Extension de mémoire et mémoire virtuelle,

SYE

11 - Extension de mémoire

Abstract

Ce document traite de la gestion de la mémoire virtuelle et physique dans un système paginé. Il explore comment la mémoire est segmentée en pages fixes, avec une gestion transparente des adresses par la MMU, et aborde des concepts tels que le swapping pour déplacer des pages entre RAM et disque en cas de surcharge. Il détaille des structures de données comme les bitmaps et les listes pour gérer les pages libres, ainsi que des mécanismes comme la pagination à la demande pour optimiser l'utilisation de la mémoire. Enfin, il explique les fautes de page, leur gestion par le système d'exploitation et les algorithmes de remplacement nécessaires lorsque la RAM est pleine, tout en décrivant les attributs des PTE (Page Table Entries) pour contrôler les pages.

Table des matières

1. Gestion des pages physiques

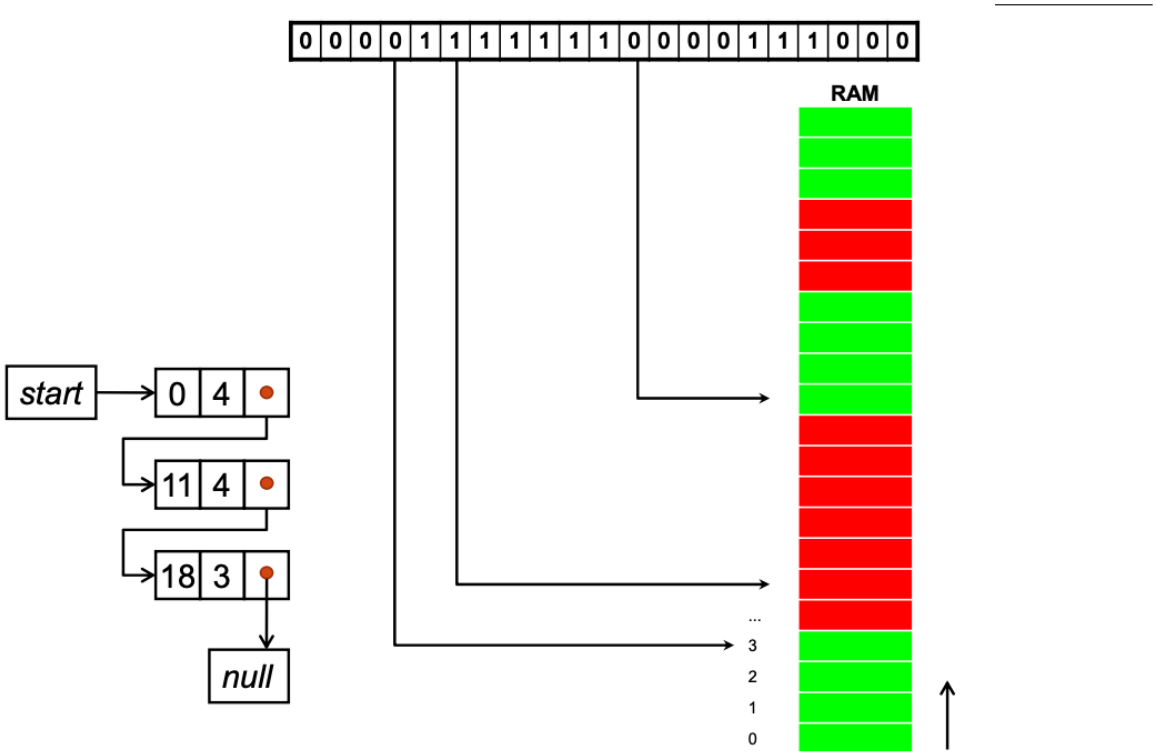
La mémoire physique (RAM) découpée en page constitue un récipient de pages. Les pages sont restituées lorsqu'un processus se termine.

1.1. Structure de données

La gestion des pages libres nécessite une structure de données adéquate. Généralement deux structures sont utilisées: un tableau de bit (bitmap) où chaque bit représente une page physique et une liste de pages libres.

1.1.1. Bitmap

Le bitmap est un tableau de bits où chaque bit représente une page physique. Un bit à 1 indique que la page est occupée, un bit à 0 indique que la page est libre. Cela permet de savoir rapidement s'il existe des pages contiguës libres. La taille de la bitmap dépendra directement de la taille de la mémoire physique.



1.1.2. Liste de pages libres

Une liste de pages libres en revanche permettra de stocker d'autres informations relatives à la page physique: son type, ses droits d'accès, etc. La liste présente l'avantage d'avoir une taille variable, et d'être réduite à néant lorsque toute la mémoire est utilisée.

1.2. Gestion mémoire pleine

Si toutes les pages physiques sont allouées, nous avons deux cas de figure:

- un memory overflow
- nécessité d'effectuer un swapping de page
  - une page peut être transférée sur le disque et libérer la place
  - il s'agit de sélectionner quelle page doit être remplacée

2. Extension de la mémoire

Un système paginé découpe la mémoire en pages fixes (ex. 4 Ko), et la MMU traduit les adresses virtuelles en adresses physiques de manière transparente. Les pages physiques peuvent être déplacées en RAM ou sur un support de stockage via le swapping. Ce mécanisme permet de transférer des pages peu utilisées de la RAM vers un espace de swap (partition ou fichier dédié) et de les recharger si nécessaire. Cela étend virtuellement la mémoire au-delà de sa taille réelle, en fonction de la taille de l'espace de swap. Sur une architecture mono-cœur, une seule page est utilisée à la fois, tandis que plusieurs pages peuvent être utilisées simultanément sur une architecture multi-cœur.

Les pages physiques peuvent donc subir des mouvements:
- déplacement d'une page en RAM
- déplacement d'une page (momentanément inutile) sur le disque dur
- rechargement d'une page depuis le disque dur dans la RAM
- swap au niveau des pages

À chaque mouvements il faut mettre à jour la table de page!

2.1. Pagination à la demande

La **pagination à la demande** permet de charger en mémoire uniquement les pages nécessaires à un instant donné, plutôt que l'intégralité des pages d'un processus. Cela optimise l'utilisation de la mémoire en ne chargeant que les pages correspondant aux instructions ou données en cours d'exécution, ainsi que celles susceptibles d'être utilisées prochainement. Les pages contenant du code sont souvent récupérées directement depuis le fichier binaire sur disque, sans être stockées dans l'espace de swap, car elles ne sont pas modifiées. Cette gestion nécessite une forte interaction entre le gestionnaire mémoire et le système de fichiers, notamment via des fichiers mappés, où le contenu est projeté en mémoire sous forme de blocs similaires aux pages mémoire.

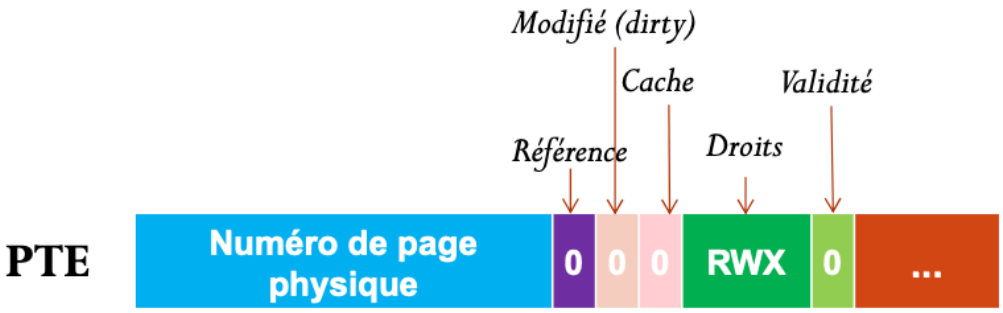
2.2. Introcution au faute de page

Une **faute de page** survient lorsqu'une page est demandée par un processus mais n'est pas présente en mémoire. Cela peut être dû à une page non chargée, à une page déplacée sur le disque ou à une page non allouée. La faute de page est gérée par le système d'exploitation, qui charge la page depuis le disque si nécessaire, puis reprend l'exécution du processus. La gestion des fautes de page est un élément clé de la gestion de la mémoire virtuelle, car elle permet d'optimiser l'utilisation de la mémoire en ne chargeant que les pages nécessaires à un instant donné.

Nous retrouvons donc deux mécanismes sous-jacents à la gestion de la mémoire virtuelle:
- **faute de page**
  - la page n'est pas présente dans la RAM
  - exception (interruption logicielle)
- **remplacement de page**
  - dans le cas où la RAM est pleine
  - algorithme de remplacement de page

3. Concept de faute de page

Une PTE contient des bits d'attributs:



- Bit de **référence** (R): indique si la page a été référencée et est mis à 1 lors du référencement
- Bit de **modification (dirty)** (M): indique si la page a été modifiée, est mis à 1 lors de la modification
- Bit de **cache** (C): indique si la page est présente dans le cache
- Bits de **protection** (RWX): indiquent les droits d'accès à la page (lecture, écriture, exécution)
- Bit de **présence** (P): indique si la page est présente en mémoire

3.1. Exemple

Lorsqu'une page non présente en RAM est demandée, une **faute de page** se produit :

1. L'instruction accède à une adresse mémoire que la MMU traduit. La table des pages est consultée et le bit de validité indique que la page est absente.
2. Une interruption de faute de page est levée.
3. La routine de service récupère la page manquante, éventuellement depuis le swap.
4. L'allocateur mémoire charge la page en RAM. Si la mémoire est pleine, une page est évincée selon un algorithme de remplacement.
5. La table des pages est mise à jour pour associer la page virtuelle à une page physique.
6. L'instruction à l'origine de la faute est réexécutée, et l'accès mémoire réussit.