

Concepts Avancés de Bases de données

Joaquim LEFRANC et Jérôme Skoda

October 21, 2017

Abstract

Comment avons nous pu écrire autant de chose sur seulement deux boucles imbriqués et une fonction de merge de moins de 20 lignes?

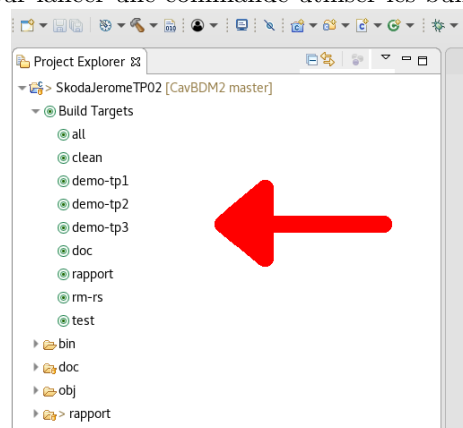
1 Comment compiler le projet

1.1 Avec le terminal

- make all : Compile tout les fichiers
- make test : Lancement de la série de tests automatiques
- make doc : Génération de la documentation (doxygen)
- make rapport : Génération du rapport (latex)
- make clean : Nettoyage du projet (supression des objets et binaires)
- make demo-tp1 : Lancer la démo tp1
- make demo-tp2 : Lancer la démo tp2
- make demo-tp3 : Lancer la démo tp2
- make rm-rs : Supprime le fichier res/RS.txt

1.2 Avec ECLIPSE

Pour lancer une commande utiliser les builds targets.



2 Arborescence

- bin : Binaire exécutable
 - demo : Exécutable de démonstration
 - test : Exécutable de test
- doc : Documentation doxygen sous différents formats
- rapport : Source du rapport
- res : Ressources nécessaires au projet (fichier de bdd)
- script : Script utilisé pour les tests
- src : Source du projet
 - bdd : Source de la bibliothèque
 - demo : Sources des différentes démonstrations d'utilisation
 - test : Sources des différents tests
- sujet.pdf : Sujet du projet
- README.md : Le readme du projet
- rapport.pdf : C'est moi
- refman.pdf : Documentation format pdf

3 Caractéristiques

- Le code est organisé
- Il y a des codes des tests
- Il y a la doc
- Il y a un rapport
- Et il y a plein d'autres choses

4 Démonstration

Les sources de démonstration sont disponibles dans: src/demo Les exécutables de test sont générés dans: bin/demo La commande make pour lancer les démos sont: make demo-tp1, make demo-tp2, make demo-tp3

- tp1-natural-join: Natural join R et S
- tp2-merge-join-without-duplicate: Merge join sans duplication
- tp3-merge-join-with-duplicate: Merge join avec duplication

5 Test unitaire

Les sources de test sont disponibles dans: src/test Les exécutable de test sont générés dans: bin/test Le script de test est dans script/test.sh La commande make pour lancer les tests est: make test

- 0-storeFileBuffer: Ecriture d'un buffer dans un fichier
- 1-natural-join-1: Natural join R et S
- 2-natural-join-2: Natural join S et R
- 3-buf-quick-sort: Fonction de tri d'un buffer
- 4-merge-join-without-duplicate-1: Merge join sans duplication R et S
- 5-merge-join-without-duplicate-2: Merge join sans duplication S et R
- 6-merge-join-with-duplicate-1: Merge join avec duplication R et S
- 7-merge-join-with-duplicate-2: Merge join avec duplication S et R



```
~/Documents/Master2/CavBDM2/tp2
> make test
Script de test automatisé
[PASS] buf-quick-sort
[PASS] merge-join-1
[PASS] merge-join-2
[PASS] natural-join-1
[PASS] natural-join-2
[PASS] storeFileBuffer

Script de test terminé
```

6.10	Référence du fichier <code>src/bdd/mergeJoinWithoutDuplicate.h</code>	13
6.10.1	Documentation des fonctions	14
6.11	Référence du fichier <code>src/bdd/naturalJoin.c</code>	14
6.11.1	Documentation des fonctions	14
6.12	Référence du fichier <code>src/bdd/naturalJoin.h</code>	15
6.12.1	Documentation des fonctions	15
6.13	Référence du fichier <code>src/bdd/quicksort.c</code>	16
6.13.1	Documentation des fonctions	16
6.14	Référence du fichier <code>src/bdd/quicksort.h</code>	17
6.14.1	Documentation des fonctions	17
6.15	Référence du fichier <code>src/demo/tp1-natural-join.c</code>	17
6.15.1	Documentation des fonctions	17
6.16	Référence du fichier <code>src/demo/tp2-merge-join-without-duplicate.c</code>	18
6.16.1	Documentation des fonctions	18
6.17	Référence du fichier <code>src/demo/tp3-merge-join-with-duplicate.c</code>	18
6.17.1	Documentation des fonctions	18
6.18	Référence du fichier <code>src/test/0-storeFileBuffer.c</code>	18
6.18.1	Documentation des fonctions	19
6.19	Référence du fichier <code>src/test/1-natural-join-1.c</code>	19
6.19.1	Documentation des fonctions	19
6.20	Référence du fichier <code>src/test/2-natural-join-2.c</code>	19
6.20.1	Documentation des fonctions	19
6.21	Référence du fichier <code>src/test/3-buf-quick-sort.c</code>	20
6.21.1	Documentation des fonctions	20
6.22	Référence du fichier <code>src/test/4-merge-join-without-duplicate-1.c</code>	20
6.22.1	Documentation des fonctions	20
6.23	Référence du fichier <code>src/test/5-merge-join-without-duplicate-2.c</code>	20
6.23.1	Documentation des fonctions	21
6.24	Référence du fichier <code>src/test/6-merge-join-with-duplicate-1.c</code>	21
6.24.1	Documentation des fonctions	21
6.25	Référence du fichier <code>src/test/7-merge-join-with-duplicate-2.c</code>	21
6.25.1	Documentation des fonctions	21

1 Attention

Ce projet a été conçu pour fonctionner avec `make`

Il est recommandé pour les utilisateurs n'aimant pas le terminal de manipuler le projet uniquement avec **eclipse**

Des *builds targets* sont configurés, il suffit de double cliquer dessus pour lancer la commande `make` correspondante

Commandes

- **make all** : Compile tout les fichiers
- **make test** : Lancement de la série de tests automatiques
- **make doc** : Génération de la documentation (doxygen)
- **make rapport** : Génération du rapport (latex)
- **make clean** : Nettoyage du projet (supression des objets et binaires)
- **make demo-tp1**: Lancer la démo tp1
- **make demo-tp2**: Lancer la démo tp2
- **make rm-rs**: Supprime le fichier `res/RS.txt`

Arborescence

- **bin** : Binaire exécutable
 - **demo** : Exécutable de démonstration
 - **test** : Exécutable de test
- **doc** : Documentation doxygen sous differents formats
- **rapport** : Source du rapport
- **res** : Ressources necessaire au projet (fichier de bdd)
- **script** : Script utilisé pour les test
- **src** : Source du projet
 - **bdd** : Source de la bibliothèque
 - **demo** : Sources des differentes démonstrations d'utilisation
 - **test** : Sources des dufferents tests
- *sujet.pdf* : Sujet du projet
- *README.md* : C'est moi
- *rappot.pdf* : Le rapport du projet

Tests disponibles

-
-
-
-
-
-

2 Navigation

- **bdd** : Source de la BDD
- **demo** : Exemple d'utilisation de la BDD
- **test** : Test unitaire de la BDD

3 Index des classes

3.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

buf	
Structure de donnée représentant un buffer	4

4 Index des fichiers

4.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

src/bdd/buffer.c	5
src/bdd/buffer.h	5
src/bdd/hexdump.c	10
src/bdd/hexdump.h	10
src/bdd/mergeJoinWithDuplicate.c	11
src/bdd/mergeJoinWithDuplicate.h	12
src/bdd/mergeJoinWithoutDuplicate.c	13
src/bdd/mergeJoinWithoutDuplicate.h	13

src/bdd/naturalJoin.c	14
src/bdd/naturalJoin.h	15
src/bdd/quicksort.c	16
src/bdd/quicksort.h	17
src/demo/tp1-natural-join.c	17
src/demo/tp2-merge-join-without-duplicate.c	18
src/demo/tp3-merge-join-with-duplicate.c	18
src/test/0-storeFileBuffer.c	18
src/test/1-natural-join-1.c	19
src/test/2-natural-join-2.c	19
src/test/3-buf-quick-sort.c	20
src/test/4-merge-join-without-duplicate-1.c	20
src/test/5-merge-join-without-duplicate-2.c	20
src/test/6-merge-join-with-duplicate-1.c	21
src/test/7-merge-join-with-duplicate-2.c	21

5 Documentation des classes

5.1 Référence de la structure buf

Structure de donnée représentant un buffer.

Attributs publics

- [char * v](#)
Value: Valeur.
- [size_t s](#)
Size: Taille.
- [size_t c](#)
Count: Nombre de valeur entrée.

5.1.1 Description détaillée

Structure de donnée représentant un buffer.

TP n°: 3

Titre du TP : Merge Join Duplicate

Date : 21/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

5.1.2 Documentation des données membres

5.1.2.1 c

```
size_t buf::c
```

Count: Nombre de valeur entrée.

5.1.2.2 s

```
size_t buf::s
```

Size: Taille.

5.1.2.3 v

```
char* buf::v
```

Value: Valeur.

La documentation de cette structure a été générée à partir du fichier suivant :

- [src/bdd/buffer.c](#)

6 Documentation des fichiers

6.1 Référence du fichier README.md

6.2 Référence du fichier src/README.md

6.3 Référence du fichier src/bdd/buffer.c

```
#include "buffer.h"
#include "hexdump.h"
#include "quicksort.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Graphe des dépendances par inclusion de buffer.c:

6.4 Référence du fichier src/bdd/buffer.h

```
#include <stdio.h>
```

Graphe des dépendances par inclusion de buffer.h: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Fonctions

- struct `buf` * `buf_create` (size_t size)
- void `buf_destroy` (struct `buf` *`buf`)
- char `buf_put` (struct `buf` *`buf`, char value)
- void `storeFileBuffer` (FILE *`fp`, struct `buf` *`buf`)
Enregistre un le 1er caractère de chaque ligne d'un fichier dans un buffer.
- void `writeBufferInFile` (FILE *`fp`, const struct `buf` *`buf`)
- void `buf_quicksort` (struct `buf` *`buf`)
Quick sort buffer.
- void `buf_dump` (const struct `buf` *`buf`)
Hex dump buffer.
- struct `buf` * `storeFileBufferOC` (const char *`file_name`, size_t `buffer_size`)
Ouvre un fichier Stoque son contenu dans un buffer Ferme le fichier.
- char `writeBufferInFileOC` (const char *`file_name`, const struct `buf` *)
Ouvre un fichier Ecrit le contenu du buffer dans le fichier Ferme le fichier.
- char `buf_val` (const struct `buf` *`buf`, int index)
Retourne la valeur à la position d'un buffer.
- size_t `buf_count` (const struct `buf` *`buf`)
Retourne le nombre de caractere enregistré dans le buffer.

6.4.1 Documentation des fonctions

6.4.1.1 `buf_count()`

```
size_t buf_count (
    const struct buf * buf )
```

Retourne le nombre de caractere enregistré dans le buffer.

Paramètres

in	<i>buf</i>	Buffer
----	------------	--------

Renvoie

Nombre de caractere enregistré

6.4.1.2 `buf_create()`

```
struct buf* buf_create (
    size_t size )
```

Creation d'un buffer

Paramètres

in	<i>size</i>	Taille du buffer
----	-------------	------------------

6.4.1.3 buf_destroy()

```
void buf_destroy (
    struct buf * buf )
```

Détruit le buffer

Paramètres

in	<i>buf</i>	buffer à détruire
----	------------	-------------------

6.4.1.4 buf_dump()

```
void buf_dump (
    const struct buf * buf )
```

Hex dump buffer.

Paramètres

in	<i>buf</i>	buffer à trier
----	------------	----------------

6.4.1.5 buf_put()

```
char buf_put (
    struct buf * buf,
    char value )
```

Ajoute un caractere dans le buffer s'il reste de la place

Paramètres

in	<i>buf</i>	buffer d'entrée
in	<i>value</i>	valeur à entrer

Renvoie

0 succès -1 erreur: buffer plein

6.4.1.6 buf_quicksort()

```
void buf_quicksort (
    struct buf * buf )
```

Quick sort buffer.

Paramètres

	<i>buf</i>	Trie le buffer en entrée
in	<i>buf</i>	buffer à trier

6.4.1.7 buf_val()

```
char buf_val (
    const struct buf * buf,
    int index )
```

Retourne la valeur à la position d'un buffer.

Paramètres

in	<i>buf</i>	Buffer
in	<i>index</i>	Index of value

Renvoie

Value

Paramètres

in	<i>buf</i>	Buffer
in	<i>index</i>	Index of value

Renvoie

Value Si -1: Erreur index > count

6.4.1.8 storeFileBuffer()

```
void storeFileBuffer (
    FILE * fp,
    struct buf * buf )
```

Enregistre un le 1er caractère de chaque ligne d'un fichier dans un buffer.

Paramètres

in	<i>Fichier</i>	d'entrée
out	<i>Buffer</i>	de sortie

6.4.1.9 storeFileBufferOC()

```
struct buf* storeFileBufferOC (
    const char * file_name,
    size_t buffer_size )
```

Ouvre un fichier Stoque son contenu dans un buffer Ferme le fichier.

Paramètres

in	<i>File_name</i>	Fichier
in	<i>buffer_size</i>	Taille du buffer, si trop petit pour contenir le fichier entièrement alors le reste du fichier est ignoré

Renvoie

buffer si NULL alors erreur de lecture du fichier

6.4.1.10 writeBufferInFile()

```
void writeBufferInFile (
    FILE * fp,
    const struct buf * buf )
```

Ecrit un buffer dans un fichier

Paramètres

out	<i>fp</i>	fichier de sortie
in	<i>buf</i>	fichier d'entrée

6.4.1.11 writeBufferInFileOC()

```
char writeBufferInFileOC (
    const char * file_name,
    const struct buf * buf )
```

Ouvre un fichier Ecrit le contenu du buffer dans le fichier Ferme le fichier.

Paramètres

in	<i>File_name</i>	Fichier
in	<i>buf</i>	Buffer à écrire dans le fichier

Renvoie

-1 si erreur dans l'ouverture du fichier

6.5 Référence du fichier src/bdd/hexdump.c

```
#include <stdio.h>
```

Graphe des dépendances par inclusion de hexdump.c:

Fonctions

- void [hexDump](#) (char *desc, const void *addr, int len)

6.5.1 Documentation des fonctions**6.5.1.1 hexDump()**

```
void hexDump (
    char * desc,
    const void * addr,
    int len )
```

TP n°: 3

Titre du TP : Merge Join Duplicate

Date : 21/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

6.6 Référence du fichier src/bdd/hexdump.h

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Fonctions

- void [hexDump](#) (char *desc, const void *addr, int len)