Projet d'algorithmique

Cours d'Informatique de Deuxième Année

—Deug Sciences S3-MI—

Simulation d'une bibliothèque

L'objet du projet est de simuler une journée dans une bibliothèque. Vous devrez gérer les livres, les emprunts, les inscriptions et la file d'attente au guichet de la bibliothèque.

- ce travail doit être rendu au plus tard le lundi 4 janvier 1999 au bureau 201 à l'IFI (aile B). La soutenance aura lieu lors de la dernière séance de Travaux Pratiques (le jeudi 7 janvier 1999) ;
- vous devrez vous inscrire sur la liste de soutenance en même temps que vous rendrez le projet ;
- la clarté et le soin apportés à la rédaction de votre travail, ainsi que la lisibilité de votre *listing* seront pris en compte ;
- ce travail peut être réalisé en binôme, chaque personne devant être présente à la soutenance et pouvant être interrogée individuellement ;
- la partie programmation devra être réalisée en *C ANSI*.

---

1 Idée générale, définitions

Une *bibliothèque* est un lieu où sont stockés des livres que l'on peut emprunter, à condition d'être inscrit dans le fichier de la bibliothèque. Les livres empruntés le sont pour une durée déterminée, et un inscrit ne peut pas avoir emprunté plus d'un certain nombre constant `MAXEMPRUNTS` de livres à un instant donné. Pour emprunter ou rendre un livre, il faut faire la queue au guichet de la bibliothèque.

1.1 Le stock de livres

Nous supposerons pour simplifier que le stock de livres de la bibliothèque n'évolue pas : la bibliothèque n'achète pas de livres, et ces derniers ne disparaissent pas. Nous fixerons le nombre de livres à une constante `MAXLIVRES`. Toujours par soucis de simplification, nous référerons toujours un livre par sa cote, mettons un entier compris entre 0 et `MAXLIVRES` $- 1$. Tous les livres se trouvent dans le fichier des livres de l'ordinateur de la bibliothèque. Ce fichier indique, pour chaque livre, s'il est emprunté, et par qui : à chaque cote de livre est associé le numéro d'étudiant de l'emprunteur s'il a été emprunté. On utilisera un numéro spécial pour indiquer que le livre est disponible.

1.2 Le fichier des inscrits

Le fichier des inscrits contient, pour chaque inscrit :

- son numéro d'étudiant,
- ses nom et prénom,

- pour chaque livre emprunté, sa cote et la date d'emprunt,
- une date limite d'interdiction d'emprunt éventuelle, utilisée pour pénaliser l'étudiant qui a rendu un ou plusieurs livres en retard.

1.3 La file d'attente au guichet

Pour emprunter ou rendre un livre, il faut faire la queue au guichet de la bibliothèque. Un étudiant qui s'insère dans la queue le fait toujours en dernière position. Quand il s'insère dans la queue, il a déjà décidé des actions qu'il va faire au guichet :

- emprunter un livre dont il connaît déjà la cote,
- rendre un livre,
- s'inscrire,
- se désinscrire.

Au guichet on peut faire plusieurs de ses quatre opérations, mais on ne peut faire une opération donnée qu'une fois : par exemple, on peut rendre des livres, se désinscrire, se réinscrire et emprunter, mais on ne peut pas s'inscrire deux fois. Au niveau des actions, l'emprunt de plusieurs livres est considéré comme une seule action. Idem pour rendre des livres.

Pour chaque entité de la queue on a donc les informations suivantes :

- son numéro d'étudiant,
- les actions à faire au guichet,
- si on veut rendre des livres, les cotes des livres à rendre,
- si on veut emprunter des livres, les cotes des livres à emprunter.

Pour simplifier on supposera dans un premier temps que les gens dans la queue ne sont pas impatients : la gestion des impatients est en option au projet (voir la section "Pour en faire plus"). La personne qui se trouve au guichet de la bibliothèque se nomme Julie. Julie s'occupe toujours de l'étudiant en première position dans la queue.

Nous décrivons maintenant les différentes actions possibles au guichet.

1.3.1 S'inscrire

Quand un étudiant s'inscrit à la bibliothèque, Julie lui demande les informations nécessaires pour renseigner son fichier des inscrits. Elle réalise alors la saisie de ces informations au clavier de son ordinateur. Au niveau du projet, ce renseignement du fichier des inscrits de traduira par une saisie réelle.

1.3.2 Se désinscrire

Un étudiant inscrit peut décider de se désinscrire. Avant de se désinscrire il devra rendre tous les livres qu'il a empruntés. On ne peut se désinscrire en période de pénalité. La désinscription se traduit par la disparition de l'étudiant dans le fichier des inscrits. Après une désinscription il faut refaire la queue si on souhaite s'inscrire à nouveau.

1.3.3 Emprunter un livre

Pour emprunter un livre il faut être inscrit. Un étudiant ne peut avoir emprunté plus de `MAXEMPRUNTS` livres à un instant donné, et on ne peut pas emprunter de livres si on est en période de pénalité. Un étudiant ne peut convoiter plus de `MAXEMPRUNTS` livres au guichet, non compris ceux qu'il a déjà empruntés. Par exemple, si `MAXEMPRUNTS` vaut 3, un étudiant qui possède déjà deux livres peut en demander au maximum 3 au guichet : c'est à Julie de faire en sorte de ne lui en donner qu'un au plus.

Bien entendu, on ne peut avoir emprunté deux fois le même livre à un instant donné. Une personne au guichet qui veut rendre des livres et en emprunter doit d'abord rendre ses livres. Une personne ne peut emprunter un livre qu'elle vient juste de rendre : si elle veut l'emprunter à nouveau elle doit refaire la queue. Un étudiant en période de pénalité ne peut pas emprunter.

1.3.4 Rendre un livre

On peut rendre un livre sans être inscrit : on n'oubliera pas, dans ce cas, de vérifier si le livre a été rendu en retard ou non, et de pénaliser l'emprunteur du livre si c'est le cas.

1.3.5 Le calcul des pénalités

Un étudiant qui à emprunté un livre et qui le rend plus de `DUREEEMPRUNT` jours après a une pénalité de `PENALITE` jours, en plus des pénalités qu'il avait déjà pour les autres livres qu'il a rendus en retard le cas échéant. Par soucis de simplification nous supposerons que tous les mois ont 30 jours.

2 Exemple

Supposons `MAXLIVRES` > 5, `MAXEMPRUNTS` $= 3$, `DUREEEMPRUNT` $= 15$ jours, qu'il n'y ait que deux inscrits à la bibliothèque et que la date du jour soit le 10/11/98 :

Num étudiant: 7
Prénom Nom: Jean Laquille
1er livre emprunté: 3 le 1/11/98
2ème livre emprunté: 5 le 1/11/98
3ème livre emprunté: 0 le 1/11/98
Pénalité: rien

Num étudiant: 4
Prénom Nom: Camille Dutour
1er livre emprunté: 4 le 1/10/98
2ème livre emprunté: 1 le 1/10/98
3ème livre emprunté: non
Pénalité: encore 10 jours

La queue est composée de 3 étudiants, dont les numéros dans l'ordre d'arrivée au guichet sont 7,4 et 5, et qui souhaitent faire les actions suivantes :

- 7 : rendre 5 et 0, se désinscrire, s'inscrire, emprunter 4 et 0.
- 4 : rendre 1, emprunter 5 et 2.
- 5 : se désinscrire, s'inscrire, emprunter 0, 2 et 3.

Voici dans l'ordre ce qu'il se passe au guichet :

- 7 arrive au guichet,
- 7 rend 5 et 0,
- 7 souhaite se désinscrire mais ne peux pas car il possède un livre,

- 7 souhaite s'inscrire mais est déjà inscrit,
- 7 souhaite emprunter 4 mais ce livre est déjà emprunté par 2,
- 7 souhaite emprunter 0 mais ne peux pas car il vient juste de le rendre,
- 7 quitte le guichet,
- 4 arrive au guichet,
- 4 rend 1 en retard : il est pénalisé de `PENALITE` jours,
- 4 souhaite emprunter 5 et 2 mais ne peux pas car il est pénalisé,
- 4 quitte le guichet,
- 5 arrive au guichet,
- 5 souhaite se désinscrire mais ne peux pas car il n'est pas inscrit,
- 5 s'inscrit : Julie lui demande les informations nécessaires au renseignement de son fichier,
- 5 emprunte 0 et 2,
- 5 souhaite emprunter 3 mais ce livre est déjà emprunté par 7,
- 5 quitte le guichet.

Après exécution de ces instructions, le fichier des inscrits est devenu :

Num étudiant: 7
Prénom Nom: Jean Laquille
1er livre emprunté: 3 le 1/11/98
2ème livre emprunté: non
3ème livre emprunté: non
Pénalité: non

Num étudiant: 5
Prénom Nom: Désiré Lafont
1er livre emprunté: 0 le 10/11/98
2ème livre emprunté: 2 le 10/11/98
3ème livre emprunté: non
Pénalité: non

Num étudiant: 4
Prénom Nom: Camille Dutour
1er livre emprunté: 4 le 1/10/98
2ème livre emprunté: non
3ème livre emprunté: non
Pénalité: 10+`PENALITE` jours

3 Ce qu'il vous est demandé

Vous devez écrire un programme qui gère la base des livres, le fichier des inscrits et qui simule la file d'attente, et les actions de chacun des étudiants au guichet. L'ajout de règles supplémentaires de bon sens est laissé à votre appréciation. Vous devrez faire un affichage des actions exécutées au guichet semblable à l'exemple. Vous devrez également intégrer à la simulation des opérations d'entrées/sorties sur disque, qui permettent de sauvegarder l'état de la base des livres, de la base des inscrits, de la date et de la file d'attente (c'est-à-dire l'état général de la simulation) dans un ou plusieurs fichiers.

Votre projet devra disposer du menu suivant, et devra exécuter les actions associées :

1. Lire les fichiers de données d'initialisation de la simulation.

2. Consultation de la base des inscrits.
3. Consultation de la base des livres. Devra notamment indiquer la date de retour d'un livre emprunté.
4. Ajouter un étudiant dans la queue.
5. Consultation de la file d'attente.
6. Traitement de la prochaine opération (de la file d'attente).
7. Changer la date du jour.
8. Modification d'une entité de la file d'attente.
9. Sauvegarder l'état général de la simulation.
10. Quitter.

Vous devrez fournir une documentation où les structures de données employées seront décrites et justifiées. Les algorithmes principaux utilisés devront être expliqués, et leur complexité calculée. Naturellement, vous devrez employer des structures de données claires et efficaces associés à des algorithmes aussi performants que possible.

Il est très important que le code soit clair et correctement commenté. Il est plus important d'avoir un projet bien documenté avec un source bien commenté, de bons algorithmes et structures de données que d'implémenter toutes les options du projet.

4 Pour en faire plus

Pour compliquer un peu le projet on y ajoute une notion de temps : chaque action au guichet prend un temps donné t :

- pour l'inscription, $t = 4$,
- pour la désinscription, $t = 2$,
- pour l'emprunt d'un livre, $t = 3$,
- pour rendre un livre, $t = 1$.

Une jounée fait `DUREEJOURNEE` unités de temps.

On devra ajouter deux choix au menu principal :

- Faire passer une unité de temps.
- Faire passer autant d'unité de temps que nécessaire pour que Julie puisse s'occuper du premier étudiant dans la queue.

Le temps va servir à ajouter des *impatients* et des *resquilleurs* dans la file d'attente. Pour ce faire, pour chaque entité de la queue au guichet on ajoute les informations suivantes :

- le nombre d'unité de temps que l'entité veut bien consacrer à faire la queue,
- le nombre de personnes que l'entité souhaite dépasser à chaque unité de temps dans la queue,

- le nombre de personnes qui ont dépassé l'entité dans la queue.

Vous pouvez ajouter à votre projet la gestion des entités dont la description suit. Il n'est pas nécessaire de les gérer toutes, mais il est important que la première gestion ajoutée soit celle des impatients et la seconde celle des resquilleurs car ce classement est établit par ordre croissant de difficulté.

4.1 Les impatients

Un étudiant qui se présente à la bibliothèque n'a pas toujours le temps de faire la queue pour emprunter un livre, par exemple. Quand il prend sa place dans la queue il décide du temps qu'il va y rester. Une fois qu'il a pris cette décision il ne peut plus changer d'avis. À chaque unité de temps qui s'écoule, le temps qui lui reste à passer dans la queue diminue de un. Quand ce temps devient nul l'étudiant quitte la queue, sans avoir pu rendre son livre. On utilisera une valeur spéciale du nombre d'unité de temps pour indiquer que l'entité fait la queue coûte que coûte. Dans ce cas elle n'est pas considérée comme impatiente.

4.2 Les resquilleurs

Comme d'habitude, des gens peu scrupuleux essaient de dépasser des gens dans la queue. Quand une de ces personnes prend sa place dans la queue elle décide du nombre de personnes qu'elle va dépasser à chaque unité de temps. Si ce nombre est différent de 0 alors l'entité est un resquilleur. Une fois qu'elle a pris cette décision elle ne peut plus changer d'avis. À chaque unité de temps qui s'écoule, elle dépasse le nombre de personnes qu'elle avait décidé de dépasser. Attention cependant : chaque étudiant de la queue compte le nombre de resquilleurs qui l'ont dépassé. Quand ce nombre devient trop grand, mettons à partir d'un seuil constant `TOLERANCE`, l'étudiant ne se laisse plus dépasser. Toutefois, le resquilleur ne cessera d'essayer de le dépasser, même s'il subit un échec à chaque fois. Notons qu'un resquilleur peut aussi être impatient.