

## TME 4BIS - AJOUT D'UNE FONCTION `MYSLEEP()` DANS `LIBSCHED`

La version 0 de `libsched` termine l'ordonnancement dès qu'il n'y a plus de tâche prête. Ceci ne permet pas de prendre en compte le fait qu'une tâche peut être bloquée ou endormie, donc non terminée, alors qu'aucune autre tâche n'est prête.

L'objectif de ce TME est de permettre la prise en compte de ce type de mécanismes. Pour cela, on utilise une nouvelle version de `libsched` dans laquelle on veut ajouter une fonction `mysleep()` (similaire à la fonction `sleep()` de la bibliothèque standard de C). La fonction `mysleep()` doit permettre à une tâche de s'endormir pendant un nombre de secondes donné.

### Modifications de `libsched` :

la version 1.1 de `libsched` crée systématiquement une tâche `idle` (cette tâche est créée lors de l'initialisation de la bibliothèque avant que le programme principal ne s'exécute). La tâche `idle` est toujours prête, mais elle ne s'exécute que s'il n'y pas de tâche prête parmi celles créées par l'utilisateur.

**Attention :** si votre fonction d'élection gère des priorités dynamiques, assurez-vous que la priorité de la tâche `idle` (indice 0 dans `Tproc[]`) reste toujours `MINPRIO-1`. Cet invariant est nécessaire pour qu'elle soit exécutée seulement quand toutes les autres tâches se trouvent dans l'état `SLEEP`.

Lorsque la tâche `idle` est exécutée, on lui attribue le processeur pendant un quantum, comme pour n'importe quelle tâche.

Dans `struct proc` :

`flag` peut maintenant prendre aussi la valeur `SLEEP` lorsqu'une tâche est endormie.

Il y a un champ supplémentaire :

```
double wakeuptime; // instant auquel la tache doit etre reveillee
```

### **Nouvelles fonctions de `libsched` :**

`resched()` l'appel à cette fonction provoque une nouvelle élection. La tâche élue peut être une des tâches prêtes, la tâche courante, ou la tâche `idle` s'il n'y pas de tâche prête à cet instant.

`finquantum()` cette fonction est appelée par `libsched` à chaque fois qu'une tâche a épuisé son quantum. Elle remet à l'état `RUN` les tâches dont la période de sommeil est terminée, mais **elle n'appelle pas `resched()`**. Ceci est fait par `libsched` après que `finquantum()` retourne.

## 1 IMPLEMENTATION DE `MYSLEEP()`

### 1.1

Copiez le répertoire `sleep` présent en `/Infos/lmd/2005/licence/ue/li324-2006fev/libsched-1.1`

---

**1.2**

Complétez la fonction `finquantum()` dans le fichier `sleeptest.c`.

---

**1.3**

Ecrivez la fonction `mysleep()`. Cette fonction doit calculer l'instant où la tâche qui l'appelle devra être réveillée, puis provoquer une élection parmi l'ensemble des tâches prêtes.