

Gestion d'une bibliothèque

Remarque : Ce TP se déroulera en 5 parties. Les 3 premières parties sont notées pendant la séance. Les parties IV et V doivent être terminées et déposées sous moodle

Objectifs

- Manipulation des struct
- Manipulation des fonctions
- Utilisation de pointeurs

Compétences attendues

- Compilation séparée et modularité
- Faire des tests
- Débogage d'une application
- Faire un compte-rendu

Rappels sur la compilation séparée

Les grands projets informatique font appel à des équipes de développement. Cela amène naturellement à découper le développement en différents modules qui sont confiés à chaque membre de l'équipe. Cela amène à considérer 3 catégories de fichiers dans une application
le fichier de déclaration de la structure de données manipulées par le module que nous nommerons de manière générique <modulei>.h,
le fichier de définition des traitements du module que nous nommerons <modulei>.c
le fichier relatif au programme principal que nous désignerons <principal>.c.

Après l'édition de ces fichiers nous devons compiler <modulei>.c par la commande

gcc -c <modulei>.c

Le résultat est le fichier <modulei>.o

Remarque : Bien faire attention à l'option de compilation -c qui limite l'exécution de gcc à mettre en oeuvre les étapes de pré-processing et de compilation. Le fichier « .o » n'est généré que s'il n'y a pas d'erreurs

On peut ensuite construire l'application exécutable en utilisant les fichiers objets obtenus à partir de chaque module. On exécute alors la commande :

gcc <modulei>.o ... <modulek>.o <principal>.c -o <principal>.exe

On notera l'utilisation de l'option « -o » qui demande à gcc d'exécuter les 3 étapes de son fonctionnement : pré-processing, compilation et édition de liens.

Remarque : On peut compiler le fichier <principal>.c avec la commande « gcc -c <principal>.c » avant l'option de l'ensemble des modules. Cela permet de corriger les erreurs de compilation. Mais ensuite, pour l'obtention du programme principal, il faut repartir du fichier source <principal>.c et non pas du fichier objet.

Enonce du sujet du TP7

La bibliothèque RAPTOUT vous a contacté pour lui construire une application lui permettant de gérer ses livres. Chaque livre est caractérisé par une **fiche** contenant les informations suivantes :

- Le **titre** du livre,
- Le **nom de l'auteur** du livre,
- Le **nom de l'éditeur** du livre,
- L' **année** de publication livre.

Comme plusieurs livres peuvent avoir le même auteur ou le même éditeur, chaque livre sera caractérisé de manière unique par un **code** livre.

Les livres sont empruntés par des membres caractérisés par le champ **emprunteur**.

L'application devra permettre :

1. d'ajouter un nouveau livre dans le répertoire (fonction **ajouter**),
2. de modifier **l'emprunteur**,
3. de supprimer un livre (fonction **supprimer**),
4. de rechercher un livre en fonction de son code (fonction **rechercher**),
5. de lister tous les livres ayant le même auteur(fonction **editerauteur**)
6. d'afficher à l'écran le nom de tous les livres du répertoire (fonction **lister**),
7. d'éditer toutes les fiches des livres du répertoire (fonction **editerall**).

Travail à réaliser

- I. Analyser le cahier de charges de l'application pour définir la structure de donnée nécessaire pour l'implémenter (15mn).
- II. Définir un graphe d'appel des traitements. On cherchera à factoriser un maximum de traitements élémentaires. (30mn)
- III. Ecrire le squelette de l'ensemble des traitements (30mn)
- IV. Coder l'application de manière à tester au fur et à mesure chaque traitement identifié à l'étape 2. (2h)
- V. Réaliser le compte-rendu de l'application (30mn)