

# Práctica 3

## Parte Extra

### Planificación multinivel con realimentación

En esta práctica debéis implementar una política de planificación basada en Round Robin pero con quanto dinámico. Para ello incluiremos el fichero de política sched\_RRdynQ.c y rellenaremos los métodos.

En dicha política se definen las siguientes variables globales y estructura:

```
#define DEFAULT_QUANTUM    3

/* Global RR quantum parameter */
const int dynq_rr_quantum=DEFAULT_QUANTUM;

/* Global max slice */
const int global_max_slice=DEFAULT_QUANTUM+2;
const int global_min_slice=2;

struct RR_dynQ_data {
    int  remaining_ticks_slice;    // Remaining ticks
    int  current_slice;           // Initial ticks when extracted from runqueue
};
```

Al igual que en Round Robin, es necesario reservar espacio para una estructura de tipo RR\_dynQ\_data para cada tarea. Los campos de esta estructura contendrán el número de ticks que restan antes de finalizar el quanto (como en Round Robin), y el número de ticks que se asignaron a la tarea al ser extraída de la runqueue. Este último parámetro, current\_slice, toma un valor inicial igual a dynq\_rr\_quantum pero podrá ser modificado dentro de la rutina task\_tick\_rrdynQ en las siguientes circunstancias:

- Si a la tarea se ha terminado el quanto y necesita más ticks para finalizar la ráfaga de CPU, current\_slice decrementará en uno (saturando en global\_min\_slice). Se pueden consultar el número de ticks que faltan para terminar la ráfaga en el campo runnable\_ticks\_left de la tarea.
- Si la tarea finaliza la ráfaga y el quanto a la vez, task\_tick\_rrdynQ se mantendrá constante.
- Si la tarea finaliza la ráfaga y todavía disponía de más ticks para finalizar el quanto, task\_tick\_rrdynQ se incrementará en uno (saturando en global\_max\_slice).

**Recuerda:** Debéis modificar el Makefile para incluir el nuevo fichero (sched\_rr\_dq.c) en la compilación y también el fichero de cabecera sched.h para que esté disponible.



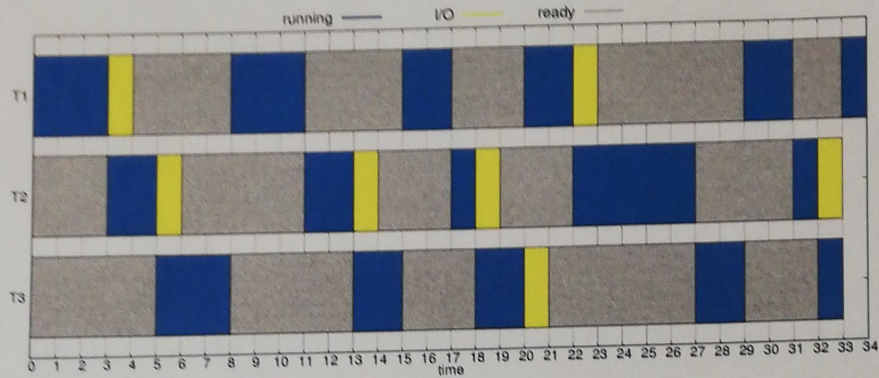
## Ejemplo

Ejemplo de simulación con el siguiente fichero:

```
T1 1 0 3 1 7 1 3
T2 1 0 2 1 2 1 1 6 1
T3 1 0 7 1 3
```

Una sola CPU:

```
./schedsim -s RR_DYNQ -i ./examples/example_RR_dynQ.txt
```



Ejemplo de simulación con dos CPUs:

```
./schedsim -n 2 -s RR_DYNQ -i ./examples/example_RR_dynQ.txt
```

