

*Série de Travaux Dirigés : 3 - MPI - fonctions de communications point à point*

---

Pour l'ensemble de ce TD, vous ne devez utiliser que des fonctions de communications point à point.

**Exercice 1. hello word**

Compilez et exécutez le programme `hello.c` (disponible sous `celene`). Créez votre `makefile` si vous le souhaitez.

**Exercice 2. Shift circulaire**

Ecrivez la fonction qui permettrait de faire un échange circulaire d'un tableau de données utilisant des envois bloquants synchrones.

**Exercice 3. Fonction d'échanges**

Écrire une fonction C+MPI *GaucheDroite* (qui sera utilisée comme fonction collective) de profil :

```
void GaucheDroite(int * in, int * out)
```

qui suppose un tableau *in* de 20 entiers sur chaque processus. L'effet de cette fonction est de transmettre, pour chaque processus, la première moitié de son tableau vers le processus voisin de gauche et l'autre moitié vers le voisin de droite. Le processus 0 n'a pas de voisin de gauche (donc rien à faire vers la gauche) et le dernier processus n'a pas de voisin de droite. Les deux moitiés sont stockées dans le même tableau *out*, la moitié venant du voisin de droite étant après la moitié venant du voisin de gauche. Le communicateur utilisé est *MPI.COMM\_WORLD*.

Refaire la même question que ci-dessus mais en traitant les voisins de manière cyclique : le dernier processus est le voisin de gauche de 0, etc.

**Exercice 4. Fonction échange total**

Ecrivez une fonction *EchangeTotal* qui réalise un échange total de données pour reconstituer un tableau dont les *n* premiers éléments sont envoyés par 0 etc. A la fin de l'échange tous les processeurs ont l'intégralité du tableau.

```
void EchangeTotal(int *tabin, int *tabout, int n)
```

**Exercice 5. Fonction Reduction**

Ecrire une fonction *Reduction* qui réalise la réduction (la somme par exemple) des données réparties sur tous les processeurs et qui met le résultat dans un autre buffer sur le processus *dest* donné en paramètre.

```
void Reduction(int *a, int *b, int dest)
```