TP2 Nachos : Introduction des fichiers mappés

1. Introduction

L'objectif de ce TP est de mettre en œuvre les fichiers mappés dans Nachos.

2. Travail réalisé

Nous avons pu mettre en place le nouvel appel système nommé Mmap. Cette méthode permet d'associer un certain nombre pages consécutives dans l'espace d'adressage d'un processus. Ces pages sont associés aux adresses disques d'un fichier qu'on désire mapper. L'espace d'adressage d'un processus maintient donc un tableau des fichiers mappés contenant pour chacun l'adresse virtuelle de la première page mappée, le taille de l'allocation et le descripteur du fichier mappé.
Le mappage a donc été réalisé ainsi que la recopie des page modifiées du fichier mappé lors de la fin d'un processus. Nous avons observé qu'une réduction du nombre de page physiques trop petite (<50) empêchaît le shell de démarrer (remarque valable pour le TP1), cela nous as donc empêché de tester la recopie des pages mappées dans PhysMem::EvictPage.
Nous avons remarqué qu'il ne fallait pas fermer le fichier avec la primitive Close() avant de terminer le processus pour pouvoir recopier les pages mappées (le descripteur de fichier serait devenu faux sinon).

3. Difficultés rencontrés

- Nous avons perdu un certain temps à debugger un bug qui n'en était pas un : un débordement de pile « sauvage » entraînait un crash sans un message de type « Stack Overflow » car on dépassait la zone de « sécurité » permettant ce test dans notre programme de test.

4. Tests effectués

Nous avons mis en place un programme de test qui a pour but de trier un tableau tableau issu d'un fichier que l'on mappe dans la mémoire. Ce tri s'effectue correctement et avec les bonnes valeurs.

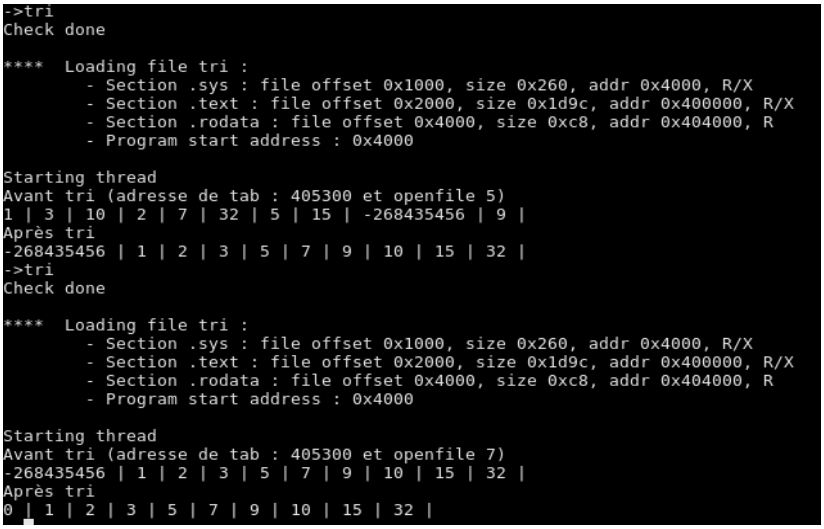

Illustration 1: Exemple d'exécution du programme de tri dans le shell nachos

5. Conclusion

Ce TP nous a permis de mieux comprendre les mécanismes mis en oeuvre au niveau gestion mémoire pour la mise en place des fichiers mappés.