

TP2 Listes contiguës, fichiers

Objectifs

- Savoir utiliser les listes contiguës en C.
- Utiliser les opérations de lecture et d'écriture sur les fichiers en C.
- Utiliser les arguments de la ligne de commande.

Contexte et préparation Le contexte de ce TP est le même que celui du TP1, à savoir une application de gestion d'annuaire. Pour les manipulations de fichiers et les arguments du main, on pourra se reporter au cours (!).

IMPORTANT ! Ce TP utilise les types de données et les fonctions du TP1.

Avant de commencer, donc :

- Créer un répertoire TP2 dans votre répertoire de PA et y copier le fichier annuaire `annu.txt` et votre `.c` du TP précédent.
- Récupérer sur le compte web polytech de Laure Gonnord, le fichier `fichiers_en_c.txt` présent à l'adresse

`~lgonnord/PA2012/TP2/`

et le mettre dans ce même répertoire. Ce fichier, écrit par B. Carré, vous montre quelques exemples de manipulation des fichiers en C.

1 Questions du TP (à faire impérativement)

Toutes les questions seront testées au fur et à mesure (appels dans le main), *comme d'habitude!*

1.1 Chargement à partir d'un fichier

La version demandée dans le précédent TP ne permet pas les opérations interactives sur l'annuaire, l'entrée standard étant redirigée pour le chargement des données. Pour libérer l'entrée standard, nous allons donc manipuler explicitement le fichier de données en C.

1. En vous inspirant des exemples fournis, écrire une nouvelle fonction de chargement de l'annuaire qui permet de lire les données à partir d'un fichier texte dont le nom est passé en paramètre. Tester en passant directement le nom "annu.txt" en argument de la fonction.
2. Utiliser les paramètres de la fonction `main` (`argc`, `argv`) pour vérifier que l'argument a été correctement fourni au lancement et récupérer le nom du fichier de données passé en argument. À ce stade, le lancement de votre programme `annuaire` doit se faire de la façon suivante :

```
./annuaire annu.txt
```

1.2 Recherche dans l'annuaire

Pour cette partie, il faut que l'annuaire soit trié selon les noms des personnes. Si vous n'avez pas écrit de fonction de tri dans le premier TP, modifiez le fichier `annu.txt` pour qu'il soit trié.

Remarque : la recherche dichotomique dans un tableau trié est un algorithme classique vu en TD.

1. Écrire une fonction `rechDicho` qui prend en paramètre un nom de personne et un annuaire, effectue une recherche dichotomique de cette personne dans l'annuaire et retourne l'indice de rangement de la personne dans la liste contiguée si elle est présente, -1 sinon.
2. Tester la fonction `rechDicho` en demandant un nom de personne à l'utilisateur et en affichant après le retour de la fonction les informations trouvées (nom, prénom, date de naissance, numéro de téléphone) ou "inconnu" sinon.

1.3 Modification de l'annuaire et sauvegarde dans un fichier

1. Écrire une fonction qui donne la possibilité à l'utilisateur de modifier le numéro de téléphone d'une personne recherchée dans l'annuaire comme précédemment. Tester la fonction.
2. L'annuaire pouvant être modifié, écrire une fonction permettant de sauvegarder l'annuaire dans un (autre) fichier texte, selon le même format que celui du fichier donné en argument.

IMPORTANT ! Le TP3 aura besoin des déclarations et fonctions suivantes écrites lors des deux séances TP1 et TP2 :

- déclarations des types
- initialisation de la structure à annuaire l'aide d'un fichier
- impression de l'annuaire
- recherche d'un nom dans l'annuaire.

Il est obligatoire d'avoir écrit ces fonctionnalités sous peine de ne pouvoir aborder le TP3.

2 Questions s'il vous reste du temps

On va faire un joli menu pour l'application annuaire :

1. Écrire une fonction menu qui propose à l'utilisateur les différentes fonctionnalités possibles de l'application (affichage de l'annuaire, recherche d'une personne, modification du numéro de téléphone d'une personne donnée, quitter l'application avec ou sans sauvegarde de l'annuaire) et qui retourne le choix fait par l'utilisateur.
2. Modifier la fonction main de l'application pour afficher le menu utilisateur et effectuer le traitement correspondant au choix de l'utilisateur.