FDI-DEVC / my_crd

Paramètres

Rendu	https://rendu-git.etna-alternance.net/module-8473/activity-45914/group-918488
Fichiers à rendre	• /Makefile • /*.c • /*.h • libmy/*.c • libmy/Makefile • /Release/my_crd
Nom de l'executable	my_crd
Correction	Soutenance
Environnement	Debian 9.7
Effectif	Seul

Objectifs

Maîtriser le C

Installer une machine virtuelle Debian 9 64bit

Installer un environnement de compilation (build-essential)

Proposer une solution technique à une problématique

Profiler et optimiser la rapidité de traitement d'une solution technique

Attention

- Votre dépôt doit contenir vos sources ainsi qu'un seul Makefile
- Votre Makefile doit générer votre exécutable `my_crd` dans un répertoire `Release` à la racine de votre dépôt : `/Release/my_crd`
- Vous devez réaliser vos propres conteneurs (tableau, liste chaînées, structures, etc)
- L'implémentation de quelconques librairies est interdite
- Votre programme doit être dynamique, le nombre de commandes maximales à exécuter

- Votre programme doit être dynamique, le nombre de commandes maximales à exécuter est indéterminé

Consignes

Le projet my_crd est un projet d'optimisation, la tache à effectuer est simple, mais il va falloir l'executer le plus rapidement possible.

Le programme doit implémenter la totalité des opérations nécessaires pour exécuter les commandes qui lui sont passées :

- Ajouter une entrée
- Supprimer une entrée
- Rechercher une entrée

Le programme ne prend aucun argument. Son fichier de commandes est lu sur l'entrée standard. Les réponses sont fournies sur la sortie standard. Vous n'avez pas à stocker les valeurs entre deux exécutions. En fonction des commandes passées, les réponses diffèrent.

Informations

Fonctions autorisées :

- write
- read
- malloc
- free
- getline

Vous pouvez aussi utiliser le code qui vous ai fourni dans le fichier `my_readline.c`

Commandes

Commande de création :

- si la clé existe, la valeur est mise à jour
- si la clé n'existe pas, la paire clé/valeur est ajoutée
- dans les deux cas, la clé est retournée

Commande de suppression

:

- si la clé existe, sa valeur est retournée
- si la clé n'existe pas, -1 est retourné

Commande de recherche :

- si la clé existe, sa valeur est retournée
- si la clé n'existe pas, -1 est retourné

- si la clé n'existe pas, "-1" est retourné

Format des commandes

Format d'une commande de création :

```
<clé><espace><valeur>\n
```

Format d'une commande de suppression :

```
<clé><espace>D\n
```

Format d'une commande de recherche :

```
<clé>\n
```

Exemple de fichier de commande :

```
32
32 D
34 4
34
34 5
34
34 D
34
```

Sortie attendue :

```
-1
-1
34
4
34
5
5
-1
```

Exemple d'exécution complète :

```
$>./MyCrd < commande-test.lst
-1
-1
34
```

```
4
34
5
5
-1
```

Moulinette de performance

Une moulinette compilera vos projets toutes les 6 heures et les chronométrera lors de l'exécution de fichiers de commandes très volumineux.

A la suite de quoi, un classement des performances de tout les projets sera publié.

Il vous faudra alors faire tout votre possible pour être le plus haut dans le classement.

Améliorez votre projet, jusqu'à devenir le plus rapide de la promotion.

Soutenance

La soutenance sera l'occasion pour vous de nous expliquer :

- le cheminement de vos différents tests pour obtenir le programme le plus rapide
- votre organisation
- la méthode utilisée pour trouver les meilleurs optimisation à réaliser
- vos conclusions sur les algorithmes qu'il est pertinent d'utiliser (ou pas) dans une telle situation.