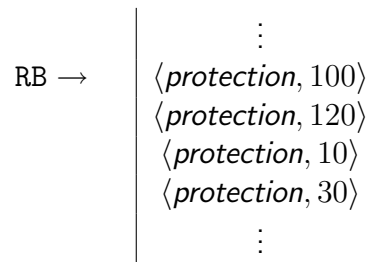


TD7 : Pagination & remplacement de pages

1 Transformation des adresses paginées

Prenons un système de mémoire paginée dans lequel les adresses sont codées sur 32 bits et chaque page mesure 2^{10} octets. Le système associe à chaque processus une table des pages comme dans l'exemple ci-dessous :



Chaque entrée de la table des pages est un couple

$\langle \textit{protection} \textit{ (8 bits)}, \textit{numéro de page physique (24 bits)} \rangle$.

Dans ces conditions,

- quelle est la taille maximum de la mémoire logique d'un processus et la taille maximum de la mémoire physique ?
- quelles sont les adresses physiques qui correspondent aux adresses logiques 2500, 750, 3072 et 1025 ?

2 Les algorithmes de remplacement

En cas de défaut de page dans un système paginé, les algorithmes de remplacement doivent déterminer une page physique victime qui sera sauvée sur disque (si nécessaire) et remplacée par la page virtuelle que l'on désire consulter. Il existe principalement, quatre stratégies pour choisir la page victime :

- Algorithme FIFO : La victime est la page physique la plus anciennement chargée.
- Algorithme LRU (Least Recently Used).
- Algorithme FINUFO (First In Not Used First Out) ou algorithme de la 2ème chance.
- Algorithme LFU (Least Frequently Used).

On vous donne les constantes NB_PP (nombre de pages physiques), NB_PV (nombre de pages virtuelles) et les tables ci-dessous.

```
struct {
    int presente;
    int npp;
} dpv [NB_PV];

struct {
    int libre;
    int npv;
    int modif;
    int info;
} dpp [NB_PP];
```

2.1 Transformation des adresses

A l'aide de ces déclarations, on vous demande d'écrire la fonction `int npv2npp(int npv)` qui détermine un numéro de page physique (`npp`) à partir d'un numéro de page virtuelle (`npv`). Elle utilise la fonction `default_de_page(int npv)` qui traite l'apparition d'un défaut de page sur la page virtuelle `npv` en la chargeant dans une page physique.

2.2 Traitement du défaut de page

Écrivez la routine `default_de_page(int npv)` en utilisant les fonctions ci-dessous.

<pre>void sauver_page (int npp, int npv) ; void charger_page (int npv, int npp) ; int choisir_victime (void) ; int nb_pp_occupees ;</pre>	<pre>mém. centrale → mém. secondaire mém. centrale ← mém. secondaire choisir une page physique victime</pre>
---	--

2.3 Choix de la victime

La fonction `choisir_victime()` choisit la page physique victime en cas de défaut de page. On vous demande de programmer cette fonction pour les quatre algorithmes décrits précédemment.

- Pour la version FIFO vous utiliserez une variable globale `victime` que donne la dernière page physique victime.
- Pour la version FINUFO vous utiliserez une variable globale `finufo` qui donne la dernière page physique victime.