

Systèmes d'exploitation (SYE)

Profs Daniel Rossier, Alexandre Corbaz, Fiorenzo Gamba
Assistants : Anthony Jaccard, Basile Cuneo, Jean-Pierre Miceli

Mécanismes de pagination

lab11 (11.12.2023)

Objectifs de laboratoire

Ce laboratoire vous permettra d'exercer la notion de pagination et de traduction d'adresse en vous demandant de simuler une partie des fonctionnalités du composant chargé de la gestion des adresses virtuelles: la MMU

Récupération et rendu du laboratoire

Pour récupérer la branche de ce laboratoire, utilisez les commandes:

1. `"git fetch upstream"`
2. `"git checkout labXX"`

Ce laboratoire sera à rendre selon la méthode vue lors du précédent laboratoire, à savoir

1. Ajouter les fichiers modifiés avec `"git add <vos_fichiers>"`
2. Commiter les changements avec `"git commit -m '<message de commit>'"`
3. Envoyer les changements sur votre repo gitlab avec `"git push -u origin"` ou `"git push --set-upstream origin"`

Etape 1 – Implémentation d'un algorithme de traduction d'adresse

Cette étape consiste à implémenter le mécanisme de traduction d'adresse opéré par la MMU durant les opérations de lecture/écriture à une adresse virtuelle.

Pour cela, le fichier « **usr/host/memsim.c** » déclare une mémoire fictive (variable globale « **main_mem** ») de 64 KB organisée en **256 pages** de **256 bytes**. Dans le cadre de ce laboratoire, l'espace d'adressage virtuel sera de même taille que l'espace d'adressage physique. L'adressage est sur **16 bits** et la pagination est à un niveau.

Les prototypes des fonctions à implémenter sont disponibles dans le code. L'initialisation de la table des pages (variable « **page_table** ») est déjà implémentée.

- ⇒ La table des pages est initialisée de telle sorte à ce que toutes les pages virtuelles soient mappées dans l'espace mémoire physique (il y a donc déjà un mappage existant).
- ⇒ L'architecture de l'adresse virtuelle est de type **(8 | 8)**.
- ⇒ Les PTEs sont sur 16 bits, bien qu'il ne soit pas utile de disposer d'autant de bits pour encoder le numéro de page physique.
- ⇒ Examiner comment les bits sont utilisés à ce stade.

- a) Implémenter la conversion d'une adresse physique en pte dans la fonction « **virt_to_pte()** »
- b) Implémenter l'écriture d'un byte en mémoire dans la fonction « **store_byte()** ».
- c) Implémenter la lecture d'un byte en mémoire dans la fonction « **get_byte()** ».

- d) Une fonction permettant de printer une page mémoire « **print_page()** » est fournie et peut être utilisé pour débbugger.
- e) Tester et valider en executant memsim. La fonction « **test_mem()** » sera exécuté pour vérifier le remplissage de la mémoire