Polytechnique Montréal

Dpartement de Génie Informatique et Génie Logiciel

INF2610 - Noyau d'un système d'exploitation

TP 4

Gestion de la mémoire

Question	Description	Pointage
1	Gestion de la mémoire	16

Structure du TP

Ce travail est composé d'une question visant à mieux comprendre le fonctionnement général de la gestion de la mémoire et traduction d'adresses.

Il vous est toujours possible de regarder si vous avez réussi ou non une question en exécutant le script **./evalue.sh**.

À La fin du TP, vous devrez lancer la commande *make handin* dans le répertoire principal du tp afin de créer l'archive *handin.tar.gz* que vous devrez remettre sur autolab.

Prenez le temps de lire attentivement l'énoncé.
BON TP! :)

Gestion de la mémoire

On considère un système à mémoire paginée pour lequel la taille d'une page est 2^{10} (1024) octets.

Le code à compléter se trouve dans le fichier **GestionMemoire.c**.

L'objectif de cette question est d'implémenter les fonctions de recherche dans le TLB et dans la table des pages en suivant les étapes suivantes :

- Complétez la fonction *calculerNumeroDePage* du fichier GestionMemoire.c. Cette fonction prend en paramètre une adresse virtuelle et doit renvoyer le numéro de page correspondant à cette adresse virtuelle. (Vous pourrez évidemment utiliser cette fonction pour les adresses physiques, car **la taille d'une page correspond à celle d'un cadre**). (/1)

- Complétez la fonction *calculerDeplacementDansLaPage* du fichier GestionMemoire.c. Cette fonction prend en paramètre une adresse virtuelle et doit renvoyer le déplacement dans la page correspondant à cette adresse virtuelle. (Vous pourrez évidemment utiliser cette fonction pour les adresses physiques, car **la taille d'une page correspond à celle d'un cadre**). (/1)

- Complétez la fonction *calculerAdresseComplete* du fichier GestionMemoire.c. Cette fonction prend en paramètre un numéro de cadre ainsi qu'un déplacement dans le cadre et doit renvoyer l'adresse avec le numéro donné et le déplacement (*offset*) donné. (/1)

Après ces trois étapes, vous allez devoir implémenter la recherche dans le TLB et la table des pages.

- Complétez la fonction *rechercherTLB* du fichier GestionMemoire.c. Cette fonction prend en paramètre une structure *RequeteMemoire* et une structure *SystemeMemoire*. Vous devez stocker l'adresse physique calculée dans l'attribut *adressePhysique* de l'objet de type *struct RequeteMemoire* passé en paramètre ou 0 si aucune traduction n'est possible. Vous devez également modifier la date de dernier accès à l'entrée du TLB à la date de la requête (attribut date de req). (/3)

- Complétez la fonction *rechercherPT* du fichier GestionMemoire.c. Cette fonction prend en paramètre une structure *RequeteMemoire* et une structure *SystemeMemoire* et sauvegarde l'adresse physique correspondant à l'adresse virtuelle de la structure request ou 0 si aucune traduction n'est possible pour le moment dans l'attribut *adressePhysique* de l'objet de type *struct RequeteMemoire* passé en paramètre. Il n'y a aucune date à modifier dans cette fonction (voir structure de la table des pages). (/3)

- Complétez la fonction *ajouterDansMemoire* du fichier GestionMemoire.c. Cette fonction prend en paramètre une structure *RequeteMemoire* et une structure *SystemeMemoire* et ajoute la page virtuelle dans le premier cadre disponible en mémoire principale. Vous devez, comme pour la fonction *rechercherTLB*, modifier l'attribut adressePhysique de l'objet de type *struct RequeteMemoire* passé en paramètre. Ne considérez pas un cas où la mémoire est pleine et qu'il faut

remplacer une entrée. On vous demande seulement de coder l'ajout en mémoire. Vous **devez mettre à jour les dates de création et de dernier accès de la nouvelle donnée en mémoire.**

On vous informe maintenant que la politique de remplacement du TLB est FIFO (First In First Out).

- Complétez la fonction *mettreAJourTLB* afin de remplacer l'entrée correctement en selon la politique FIFO. (/4)

ATTENTION
Toutes les structures sont initialisées. Vous n'avez pas à allouer de la mémoire avec *malloc* ou *calloc*.
ATTENTION
Vous avez accès à la taille du TLB avec la macro $TAILLE\_TLB$, à la taille de la mémoire avec $TAILLE\_MEMOIRE$ et à la taille de la table des pages avec la macro $TAILLE\_PT$.
Voici la définition de chaque structure :

```
struct RequeteMemoire {
  u_int8_t estDansTLB;
  u_int8_t estDansTablePages;
  unsigned long adressePhysique;
  unsigned long adresseVirtuelle;
  unsigned long date;
};
```

```
struct SystemeMemoire {
  struct TLB* tlb;
  struct TablePages* tp;
};
```

```
struct TLB {
  unsigned int* numeroPage;
  unsigned int* numeroCadre;
  u_int8_t* entreeValide;
  unsigned long* dernierAcces;
  unsigned long* dateCreation;
};
```

```
struct TablePages {
  unsigned int* numeroCadre;
  u_int8_t* entreeValide;
};
struct Memoire {
  unsigned int* numeroPage;
    unsigned long* dateCreation;
    unsigned long* dernierAcces;
    u_int8_t* utilisee;
};
```