Documentation Mini Projet Langage C

Caculli Giorgio, Jedrzej Tyranowski
Haute École de Louvain en Hainaut (HELHa)

13 décembre 2020

Résumé

Documentation pour le projet de Langage procédural sur les listes chaînées. Projet basé sur le concept d'un centre de formations.

Mots-clés : liste, chaînée, c, noeud, centre, formation, tête

Table des matières

1 Introduction

1.1 Le langage C

La langage de programmation utilisé lors du développement et la mise en œuvre du programme est le `ANSI-C`. Les différentes versions du langage disponibles lors du développement de ce programme sont :

- **ANSI-C** : La première vérsion standardisée par le **American National Standard Institute**, abrégé en **ANSI** dans ce document, du langage `C` publiée en 1990.
- **C-99** : Révision de la version ANSI pour permettre aux développeurs d'utiliser les commentaires `//`, les booléans grâce à la librairie `<stdbool.h>`, la déclaration des int directement dans la boucle for, et d'autres modérnisations de la syntaxe.
- **C-11** : Mise à jour du langage `C` pour permettre le support des `thread` afin de pouvoir faire du multi-threading.
- **C-17** : Révision de la version **C-11** qui n'ajoute aucune nouvelle fonctionnalité, mais corriges beaucoup bugs présents dans la version 11.

1.2 Fonctions générales utilisées

1.2.1 Qu'est-ce l'allocation de mémoire dynamique ?

1.2.2 Qu'est-ce `malloc()`

1.2.3 Qu'est-ce `calloc()`

1.2.4 Pourquoi utiliser `calloc()`

1.2.5 Pourquoi utiliser `malloc()`

1.2.6 Syntaxe de `calloc()`

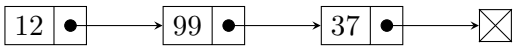
1.2.7 Exemples de `calloc()`

1.2.8 Syntaxe de `malloc()`

1.2.9 Exemples de `malloc()`

1.2.10 Différences entre `calloc()` et `malloc()`

2 Listes chaînées



2.1 Création d'un nouveau nœud

2.2 Insertion d'un nœud dans une liste chaînée

2.3 Suppression d'un nœud d'une liste chaînée

2.4 Affichage d'une liste chaînée

3 Énoncé

4 Programme

4.1 Mode d'emploi

5 Code

5.1 Structures

```c
typedef struct personne
{
  int id;
  char nom[25];
  char prenom[25];
  int formateur;
  int nb_formations;
  int formations[30];
  int nb_jours_indisponible;
  int jours_indisponible[7];
  int reduction;
  int val_reduction;
} personne;
```

```c
typedef struct noeud_db_personne
{
  personne *p;
  struct noeud_db_personne *next;
} noeud_db_personne;
```

```c
typedef struct db_personne
{
  noeud_db_personne *head;
} db_personne;
```

```c
typedef struct noeud_formation
{
  personne *p;
  struct noeud_formation *next;
} noeud_formation;
```

```c
typedef struct formation
{
  int id;
  char nom[40];
  float prix;
  int nb_jours;
  int jours[7];
  float heures[24];
  float durees[10];
  int nb_prerequis;
  int prerequis[10];
  noeud_formation *head;
} formation;
```

```c
typedef struct noeud_db_formation
{
  formation *f;
  struct noeud_db_formation *next;
} noeud_db_formation;
```

```c
typedef struct db_formation
{
  noeud_db_formation *head;
} db_formation;
```

5.2 Fonctions

```
1 personne *creer_personne( char nom[], char prenom[], int formateur );
```

```
1 void afficher_personne( personne *p );
```

```
1 db_personne *creer_db_personne();
```

```
1 void ajouter_db_personne( db_personne *db, personne *p );
```

```
1 int supprimer_db_personne( db_personne *dbp, int id );
```

```
1 void afficher_db_personne( db_personne *db );
```

```
1 personne *get_personne( db_personne *db, char nom[], char prenom[], int formateur );
```

```
1 formation *creer_formation( char nom[], float prix );
```

```
1 int ajouter_formation( formation *f, personne *p );
```

```
1 int supprimer_personne_de_formation( formation *f, int id );
```

```
1 void afficher_formation( formation *f );
```

```
1 db_formation *creer_db_formation();
```

```
1 void ajouter_db_formation( db_formation *db, formation *f );
```

```
1 int supprimer_db_formation( db_formation *dbf, int id );
```

```
1 formation *get_formation( db_formation *dbf, char nom_formation[] );
```

```
1 void afficher_db_formation( db_formation *dbf );
```

```
1 void menu_creer_formation( db_formation *f );
```

```
1 void menu_creer_personne( db_personne *p );
```

```
1 int menu_creer( db_formation *f, db_personne *p );
```

```
1 void menu_ajouter_formation( db_formation *f, db_personne *p );
```

```
1 void menu_supprimer_personne( db_formation *dbf, db_personne *dbp );
```

```
1 void menu_supprimer_formation( db_formation *dbf, db_personne *dbp );
```

```
1 int menu_supprimer_personne_de_formation( db_formation *dbf );
```

```
1 int menu_supprimer( db_formation *dbf, db_personne *dbp );
```

```
1 int menu_affichage( db_formation *f, db_personne *p );
```

```
1 int menu( db_formation *f, db_personne *p );
```

```
1 int main( void );
```

Glossaire

ANSI American National Standard Institute. 3