

Gestion mémoire

V. FELEA & A. HUGEAT

Ces exercices concernent la partie de conception des systèmes, pour le volet gestion mémoire. D'une part, nous nous intéressons aux commandes/fichiers système permettant d'obtenir des informations sur la gestion mémoire. D'autre part, les problèmes nécessitant une programmation permettront d'approfondir les notions de cours et l'analyse faite en TD sur les concepts mémoire pour les systèmes d'exploitation.

1. Commandes système Linux

Cet exercice propose de visualiser les informations liées à la gestion mémoire, par l'intermédiaire des commandes système et/ou du contenu des fichiers système.

Remarque Des informations plus amples sur les commandes/fichiers système donnant des informations sur la mémoire se trouvent dans le document déposé sur Moodle.

(a) Informations générales sur le système - commandes **uname** et **uptime**

Q1 Noter les informations fournies par **uname** ainsi que leur signification. Donner également la version courante de la distribution Linux utilisée.

Q2 Même question pour la commande **uptime**.

(b) Mémoire - généralités

Q3 Quelle est la taille de la mémoire physique de la machine ?

Q4 En utilisant la commande **ps**, afficher la liste de tous les processus avec le format :

`commande tailleV tailleR Pmem`

où **tailleV** est la taille de la mémoire virtuelle du processus, **tailleR** est la taille de la mémoire physique utilisée par le processus, et **Rmem** est le pourcentage de mémoire physique utilisée par le processus par rapport à la mémoire physique totale du système.

Q5 Que fait la commande **free** ? Comment faire pour obtenir un nouvel affichage toutes les 5 secondes ? Que pouvons-nous constater selon les affichages successifs ?

Q6 Quelles informations peuvent être obtenues en utilisant la commande **vmstat** ? Noter en particulier celles relatives à la gestion de la mémoire.

(c) Pagination

Q7 Quelle est la taille d'une page utilisée par le système ? (utiliser la commande **getconf**)

- Q8 Le nombre de défauts de page d'un processus en exécution peut être visualisé grâce à la commande `ps`. Choisir un processus s'exécutant dans le système et donner cette information.
- Q9 La commande `time(1)` permet d'exécuter des programmes et résume l'utilisation des ressources. Plus particulièrement, l'option `-v` donne plusieurs statistiques, dont le nombre de défauts de page. Exécuter deux fois successivement la commande `xclock` et comparer le nombre de défauts de pages obtenus. Expliquer. **Attention.** `time` est un mot réservé dans les interpréteurs de commandes (shells). Pour utiliser la commande `time` : `command time [options] [command]` ou `/usr/bin/time [options] [command]`.

2. Allocation pas subdivisions (Buddy Allocation)

Rappel de l'algorithme d'allocation par subdivision de Knuth.

Dans les méthodes d'allocation par subdivision, les tailles de mémoires allouées sont variables, mais multiples d'une certaine unité d'allocation et les tailles permises sont définies par une relation de récurrence. Un mécanisme d'allocation utilisé par le système Linux est l'allocation buddy où les tailles des zones disponibles sont des puissances de 2 (2^0 , 2^1 , 2^2 , etc.).

Pour chacune des tailles, une liste des blocs libres est conservée dans une des cases d'une table appelée *Table des Zones Libres* (TZL). Dans le système binaire, il y a donc une liste des blocs de taille 2^0 dans la case 0, 2^1 dans la case 1, 2^2 dans la case 2, ..., 2^i dans la case i . Les informations sur les zones libres sont stockées dans les zones libres elles-mêmes, en particulier le pointeur vers le bloc libre suivant dans la liste.

Lorsqu'il y a une nouvelle demande d'une zone de taille arrondie supérieure à 2^k et que ce type de zone n'existe pas, une zone libre de taille 2^p ($p \geq k + 1$, avec le plus petit p) sera divisée en 2 zones compagnons. Ce mécanisme est itéré sur une seule des deux zones libres créées, jusqu'à obtenir une zone de taille 2^k .

Lorsqu'un processus finit son exécution et libère l'espace occupé, la zone d'allocation devient libre et si la zone compagnon est libre, les deux sont fusionnées. Ce procédé est itéré tant qu'une fusion est possible.

Il est demandé d'implémenter le gestionnaire d'allocation mémoire avec l'algorithme de Buddy allocation. Dans un premier temps, une seule demande d'allocation sera simulée. Dans un deuxième temps, plusieurs demandes successives seront prises en compte, sans la libération de la mémoire.

3. Transcodage d'adresses dans un système paginé

Implémenter un programme simulant le transcodage d'adresses dans un système paginé. Considérer essentiellement la pagination à un niveau. Le cas d'une pagination à deux niveaux pourra également être traité.

4. Algorithmes de remplacement de pages

Implémenter les algorithmes optimal, LRU et FIFO de remplacement de pages. Calculer pour chaque jeu de données (nombre de cadres disponibles et succession de pages demandées) le taux de défauts de pages.