

TP3 Allocation dynamique de mémoire

Objectifs

- Savoir utiliser les listes contiguës en C.
- Savoir utiliser les fonctions d'allocation dynamique de mémoire pour adapter la taille des listes contiguës au contexte.

Contexte et préparation Le contexte de ce TP est le même que celui des 2 TP précédents, à savoir une application de gestion d'annuaire.

L'application réalisée jusqu'à présent permet les ajouts / retraites interactifs de personnes dans l'annuaire. Le nombre de personnes dans l'annuaire peut donc être très variable. Dans l'application réalisée, l'annuaire est représenté par une liste contiguë dont l'espace de stockage est fixé par l'application. La liste contiguë permet les opérations d'ajout mais dans la limite de la taille de l'espace de stockage (au pire, aucune opération d'ajout n'est possible si la taille de l'espace de stockage correspond exactement au nombre de personnes rangées dans l'annuaire).

Afin d'optimiser la gestion de l'espace de stockage de l'annuaire, nous allons effectuer de l'allocation dynamique de mémoire.

Pour réaliser ce TP, vous avez besoin des fonctions des deux TP précédents. Avant de commencer, donc :

- Création d'un répertoire TP3, récupération de `annu.txt`
- Copier votre fichier `.c` du tp2.

1 Questions du TP (à faire impérativement)

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

Allocation dynamique "initiale"

1. Modifier le fichier de données de façon à ce que soit fourni sur la première ligne du fichier, le nombre de personnes rangées dans l'annuaire.
2. Modifier la définition du type `annuaire` de façon à ce que le champ `espace` de la liste contiguë soit alloué dynamiquement. (ie : le champ `espace` n'est plus de type `Tableau de Personne` mais de type `Pointeur de Personne`). Que doit-on modifier dans les fonctions pour que le programme compile/marche encore ?
3. Écrire une fonction `allocAnnuaire` qui reçoit en paramètre une structure `annuaire`, un entier `nb` et qui alloue dynamiquement l'espace de stockage de la liste contiguë de taille $(nb+EXT)*sizeof(Personne)$ où `EXT` est une constante du programme, fixant la taille de la zone d'extension de l'annuaire.
4. Tester cette fonction dans le main en appelant cette fonction avec `nb` le nombre récupéré sur la première ligne du fichier de données.

Remarque : Ainsi, quel que soit le nombre `nb` de personnes rangées dans l'annuaire, l'utilisateur aura possibilité d'ajouter `EXT` personnes.

5. Modifier la fonction de sauvegarde de l'annuaire de façon à écrire sur la première ligne du fichier le nombre de personnes que contient l'annuaire.

2 Questions s'il vous reste du temps

Allocation dynamique "à la demande" On peut améliorer la gestion de l'espace mémoire : dans la version précédente, la fonction `allocAnnuaire` alloue dynamiquement un espace mémoire de taille `(nb+EXT)*sizeof(Personne)`. On souhaite dans cette 2ème étape que l'espace alloué dynamiquement ne soit plus un Tableau de `(nb+EXT) Personne` mais un Tableau de `(nb+EXT) Pointeur de Personne!`. Ainsi, pour n personnes contenues dans l'annuaire, il y aura exactement n structures en mémoire par rapport à `nb+EXT` dans tous les cas avec la représentation précédente. Ceci réclame d'allouer (resp. libérer) dynamiquement les structures `Personne` lors du chargement et de l'ajout (resp. la suppression) de personnes.

Après avoir fait une sauvegarde de votre fichier `C`, modifiez l'application de façon à réaliser cette gestion mémoire.