

INFO-F-201 – Systèmes d'exploitations

Projet de programmation système en C

TinyDB

1 Enoncé

L'ULB veut rénover son système de base de données d'étudiants et fait donc appel à vos compétences en programmation système en C pour reprendre un ancien projet de base de données laissé à l'abandon depuis plusieurs années. Au vu du nombre croissant d'étudiants, la base de données doit être particulièrement rapide et performante, exécutant de ce fait plusieurs requêtes simultanément. Comme vous ne savez pas sur quelle machine va tourner votre programme, **vous ne pouvez utiliser que la librairie standard C et l'API POSIX.**

2 Requirements

2.1 La table d'étudiants

Une base de données se compose de différentes tables liées entre elles par des relations. Dans ce cas-ci, il n'y aura que la table étudiants avec 5 champs différents illustrés ci-dessous.

student	
Champ	Type
id	unsigned int
fname	char[64]
lname	char[64]
section	char[64]
birthdate	struct tm

TABLE 1 – La table d'étudiants avec le type de chaque champ.

NB : le fichier de header `student.h` définit déjà une structure de type `student_t` qui correspond à cette table.

2.2 Lancement du programme

Votre programme s'appellera 'tinydb' et devra accepter un paramètre optionnel, le chemin vers la base de données. Le programme se lance donc de l'une des deux manières ci-dessous.

```
$ ./tinydb <chemin_vers_la_base_de_donnees>  
$ ./tinydb
```

2.3 Comportement du programme

Le programme se présente sous forme d'une ligne de commande qui accepte des requêtes entrées par l'utilisateur sur STDIN comme ci-dessous.

```
$/tinydb
Welcome to the Tiny Database!
Loading the database...
Done!
Please enter your requests.
> select fname=Yannick
logs/1600954212483482624-select.txt
...
```

De la même manière, le programme doit pouvoir se lancer en redirigeant un fichier sur STDIN comme ci-dessous.

```
$/tinydb < queries.txt
Welcome to the Tiny Database! Please enter your requests.
Running query 'select fname=Cedric'
Running query 'update id=88 set lname=Quevrin'
Running query 'select fname=Damien'
...
```

2.4 Resultat des requêtes

Le résultat d'une requête se trouvera dans le répertoire logs, dans un fichier dont la nomenclature est logs/<timestamp>-<type_de_requete>.txt.
Par exemple :

```
$ ls logs/
1600954212483482624-select.txt
1600954212483531264-update.txt
1600954212483538944-delete.txt
1600954212483565312-insert.txt
```

La première ligne du fichier de log résum la requête (requête effectuée, nombre de résultats, durée de la requête, ...); les lignes suivantes sont la liste des résultats comme présenté ci-dessous.

```
Query "select fname=Hadrien" completed in 0.026368ms with 1 results.
864030: Hadrien Legast in section info, born on the 30/9/1998
```

2.5 Format des requêtes

Vous devez gérer quatre types de requêtes dont vous trouverez le format ci-dessous. Des exemples se trouvent dans le répertoire tests/queries.

- `select <champ>=<valeur>`. Cette requête renvoie la liste des étudiants qui correspondent à l'unique filtre `<champ>=<valeur>`.

- `insert <fname> <lname> <id> <section> <birthdate>`. Cette requête insère un nouvel étudiant dans la base de données.
- `delete <champ>=<valeur>`. Cette requête supprime tous les étudiants qui correspondent au filtre donné.
- `update <filtre>=<valeur> set <champ_modifie>=<valeur_modifiee>`. Cette requête modifie tous les étudiants correspondant au filtre `<filtre>=<valeur>`, en donnant la valeur `<valeur_modifiee>` au champ `<champ_modifie>`.

2.6 Fin du programme

Le programme se termine lorsque l'utilisateur appuie sur CTRL+D, sur CTRL+C ou quand le fichier donné en entrée a été lu en entier. Dans tous les cas, le programme se termine *proprement* en sauvegardant le contenu de la base de données sur le disque, comme ci-dessous.

```
> select fname=Yannick
logs/1600954212483482624-select.txt
> ^C
Waiting for requests to terminate...
Comitting database changes to the disk...
Done.
```

3 Ce qui est mis à votre disposition

Vous pouvez télécharger la base du projet sur l'Université Virtuelle afin de ne pas commencer de zéro. Bien qu'il n'y ait pas d'obligation de les utiliser, je vous invite à y jeter un oeil. Ce répertoire contient

- quelques headers et fichiers précompilés pour ne pas partir de zéro
- une base de données de plusieurs milliers d'étudiants dans `students.bin`
- des tests automatiques (cf : pondération)
- un makefile à compléter

4 Critères d'évaluation

Tout d'abord, votre projet doit compiler sans erreur et passer les tests fournis avec la commande `make tests -B`. Si le code ne compile pas, vous recevrez une note de 0/20. Si les tests ne passent pas, votre note sera automatiquement inférieure à 10/20. Ces tests sont basiques et vérifient uniquement que votre base de données fonctionne correctement. D'autres tests plus complexes seront évidemment utilisés. N'hésitez pas à écrire vos propres tests et à nous les fournir.

4.1 Pondération

- Ce projet de programmation système va principalement évaluer votre compétence à manier correctement les outils liés aux systèmes d'exploitation (threads, fichiers, sémaphores, ...) dans le langage C. La pertinence des outils utilisés ainsi que la manière dont ils sont utilisés (trop, pas assez, au mauvais endroit, trop longtemps, ...) sont évalués. **/10**
- Ce projet doit contenir un rapport dont la longueur attendue est d'une à trois pages (max 5). Ce rapport décrira succinctement le projet, les choix d'implémentation si nécessaire, les difficultés rencontrées et les solutions originales que vous avez fournies. **/3**
- Votre code sera aussi évidemment examiné en termes de clarté, de documentation, de commentaires, de structure et de bonnes pratiques. **/5**
- Votre code compile sans warning avec les flags
`-pthread -std=c11 -Wall -Wpedantic` **/2**

5 Remise du projet

Vous devez remettre un fichier zip contenant

- vos sources
- un Makefile
- votre rapport au format PDF
- les tests que vous auriez écrits

N'incluez ni le fichier de base de données de milliers d'étudiants ni vos logs dans le fichier zip!

Vous devez soumettre votre projet sur l'**Université Virtuelle** pour le **6 décembre 2020 23h59** au plus tard.

Retards

Tout retard sera sanctionné d'un point par tranche de 4h de retard, avec un maximum de 24h de retard et devra être soumis sur l'université virtuelle.