

Sistemas Operacionais

Segunda Lista de Exercícios

Norton Trevisan Roman
Clodoaldo Aparecido de Moraes Lima

14 de novembro de 2013

1. Diferencie os escalonamentos preemptivos e não-preemptivos.
2. Os computadores CDC 6600 podiam lidar simultaneamente com até 10 processos de E/S, usando uma forma interessante de round robin chamada **compartilhamento de processador**. Um chaveamento de processo ocorria depois de cada instrução; assim, a instrução 1 vinha do processo 1, a instrução 2 do processo 2 etc. O chaveamento era feito por um hardware especial e a sobrecarga era zero. Se um processo precisasse de T segundos para terminar sua execução, na ausência de competição, quanto tempo seria necessário se o compartilhamento do processador fosse usado com n processos?
3. Escalonadores round-robin normalmente mantêm uma lista de todos os processos que podem ser rodados, com cada processo ocorrendo exatamente uma vez na lista. O que aconteceria se um processo ocorresse duas vezes na lista? Você consegue pensar em uma razão para que se permita isso?
4. As medidas de um certo sistema mostram que o processo médio executa por um tempo T antes de ser bloqueado para E/S. Um chaveamento de processos requer um tempo S efetivamente gasto (sobrecarga). Para o escalonamento circular (round robin) com um quantum Q , dê uma fórmula para a eficiência (ou seja, o tempo útil dividido pelo tempo total de CPU) da CPU, para rodar um processo médio, em cada um dos seguintes casos:
 - (a) $Q = \infty$
 - (b) $Q > T$
 - (c) $S < Q < T$
 - (d) $Q = S$
 - (e) Q próximo de 0
5. Em um sistema batch, cinco tarefas estão esperando para serem executadas. Seus tempos de execução previstos são 9, 6, 3, 5 e X. Em que ordem elas deveriam ser executadas para minimizar o tempo médio de resposta?
6. Cinco tarefas em lote, A a E, chegam a um centro de computação quase ao mesmo tempo. Elas têm tempos de execução estimados em 10, 6, 2, 4 e 8 min. Suas prioridades, determinadas externamente, são 3, 5, 2, 1 e 4, respectivamente, com 5 sendo a mais alta. Para cada um dos seguintes algoritmos, determine o tempo médio de execução completa (mean turnaround time) desses processos. Ignore o tempo gasto com a troca de processos.
 - (a) round robin
 - (b) Escalonamento por prioridades
 - (c) First-come, First-served (na ordem 10, 6, 2, 4, 8)
 - (d) Shortest job first

Para (a), assuma que o sistema aceita multiprogramação, e que cada processo recebe uma fatia justa da CPU. Para (b) a (d) assuma que somente um processo pode rodar por vez, rodando até o fim. Todos os processos são CPU bound (sem E/S).

7. Um processo rodando com um escalonador de múltiplas filas (CTSS) necessita de 30 quanta para ser finalizado. Quantas vezes ele será colocado para rodar, incluindo a primeira vez (antes que tenha começado a rodar)?
8. Um sistema de tempo real tem quatro eventos periódicos com períodos de 50, 100, 200 e 250 ms cada. Suponha que os quatro eventos requeiram 35, 20, 10 e x ms de tempo de CPU, respectivamente. Qual o maior valor de x para que o sistema seja escalonável?
9. Com o valor de x do exercício anterior, ilustre o escalonamento dos processos segundo
 - (a) Rate Monotonic Scheduling
 - (b) Earliest Deadline First

Para tanto, suponha que todos os processos tentam rodar no instante 0.

10. Um sistema de tempo real precisa controlar duas chamadas de voz, cada uma executada a cada 5 ms e consumindo 1 ms do tempo da CPU por surto, além de um vídeo de 25 quadros/s, sendo que cada quadro requer 20 ms do tempo da CPU. Esse sistema pode ser escalonado? Justifique.
11. Ilustre o escalonamento dos processos do exercício anterior segundo
 - (a) Rate Monotonic Scheduling
 - (b) Earliest Deadline First

Para tanto, suponha que todos os processos tentam rodar no instante 0.

12. Considere que cinco processos sejam criados no instante de tempo 0 (P1, P2, P3, P4 e P5) e possuam as características descritas na tabela a seguir:

<i>Processo</i>	Tempo de UCP	Prioridade
P ₁	10	3
P ₂	14	4
P ₃	5	1
P ₄	7	2
P ₅	20	5

Qual o tempo de turnaround médio segundo os seguintes algoritmos:

- First Come, First Served
 - Shortest job first
 - Prioridade (número menor implica prioridade maior)
 - Circular com fatia de tempo igual a 2 u.t. (pressupondo multiprogramação)
13. Considere um sistema operacional que implemente escalonamento circular (round robin) com fatia de tempo igual a 10 u.t.. Em um determinado instante de tempo, existem apenas três processos (P1, P2 e P3) na fila de prontos, e o tempo de CPU de cada processo é 18, 4 e 13 u.t, respectivamente. Qual o estado de cada processo no instante de tempo T, considerando a execução dos processos P1, P2 e P3, nesta ordem, e que nenhuma operação de E/S é realizada?
 - (a) T = 8 u.t.
 - (b) T = 11 u.t.
 - (c) T = 33 u.t.
 14. Considere um sistema operacional que implemente escalonamento circular com fatia de tempo igual a 10 u.t. Em um determinado instante T = 0 u.t., existem apenas três processos (P1, P2 e P3) na fila de pronto, e o tempo de CPU de cada processo é 14, 4 e 12 u.t, respectivamente. Qual o estado de cada processo no instante de tempo T, considerando a execução dos processos P1, P2 e P3, nesta ordem, e que apenas o processo P1 realiza operações de E/S? Cada operação de E/S é executada após 5 u.t. e consome 10 u.t.

- (a) $T = 8$ u.t.
- (b) $T = 18$ u.t.
- (c) $T = 28$ u.t.

15. Existem quatro processos (P_1 , P_2 , P_3 e P_4) na fila de pronto, com tempos de UCP estimados em 9, 6, 3 e 5, respectivamente. Em que ordem os processos devem ser executados para minimizar o tempo de turnaround dos processos?

16. Considere a tabela a seguir onde

<i>Processo</i>	Tempo de UCP	Prioridade
P_1	40	4
P_2	20	2
P_3	50	1
P_4	30	3

Qual o tempo de turnaround médio dos processos considerando o tempo de troca de contexto igual a 0 e a 5 u.t. para os seguintes escalonamentos:

- (a) FCFS
- (b) SJF
- (c) Circular com fatia de tempo igual a 20 u.t.

17. De volta à tabela:

<i>Processo</i>	Prioridade
P_1	4
P_2	2
P_3	1
P_4	3

Seguindo o algoritmo de loteria, mostre a ordem de escalonamento dos processos acima, sabendo que a cada um é atribuído um número de bilhetes igual à sua prioridade, na ordem $P_1 \dots P_4$, e que os bilhetes sorteados são 10, 5, 1, 9, 4, 2, 6, 8, 3, 7.

18. Um usuário possui 5 processos (A_1 a A_5), enquanto que outro possui 2 (B_1 e B_2). Ilustre uma possível ordem de escalonamento, para o algoritmo fair-share, em que são feitas as seguintes promessas ao usuário:

- (a) Cada usuário terá a mesma quantia da CPU
- (b) O usuário A terá o dobro do tempo de CPU que o usuário B
- (c) O usuário B terá o triplo do tempo de CPU que o usuário A

19. Em um sistema operacional, o escalonador utiliza duas filas. A fila A contém os processos do pessoal do CPD e a fila B contém os processos dos alunos. O algoritmo dentro das filas é fatia de tempo (Round Robin). De cada 11 unidades de tempo de processador, 7 são fornecidas para os processos da fila A, e 4 para os processos da fila B. O tempo de cada fila é dividido entre os processos também por fatias de tempo, com fatias de 2 unidades para todos. A tabela abaixo mostra o conteúdo das duas filas no instante zero. Considere que está iniciando um ciclo de 11 unidades, e agora a fila A vai receber as suas 7 unidades de tempo. Mostre a sequência de execução dos processos, com os momentos em que é feita a troca. Obs. Se terminar a fatia de tempo da fila X no meio da fatia de tempo de um dos processos, o processador passa para a outra fila. Entretanto, esse processo permanece como primeiro da fila X e recebe novo quantum.

	Fila A			Fila B		
<i>Processo</i>	P_1	P_2	P_3	P_4	P_5	P_6
<i>Duração</i>	6	5	7	3	8	4

20. O sistema operacional Linux utiliza dois algoritmos de escalonamentos, um de tempo compartilhado e outro baseado em prioridades para tratar tarefas de tempo real. Cada processo é associado a uma classe de escalonamento. No algoritmo de escalonamento para processo de tempo compartilhado, o Linux usa um algoritmo por prioridade, baseado em créditos. Cada processo possui um determinado número de créditos (inicialmente, o número de créditos é igual à prioridade do processo); o processo com maior número de créditos na fila de prontos é selecionado pelo escalonador. A cada interrupção do clock (1ms), o processo em execução perde um crédito; quando seu crédito chega a zero, o escalonador é ativado para selecionar outro processo para ganhar o processador. Se nenhum processo na fila de prontos tiver créditos, o algoritmo faz uma nova atribuição de créditos a todos os processos (inclusive aos processos bloqueados), de acordo com a seguinte regra: $\text{créditos} = \text{créditos}/2 + \text{prioridade}$. Avalie o algoritmo de escalonamento por prioridade usado pelo Linux e o Algoritmo por Alternância Circular (quantum = 3ms) quanto à utilização da CPU e tempo médio de resposta. Para tanto, considere o seguinte volume de trabalho:

<i>Ordem</i>	<i>Processo</i>	<i>Surto de CPU*</i>	<i>Duração de E/S</i>	<i>Tempo total de CPU</i>	<i>Prioridade</i>
1	A	2 ms	5 ms	6 ms	3
2	B	3 ms	10 ms	6 ms	3
3	C	–	–	14 ms	3
4	D	–	–	10 ms	3

(*) tempo de CPU necessário antes de cada solicitação de E/S (processos A e B ficam alternando entre surtos de CPU e em operações de E/S)