

Asignatura: Programación de Sistemas y Dispositivos

Curso: **2019/2020**

Examen: Final

Fecha: 28-05-2020

Semestre: 2

Convocatoria: Ordinaria

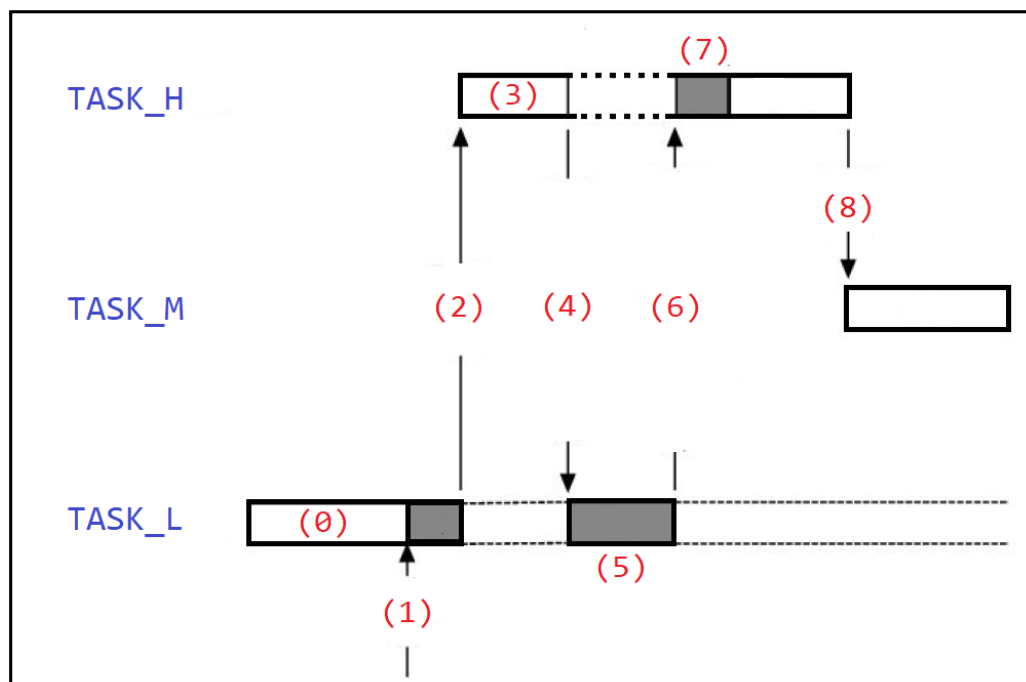
1. Explica las diferencias fundamentales entre los *RTOS* y los *GPOS*. (1pt)
2. Se tiene un microcontrolador de *32-bit*, cuya frecuencia de funcionamiento es de 200MHz. El proyecto requiere configurar una aplicación, de tal modo que se ejecute cada 10 milisegundos, mediante la programación de un *Timer* de *16-bit* y el uso de su respectiva *ISR*. Si la frecuencia efectiva de reloj del *Timer* es 4 veces inferior a la frecuencia de la *CPU* y los Prescalers disponibles son 1, 2, 4, 16, 64 y 256:
  - a. ¿Podrían aplicarse todos los valores de prescaler indicados para generar los 10ms? Razonar la respuesta. (1pt)
  - b. ¿Qué valor de contador de registro habría que configurar para asegurar los 10ms? (Calculad este valor para el prescaler más bajo que pueda ser aplicado). (1pt)
3. En lo referente a los sistemas *Foreground-Background*:
  - a. Explica en detalle las ventajas y desventajas que conlleva su uso. (0.5pt)
  - b. ¿Con qué otro nombre se conocen estos sistemas? (0.5pt)
  - c. ¿En qué consiste la técnica de *polling* y por qué se suele hacer un uso abusivo en estos sistemas? (0.5pt)
  - d. En comparativa con los *RTOS*, ¿Qué sistema posee una respuesta más rápida a la hora saltar o salir del vector de interrupciones *ISR*? Argumenta la respuesta. (0.5pt)
4. En *RTOS*, y en particular a sus objetos de *Kernel*:
  - a. ¿Qué diferencias existen entre los *Semáforos binarios* y los *Mutex*? Razona en que situaciones valorarías más el uso de uno frente al otro. (0.5pt)
  - b. ¿Por qué razón, en estos sistemas, a la hora de sincronizar una *Task* con una *ISR* o una *Task* con otra *Task*, se opta por el uso de un *Semáforo* y no un simple *flag* declarado como variable global? ¿Cuál es la ventaja que se busca? (0.5pt)
  - c. A nivel funcional, ¿Qué similitud tienen los *Message Queues* y los *Semáforos*? Además, ¿Qué ventaja adicional conlleva el uso de *Message Queues* frente a estos? (0.5pt)



- d. Una empresa ha decidido, por primera vez, hacer uso de los *RTOS* y acaba de desarrollar un algoritmo que genera un valor de vital importancia para asegurar el correcto funcionamiento de una aeronave. ¿Por qué recomendarías a esta empresa usar *Message Queues* en lugar de usar una simple variable global, a la hora de transferir o compartir dicho valor entre *Tasks* e *ISRs*? (0.5pt)
5. En el siguiente gráfico se muestran los estados actuales de 3 tareas gestionadas por un *RTOS* con scheduler de tipo prioritario. Se pide explicar, en detalle, lo que ocurre en cada punto indicado en la gráfica (2pt) y contestar a las dos preguntas situadas debajo de la misma (1pt). Tened en cuenta las siguientes consideraciones:
- Task\_H* y *Task\_L* son las únicas tareas que quieren usar el mismo y único recurso compartido (*Uart*).
  - Las zonas marcadas en gris corresponden a zonas donde se está haciendo uso del recurso compartido.

Solución para el primer caso:

(0): *TASK\_L* toma el control de la CPU y comienza a ejecutarse (*Running State*), ya que es la única tarea que estaba lista para ser ejecutada (*Ready-to-Run State*). *TASK\_M* y *TASK\_H* tienen prioridades superiores, pero permanecen a la espera de su turno (*Pending State*).



- En lo referente al fenómeno de inversión de prioridades y a la gráfica anterior. ¿Se está produciendo o se está evitando dicho fenómeno? (0.5pt)
- ¿Qué objeto de *kernel* se está usando, un *Semáforo* o un *Mutex*? (0.5pt)