623 Consommation mémoire

INF3173
Principes des systèmes d'exploitation

Jean Privat

Université du Québec à Montréal

Hiver 2021

Allocation paresseuse aux processus

Les processus

Ont tendance à allouer de la mémoire

- Qu'ils n'utilisent pas entièrement
- Qu'ils n'utilisent pas de suite
- Qu'ils n'utilisent pas du tout

Le système d'exploitation peut promettre

- Reçoit les demandes des processus
- Répond positivement et alloue des pages virtuelles
- N'associe pas encore de page physique

Puis livrer plus tard

- C'est seulement lors du premier accès à la mémoire
- Que le système d'exploitation associera une page physique

Mise en œuvre de l'allocation paresseuse

- La zone mémoire virtuelle est créée (ou agrandie)
- Les pages logiques sont laissées invalides dans la table des pages

Au premier accès

- MMU lève une faute
- Système d'exploitation
  - Attrape l'interruption
  - Alloue une page physique
  - Met à jour la table des pages
  - Redonne la main au processus
- Processus réussit son instruction (cette fois)

$\rightarrow$ Défaut de page mineur

Page « zéro » (Linux)

- La « page zero » est une page physique globale et unique
- En lecture seule
- Ne contient que des 0
- Associée aux pages nouvellement allouées aux processus
- Les processus peuvent déjà y lire des zéros
- Nettoyage de mémoire « gratuit »
- Lors de la première écriture,
  On alloue une page physique privée (COW)

Questions

- Est-ce que la mise à zéro est vraiment gratuite ?

zero.c

```c
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
int main(int argc, char **argv) {
  size_t size = 3l << 30;
  size_t page = sysconf(_SC_PAGESIZE);
  long cpt = 0, opt;

  char *ptr = malloc(size);
  if (ptr==NULL) { perror("malloc"); return 1; }

  while((opt = getopt(argc, argv, "rwp")) != -1) {
    if (opt == 'r')
      for(size_t i=0; i<size; i += page)
        cpt += ptr[i];
    if (opt == 'w')
      for(size_t i=0; i<size; i += page)
        ptr[i] = 1;
    if (opt == 'p') pause();
  }

  free(ptr);
  return cpt;
}
```

Mémoire infinie

Thrashing

- Presque plus de place en RAM
- Le système d'exploitation passe son temps à sauver et charger des pages du disque
- Dégradation importante des performances et de l'interactivité

OOM (*Out of memory*)

- Situation où il n'y a plus de mémoire: ni RAM ni en swap
- Le système d'exploitation en a besoin maintenant
- Ou en avait promis aux processus qui y accèdent

La mémoire ne peut être « reprise » aux processus

- Car le `malloc(3)`/`mmap(2)`/`brk(2)` historique a réussi
- Donc les pages existent dans l'espace virtuel des processus
- Les reprendre causerait un dysfonctionnement

Solution : extermination

OOM killer (Linux)

- Détermine quels processus terminer de force
- Objectif: permettre au système d'exploitation de survivre

Terminer oui, mais qui ?

- Nombreuses heuristiques pour terminer le « meilleur » processus
- Idéalement le processus responsable du OOM
- Parfois se trompe de processus (oups!)

Configuration OOM

- `/proc/PID/oom_score_adj` : bonus/malus (configurable)
- `/proc/PID/oom_score`: score actuel du processus (si OOM maintenant)
- `choom(1)` affiche et configure le score

Surengagement (*overcommitment*)

- « Prêter de la mémoire qu'on a pas »
- C'est pas toujours une bonne idée
- Une des causes principales des OOM

Linux

- `/proc/sys/vm/overcommit_memory` change la politique
  - 0 (défaut) = refuse seulement les demandes absurdes
  - 1 = accepte toutes les demandes d'allocation
  - 2 = pas de surengagement au-delà d'une certaine limite
- `/proc/sys/vm/overcommit_ratio` ou `/proc/sys/vm/overcommit_kbytes` Configure la limite si le mode est 2 (+50% par défaut)
- `CommitLimit` et `Committed_AS` de `/proc/meminfo`

Récapitulatif : la mémoire c'est compliquée

Plein de combinaisons pour chaque page virtuelle

- Privé vs. partagé
- Initialisé vs. non-initialisé
- Fichier projeté vs. anonyme
- Résidente en mémoire, ou sur disque
- Sale (*dirty*) = fichier à mettre à jour sur le disque, ou non
- Etc.

Plus

- Les structures supplémentaires (table des pages, etc.)

Conclusion

- La mémoire c'est compliquée
- « Compter » la mémoire utilisée ou libre
  C'est compliqué aussi