

**Exame – Época Normal**

10/Jan/2014, duração: 2H00

N.º mec.: \_\_\_\_\_ Nome: \_\_\_\_\_

---

1- [4] Para aquecer ...

- a) [0.5] No âmbito da ciência Tempo-Real, os termos “tempo-real” e “rápido” são equivalentes? Exemplifique.
- b) [0.5] Num sistema de prioridades fixas sem preempção em que cada tarefa acede apenas a um recurso partilhado, o protocolo de acesso a regiões críticas mais adequado é o PIP?
- c) [0.5] Para regular o acesso a uma variável partilhada (forma simples de *shared memory*) com dimensão de um byte o método mais eficiente é desligar as interrupções?
- d) [0.5] Um servidor serve exclusivamente para limitar a utilização de CPU de uma tarefa aperiódica?
- e) [0.5] Os algoritmos RM e EDF são considerados ótimos. Pode então afirmar-se que o uso de um ou outro é equivalente no que concerne à escalonabilidade de um sistema?
- f) [0.5] Num CPU sem *cache* nem *pipelining*, e em que as interrupções estão desligadas, o tempo de execução de uma tarefa é constante. Verdadeiro ou falso? Justifique.
- g) [0.5] No S.O. Linux em que condições é que é equivalente usar os escalonadores “SCHED\_RR” e “SCHED\_FIFO”?
- h) [0.5] Complete a frase: “O escalonamento baseado em prioridades fixas apresenta, em relação ao escalonamento baseado em prioridades dinâmicas, as seguintes 3 vantagens: .... “

2- [1.0] Considere um sistema tempo-real em que há diversas tarefas, com prioridades distintas, que são criadas dinamicamente. Estas tarefas necessitam de aceder a um conjunto de recursos partilhados. Indique, justificando, se neste cenário seria mais vantajoso optar pelo protocolo “PIP” ou “PCP”.

3- [7] Considere uma aplicação tempo-real, escalonada segundo o critério RM, composta pelo conjunto de tarefas descritas na tabela seguinte. Cada tarefa encontra-se caracterizada pelo seu período ( $T_i$ ) e tempo de execução de pior caso ( $C_i$ ). As tarefas  $\tau_1$  a  $\tau_3$  são periódicas. A tarefa  $\tau_a$  é aperiódica. O “tick” do sistema é 1 unidade de tempo e existe preempção.

Tarefa	$T_i=(D_i)$	$C_i$
$\tau_1$	5	1
$\tau_2$	7	2
$\tau_3$	?	4
$\tau_a$	---	1

- [1.5] Calcule o tempo de resposta da tarefa  $\tau_3$  usando o método analítico adequado. Qual o período mínimo que a tarefa  $\tau_3$  pode ter?
- [1.5] Desenhe um diagrama de Gantt que ilustre a situação calculada na alínea anterior.
- [1.5] Considere que  $T_3= 20$ . Usando critérios baseados em taxa de utilização, determine os parâmetros de um servidor do tipo “polling server” que minimize o tempo de resposta da tarefa  $\tau_a$
- [1.5] Considere que ocorreram 5 instâncias consecutivas da tarefa  $\tau_a$ . Calcule qual o tempo máximo que qualquer destas instâncias demora a ser processada. Caso não tenha resolvido a alínea anterior considere um servidor com período 2 e capacidade 1.
- [1] Resolveu duplicar-se o período do servidor e passar a sua capacidade para metade. Qual o impacto desta alteração?

4 – [6] Considere o conjunto de cinco tarefas descrito na tabela abaixo. Cada tarefa encontra-se caracterizada pelo seu período ( $T_i$ ) tempo de execução de pior caso ( $C_i$ ) e deadline relativa ( $D_i$ ). Pretende escalonar-se este conjunto de tarefas usando escalonamento estático cíclico.

Tarefa	$T_i=(D_i)$	$C_i$	Tarefa	$T_i=(D_i)$	$C_i$
$\tau_1$	1	0.2	$\tau_4$	4	0.2
$\tau_2$	2	0.6	$\tau_5$	6	0.5
$\tau_3$	3	0.3			

- a) [3.5] Calcule a tabela de escalonamento, indicando explicitamente os parâmetros relevantes.
  - b) [1] Calcule o tempo de resposta de cada uma das tarefas.
  - c) [1.5] Ao escalonamento anterior pretende adicionar-se uma tarefa  $\tau_6$ , com  $T_6=D_6=12$  e  $C_6=1$ .  
É possível? Em caso afirmativo, descreva em detalhe a forma de o fazer.
- 5- [2] Pretende desenvolver-se uma aplicação, usando o SO FreeRTOS, que contém tarefas periódicas. Estas tarefas apresentam um tempo de execução que varia significativamente de instância para instância. Sabendo que é importante, para a aplicação, que a activação das tarefas seja o mais regular possível, apresente o código adequado para a estrutura do corpo destas tarefas. A funcionalidade associada a cada uma das tarefas pode ser modelada pela função “task\_load();”.