

Sistemas de Programação II

INF 02828

(Engenharia de Computação / Elétrica)

Sistemas Operacionais

INF 02780

(Ciência da Computação)

Exercício - Escalonamento

Período: 2010/1

1. 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 processos de tempo compartilhado, o Linux usa um algoritmo por prioridades, 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 o maior número de créditos na fila de prontos é selecionado pelo escalonador. A cada interrupção do temporizador (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 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 Revezamento (quantum = 3ms) quanto à utilização da CPU e tempo total de execução dos processos (turnaround). Para tanto, considere o seguinte volume de trabalho (processos de A a D chegam no sistema ao mesmo tempo):

ordem	processo	surto de CPU*	duração de I/O	tempo total de CPU	prioridade
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 I/O (processos A e B ficam alternando entre surtos de CPU e em operações de I/O).

2. Cinco processos, de A até E, chegam ao computador ao mesmo tempo. Eles têm seus tempos de processamento estimados em 10, 6, 2, 4 e 8 minutos respectivamente. Suas prioridades (atribuídas externamente) são 3, 5, 2, 1 e 4, respectivamente, sendo 5 o representante da prioridade mais alta. Nenhum dos processos faz I/O. Para cada um dos algoritmos de escalonamento abaixo, determine o tempo médio de turnaround dos processos. Ignore o overhead causado pela troca de contexto.

- (a) Round Robin
- (b) Escalonamento com prioridade
- (c) FIFO (ordem de execução: 10, 6, 2, 4, 8)
- (d) SJF

Para (a), assuma que a fila começa em A, indo em ordem até E (quantum = 4).