ils sont associés pour accélérer le temps d'accès aux informations stockées.

Exemples 11 :

- Caches L1, L2 et L3 pour la RAM
- TLB pour la MMU (voir partie II)
- Caches d'inodes pour les systèmes de fichiers

II] Mémoire pour le système d'exploitation :

1.) Introduction à l'abstraction de la mémoire :

Proposition 12 : Jusqu'à dans les années 80, tous les ordinateurs n'avaient pas d'abstraction de la mémoire. Tous les programmes avaient accès à toute la mémoire physique. Il était alors difficile d'exécuter plusieurs processeurs en même temps.

Problèmes 13 :

- Difficulté de mise en place d'une exécution parallèle.
- Très mauvaise sécurité, que ce soit au niveau des secrets ou au niveau du bon déroulement des programmes.

2.) Espaces d'adressage :

Definition 14: Un **espace d'adressage** est une abstraction de la mémoire qu'un processus peut utiliser. C'est alors l'ensemble des adresses qu'il a à sa disposition pour stocker des informations.

Proposition 15: Chaque processus a son propre espace d'adressage, indépendant des autres (sauf en cas de partage de mémoire).

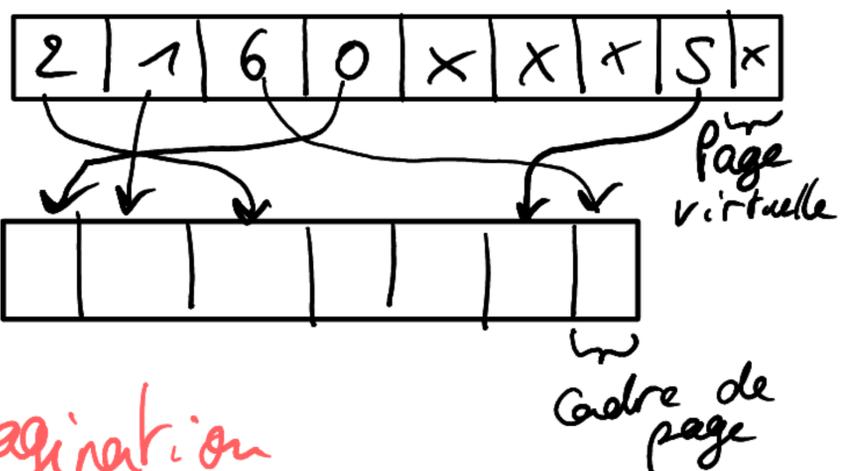
Proposition 16: Pour gérer la mémoire de plusieurs processus, deux approches peuvent être utilisées: le **swapping**, consistant à la copie de la totalité de la mémoire à chaque changement de processus, ou le **mémoire virtuelle**.

3.) Mémoire virtuelle:

Proposition 17: La mémoire virtuelle repose sur la découpe de l'espace d'adressage en **pages**, espaces d'adresses contiguës, généralement de quelques Kio. Ces pages sont mappées sur la mémoire physique, mais toutes ne le sont pas.

Exemple 18: Mémoire virtuelle

Mémoire physique



Exemple de **pagination**

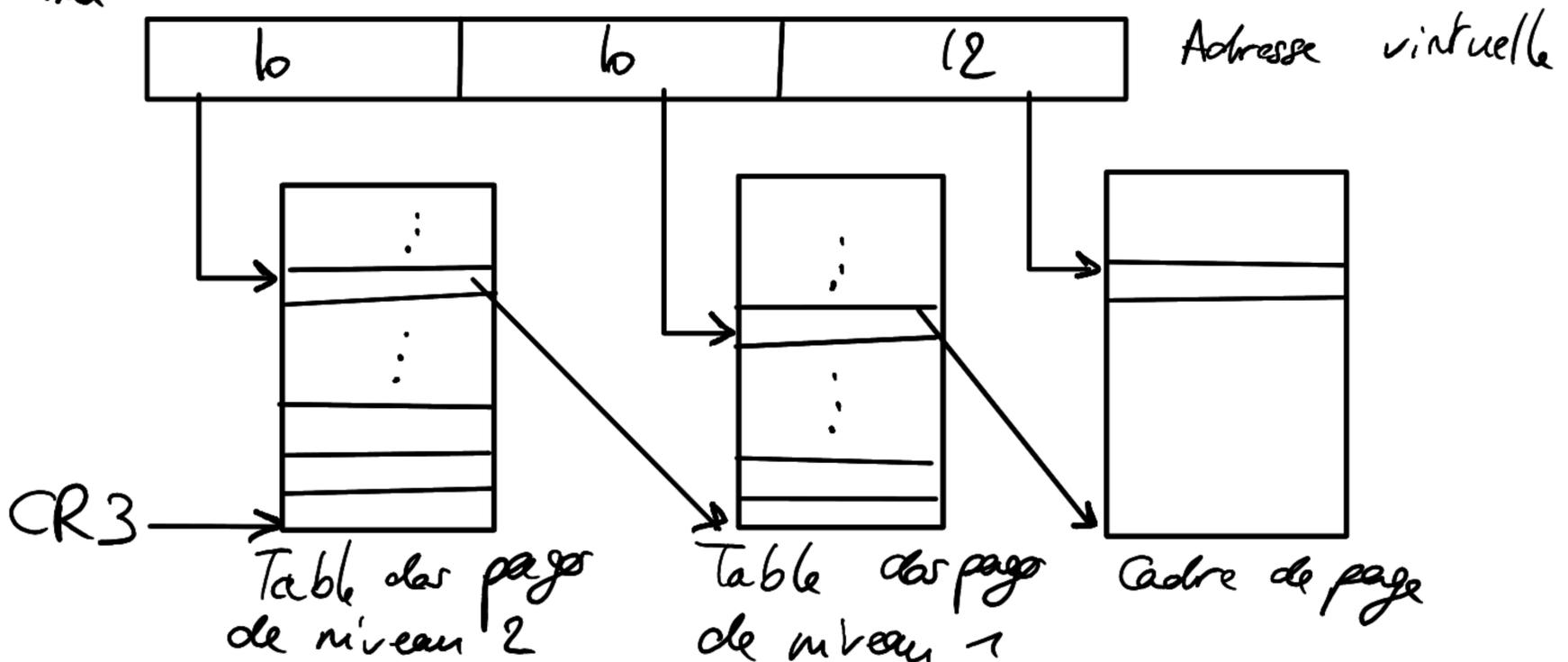
Proposition 19: Les adresses générées par les processus sont appelées des **adresses virtuelles** et forment l'espace d'adressage virtuel.

Proposition 20 : Dans les ordinateurs modernes, un composant électronique, la **MMU** (**Memory Management Unit**) est chargé de faire la traduction entre les adresses virtuelles et les adresses physiques. Pour accélérer, un cache est souvent associé à la MMU: le **TLB** (**Translation Lookaside Buffer**).

Proposition 21 : La correspondance entre les pages virtuelles et les cadres de page est indiquée dans la **table des pages**, unique à chaque processus. Celle-ci est inscrite dans la mémoire et est pointée par l'adresse contenue dans le registre **CR3**.

Remarque 22 : Dans le cas de grandes mémoires (plusieurs Go), on utilise des tables de pages multiniveaux.

Schéma 23 : Traduction d'une adresse virtuelle en adresse physique



Proposition 24 : Les pages étant allouées et désallouées au fur et à mesure de l'exécution des processus, il est nécessaire de réutiliser des pages préalablement désallouées. On utilise pour cela des algorithmes de **remplacement des pages**.

Exemple 25 : Algorithmes de remplacement des pages

- Algorithme premier arrivé, premier servi
- Algorithme de l'horloge.

4.) Processus en mémoire:

DEV: Organisation d'un processus en mémoire

Proposition 26: Les processus dans Linux peuvent accéder à des données stockées dans des fichiers mappés dans leur mémoire. On les appelle les **memory-mapped files**.

Exemple: Bibliothèques partagées

Proposition 28: Sous Linux, il est possible de partager des parties de la mémoire entre plusieurs processus. C'est notamment le cas pour les segments de code, ou pour les fichiers ouverts.

Schema 29: Partage d'un fichier mappé

