PGENE503 – Sistemas em Tempo Real
Terceira Lista de Exercícios

1) Descreva as diferenças entre tarefas periódicas, aperiódicas e esporádicas esclarecendo as restrições temporais que caracterizam os comportamentos temporais das mesmas.

2) Em que situações se justificam a utilização de um *executivo cíclico de tempo-real*? Que vantagens podem trazer essa utilização?

3) Com um conjunto fixo de tarefas periódicas é possível projetar um escalonamento completo tal que a execução repetida deste escalonamento causará que todos as tarefas executem em uma taxa correta. Sendo assim, considere o seguinte conjunto de tarefas que é mostrado na tabela abaixo:

Tarefa	**Período (T)**	**Tempo de Execução (C)**
A	25	10
B	25	8
C	50	5
D	50	4
E	100	2

É possível escalonar este conjunto de tarefas usando o algoritmo do executivo cíclico? Caso positivo, forneça:

a) As provas formais de que as tarefas são de fato escalonáveis.
b) O diagrama temporal para ilustrar as tarefas que o processador está executando em um determinado momento.
c) O código em ANSI-C ou C++ para implementar o programa executor cíclico (ou despachante) para a escala obtida no item b)

4) O que se entende por inversão de prioridades? E *herança de prioridades*?

5) Verifique a escalonabilidade dos dois conjuntos de tarefas periódicas descritos nas tabelas abaixo usando o teste do *Rate Monotonic* (RM). Se o conjunto for escalonável descreva para cada tabela a escala obtida, na forma de um diagrama de *Gantt*, quando usada a política RM.

Tabela 1: Conjunto de tarefas do sistema

	Computação	Período	Deadline
Tarefa A	3	7	7
Tarefa B	2	12	12
Tarefa C	2	20	20

Tabela 2: Conjunto de tarefas do sistema

	Computação	Período	Deadline
Tarefa 1	1	4	4
Tarefa 2	2	6	6
Tarefa 3	3	10	7

6) Implemente o algoritmo do Rate Monotonic (RM) usando a linguagem ANSI-C ou C++. Considere que o seu algoritmo recebe como entrada um conjunto de tarefas usando o mesmo formato da Tabela 1 e fornece como saída um diagrama de *Gantt*. Lembre que o algoritmo do RM define a ordem de atribuição de prioridades a um conjunto de tarefas, na ordem inversa da periodicidade das tarefas. Além disso, verifique a escalonabilidade do conjunto de tarefas antes de tentar escaloná-las usando o algoritmo do RM. Mensure o tempo de computação (usando o comando *time* do UNIX) da sua implementação para o conjunto de tarefas das Tabelas 1, 2 e 6.

7) Verifique a escalonabilidade do conjunto de tarefas do exercício 5 considerando agora as políticas de escalonamento do *Deadline Monotonic* (DM), *Least Laxity* e *Earliest Deadline First* (EDF). Se o conjunto for escalonável descreva a escala correspondente na forma de um *diagrama de Gantt* para cada uma das políticas. Escreva um programa para cada política de escalonamento com o intuito de automatizar a checagem da escalonabilidade e geração do *diagrama de Gantt.*

8) Estude sobre o teste de escalonabilidade tomando como base o protocolo prioridade teto (PCP). Três tarefas periódicas T_1, T_2 e T_3 compartilham os recursos R_1, R_2 e R_3. As restrições temporais das tarefas e as durações de suas seções críticas que atuam nos recursos compartilhados são indicadas nas tabelas 3 e 4 respectivamente. Com base nestes dados:

a) Calcule os bloqueios B_i, para cada tarefa T_i.
b) Verifique se o conjunto de tarefas é escalonável segundo o RM

Tabela 3: Restrições temporais das tarefas

	Computação	Período	Prioridade
Tarefa 1	5	20	1
Tarefa 2	6	30	2
Tarefa 3	10	35	3

Tabela 4: Duração das seções críticas

	R1	R2	R3
Tarefa 1	1	1	0
Tarefa 2	3	0	1
Tarefa 3	0	4	4

9) Esboce o diagrama da capacidade do servidor *sporadic* para o seguinte conjunto de tarefas considerando que a tarefa aperiódica C é ativada no tempo t0:

Tabela 5: Conjunto de tarefas do sistema

	Computação	Período	Deadline	Prioridade
Tarefa periódica A	4	12	12	2
Tarefa periódica B	4	20	20	3
Tarefa servidora SS	8	32	10	1
Tarefa aperiódica C	8	---	10	---

10) Considere um robô móvel projetado para atingir a velocidade de 1m/s e cujos sensores de obstáculos reagem apenas na gama de distâncias de 5 a 25cm. Qual a taxa mínima de amostragem dos sensores de obstáculos para que estes sejam detectados? (considere amostragem sem *jitter*).

11) Um sistema de esteiras rolantes numa determinada indústria tem 3 esteiras encadeadas para transportar cargas variáveis. De forma que não haja choques entre materiais nele transportados, as respectivas velocidades são controladas em malha fechada por um controlador centralizado, sendo ainda efetuado um controle global para garantir que as 3 esteiras rolem à mesma velocidade. Existe ainda um botão de emergência que deverá ser atendido pelo controlador que deverá parar de imediato as várias esteiras em simultâneo. O controlador está construído sobre um sistema de tempo-real e usa 4 tarefas periódicas para as 4 malhas de controle e uma aperiódica para atendimento do botão de emergência. As características das tarefas em termos de tempo de execução, período e deadline estão representadas na tabela abaixo (unidade *ms*):

Tabela 6: Conjunto de tarefas do sistema

	Computação	Período	Deadline	Comentário
T1	3	10	10	Malha da esteira 1
T2	3	10	10	Malha da esteira 2
T3	3	10	10	Malha da esteira 3
T4	2	100	100	Controle global
TA	1	---	---	Botão emergência

a) Determine se o conjunto das tarefas periódicas é escalonável usando o critério *Rate Monotonic* baseado em taxa de utilização.
b) Considere as tarefas periódicas todas ativadas em simultâneo no instante t=0. Determine os respectivos tempos de resposta.
c) Esboce um gráfico de *Gantt* da execução das tarefas periódicas durante o tempo necessário à verificação da escalonabilidade do conjunto.
d) Para efetuar a execução da tarefa aperiódica pretende-se utilizar um servidor adequado. Diga que tipo de servidor poderá ser usado, qual a prioridade a que deverá ser executado e quais as respectivas características.
e) Sabendo que as tarefas se comunicam através de uma porta de I/O comum a todas (inclusive a aperiódica), e que esta porta de I/O é usada em modo exclusivo durante 1ms, qual o bloqueio causado às várias tarefas, se for usado semáforos com o protocolo *Priority Inheritance Protocol*? (considere que o tempo de computação total da tarefa TA é acessando a região crítica).

12) Calcule o WCET da função *linear_search* (mostrado abaixo) usando análise estática.

```
#define SIZE 10
int linear_search(int *a, int n, int q) {
  unsigned int j=0;
  while (j<n && a[j]!=q) {
    j++;
  }
  if (j<SIZE) return 1;
  else return 0;
}
int main() {
  int a[SIZE];
  a[SIZE/2]=3;
  assert(linear_search(a,SIZE,3));
}
```

13) Calcule o WCET do código abaixo considerando que as operações de *lock* e *unlock* consomem 3 u.t., a operação de incremento e comparação consomem 2 u.t. e 1 u.t. respectivamente.

```
void *thread_A(void *arg) {
  pthread_mutex_lock(&mutex);
  A_count++;
  if (A_count == 1)
    pthread_mutex_lock(&lock);
  pthread_mutex_unlock(&mutex);

  pthread_mutex_lock(&mutex);
  A_count--;
  if (A_count == 0)
    pthread_mutex_unlock(&lock);
  pthread_mutex_unlock(&mutex);
}
```

```
void *thread_B(void *arg) {
  pthread_mutex_lock(&mutex);
  B_count++;
  if (B_count == 1)
    pthread_mutex_lock(&lock);
  pthread_mutex_unlock(&mutex);

  pthread_mutex_lock(&mutex);
  B_count--;
  if (B_count == 0)
    pthread_mutex_unlock(&lock);
  pthread_mutex_unlock(&mutex);
}
```

14) Exiba uma seqüência de execução que leve a bloqueio fatal para o código do exercício 13. Assuma que a prioridade da *thread* A é maior do que a prioridade da *thread* B. Depois disso, para tentar resolver o problema do bloqueio fatal do exercício 13, forneça o diagrama de *Gantt* assumindo:

a) O uso do protocolo de herança de prioridade
b) O uso do protocolo da prioridade teto

15) Implemente o algoritmo de herança de prioridade usando a linguagem ANSI-C ou C++. O seu algoritmo deve receber como entrada as Tabelas 3 e 4 e deve fornecer como saída um diagrama de *Gantt.*

16) Compare o desempenho dos algoritmos desenvolvidos nas questões 6 e 7 com os respectivos algoritmos implementados no Real Time Application Interface (RTAI) que está disponível em https://www.rtai.org/.

Data de entrega: 12 de novembro de 2013 (terça-feira).
Após esta data será descontado 2 pontos por dia de atraso.
A lista de exercícios deve ser resolvida e entregue individualmente.

29/09/2012