

Évaluation de TP sur Machine (1h45) - Index de mots

Contexte

Soit un fichier texte, le but est de construire un index des mots de ce fichier ordonné alphabétiquement en leur associant le nombre de fois où ils apparaissent.

IMPORTANT ! Vous nous rendrez un contrôle TP qui **COMPILE correctement** (et avec des commentaires !), dans lequel est facile de voir ce que vous avez réussi à faire. Les fonctions qui ne fonctionnent pas seront commentées avec la mention : “ne fonctionne pas !” (et éventuellement des explications ...). **Un contrôle TP qui ne compile pas aura au grand maximum la note de 6/20**

1 Préparation et consignes

Avant de commencer, copier le répertoire (entier !) :

```
/home/imaEns/lgonnord/PA/controle2012
```

sur votre compte examen (pas sur le bureau !). Vous travaillerez uniquement dans ce répertoire de votre compte examen (`controle2012`, donc !). Les fichiers situés à l'extérieur de ce répertoire seront ignorés. **Sauvez souvent vos codes !**

Le répertoire contient :

- `text.txt` : un fichier texte exemple.
- `result.txt` : le fichier index correspondant, qui montre ce que votre programme doit produire. *Faites attention à ne pas écraser ce fichier lors de vos travaux!*
- `index.h` : ce fichier contient les spécifications de la SD à implémenter.
- `main.c` : contient le squelette du programme à réaliser.
- `Makefile` : permet de réaliser la compilation séparée de `index+main`.

Structure de donnée index L'index est implémenté sous la forme d'une liste chaînée d'Entry (cf `index.h`). Une Entry est une structure (`word`, `counter`) qui associe à un mot (champ `word`) le nombre de fois (champ `counter`) où il apparaît dans le fichier. L'index doit être ordonné de manière croissante sur les mots.

2 Travail à réaliser

- Écrire dans un fichier nommé `index.c` les fonctions d'entête fournies dans `index.h`. Les spécifications de ces fonctions (ce qu'elles doivent faire!) sont fournies en commentaire dans `index.h`.
- Dans le même temps, tester en complétant le programme `main.c`.
- Votre programme devra compiler en utilisant le Makefile fourni (sans modification).

3 Annexe : le .h fourni

```
/*
 * index.h — PLEASE DO NOT MODIFY AT ALL !
 */

#ifndef __INDEX__
#define __INDEX__

#define WORD_MAX_SIZE 30

#include <stdio.h>

// Data structure
typedef struct entry
{
    char word[WORD_MAX_SIZE];
    int counter;
} Entry;

typedef struct cell
{
    struct cell *next;
    Entry value;
} Cell;

typedef Cell *PtCell, *Index;

// Updates the index with the given word
// —if the word already exists : increments its counter
// —otherwise : adds the new word alphabetically in the index
// and initializes its counter to 1
void update (char * word, Index * ind);

// Builds an index of words from the given FILE pointer
//It is MANDATORY to use the update function
void load (FILE * fp, Index * ind);

// Prints the index on the standard output (terminal)
// each cell is printed on a newline with the following syntax :
//word:counter
void print_index (Index ind);

// Frees the index
void free_index (Index * ind);

#endif
```