Sistemas de Tempo Real – Prova P1 (Rômulo) – 2013
Exercícios

---

1) O que significa uma anomalia temporal em um processador ? E como anomalias temporais afetam a determinação do tempo de resposta no pior caso ?

2) Em que situações o tempo de execução de uma instrução de máquina varia ?

3) Qual a utilidade das anotações na análise do tempo de execução no pior caso ?

4) Explique o conceito de contexto de fluxo (flow context).

1) Explique os principais passos da sincronização de relógios via rede

2) Discuta os vários fatores que afetam a qualidade de um timestamp (a) obtido localmente e (b) obtido através da rede.

3) Liste os fatores estáticos e dinâmicos que afetam a frequência de oscilação de um cristal de quarto ?

4) Explique os 4 tipos de erros na sincronização de relógios considerados pelo PTP.

TERMINOLOGIA:

O conjunto de tarefas

- é escalonável pelo algoritmo Z

- é viável na classe de algoritmos (x,y)-restritos

- pertence ao conjunto <x,y>

1) Um algoritmo Z é (x,y)-restricted. Isto significa que:

(a) O algoritmo Z consegue escalonar qualquer conjunto de tarefas que seja escalonável por algoritmos da classe (x,y)-restricted.

(b) O algoritmo Z consegue escalonar todos os conjuntos de tarefas pertencentes ao conjunto <x, y>.

(c) Não é possível afirmar quantos conjuntos de tarefas pertencentes ao conjunto <x,y> o algoritmo Z consegue escalonar.

(d) O algoritmo Z não consegue escalonar conjuntos de tarefas pertencentes ao conjunto <x,y>.

2) Se o algoritmo Z é ótimo com respeito à classe dos algoritmos (x,y)-restrito, então pode-se afirmar que:

(a) O algoritmo Z será ótimo em qualquer classe (k,y)-restrito, se k < x.

(b) O algoritmo Z será ótimo em qualquer classe (k,y)-restrito, se k > x.

(c) Qualquer conjunto de tarefas viável na classe (x,y)-restrito pode ser escalonado pelo algoritmo Z.

(d) Qualquer conjunto de tarefas escalonável por algum algorithm também será escalonado com sucesso pelo algoritmo Z.

3) No esquema particionado, considere a escala de tempo construida em um processador específico pelo EDF (Earliest Deadline First), para um dado conjunto de tarefas periódicas. Será sempre verdade que:

(a) Existe alguma política de atribuição de prioridades fixas para as tarefas capaz de reproduzir esta escala de tempo.

(b) Talvez exista alguma política de atribuição de prioridades fixas para as tarefas capaz de reproduzir esta escala de tempo.

(c) Não existe conjunto de tarefas tal que exista alguma política de atribuição de prioridades fixas para as tarefas capaz de reproduzir esta escala de tempo.

(d) Esta escala de tempo não é considerada uma escala de tempo válida para algoritmos de escalonamento da classe (3,x)-restrito.

4) Com respeito ao escalonamento particionado de multiprocessadores, pode-se afirmar que:

(a) Testes de escalonabilidade para monoprocessadores são inúteis.

(b) O algoritmo Best-Fit gera uma alocação ótima de tarefas em processadores.

(c) O teste de escalonabilidade de um processador precisa considerar as tarefas alocadas aos demais processadores.

(d) Heuristicas como Best-Fit podem não gerar uma alocação viável, ainda que uma exista.

5) Um algoritmo de escalonamento ser work-conserving significa que:

(a) Os processadores não ficam ociosos em nenhum momento.

(b) Pelo menos um processador não está ocioso a cada momento.

(c) Nenhum processador fica ocioso se houver algum job apto no sistema, não importa a classe de escalonamento.

(d) Nenhum processador fica ocioso se houver algum job apto capaz de executar naquele processador em questão.

6) Duas classes de algoritmos de escalonamento serem incomparáveis significa que:

(a) Todos os conjuntos de tarefas viáveis por uma também são viáveis pela outra, e vice-versa.

(b) Nenhum conjunto de tarefas viável por uma é também viável pela outra, e vice-versa.

(c) Existe pelo menos um conjunto de tarefas viável por uma que não é viável pela outra, e vice-versa.

(d) Todos os conjuntos de tarefas viáveis por uma também são viáveis pela outra, mas nem todos os conjuntos de tarefas viáveis pela outra são viáveis pela primeira.

7) No teorema 1 é possível afirmar que <1,1> C= <2,1> pois

(a) Toda escala de tempo gerada por um algoritmo de prioridade fixa também é uma escala que poderia ser gerada por um algoritmo de prioridade variável entre tarefas mas fixa entre jobs.

(b) Toda escala de tempo gerada por um algoritmo de prioridade fixa também é uma escala que poderia ser gerada pelo algoritmo EDF.

(c) Toda escala de tempo gerada por um algoritmo de prioridade variável entre tarefas mas fixa entre jobs também é uma escala que poderia ser gerada por um algoritmo de de prioridade fixa.

(d) Toda escala de tempo gerada por um algoritmo de prioridade variável entre tarefas mas fixa entre jobs é uma escala ótima, mesmo no contexto <1,1>.

8) Para provar que um conjunto de tarefas pertence ao conjunto <x,y> é necessário mostrar que:

(a) Este conjunto de tarefas é escalonável por qualquer algoritmo de escalonamento (x,y)-restrito.

(b) Este conjunto de tarefas é escalonável por pelo menos um algoritmo de escalonamento (x,y)-restrito.

(c) Este conjunto de tarefas é escalonável por qualquer algoritmo de escalonamento (z,k)-restrito, onde “z >= x” e “k >= y”.

(d) Este conjunto de tarefas é escalonável por qualquer algoritmo de escalonamento (z,k)-restrito, onde “z <= x” e “k <= y”.

9) Para mostrar que um conjunto de tarefas é viável na classe de algoritmos (2,1)-restrito:

- É necessário usar EDF ?

- Existe vantagem em usar outro algoritmo de escalonamento desta mesma classe ?

- É possível que seja escalonável por outro algoritmo desta classe e não seja escalonável por EDF ?

(a) Não, Não, Não

(b) Não, Sim, Sim

(c) Sim, Não, Não

(d) Sim, Não, Sim

(e) Sim, Sim, Sim

10) Com respeito a tabela 3, marque a falsa:

(a) Qualquer conjunto de tarefas escalonável com prioridade fixa também é escalonável com EDF, mantida a mesma política de migração.

(b) Qualquer conjunto de tarefas pertencente a algum conjunto <x,y> pode ser escalonado por um algoritmo que decide atribuir novas prioridades ou migrar os jobs a qualquer momento.

(c) O uso de EDF não é capaz de compensar a proibição de migração no que diz respeito à escalonabilidade de um conjunto de tarefas.

(d) Quando dois conjuntos <x,y> e <k,z> são incompatíveis então não existe um conjunto de tarefas específico que seja viável nas classes de algoritmos (x,y)-restrito e (k,z)-restrito.

11) Mostre por que o LLF (Least-Laxity-First) gera mais overhead do que o EDF através de um exemplo.

12) Aplique o teste de escalonabilidade do teorema 12 no conjunto de tarefas abaixo:

T1=(1,2)   T2=(3,6)   T3=(1,3)   T4=(2,3)   T5=(5,6)