

R2.04 : programmation bas niveau en langage C

TP2 bis – allocation dynamique

IUTLCO - Département Info – BUT Informatique

But du TP :

Il consiste à manipuler :

- les pointeurs
- l'allocation dynamique

Le programme ne gère pas le fait que le contenu des fichiers textes lus puisse être différent de ce qui est attendu. De même pour les entrées clavier de l'utilisateur.

Ce TP est à déposer sur Moodle à l'issue de la séance.

Explications

Ce TP fait suite au TP2 et poursuit le développement du projet **Tab foot**.

La version actuelle de la structure de donnée d'équipe de foot décrite dans le fichier *equipe.h* utilise exclusivement l'allocation statique des données, c'est à dire que les tailles de tous les éléments de la structure sont définies dans le code source avant compilation.

Dans la structure *equipe*, le nom de l'équipe contient obligatoirement 50 caractères et le tableau de joueurs est composé obligatoirement de 50 joueurs. Dans la structure *joueur*, le nom contient 30 caractères et le prénom, 50.

En fait, quelle que soit l'équipe qui sera lue, son occupation mémoire sera toujours la même.

Dans la suite, tous les tableaux statiques vont être remplacés par des tableaux dynamiques afin de n'occuper que le minimum d'espace mémoire requis.

Question 1 – nouveau module

Afin de tester l'allocation dynamique en même temps que l'allocation statique à des fins de comparaison, nous allons ajouter un module complémentaire au programme.

Dupliquer les 2 fichiers *equipe.h* et *equipe.c* dans *equipe_alloc_dyn.h* et *equipe_alloc_dyn.c*.

Il faut ensuite pouvoir différencier les types et fonctions de ce nouveau module par rapport à l'autre. Dans **les 2 nouveaux fichiers** :

1. supprimer les 2 fonctions *définir_capitaine* et la fonction *trouver_maillot_joueur*
2. ajouter *AllocDyn* à la fin des types (*joueur*, *ptrjoueur*, *equipe*, *ptrequipe*) partout où ils apparaissent : *joueurAllocDyn*, *ptrjoueurAllocDyn*, *equipeAllocDyn*, *ptrequipeAllocDyn*.
3. Ajouter *_alloc_dyn* à la fin de tous les noms de fonctions : *lire_equipe_fichier_alloc_dyn*, *lire_equipe_clavier_alloc_dyn*, ...
4. Au début de *main.c*, inclure en plus la nouvelle bibliothèque *equipe_alloc_dyn.h*
5. dans *Makefile* ajouter *equipe_alloc_dyn.o* dans les 2 lignes de la cible *foot* en haut et dans la cible *clean* en bas, puis ajouter les 2 lignes supplémentaires :
equipe_alloc_dyn.o : equipe_alloc_dyn.c equipe_alloc_dyn.h
gcc -c equipe_alloc_dyn.c -Wall
6. Recompiler le tout, il ne doit pas y avoir d'erreur

Question 2 – modification de la sdd et de la lecture du fichier

Nous avons vu en cours qu'un tableau est déclaré de manière statique par :

type tableau[taille] ;

exemple : *int tab[10] ;*

Rappel : tab est l'adresse de la première case du tableau.

La taille du tableau est écrite dans le code et ne peut donc pas changer à l'exécution.

Pour obtenir le même tableau mais avec une taille dynamique il faut utiliser la notation pointeur :

*type *tableau ;*

exemple : *int *tab ;*

Le tableau n'existe pas encore et une zone mémoire devra être allouée pour y stocker les valeurs mais cette fois, la taille peut être issue d'une variable du programme éventuellement demandée à l'utilisateur.

int taille = 10 ; // ou lu à partir du clavier avec un scanf, ou calculée, etc ...

*tab=malloc(taille * sizeof(int)) ; // allocation mémoire de 10 * la taille d'un entier (en octets)*

Dans *equipe_alloc_dyn.h*, remplacer les 4 tableaux des 2 structures par des notations pointeur.

Dans *equipe_alloc_dyn.c*, modifier *lire_equipe_fichier_alloc_dyn* afin d'allouer dynamiquement ces 4 tableaux :

1. le fscanf du nom de l'équipe doit être remplacé par 3 lignes :
 - un fscanf dans une variable chaîne de caractères temporaire de taille fixe (100 par exemple)
 - un malloc de *e→nom* dont la taille doit permettre de contenir autant de caractères que le nom lu dans la variable temporaire plus 1 caractère pour la fin de chaîne. La taille d'une chaîne de caractère est obtenu grâce à la fonction *int strlen(char *cc)* issue de la bibliothèque *string.h*.
 - la recopie de la variable temporaire dans le nom
2. nbjoueurs n'est pas un tableau donc lu normalement par le fscanf.
Par contre il doit être suivi de l'allocation du tableau de joueurs *e→joueurs* qui doit contenir exactement le nombre de joueurs requis
3. dans la boucle, le nom et le prénom du joueurs devront être alloués donc suivre la même procédure que pour le nom de l'équipe.

A partir du moment où la structure est correctement allouée, elle peut être utilisée comme la structure statique donc la fonction d'affichage *afficher_joueurs_alloc_dyn*, par exemple, doit fonctionner sans modification.

Dans *main.c*, définir une nouvelle variable *eqFootAllocDyn* pour stocker une équipe de type *equipeAllocDyn*.

Après la lecture du fichier dans la structure statique (*lire_equipe_fichier*), lire le même fichier dans la structure dynamique (*lire_equipe_fichier_alloc_dyn*), et enfin afficher cette équipe. Vérifier que tout fonctionne bien (affichage des mêmes joueurs évidemment).

Question 3 – vérification de la mémoire occupée

Vérifions maintenant l'occupation mémoire des 2 équipes *eqFoot* et *eqFootAllocDyn*.

Suite à la lecture de l'équipe *eqFoot*, afficher sa taille qui sera récupérée par *sizeof(eqFoot)* qui renvoie le nombre d'octets utilisés. Attention *sizeof* retourne un *long long unsigned int* et il faut utiliser le format *%llu* dans *printf*.

Afficher la taille de *eqFootAllocDyn* est plus complexe. En effet, *sizeof(eqFootAllocDyn)* va donner la taille de la structure *equipeAllocDyn* qui contient 2 pointeurs et un entier, mais ne récupère pas l'occupation mémoire des données pointées (les caractères du nom de l'équipe et les joueurs du tableau *joueurs*).

Écrire dans *equipe_alloc_dyn.c* une nouvelle fonction *occupation_memoire_equipe_alloc_dyn* qui calcule l'occupation mémoire d'une équipe allouée dynamiquement et retourne le résultat final. Pour cela vous devrez cumuler la taille de toutes les données des tableaux dynamiques et donc refaire les calculs qui avaient été fait dans chaque *malloc*.

Après lecture de l'équipe « LOSC.txt » dans *eqFoot* et dans *eqFootAllocDyn* vous devez obtenir les occupations mémoires suivantes :

```
8 Moussa SOW
9 Tulio DE_MELO
12 Cedric BASEYA
17 Pierre-Alain FRAU
27 Yao_Kouassi GERVAIS

Occupation memoire :
eqFoot : 4256 octets
eqFootAllocDyn : 973 octets
```

L'occupation mémoire est nettement inférieure, c'est l'un des avantages des langages de bas niveau : maîtriser l'occupation mémoire, mais au prix d'une manipulation plus « lourde ».

Question 4 (optionnelle) – modification de lecture au clavier

De la même manière que pour la question 1, modifier la lecture d'une nouvelle équipe au clavier dans une structure dynamique : *lire_equipe_clavier_alloc_dyn*.

Question 5 (optionnelle) – menu

Organiser le code de votre fonction *main* dans un menu permettant de tester facilement les différentes questions des TP2 et TP2bis.