

Teste de Tempo de Resposta

Arquivo fonte: `tempo_de_resposta.c` ou `tempo_de_resposta.cpp`

1. Tarefa

Este trabalho consiste na implementação do teste de escalonabilidade baseado no tempo de resposta máximo, aplicado a tarefas periódicas em que o *deadline* corresponde ao período, escalonadas segundo o algoritmo *Deadline Monotônico*.

O simulador deverá ler da entrada padrão um conjunto de valores que correspondem à definição de um conjunto de tarefas, conforme definido na Seção 2 (Entrada), e gerar na saída o resultado da aplicação do teste de escalonabilidade baseado no tempo de resposta máximo, tal como descrito por Farines, Fraga e Oliveira (2000), para cada conjunto de tarefas.

De acordo com Farines, Fraga e Oliveira (2000), o tempo de resposta máximo de uma tarefa i é calculado usando a seguinte fórmula iterativa:

$$R_i^{n+1} = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_j^n}{P_j} \right\rceil \times C_j$$

usando-se como ponto de partida:

$$R_i^0 = C_i$$

Onde:

- R_k^n é o tempo de resposta máximo da tarefa k na interação n ;
- C_k é o tempo de computação da tarefa k ;
- P_k é o período relativo da tarefa k ; e
- $hp(k)$ é o conjunto das tarefas com prioridade maior do que a prioridade da tarefa k .

Este método iterativo converge quando se tem uma utilização menor ou igual a 100%, sendo que o teste de escalonabilidade é feito para cada tarefa comparando-se o tempo de resposta obtido com o *deadline* relativo da tarefa: se o tempo de resposta for menor ou igual ao *deadline* relativo, então a tarefa é escalonável; caso contrário, não é.

2. Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém um número inteiro N , que indica o número de tarefas, e um número inteiro T , que indica o tempo de simulação (que NÃO será usado neste exercício). A seguir aparecem na entrada as descrições de cada uma das N tarefas. A descrição de cada tarefa é composta por três valores que correspondem respectivamente a tempo de computação da tarefa (C_i), período da tarefa (P_i) e *deadline* da tarefa (D_i). O final das entradas é indicado por $N = 0$ ou $T = 0$.

Os valores de entrada devem ser lidos da entrada padrão (normalmente o teclado) – por exemplo, com `scanf()` ou `getchar()`, em C –, de forma que seja possível redirecionar um arquivo para o processo. Não se deve utilizar arquivos de entrada, nem funções para esperar pelo pressionamento de teclas (comuns quando se depura um programa).

Exemplo de Entrada

```
3 20
2 10 6
2 10 8
```

8 20 16

3 36

3 9 6

4 18 12

4 12 10

3 210

2 10 9

4 15 8

10 35 20

0 0

3. Saída

Para cada conjunto de teste da entrada seu programa deve executar o teste de escalonabilidade baseado no tempo de resposta (conforme descrito neste documento) para cada conjunto de tarefas obtido da entrada padrão. O resultado de cada conjunto deve corresponder a uma linha para cada tarefa apresentando o tempo de resposta máximo da tarefa e um caractere para indicar se a tarefa é escalonável (“S”) ou não (“N”).

Como a entrada pode ser composta por vários conjuntos de teste, os resultados de cada conjunto deverão ser separados por uma linha em branco. A grafia mostrada no Exemplo de Saída, a seguir, que corresponde ao resultado esperado para o exemplo de entrada apresentado anteriormente, deve ser seguida rigorosamente.

Os valores de saída devem ser escritos na saída padrão (normalmente o vídeo) – por exemplo, com *printf()* ou *putchar()*, em C –, de forma que seja possível redirecionar a saída gerada pelo processo para um arquivo texto qualquer. Não se deve utilizar arquivos de saída, nem funções para limpar a tela.

Exemplo de Saída

2 S

4 S

16 S

3 S

18 N

7 S

6 S

4 S

24 N

(esta saída corresponde ao exemplo de entrada acima)

4. Restrições

$1 \leq N \leq 26$ ($N = 0$ apenas para indicar o final da entrada)

$1 \leq T \leq 100000$ ($N = 0$ apenas para indicar o final da entrada)

$1 \leq C_i \leq 100000$

$1 \leq P_i \leq 100000$

$1 \leq D_i \leq 100000$

5. Referências

FARINES, Jean-Marie; FRAGA, Joni da Silva; OLIVEIRA, Rômulo Silva de. Sistemas de Tempo Real. In: ESCOLA DE COMPUTAÇÃO, 12., 24 a 28 de julho de 2000, São Paulo. **Anais...** São Paulo: IME-USP, 2000. 208 p.

Autor: Roland Teodorowitsch