

## Teste 2 de Sistemas Operacionais

Nome: Daniel Sant' Anna Andrade

Matrícula: 20200036904

Nome: Luan Gadioli Mendonça Berto

Matrícula: 20200034042

- 1) Um item ser atômico nos diz que ele é “indivisível”, ou seja, dois processadores ou tarefas não podem entrar na mesma região do código ao mesmo tempo, conhecido como “regiões críticas”.
- 2) Sim, faz sentido usar uma barreira para sincronizá-los. Quando todos os processos precisam alcançar um mesmo estado, antes de prosseguir eles impõem uma barreira no fim de cada fase. Processos que alcançam a barreira são bloqueados e os processos são liberados quando todos alcançam a barreira. Algumas aplicações são divididas em fases, onde um processo não deve seguir para a fase seguinte até que os outros processos atinjam o estado de prontidão e ocorra a sincronização entre eles.
- 3) Se  $0 < X \leq 3$ , então (X, 3, 5, 6, 9).  
Se  $3 < X \leq 5$ , então (3, X, 5, 6, 9).  
Se  $5 < X \leq 6$ , então (3, 5, X, 6, 9).  
Se  $6 < X \leq 9$ , então (3, 5, 6, X, 9).  
Se  $X > 9$ , então (3, 5, 6, 9, X).
- 4) Nem sempre a memória principal é suficiente para comportar todos os processos, às vezes é preciso manter algumas tarefas no disco. O escalonador de processos de 2 níveis escolhe o processo que tem mais prioridade e menos tempo e coloca-o na memória principal, ficando os outros alocados em disco; com essa execução o processador evita ficar ocioso.
- 5) Nessa questão temos 3 eventos, duas chamadas de voz e uma chamada de vídeo. Para determinar se esse sistema é escalonável, deve-se atender ao seguinte critério
$$\sum_{i=1}^m \frac{C_i}{P_i} \leq 1$$
Como a chamada de vídeo executa 25 quadros/s e cada quadro requer 20ms de tempo de CPU, P será 1 segundo e C será  $25 * 20 \text{ ms}$ , dessa forma C é 500ms (0,5s).

Utilizando a fórmula:

$$\sum_{i=1}^3 \frac{0,001}{0,005} + \frac{0,001}{0,005} + \frac{0,5}{1} = 0,9 \leq 1$$

Assim, o resultado de 0,9 é menor que 1, sendo assim um sistema escalonável.

6) A fórmula para encontrar a previsão da próxima execução é

$$T_{n+1} = at_n + (1 - a)T_n$$

ou resumindo:

$$T_{n+1} = t_n/2 + T_n/2$$

Logo:

$$T_0 = t_0 = 40$$

$$t_1 = 20$$

$$t_2 = 40$$

$$t_3 = 15$$

$$T_4 = t_3/2 + t_2/4 + t_1/8 + t_0/16 + T_0/16$$

$$T_4 = 15/2 + 40/4 + 20/8 + 40/16 + 40/16$$

$$T_4 = 7,5 + 10 + 2,5 + 2,5 + 2,5$$

$$T_4 = 25ms$$

Logo, a estimativa para o próximo tempo da questão é de 25ms.

7)

a) (First-Come, First Served), ou seja, atender as tarefas em sequência, à medida em que elas se tornam prontas, conforme sua ordem de chegada.

tarefa a: 10 min

tarefa b: 16 min

tarefa c: 18 min

tarefa d: 22 min

tarefa e: 30 min

b) Esse algoritmo proporciona os menores tempos médios de espera das tarefas.

tarefa a: 30 min

tarefa b: 12 min

tarefa c: 2 min  
tarefa d: 6 min  
tarefa e: 20 min