Práctico II

Implementación de prioridades y estados de tarea

Implementación de Sistemas Operativos I
Maestría en Sistemas Embebidos
Año 2023

Docentes
Mg. Ing. Gonzalo Sanchez
Esp. Ing. Hannes Sciarrone

Tabla de contenido

Registro de cambios

Revisión	Cambios realizados	Fecha
1.0	Creación del documento	15/09/2023

Actividades a realizar

Estado de tareas

Las tareas en un RTOS cuentan con estados que indican en qué etapa se encuentra. Por lo visto en la teoría los estados posibles son:

- Ready
- Running
- Blocked
- Suspended (*opcional, no obligatorio*)

Entonces, para su RTOS se debe implementar un mecanismo que almacene el estado de las tareas y cambiar su scheduler para que actúe en consecuencia al estado de ellas.

Tarea Idle

Según lo visto en la teoría, la tarea **Idle** es justamente una tarea que se ejecuta si y sólo si todas las tareas creadas para la implementación deseada están en el estado *Blocked* esperando por algún evento que las pase a un estado de ready. Por la propia definición de la tarea **Idle** se entiende que esta debe tener la menor prioridad posible.

Para la implementación es necesario que:

- Implemente la tarea **Idle** con lo visto en la teoría.
- Si esta tarea no realiza operaciones debe llamar a la instrucción Wait For Interrupt.
- La tarea debe definirse como WEAK para poder sobreescribir su implementación.

API Delay

Para poder testear fácilmente los estados de las tareas (en especial el estado bloqueado) se requiere la implementación de la primera API de sistema operativo, que se considera la más sencilla: el Delay. La tarea que invoca un delay debe permanecer bloqueada por la cantidad de milisegundos especificada en el argumento al momento de la llamada. Se recomienda que en cada tick del sistema se actualice un contador interno que puede estar asociado a la tarea, ya que no pueden haber dos delays concurrentes en la misma tarea. Una vez finalizada la cuenta, la tarea debe pasar a estado ready y el scheduler decidirá el momento de su ejecución.

Prioridades

Los sistemas operativos de tiempo real siempre ejecutan las tareas por orden de prioridad. Esto quiere decir que para funcionar correctamente el scheduler debe ejecutar las tareas de la más prioritaria a la menos prioritaria evaluando si ellas se encuentran en el estado de ready.
Para esta etapa del trabajo práctico usted debe implementar:

- Cuatro (4) niveles de prioridad como mínimo donde, el nivel cero (0) es la más prioritaria y tres (3) será el nivel de menor prioridad.
- En el scheduler se debe evaluar cual tarea será la siguiente en ser ejecutada al tener en cuenta tanto el nivel de prioridad como el estado de las tareas.
- Entre tareas de igual prioridad la lógica de ejecución continua siendo circular (round robin). Esto quiere decir que si existen tres tareas de igual prioridad con el estado valido para su ejecución, el orden de ejecución será:

```
Tarea 1 -> Tarea 2 -> Tarea 3 -> Tarea 1 -> Tarea 2 -> …….
```