

Lista de Exercícios - Sistemas Operacionais

Aula 6: *Escalonamento ou Agendamento*

Professores: Valmir C. Barbosa e Felipe M. G. França

Assistente: Alexandre H. L. Porto

1. Suponha que um processo requeira, em média, um tempo T de execução antes que se bloqueie por E/S. Suponha também que uma comutação entre processos requeira um tempo S , o qual é efetivamente desperdiçado (*overhead*). Para o escalonamento por *round robin* com quantum Q , dê uma equação para a eficiência do processador (fração do tempo usado para executar os processos) para cada um dos seguintes casos:

(a) $Q = \infty$.

Resp.: Como T é menor do que Q (pois Q é infinitamente grande), então o processo executará no processador por um tempo T antes de bloquear, e depois será necessário fazer uma troca de contexto, o que irá requerer um tempo S . Logo, a eficiência do processador será de $T/(T + S)$.

(b) $Q > T$.

Resp.: Se Q for finito mais maior do que T , a eficiência será a mesma do Item 1a, isto é, $T/(T + S)$, pois o processo executará, assim como antes, por um tempo T antes de ser bloqueado.

(c) $S < Q < T$.

Resp.: Se Q for menor do que T , então teremos que fazer T/Q trocas de contexto até que o processo seja finalmente bloqueado.

Com isso, o tempo gasto com as trocas de contexto será de ST/Q . Logo, a eficiência do processador será de $T/(T + ST/Q)$, isto é, $Q/(Q + S)$.

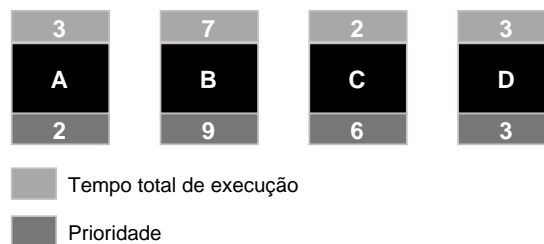
(d) $Q = S$.

Resp.: Assim como no caso anterior, teremos que executar $T/Q = T/S$ trocas de contexto. Logo, para obter a eficiência do processador, basta substituir, na equação dada no Item 1c, Q por S . Ao fazer isso, vemos que a eficiência do processador será de $S/(S + S) = S/2S = 1/2$.

(e) Q perto de zero.

Resp.: Quando Q tender a zero, o número de trocas de contexto tenderá a infinito. Com isso, a eficiência do processador tenderá a zero, pois quanto mais Q tender a 0, mais tempo será gasto com as trocas de contexto.

2. Suponha que os processos da figura dada a seguir tenham acabado de entrar no estado pronto, e que sejam os únicos processos em execução. Quais seriam os tempos de término destes processos, se o sistema operacional usasse, ao escalonar os processos:



Resp.: Existem várias respostas para cada um dos itens desta questão. Em cada um dos itens, vocês somente precisam fornecer uma destas respostas, que será considerada como correta se, dada a sua suposição para o item, a sua resposta estiver coerente com o algoritmo especificado no item e a sua suposição. Supondo que os tempos dos processos são dados na questão em uma certa unidade de tempo, vocês precisarão supor, no item (a) da questão, quantas unidades de tempo equivalem a um quantum. Já no item (b), vocês precisarão supor quantas unidades de

tempo equivalem à redução, em 1 unidade, da prioridade do processo em execução no processador. Além disso, no item (a), vocês precisam escolher uma ordem em que os processos começam a sua execução.

(a) o algoritmo de *round robin*.

Resp.: Na nossa resposta, vamos supor que cada quantum equivale a 1 unidade de tempo, e que as quatro escolhas iniciais do escalonador foram, em ordem, A, B, C e D. Lembrando que no algoritmo de *round robin* cada processo executa alternadamente no processador até terminar, obteremos a seqüência de execução dada na figura a seguir. A primeira linha mostra, da esquerda para a direita, a ordem de execução dos processos no processador. A segunda linha mostra, em cada coluna, o tempo decorrido após a execução do processo desta coluna na primeira linha. Como podemos ver pela figura, os tempos de término dos processos A, B, C e D serão de, respectivamente, 9, 15, 7 e 11 unidades de tempo.

-	A	B	C	D	A	B	C	D	A	B	D	B	B	B	B
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

(b) o algoritmo de prioridades.

Resp.: Nesta resposta, vamos supor que cada redução de 1 unidade da prioridade do processo em execução equivale a 1 unidade de tempo. Na Aula 6 vimos que, no algoritmo de prioridades, um processo executa até que a sua prioridade deixe de ser a maior de todas. Na figura a seguir mostramos, na primeira linha, a prioridade do processo antes de ele executar no processador. Na segunda linha mostramos, da esquerda para a direita, a ordem de execução dos processos no processador. Finalmente na última linha mostramos, para cada coluna, o tempo após o processo da coluna ter sido executado. Pela figura, vemos que os tempos de término dos processos A, B, C e D são de, respectivamente, 15, 9, 6 e 14 unidades de tempo.

-	9	8	7	6	6	5	5	4	3	3	2	2	1	1	0
-	B	B	B	B	C	C	B	B	B	D	D	A	A	D	A
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

(c) o algoritmo do trabalho mais curto primeiro.

Resp.: Como vimos na Aula 6, no algoritmo do trabalho mais curto primeiro, os processos são executados, em ordem crescente, de acordo com os seus tempos de execução. Além disso, quando um processo começa a executar, ele executa exclusivamente no processador até terminar. Então, uma das possíveis ordens de execução seria C, A, D e B (a outra seria C, D, A e B). O processo C então executaria do tempo 0 até o 2, o processo A do tempo 2 até o 5, o processo D do tempo 5 até o 8 e, finalmente, o processo B do tempo 8 até o 15. Logo, os tempos de término dos processos A, B, C e D seriam de, respectivamente, 5, 15, 2 e 8 unidades de tempo.

- Suponha que o sistema operacional use o algoritmo por prioridades ao escalonar os processos. Suponha ainda que a prioridade do processo em execução seja reduzida a cada unidade de tempo e que ele execute enquanto a sua prioridade é a maior. Será possível definirmos prioridades de modo que os processos do sistema sejam escalonados como seriam pelo algoritmo do trabalho mais curto primeiro, se soubermos o tempo de execução de todos os processos a serem executados? Justifique a sua resposta.

Resp.: Sim, será possível, se atribuirmos as prioridades corretas aos processos. Pelo enunciado da questão, vemos que a priori conhecemos quantos processos vão executar no sistema operacional. Vamos supor que existem n processos, e que P_1, P_2, \dots, P_n são os processos ordenados segundo uma ordem não-decrescente de acordo com os seus tempos de execução. Vamos supor também que o tempo de execução do processo P_i , $1 \leq i \leq n$, desta ordenação seja t_i unidades de tempo. Seja T a soma destes tempos, isto é, $T = \sum_{1 \leq i \leq n} t_i$. Ao processo P_1 vamos atribuir a prioridade $q_1 = T$, e ao processo P_i , $1 < i \leq n$, a prioridade $q_i = T - \sum_{1 \leq j \leq i-1} t_j$. Usando estas prioridades, os processos executarão de acordo com o algoritmo do trabalho mais curto primeiro, como veremos a seguir através de um exemplo. Suponha que 5 processos A, B, C, D e E, cujos tempos de execução são de, respectivamente, 3, 2, 4, 2

e 1 unidades de tempo, vão ser executados no sistema operacional. Ordenando estes processos em ordem não-decrescente de acordo com os tempos de execução, obteremos a ordem E, B, D, A e C para os processos (a ordem também poderia ser E, D, B, A e C). Logo $P_1 = E$, $P_2 = B$, $P_3 = D$, $P_4 = A$ e $P_5 = C$. A soma T dos tempos será $1 + 2 + 2 + 3 + 4 = 12$. As prioridades atribuídas aos processos serão $q_1 = 12$, $q_2 = 12 - 1 = 11$, $q_3 = 12 - (1 + 2) = 12 - 3 = 9$, $q_4 = 12 - (1 + 2 + 2) = 12 - 5 = 7$ e $q_5 = 12 - (1 + 2 + 2 + 3) = 12 - 8 = 4$ e, com isso, obteremos a ordem de execução dada na tabela a seguir. Nesta tabela damos as prioridades na primeira linha, os processos na segunda linha, e as unidades de tempo na última linha. Como podemos notar por esta tabela, cada processo executará até o seu término sem ser interrompido por um outro processo. Logo, as prioridades foram corretamente escolhidas, pois a ordem de execução dos processos, segundo o algoritmo do trabalho mais curto primeiro, será E, B, D, A e C, que será exatamente a ordem em que os processos serão executados pelo algoritmo de prioridades.

12	11	10	9	8	7	6	5	4	3	2	1
E	B	B	D	D	A	A	A	C	C	C	C
0	1	2	3	4	5	6	7	8	9	10	11

4. Suponha que um sistema operacional use o algoritmo de escalonamento por *round robin*, sendo que cada quantum equivale a duas unidades de tempo. Suponha ainda que os processos tenham sido escalonados no processador na ordem ABABABABBB, e que os processos usem todos os seus quanta. Qual será a nova sequência de escalonamento se o sistema operacional agora usar o algoritmo por prioridades, sendo que um processo executa até a sua prioridade, que é aumentada de 1 unidade a cada unidade de tempo, deixe de ser a menor, e que as prioridades iniciais dos processos A e B são, respectivamente, de 7 e 5?

Resp.: Pela ordem de execução, vemos que o processo A executou por 4 quanta e o processo B por 6 quanta. Agora, como cada quanta corresponde a duas unidades de tempo, e como os processos A e B usaram todos os seus quanta, então A e B executaram por, respectivamente, 8 e 12 unidades de tempo no processador. As duas tabelas dadas a seguir mostram a ordem de execução se agora usarmos o algoritmo por prioridades, sendo que cada processo executa até terminar

ou até algum outro processo possuir prioridade menor, e supondo que a prioridade é aumentada de uma unidade a cada unidade de tempo em execução. Para cada tabela, a primeira linha mostra a unidade de tempo em que cada processo foi executado. A segunda linha mostra os processos executados em cada unidade de tempo. Finalmente, a terceira linha mostra, para o processo de cada coluna, a prioridade deste processo antes de ele executar na unidade de tempo desta mesma coluna. Como podemos ver pelas tabelas, a sequência de execução agora será BBBAABBAABBAABBAABBB.

0	1	2	3	4	5	6	7	8	9
B	B	B	A	A	B	B	A	A	B
5	6	7	7	8	8	9	9	10	10

10	11	12	13	14	15	16	17	18	19
B	A	A	B	B	A	A	B	B	B
11	11	12	12	13	13	14	14	15	16