



Réalisation d'une File

Objectifs

1. passage par référence,
2. codes d'erreur.

Ce TP est consacré à une mise en œuvre du Type Abstrait de Données (T.A.D.) FILE. Cette réalisation utilise un tableau et implémente une FILE à capacité limitée.

BIEN LIRE LE SUJET EN ENTIER AVANT DE COMMENCER

1 Introduction

La communication et la synchronisation entre processus adoptent très souvent un schéma générique connu sous le nom de « producteur/consommateur ». Ce schéma est notamment celui mis en œuvre dans le système Unix avec la notion de communication par pipe.

Dans ce schéma, un processus au moins, qualifié de producteur, produit des informations (lignes de texte par exemple) dans un tampon de taille limitée et un processus au moins, qualifié de consommateur, consomme les informations dans **l'ordre chronologique de leur production**.

L'implantation d'un tel schéma de communication s'appuie sur une structure de données partagée de type file d'attente FIFO (first-in, first-out). Il s'agit de gérer de façon circulaire l'espace disponible dans le tampon mémoire partagé par le(s) processus producteur(s) et consommateur(s).

L'objectif de ce TP est de développer, de façon très simplifiée, un type abstrait ayant les propriétés d'une file d'attente fifo. En particulier, le problème de la synchronisation des processus usagers ne sera pas abordé : par exemple, le blocage du processus producteur si la file fifo est pleine. On se contentera ici de signaler l'échec de l'opération.

De plus, les opérations d'insertion en tête de file et d'extraction en queue de file porteront toujours sur un élément simple, ici un entier.

2 Rappel des opérations de manipulation du T.A.D. File

Les opérations qui permettent de manipuler une FILE sont les suivantes :

- un constructeur qui fournit une nouvelle FILE ;
- une opération qui "défile" un élément (lecture dans la FILE) ;
- une opération qui "enfile" un élément (écriture dans la FILE) ;
- une consultation de la tête de la FILE ;
- un affichage du contenu de la FILE ;
- un test de l'état de la FILE (Vide, Pleine, Ni l'un Ni l'autre).

3 Principe d'implémentation

L'implémentation de la FIFO s'appuie sur un tableau (FIFO à capacité limitée) avec un fonctionnement circulaire.

Vous utiliserez :

- un vecteur (tableau) d'entiers de taille n ;
- deux variables entières désignant la "tête" et la "queue" de la FILE. La "tête" est l'indice de la case du prochain entier à lire et la "queue" est l'indice de la prochaine case où écrire ; ces indices varient donc de 0 à $n - 1$;
- une valeur entière qui stocke le nombre d'éléments de la FIFO.

Ainsi une lecture (resp. d'écriture) consistera à mettre à jour (c'est-à-dire d'incrémenter) la tête (resp. la queue) de la FILE modulo n et à gérer le nombre d'éléments.

Par convention, une tentative pour écrire une valeur dans la FILE alors qu'elle est pleine laissera la FILE intacte et sera identifiée comme «opération impossible». Il en sera de même avec une tentative de lecture dans une FILE vide.

La Fig. 1 résume ces propositions.

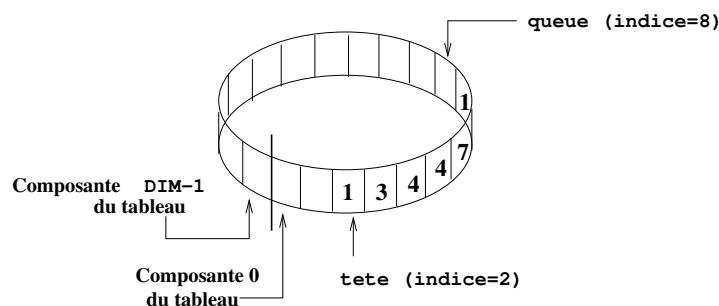


FIGURE 1 – La FILE circulaire.

4 Travail à réaliser en séance

Nous vous demandons d'associer à chacune des opérations de manipulation de la FILE, une fonction C pouvant avoir des paramètres données/résultats.

Toutes ces fonctions devront renvoyer un code d'erreur (par exemple en cas de lecture sur une FILE vide).

Vous vous appliquerez à ne passer que l'information nécessaire au moment de l'appel à la fonction (l'adresse de la variable si vous souhaitez modifier cette variable grâce à la fonction ou sa *valeur* sinon).

5 Éléments de code

Pour vous aider, nous vous donnons un fichier `tp2_file.c` à récupérer sur MOODLE qui propose un menu interactif sous forme d'une boucle dans un programme principal.

À chaque itération, le menu proposera d'effectuer une des opérations suivantes :

- construire (initialiser) la FILE ;
- "enfiler" un élément si l'état de la FILE le permet ;
- "défiler" un élément si l'état de la FILE le permet ;
- afficher le contenu de la FILE ;
- terminer.

Concentrez-vous sur les opérations de ce menu ; si vous avez le temps, vous pourrez rajouter la consultation de la tête et celle de l'état de la FILE.

6 Amélioration (peut se faire hors séance)

Regroupez les variables qui définissent la FILE en une structure et reprenez les fonctions de manipulation de la FILE en prenant en compte ce changement.