DHUMES Felix  
GAYTON Mathis

SDD TP1 – Compte rendu

I – Présentation générale

Le but de ce TP est de créer un module de gestion de messages à l’aide d’une liste chainée. La structure de données comporte plusieurs données telles que les dates de début et de fin de validité du message. A l’aide du module, on pourra afficher seulement les messages valides, chercher un motif dans une chaine par exemple.

Description et schéma de la structure de données et des fichiers de données utilisés  
  
Structure d’un message :

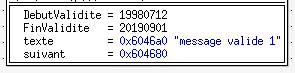
struct message : Chaque message est composé de 4 variables

- DebutValidite : entier représentant la date de début de validité du message sous la forme aaaammjj

- FinValidite: entier représentant la date de fin de validité du message aaaammjj

- texte : pointeur sur une chaine de caractères représentant le message (de taille 100 au maximum)

- suivant : variable de type message \* qui est un pointeur sur le prochain message



struct liste :

- premier : variable de type message \* qui est un pointeur sur le premier message de la liste chainée. C’est la tête réelle de la liste.



Fichiers utilisés :

Les fichiers d’entrés et de sorties possèdent des messages et sont de la forme suivante :

Sur chaque ligne on retrouve un message (sous sa forme de structure de donnée) avec sa date de début de validité, sa date de fin de validité et le message. Chacune de ces données sont séparées par un espace.

|  |
| --- |
| 20110429 20121221 message 1  19980712 20190901 message 2  20241231 20250705 message 3 |

Organisation du code source :

listeChainee.h : Contient les librairies nécessaires, les deux structures de données (message et liste), les constantes (#define) pour la taille d’un message et la taille d’une date, et les prototypes des fonctions de listeChainee.c

listeChainee.c : Contient l’implémentation des fonctions de listeChainee.h

message.h : Contient les librairies nécessaires et les prototypes de message.c

message.c : Contient l’implémentation des fonctions de message.h

fichier.h : Contient les librairies nécessaires et les prototypes de fichier.c

fichier.c : Contient l’implémentation des fonctions de fichier.h

main.h : Contient les librairies nécessaires.

main.c : Contient les tests des fonctions.

Makefile : Permet la compilation du programme et crée l’exécutable sous le nom de main

Fonctions contenues dans chaque fichier :

listeChainee.h

Void InitLCH(liste\_t\* liste);

message\_t \* CreerCellule(int DebutValidite, int FinValidite, char \* texte);

void InsertionApres(message\_t \* message, message\_t \*\* prec);

message\_t \*\* RecherchePrec(liste\_t \* liste, int valeur);

void SupprimerCellule(message\_t \*\* prec);

void ParcoursLCH(liste\_t \* liste);

void SupprimerLCH(liste\_t \* liste);

FAIRE DES DESCRIPTIONS DES FONCTIONS ?

message.h

void AfficherMessagesValides(liste\_t \* liste);

int RecupererDateSysteme();

void SupprimerMessagesObsoletes(liste\_t \* liste);

void ModifierDateDebut(liste\_t \* liste, int DateInitiale, int NouvelleDate);

void MessagesContenantMotif(liste\_t \* liste, char \* motif);

int ContientMotif(char \* s, char \* motif);

fichier.h

void LectureFichier(char \* NomFichier, liste\_t \* liste);

void suprRN(char\* s, int tailleMax);

void SauvegardeLCH(char \* NomFichier, liste\_t \* liste);

Pour lancer le programme, il faudra faire, après avoir entré les fonctions désirées dans le fichier main.c et fait make :



Si on veut pouvoir enregistrer dans un autre fichier les messages, il faudra préciser un deuxième nom de fichier :



Code Source :   
COPIER COLLER DU CODE

listeChainee.h :

#ifndef LISTE\_CHAINEE\_H

#define LISTE\_CHAINEE\_H

#include <stdlib.h>

#include <stdio.h>

|  |
| --- |
|  |

#include <string.h>

#define TAILLE\_TEXTE 100

#define TAILLE\_DATE 9

typedef struct message {

int DebutValidite;

int FinValidite;

char \* texte;

struct message \* suivant;

} message\_t;

typedef struct liste {

struct message \* premier;

} liste\_t;

void InitLCH(liste\_t \* liste);

message\_t \* CreerCellule(int DebutValidite, int FinValidite, char \* texte);

void InsertionApres(message\_t \* message, message\_t \*\* prec);

message\_t \*\* RecherchePrec(liste\_t \* liste, int valeur);

void SupprimerCellule(message\_t \*\* prec);

void ParcoursLCH(liste\_t \* liste);

void SupprimerLCH(liste\_t \* liste);

#endif // LISTE\_CHAINEE\_H

listeChainee.c :

Compte rendu d’éxécution :   
  
Makefile :   
  
all: main.c listeChainee.c fichier.c message.c

gcc -g -Wall -Wextra -Wpedantic -ansi -Wno-endif-labels -o main main.c listeChainee.c fichier.c message.c

clean: rm main