



Rémy Malgouyres  
Université Clermont 1  
<http://laic.u-clermont1.fr/~mr/>

## Programmation système

# TP n° 2 Lancer le quicksort dans un thread

Durée 1 semaine

### Objectifs :

Le but de ce TP est de pratiquer les pointeurs de fonctions, d'apprendre à utiliser la fonction `qsort`, et de lancer un calcul dans un thread.

**Note.** Chaque semaine, le listing commenté du TP est à rendre en début de la séance de la semaine suivante.

## 1 Fonctions de comparaison

Une fonction de comparaison est une fonction qui prend deux paramètres  $(a, b)$  d'un certain type et renvoie :

- 0 si  $a = b$  ;
- Un nombre négatif si  $a < b$  ;
- Un nombre positif si  $a > b$ .

Pour faire une fonction de comparaison un peu générale (dont le prototype fonctionne pour différents types), nous allons mettre  $a$  et  $b$  de type `void*`. La fonction de comparaison aura alors pour prototype :

```
int comparaison(const void *a, const void *b)
```

**Exercice 1** Pour chacune des fonction de comparaison suivantes, on testera la fonction dans le main.

- Faire une fonction de comparaison pour des entiers.
- Faire une fonction de comparaison pour des chaînes de caractères.
- Faire une fonction de comparaison pour le type `Date` suivant :

```
typedef struct  
{  
    int jour; /* de 1 à 31 */  
    int mois; /* de 1 à 12 */  
    int annee;  
}
```

## 2 La fonction qsort

La fonction `qsort` implémente le tri rapide. Elle a pour prototype :

```
void qsort(void *tableau,  
           size_t nb_elements,  
           size_t taille_element,  
           int (*comparaison)(const void *, const void *));
```

Le premier paramètre est le tableau des éléments à trier, le deuxième paramètre est le nombre d'éléments du tableau, le troisième paramètre est la taille d'un élément (obtenu par `sizeof`), et le quatrième élément est la fonction de comparaison utilisée pour le tri.

**Exercice 2** Créer une structure `TypeTableau` qui contient :

- Un tableau d'entiers ;
- Le nombre d'éléments du tableau ;

Créer aussi une fonction qui génère un `TypeTableau` alléatoire dont la taille est passée en paramètre.

**Exercice 3** Créer une fonction de thread qui prend en paramètre un `TypeTableau` et le trie par `qsort`. Tester en créant un thread.

**Exercice 4** Modifier le programme précédent pour qu'un autre thread puisse terminer le programme si l'utilisateur appuie sur la touche 'A' (par `exit(0)`). Le programme doit afficher un message en cas d'annulation, et doit afficher le résultat du calcul sinon.

**Exercice 5** [facultatif] Modifier le programme et les structures de données pour qu'il puisse proposer le choix à l'utilisateur de trier des entiers, des chaînes de caractères ou des dates en utilisant la même fonction de thread.