

Algorithmique et Structures de Données

TD n°5

Licence Informatique 2ème année
Université de Nice-Sophia Antipolis

1 Circular Buffer : Mémoire Tampon Circulaire

En informatique, une mémoire tampon, couramment désignée par le terme anglais *buffer*, est une zone de mémoire vive ou de disque utilisée pour stocker temporairement des données, notamment entre deux processus ou matériels ne travaillant pas au même rythme. Ainsi, les données envoyées vers un périphérique (externe) sont-elles le plus souvent stockées dans des mémoires tampon en attente de leur envoi effectif pour épargner à l'ordinateur le contretemps dû à la différence de débits entre le microprocesseur interne et les différents périphériques souvent lents. De même, les données reçues de l'extérieur sont le plus souvent stockées dans des tampons en attente de leur traitement par l'ordinateur (pour des raisons d'efficacité, et aussi pour éviter qu'une réception de données trop rapprochées fasse que certaines, non traitées, ne soient perdues).

Dans ce TD, nous allons nous intéresser aux mémoires tampons circulaires.

On représente une mémoire tampon par un tableau. Une mémoire tampon circulaire utilise une mémoire tampon de taille fixe comme si ses deux extrémités étaient connectées. Autrement dit, une mémoire tampon de taille fixe est un tableau dont la fin et le début coïncident logiquement. Physiquement on garde la structure de tableau, mais on considère que l'indice suivant le dernier (i.e. n) est 1 (celui du début) et que l'indice précédant le premier (i.e. 1) est la fin (i.e. n). On peut effectuer deux opérations sur la mémoire tampon :

- la lecture d'un élément à partir de la mémoire tampon correspond à la suppression du premier élément du tableau;
- l'écriture d'un élément dans la mémoire tampon équivaut à ajouter cet élément à la fin du tableau.

On considère que l'on dispose d'un tableau de 10 éléments. Dessiner l'évolution du tableau pour les opérations suivantes.

1. On écrit successivement les valeurs 10, 2, 12, 3, 6, 8.
2. On lit la première valeur.
3. On écrit la valeur 9.
4. On lit les 3 premières valeurs.
5. On écrit les valeurs 4, 5, 1, 7, 11, 13.

En réfléchissant un peu, on s'aperçoit qu'il est bien utile de disposer de deux indices : l'un de début et l'autre de fin.

1. Donner l'algorithme qui gère l'écriture dans la mémoire tampon d'un élément. On ne traitera pas le cas du *buffer* plein
2. Même question pour la lecture à partir de la mémoire tampon.

3. Ecrire la fonction qui calcule le nombre d'éléments pris dans le buffer
4. Ecrire la fonction VIDE qui est vrai si le buffer est vide et fausse sinon
5. Ecrire la fonction PLEIN qui est vrai si le buffer est plein et faux sinon

Utiliser deux indices dont l'un désigne le premier élément et l'autre le dernier ne permet pas de distinguer le cas où le buffer est vide au cas où le buffer est plein. Montrer cela. Discuter les avantages et les inconvénients des méthodes permettant de remédier à ce problème :

1. on laisse toujours une case du tableau vide;
2. on compte le nombre d'éléments;
3. on compte le nombre d'écriture et de lecture;
4. on utilise un modulo.

Donner les algorithmes de lecture et d'écriture pour les solutions 1 et 4 ensemble.