[Portada]

[Indice]

# Introducción

Esta práctica consiste en el desarrollo de planificadores de procesos en el lenguaje de programación C. Se implementarán los siguientes planificadores:

## Round-Robin

Planificador que simplemente incorpora el algoritmo Round-Robin a través de una rodaja definida en uno de los ficheros.

## Round-Robin/FIFO con prioridades

Planificador que ejecutará Round-Robin para aquellos hilos con prioridad baja y una política FIFO para aquellos procesos con prioridad alta.

## Round-Robin/FIFO con cambios de contexto voluntarios

Planificador similar al anterior pero que permite a un hilo realizar un cambio de contexto voluntario mediante una llamada al sistema.

# Descripción del código

## Round-Robin (RR.c)

Las principales funciones implementadas son *TCB\* scheduler ()*, *void timer\_interrupt (int sig)* y *void activator (TCB\* next)*.

También se utilizará una cola (llamada *thread\_q*) que contiene todos los hilos que aun no han terminado.

Cada vez que se cree un hilo se encolara también en *thread\_q*.

### timer\_interrupt(int sig):

Va restando ticks del reloj al hilo que se esta ejecutando actualmente, cuando se le acabe la rodaja (ticks del hilo iguales a 0), se llama al planificador (*void activator (TCB\* next)*) para determinar el siguiente hilo a ejecutar y pasárselo al activador (*void activator (TCB\* next)*) para que realice los cambios de variables y contexto necesarios.

### scheduler ():

Es el planificador del algoritmo, si un proceso aún no ha terminado, se le vuelve a asignar la rodaja y se le pone al final de la cola.

Sacamos de la cola el siguiente hilo y lo devolvemos para que el activador haga sus funciones, si no hay siguiente elemento terminamos el programa y salimos del mismo.

### activator (TCB\* next):

Parte del hilo que ha pasado el planificador, cambiamos la variable running que representa el hilo actual por el nuevo y la variable current que representa el índice del hilo actual por el nuevo índice.

Si el hilo antiguo ha terminado, simplemente establecemos el contexto del nuevo mediante setcontext. En cambio, si el hilo no ha terminado, cambiamos el contexto de uno a otro mediante swapcontext.

Además, se han editado as siguientes funciones:

### init\_mythreadlib():

Para inicializar la cola de hilos.

### mythread\_create(void (\*fun\_addr)(), int priority):

Para encolar los hilos que se van creando

## Round-Robin/FIFO con prioridades (RRF.c)

Las principales funciones implementadas son *TCB\* scheduler ()*, *void timer\_interrupt (int sig)* y *void activator (TCB\* next)*.

También se utilizarán un par de colas (llamas low\_q y high\_q) para representar las listas de hilos de baja prioridad y de alta prioridad respectivamente

### timer\_interrupt(int sig):

Sirve para controlar los ticks del round robin, si actualmente se esta corriendo un hilo de prioridad alta, se sale de esta función sin más.

En caso contrario, se utiliza como en RR.c, se baja 1 al contador de ticks y cuando se acabe la rodaja vamos a coger el siguiente hilo con scheduler y manejar el contexto con activator

### scheduler ():

Planificador del algoritmo, nos permite sacar un hilo de la cola correspondiente, tenemos dos casos:

* Si el hilo actual ha terminado, sacaremos un hilo de la cola de prioridad alta (si hay) y si no hay sacaremos un hilo de la cola de prioridad baja
* Si el hilo no ha terminado, nos encontramos en un caso de un hilo de prioridad baja que se le ha terminado su rodaja, y sacamos el siguiente hilo de la cola.

### activator (TCB\* next):

Partimos de hilo que nos ha devuelto scheduler, si el hilo actual ha terminado, usamos setcontext para establecer el nuevo, cambiando variables running y current.

Si el hilo anterior simplemente ha terminado su rodaja y no ha terminado, cambiamos variables running y current y cambiamos el contexto con swapcontext

Además, se han editado las siguientes funciones:

### init\_mythreadlib():

Para inicializar las colas de hilos.

### mythread\_create(void (\*fun\_addr)(), int priority):

Para encolar los hilos que se van creando dependiendo de la prioridad que tengan

### mythread\_exit():

Para añadir una finalización del programa cuando no queden hilos

## Round-Robin/FIFO con prioridades con cambios de contexto voluntarios(RRFN.c)

# Batería de pruebas

Las pruebas se realizarán cambiando el contenido del fichero main.c

## Round-Robin (RR.c)

|  |  |
| --- | --- |
| Lista de pruebas | |
| Descripción | Resultado |
| Prueba de ejecución estándar con los hilos que están en main por defecto | Ejecución normal |
| Ejecución sin creación de hilos con mythread\_create | Ejecución solo del hilo principal |
| Ejecución con un solo hilo creado | Ejecución de forma seguida del hilo principal y el creado |
| Ejecución de tres hilos que ejecutan la misma función | Ejecución correcta de Round Robin |
| Ejecución de tres hilos cuya duración es la misma y además es menor que la rodaja | Se ejecutan de forma normal como si se tratara de un FIFO |
| Ejecución de tres hilos cuya duración es muy superior a la de rodaja y necesiten de múltiples rodajas para poder ejecutarse. Tiene como objetivo que el round robin funcione con más de un cambio de contexto para el mismo hilo | Ejecución correcta de Round Robin |

## Round-Robin/ FIFO con prioridades (RRF.c)

Las pruebas se realizarán cambiando el contenido del fichero main.c

|  |  |
| --- | --- |
| Lista de pruebas | |
| Descripción | Resultado |
| Prueba de ejecución normal | Ejecución normal, los hilos de alta prioridad se ejecutan como FIFO, los hilos de baja se ejecutan como Round Robin |
| Ejecución con hilos solo de baja prioridad | Ejecución de Round Robin como en RR.c |
| Ejecución con hilos solo de alta prioridad | Ejecución del planificador como FIFO |
| Ejecución de tres hilos que ejecutan la misma función | Ejecución correcta de Round Robin |
| Ejecución de expulsión. Creamos un hilo de alta prioridad dentro de un hilo de baja prioridad | Ejecución con la traza de expulsión por la salida estándar |
| Ejecución con un solo hilo creado | Ejecución de forma seguida del hilo principal y el creado |
| Ejecución con dos hilos ejecutando la misma función, cambiando la prioridad | Se ejecuta primero el de mayor prioridad |
| Ejecución sin creación de hilos con mythread\_create | Ejecución solo del hilo principal |

## Round-Robin/FIFO con prioridades con cambios de contexto voluntarios(RRFN.c)

[conclusiones]