8INF342 – Systèmes d'exploitation
Laboratoire #3
Gestion de la mémoire virtuelle

Automne 2023 **Professeure : Sara Séguin**

Instructions générales

- Devoir à remettre au plus tard le **7 décembre 2023 à 19h**.
- Travail à réaliser en équipe de 2 étudiants.
- Un seul rapport à remettre par équipe.
- Ce travail compte pour 12% de la note finale.

Objectifs du devoir

- Développer, en C++, un programme qui intègre les bonnes pratiques de la programmation.
- Implémenter un algorithme simulant un système de gestion de la mémoire virtuelle.
- Comprendre le fonctionnement des tables de page et du TLB (translation lookaside buffer).
- Comprendre le processus de traduction des adresses logiques en adresses physiques.

Mise en place

Le devoir est réalisé en C++ sous Linux.

Le fichier adresses.txt contient les adresses à traduire.

Le fichier correct.txt contient les adresses traduites ainsi que la valeur du byte signé, afin de valider votre programme.

Le fichier simuleDisque.bin simule le stockage des pages en mémoire secondaire.

Énoncé du devoir

Vous devez programmer, en C++, un simulateur de gestion de la mémoire virtuelle.

Vous disposez d'un fichier contenant 1000 adresses logiques représentées par des nombres entiers de 32 bits (adresses.txt)

La structure d'adresse est de 16 bits, vous devez donc prendre les 16 bits les moins significatifs de chaque adresse logique :

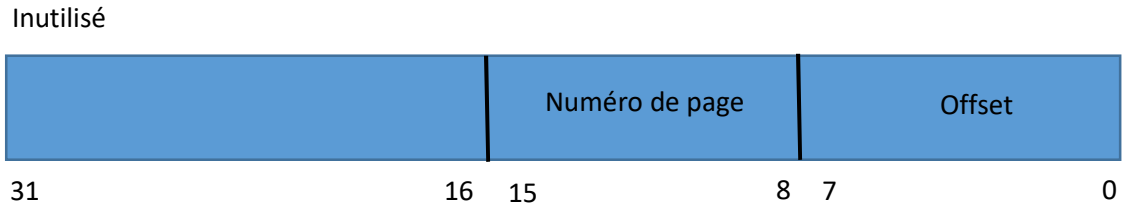

Figure 1 : Format des adresses

Votre programme doit lire les adresses logiques et les traduire en adresses physiques.

Votre espace d'adresses virtuelles est 2^{16} = 65 536 bytes.

Les spécifications suivantes doivent être respectées :

- 2^8 entrées dans la table de pages (PTE, donc il y a 256 lignes)
- Taille des pages de 2^8 bytes
- 16 entrées dans le TLB
- Taille des frames de 2^8 bytes
- 256 frames
- Mémoire physique de 65 536 bytes (256 frames x 256-byte par frame)

Puisque la taille de la mémoire physique est la même que la taille de la mémoire virtuelle, vous n'avez pas à vous soucier du remplacement des pages pour la mémoire physique. **Lorsque vous lisez une adresse virtuelle, vous pouvez simplement la charger dans le prochain frame, en considérant que la mémoire est vide initialement.**

Étape 1 : Extraction du numéro de page et de offset

Dans un premier temps, exercez-vous à extraire des numéros de page et de offset pour des nombres entiers. Par exemple, si vous prenez la valeur 1, vous pouvez l'écrire sous forme binaire 16 bits : 0000 0000 0000 0001. Vous pouvez utiliser des opérateurs de masque de bit ou de bit shift pour obtenir, par exemple : page = 0000 0000 et offset = 0000 0001. Testez plusieurs valeurs afin de vous assurer que votre code fonctionne correctement.

Étape 2 : Traduction des adresses sans TLB

Lorsque vous traduisez une adresse logique en adresse physique, vous devez lire l'adresse logique et regarder si la page se trouve en mémoire principale. Une structure de données (un tableau peut faire l'affaire) simulant la table de page doit être créée et une autre simulant la mémoire physique. Si la page se trouve en mémoire, alors le bit de présence correspondant à la page sera

égal à 1 et le numéro du frame où est chargée la page sera disponible. Si la page ne se trouve pas en mémoire, elle devra être chargée à partir du disque et un « page fault » est généré. Lorsque vous chargez une page en mémoire, vous devez copier la page de 256 bytes dans un frame disponible en mémoire principale. Allez-y séquentiellement, chargez la première page voulue dans le frame 0, la seconde dans le frame 1 et ainsi de suite. Par exemple, si la première adresse à traduire nécessite la page 20, vous chargeriez la page 20 dans le frame 0. La table de page sera aussi mise à jour en spécifiant le numéro du frame à la bonne PTE et en mettant le bit de présence à 1.

Le fichier simuleDisque.bin est un fichier binaire de 65 536 bytes. Chaque page a une taille de 256 bytes. Ainsi, lorsque vous traduisez une adresse, vous devez copier la page en mémoire physique, puis retourner l'adresse physique, en plus de la valeur du byte signé situé à l'adresse physique. Petit rappel : un byte signé peut prendre les valeurs -128 à 127. Lorsque vous êtes situé au bon endroit dans votre traduction (adresse physique), vous n'avez qu'à lire un char, qui est un byte en C++.

Vous pouvez valider vos réponses puisque le fichier correct.txt contient les adresses physiques, en plus de la valeur du byte signé situé à l'adresse physique (correct.txt).

Étape 3 : Traduction des adresses avec TLB et algorithme de remplacement des pages dans le TLB

Cette partie ajoute une difficulté en ajoutant un Translation Lookaside Buffer (TLB).

Lorsque vous traduisez une adresse logique, vous devez d'abord vérifier si la page est dans le TLB. Le TLB contient le frame associé à la page. (Si une page est une entrée du TLB, elle est donc chargée en mémoire.) Si elle y est, alors vous récupérez directement le numéro du frame. Sinon, vous devez parcourir la table de pages, et faire comme à l'étape 2.

La Figure 2 illustre le principe de fonctionnement du système de gestion de la mémoire virtuelle.

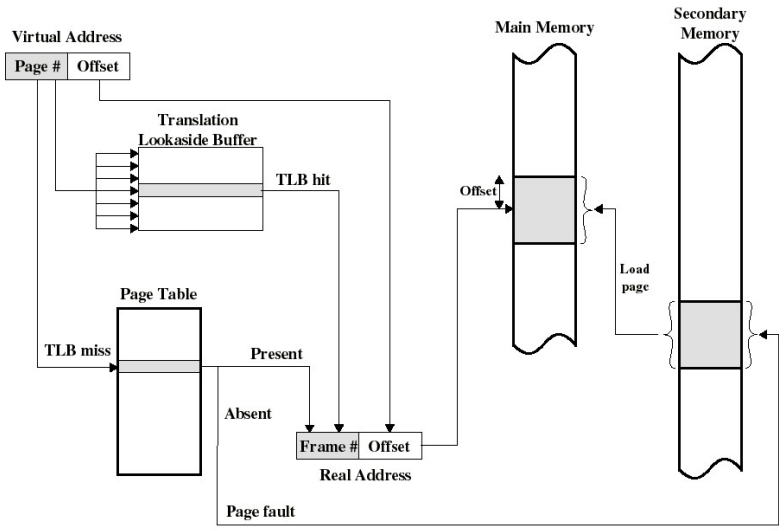

Figure 2 : Fonctionnement du gestionnaire de la mémoire virtuelle

Précisions

Lorsqu'il y a un « page fault », cela signifie que la page demandée n'est pas chargée en mémoire. Le fichier « simule_disque.bin » simule un disque dur (la mémoire secondaire) qui contient les pages conservées sur le disque. Si la page demandée est, par exemple, 24, alors vous devrez aller charger la page 24 de ce fichier. Noter que ce fichier a une taille de 65536 bytes, et qu'une page a une taille de 256 bytes (0 à 256). Lorsque vous chargez une page en mémoire, vous devez donc mettre à jour la table de pages ainsi que le TLB.

Ne pas oublier que le TLB contient seulement 16 entrées. Lorsqu'il est plein, vous devez donc implémenter un algorithme de remplacement des pages. Vous devez implémenter l'algorithme **LRU (least recently used).**

Vous devez aussi calculer les statistiques suivantes :

1. Page fault rate. Le pourcentage des références aux adresses qui ont mené à un « page fault ».
2. Succès TLB. Le pourcentage des références aux adresses qui étaient disponibles dans le TLB.

Votre programme doit créer un fichier de sortie qui indiquera, pour chaque adresse à traduire sur une ligne :

1. L'adresse virtuelle à traduire en décimal (Virtuelle : x)

2. L'adresse physique correspondante en décimal (Physique : y)
3. La valeur du byte signé associé à cette adresse physique en binaire et en décimal.

   Ainsi, une ligne pourrait ressembler à ceci :

   Virtuelle : 30557 Physique : 93 Valeur dec :0 Valeur bin : 0000 0000

Ces fonctions et/ou librairies vous seront utiles :

- seekg
- read
- std ::ifstream
- close
- std ::getline
- std ::istringstream
- std ::bitset
- std ::stoi

<u>À remettre</u>

Vous devez préparer un rapport de laboratoire comportant :

- Page-titre!
- Introduction générale. Expliquer le but du laboratoire et formulez une introduction claire et concise.
- Vous devez expliquer votre démarche (logiciels utilisés, librairies, etc.) dans une section méthodologie. Identifier clairement chacune des étapes et bien expliquer comment votre code fonctionne (pages, offset, structures utilisées, fonctionnement du remplacement, etc.)
- Une section résultats et discussion, qui détaille les résultats obtenus ainsi que les difficultés rencontrées.
- Une conclusion.
- Le code doit être commenté.
- Vous devez remettre une copie électronique de votre travail.

<u>Barème</u>

Ce rapport compte pour **12%** de la note finale.

Rapport corrigé sur **100 points.**

Rapport : 10 points

Commentaires : 10 points

Solution : 60 points

Exécution devant le chargé de TD : 20 points