Polytech Paris-Sud Systèmes
Année 2012-2013

TP5

1 Gestion mémoire

Afin de tester le gestionnaire mémoire vu en TD, on utilise un mécanisme général pour détourner les appels de fonction, celui de la variable d'environnement `LD_PRELOAD` (lisez le manuel de `ld.so (8)` pour plus d'informations). Mais avant de pouvoir remplacer totalement la gestion mémoire la libc, il nous faut rajouter les fonctions `calloc` et `realloc` à notre bibliothèque.

Remarque :
Étapes liées à la réallocation mémoire :

- *si le pointeur passé est* `NULL`, `realloc` *doit agir comme* `malloc`
- *si la taille passée est de 0,* `realloc` *doit agir comme* `free`
- *si la nouvelle taille demandée est plus petite ou égale que la taille actuelle :*
  - *on change la taille de la zone utilisée*
- *si la nouvelle taille demandée est plus grande que la taille actuelle :*
  - *il faut effectuer une nouvelle allocation suivie d'une recopie des données et terminer par une libération de la mémoire*

Remarque :
Pour utiliser `LD_PRELOAD`, *dans un autre terminal , tapez la commande :*

```
export LD_PRELOAD=./libmemory.so
```

Question 1 *Récupérer le squelette de la bibliothèque sur*

```
http://www.lri.fr/~mandel/enseignement/systeme/tp-05.tgz
```

et implémenter les fonctions `mcalloc` *et* `mrealloc`. *Lancez des commandes, e.g.,* `sleep 1`, `ls`, `xterm` *(en ordre croissant de complexité) et vérifiez leur fonctionnement.*

Question 2 *Écrire une fonction* `void print_stat ()` *qui affiche des informations sur le tas comme nombre de blocs, le nombre de blocs libres, la différence entre la mémoire allouée et le mémoire utilisée.*

Appeler la fonction `print_stat` *à chaque appel à une fonction de votre bibliothèque de gestion de mémoire.*

Question 3 *Proposer une autre stratégie d'allocation et de libération de mémoire.*

Clonage et mutation de processus

Question 4 *En utilisant les fonctions* `fork()` *et* `exec*()` *écrire un programme ayant le même comportement que le shell lors de l'exécution de la commande suivante :*

```
--> echo Debut; ls -i -l /tmp
```