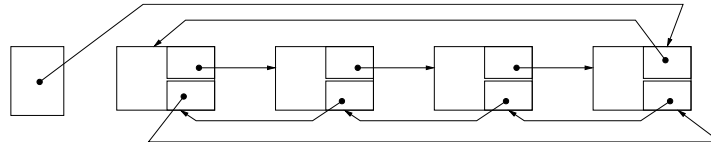


## Structures de données et algorithmes

TD : tampons circulaires bilatères

### Tampon bilatère doublement chaîné

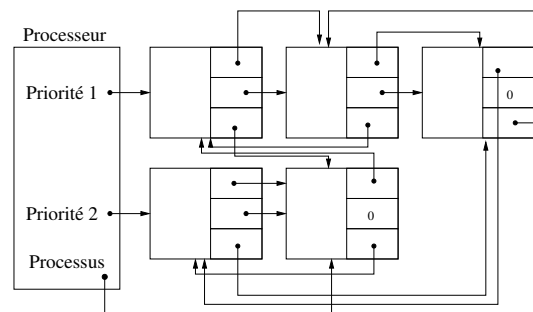
On rappelle qu'un tampon bilatère doublement chaîné est une implantation chaînée du type abstrait File. On peut représenter schématiquement un tel tampon comme suit.



1. Écrivez le fichier header de nom `tampon_bilatere.h` contenant la définition de la structure de données suggérée par le dessin ci-dessus et les prototypes des fonctions C qu'on vous demandera de programmer.
2. Rappelez la spécification vue en cours (signature, préconditions et axiomes)
3. Programmez les fonctions classiques d'adjonction en queue et de suppression de la tête.
4. Programmez une fonction pour calculer la longueur d'un tel tampon.
5. Écrivez une fonction d'insertion à la  $i$ ème place.
6. Programmez les fonctions de décalage à gauche et à droite de  $n$  places.
7. Faites un programmes de tests permettant de vérifier que votre implantation compile et fonctionne sur le jeu d'essais que vous fournirez.

## 1 Files de priorité

Dans un système à temps partagé (avec un seul processeur) l'accès au processeur se fait selon une file d'attente : le processus servi a le droit d'utiliser le processeur durant un certain quantum de temps, puis il est remis dans la file d'attente suivant sa priorité ou à la fin (tourniquet). Lorsqu'on a des classes des priorités, il y peut y avoir plusieurs files d'attente, 2 dans le dessin ci-dessus. Un processus peut éventuellement changer de classe de priorité, par exemple lorsqu'il n'a pas été servi depuis longtemps. Le système gère aussi une liste de processus en cours sous forme d'un tampons bilatère doublement chaînée. En général, un autre accès plus rapide, via une table de hachage, est prévu. Cela peut être représenté schématiquement par le dessin simplifié ci-dessous où l'on n'a que deux files de priorité.



Un processus est décrit par un bloc de contrôle avec un grand nombre d'informations (table de pagination, table de segments, tables des entrées/sorties, statistiques, priorité, ...). Nous nous contenterons de l'identifier par un entier, mais c'est en réalité, un pointeur vers une telle structure.

## 1.1 Travail préliminaire

Formez des équipes de 4 et travaillez en groupe à une spécification, éventuellement en plusieurs modules de spécification, du système de file de priorités présenté plus haut. Essayer de mettre en avant les notions importantes et les opérations qui devront être implantées. Essayez aussi de ne pas trop penser à des structures de données a priori bien que le sujet y incite fortement.

Je ne vous fais pas l'injure de vous proposer des systèmes informatiques favorisant la collaboration il y en a plein. Certains sont mieux adaptés que d'autres et peut-être moins intrusifs (Gitlab, Slack, ...).

## 1.2 Programmation

1. Définissez la structure de données correspondant au schéma donné.
2. Écrivez une fonction C pour ajouter un processus dans le système et le placer à la fin de la file de priorité 1 ou 2 (valeur donnée en argument).
3. Même chose pour retirer un processus en tête d'une telle file.
4. Faites une fonction pour retirer un processus donné (par son numéro ou par un pointeur sur la cellule correspondante) d'une file et le placer dans l'autre.
5. Écrivez une fonction pour retirer un processus du système.