

Práctica 5

Sistema Operativo de tiempo real

1. Instale 2 tareas de FreeRTOS y controle un led distinto con cada una, logrando parpadear a 10 y 0,5 Hz.
 - a. ¿Qué tamaño tiene el heap del sistema operativo?
 - b. ¿Cuál es el tamaño del stack asignado para cada tarea?
 - c. ¿Cuál es el máximo tamaño predicho que se ocupará del stack de cada tarea?
 - d. Verifique el stack realmente ocupado en tiempo de ejecución
 - e. Utilice Trace Hooks para observar con el analizador lógico la ejecución de las tareas.

2. Cree un conjunto de 3 tareas, cada una con el esquema de funcionamiento dado por el siguiente pseudocódigo para la tarea i :

```
TickType_t lastTime;  
for (;;) {  
    EncenderLED(LED_i);  
    busy_delay(e_i);  
    ApagarLED(LED_i);  
    DelayUntil( &lastTime, p_i );  
}
```

- a. Realice una demostración de la cota dada por algoritmo RMA asignando prioridades apropiadamente según el factor e_i/p_i de cada tarea y verificando el funcionamiento con trace hooks y el analizador lógico.
 - b. Modifique el código para poder pasar los datos del led y los valores de e y p como argumentos a la tarea. Para ello considere crear una estructura que englobe estos datos.
3. Implemente una tarea que imprima repetitivamente “Hola mundo” en la pantalla OLED.
 - a. ¿Cuál es el máximo tamaño predicho que se ocupará del stack de cada tarea?
 - b. Verifique la predicción de obtenida en el inciso a utilizando la “marca de agua”
 - c. ¿Cuál es el tiempo de ejecución de la tarea?
 4. Implemente un sistema que responde a la presión de 3 botones imprimiendo en la pantalla OLED consecutivamente los botones presionados. La solución se hará a partir de 2 tareas: una de control de la botonera, y una de control de la pantalla. La tarea de la botonera deberá registrar los botones presionados y enviar la información a la tarea de impresión. Cada botón se considerará activado luego de haber soltado el botón (no mientras se lo presiona). El dato del botón activado se almacenará con un enum que contendrá un elemento para cada botón que exista: izquierda, centro, derecha, ninguno. La comunicación se dará a través de un buffer FIFO.
 - a. Justifique
 - i. La asignación de stack de las tareas
 - ii. La asignación de heap del sistema operativo
 - iii. Las prioridades de las tareas.
 - b. Obtenga la predicción del stack utilizado por cada tarea

- c. Utilice la característica de debug con “marca de agua” para verificar que durante la ejecución se verifican los límites de stack.
 - d. Utilice los hooks para producir salidas con el GPIO de manera de medir el tiempo de ejecución de las tareas y obtener el factor de utilización.
5. Se desea implementar un sistema que muestrea valores a 200 Hz con el ADC y los envía por UART. Se implementa el muestreo de una señal con tasa de 5 ms, el jitter no es crítico por lo que se implementará con una tarea y "Delay Until" acumulando los resultados en una Queue. (utilizar vTaskDelayUntil de FreeRTOS, ya que osDelayUntil presenta errores). Otra tarea leerá datos de la Queue resguardándolos en un arreglo local con capacidad para almacenar 1 segundo de información. Cuando el arreglo se llena, debe imprimir por UART (sin usar DMA) la lista de todos los valores enteros leídos hasta el momento (códigos del ADC, no tensión), uno por línea, y finalmente debe imprimir el promedio (sólo la parte entera).
- a) Utilizando los siguientes supuestos:
- El muestreo y carga en el Queue toman 30 us
 - Cada valor se imprime utilizando 7 caracteres y la conversión de entero a cadena toma 20 us
- Responda y justifique:
- i. ¿Qué tasa hay que configurar en el puerto serie para que el factor de utilización esté en la zona "segura"? (Nota: Las tasas estándar son (en baudios): 4800, 9600, 14400, 19200, 38400, 57600, 115200)
 - ii. ¿Cuál sería la asignación de prioridades para garantizar el funcionamiento por la cota RMA? ¿Coincide con las necesidades de este sistema?
- b) Implemente el sistema descrito y explique su operación. No escriba código por fuera de las tareas.
- i. Justifique el tamaño del Queue implementado.
 - ii. Justifique la configuración de Stack de cada tarea y de Heap del SO, y demostrando su suficiencia con capturas de debug en tiempo de compilación y en tiempo de ejecución.
 - iii. Agregue a la implementación la capacidad de visualizar la ejecución.
 - iv. Utilice Task Hooks para visualizar los cambios de contexto con el analizador lógico, de manera de lograr realizar capturas de pantalla que permitan visualizar la ejecución de las tareas (incluyendo la IDLE). Explique lo que se observa.
 - v. Con auxilio de la visualización anterior y cualquier otra señalización que sea útil, explique si el factor de utilización es menor a 1 y su cercanía o lejanía con el estimado.