

Structures de Données TP2

Stéphane Marchand-Maillet, B. Pulfer, A. Freeman, H. Haldi¹

¹Département d'Informatique UNIGE

Mars 2023

But

Le but de ce TP est de vous faire modéliser un problème concret en termes de structures de données et vous faire implémenter des programmes les manipulant en C.

Pour les schémas, diagrammes d'activité et fiches demandées, utilisez [nomnoml](#) en vous basant sur l'exemple fourni sur ce site.

Quand il s'agit d'implémenter, veuillez vous référer aux diapositives données aux exercices sur le C. **Un squelette d'implémentation est disponible sur moodle à cet effet.**

1 Checklist d'Avion

1.1 Problème

Notre compagnie aérienne utilisait une checklist en papier afin de vérifier que tout soit bon dans un avion avant de décoller. La checklist d'un avion est une liste de tâches donnée sous forme imbriquée. Chaque tâche appartient à une catégorie et à une sous-catégorie, menant à une liste imbriquée :

- Vérification du cockpit
 - Vérification du poste pilote
 - * Vérifier le fonctionnement de la manette A
 - * Vérifier le fonctionnement du bouton B
 - Vérification du poste co-pilote
 - * Vérifier le bouton C
 - * Vérifier le bouton D
- Vérification du fuselage
 - ...

Le faire au papier est long et inefficace et brûle des arbres, les ours polaires s'étant plaint, la compagnie décide de le faire par un programme informatique.

Le but est de faire un programme d'aide au déroulement de la checklist. Ce programme doit lire la liste ligne par ligne et permettre de valider les différentes tâches à effectuer. Au cas où une tâche n'est pas validée par l'opérateur (e.g., le bouton D ne s'allume pas), toutes les tâches de la **sous-catégorie** en cours (e.g., Poste co-pilote) doivent être réévaluées. Une sous-catégorie peut avoir un nombre *arbitraire* de tâches pas forcément égal aux autres.

L'utilisateur doit pouvoir entrer un entier 1 ou 0 afin d'indiquer qu'une tâche est évaluée ou pas, afin de ce faire, utilisez la fonction `scanf()` comme illustrée dans les diapositives sur l'Introduction au C à la diapositive 7.

1.2 Modélisation et Implémentation

1. Décrire la structure de données à utiliser afin de réaliser ce programme.
2. Donner le diagramme d'activité complet du programme utilisant cette structure de données.
3. Implémentez le programme. **Basez vous sur le squelette.**

2 Les sièges

2.1 Problème

Dans les avions de notre compagnie, l'affectation des sièges se fait *premier arrivé premier servi*.

1. Donner la structure intermédiaire à utiliser dans ce cas pour simuler notre salle d'attente.
2. A partir de la modélisation des structure nécessaires, proposer le diagramme d'une implémentation qui, à chaque temps, choisi 2 entiers (entre 1 et 5) aléatoirement représentant le nombre de passagers entrants et sortants de notre salle d'attente, fixez un maximum de nombre de passagers présents dans la salle à 10.
3. Réaliser cette implémentation, faites des `usleep()` entre chaque modification de la structure afin de simuler une execution en temps réel. **Basez vous sur le squelette.**

```
#include <unistd.h>
/*...*/
while(1) {
    /*...*/
    // retirer passagers
    usleep(3000000); // arrête le programme pendant 3 secondes.
    /*...*/
    //ajouter passagers
    /*...*/
    usleep(3000000); // u => microsecondes.
}
```



Figure 1: L'An-225 transportant la *Buran* Soviétique en 1989

3 Liste chaînée

Une liste chaînée est une structure de données dynamique qui permet d'ajouter des éléments, d'en supprimer, de voir à quelle position se trouve un élément, etc.

1. Implementez la liste chaînée. La liste doit, en principe, pouvoir contenir n'importe quel type d'élément (par exemple, des personnes, des voitures, ...). **Basez vous sur le squelette.**