Parte Extra de la Práctica 3

(**4 puntos**) Implementar RR con cuanto dinámico, de forma que si una tarea agota su cuanto y no es final de ráfaga el siguiente cuanto se decremente en 1, y si no lo agota su cuanto se incremente en 1. El cuanto siempre debe estar entre 1 y un valor máximo preestablecido. Algunas indicaciones para realizar la práctica son:

- Crear el fichero de planificador sched_multiRR.c. Lo copiamos de sched_rr.c y renombramos las funciones y las variables que corresponda de acuerdo al nombre de la nueva politica (multiRR).
- Añadimos la variable (constante) global_max_slice con el valor deseado.
- En la estructura multiRR_data, añadimos los campos:
  - **current_slice** : valor actual del cuanto para la tarea
  - **quantumExhausted** : flag que podrá tomar tres valores
    - 1 (true): indicando que la tarea agotó el cuanto
    - 0 (false): indicando que la tarea no agotó el cuanto
    - -1 (new): indicando que la tarea se acaba de crear
  - **maxSlice**: máximo valor posible para el cuanto
- (**0.25 puntos**) Modificamos la función task_new_multiRR para que se inicialicen convenientemente los nuevos campos de multiRR_data. El flag quantumExhausted se marcara a -1 para indicar que es una tarea recién creada, de forma que no se incremente el cuanto al encolarla.
- (**1.5 puntos**) Modificar la función enqueue_task_multiRR, de forma que si la tarea tiene el flag quantumExhausted a 1, se le reste 1 a su current_slice antes de encolarla, y si lo tiene a 0 se le sume 1; simpre que el cuanto siga estando en los límites válidos.
- (**1.5 puntos**) Modificar la función task_tick_multiRR para que ponga a 1 el flag quantumExhausted si el tick actual agota el cuanto pero no es final de ráfaga y a 0 en caso contrario.
- (**0.75 puntos**) Actualizar el proyecto para que se compile el nuevo fichero y la politica esté registrada.