

Aluno(a).: \_\_\_\_\_

Professor.: Ronilson R. Pinho

1. Com relação ao gerenciamento de memória com paginação em sistemas operacionais, assinale a opção correta.
    - a) As páginas utilizadas por um processo, sejam de código ou de dados, devem ser obrigatoriamente armazenadas na partição de swap do disco, quando o processo não estiver sendo executado.
    - b) Todas as páginas de um processo em execução devem ser mantidas na memória física enquanto o processo não tiver terminado.
    - c) Um processo somente pode ser iniciado se o sistema operacional conseguir alocar um bloco contíguo de páginas do tamanho da memória necessária para execução do processo.
    - d) O espaço de endereçamento virtual disponível para os processos pode ser maior que a memória física disponível.
    - e) Um processo somente pode ser iniciado se o sistema operacional conseguir alocar todas as páginas de código desse processo.
  2. Sistemas operacionais de tempo real são utilizados em controle de processos automatizados, em que o tempo de resposta a determinados eventos é um fator crítico. Com relação a esse assunto, julgue os itens seguintes.
    - I. Sistemas de tempo real estritos (hard real-time) não utilizam dispositivos de memória secundária (como discos), pois estes não oferecem garantia de término das operações dentro de uma quantidade máxima de tempo.
    - II. Um sistema operacional de propósito geral pode ser modificado para ser de tempo real atribuindo-se prioridades fixas para cada um dos processos.
    - III. O escalonamento mais utilizado por sistemas operacionais de tempo real é o shortest-job-first (tarefa mais curta primeiro).
- Assinale a opção correta.
- a. Apenas um item está certo.
  - b. Apenas os itens I e II estão certos.
  - c. Apenas os itens I e III estão certos.
  - d. Apenas os itens II e III estão certos.

e. Todos os itens estão certos.

3. O problema do buffer limitado de tamanho N é um problema clássico de sincronização de processos: um grupo de processos utiliza um buffer de tamanho N para armazenar temporariamente itens produzidos; processos produtores produzem os itens, um a um, e os armazenam no buffer; processos consumidores retiram os itens do buffer, um a um, para processamento. O problema do buffer limitado de tamanho N pode ser resolvido com a utilização de semáforos, que são mecanismos de software para controle de concorrência entre processos. Duas operações são definidas para um semáforo s: wait(s) e signal(s).

Considere o problema do buffer limitado de tamanho N cujos pseudocódigos dos processos produtor e consumidor estão mostrados na tabela abaixo. Pode-se resolver esse problema com a utilização dos semáforos mutex, cheio e vazio, inicializados, respectivamente, com 1, 0 e N.

Processo produtor	Processo consumidor
Produz item	Comando_e
Comando_a	Comando_f
Comando_b	Retira do buffer
Coloca no buffer	Comando_g
Comando_c	Comando_h
Comando_d	Consome o item

A partir dessas informações, para que o problema do buffer limitado de tamanho N cujos pseudocódigos foram apresentados possa ser resolvido a partir do uso dos semáforos mutex, cheio e vazio, é necessário que comando\_a, comando\_b, comando\_c, comando\_d, comando\_e, comando\_f, comando\_g e comando\_h correspondam, respectivamente, às operações

- wait(vazio), wait(mutex), signal(mutex), signal(cheio), wait(cheio), wait(mutex), signal(mutex) e signal(vazio).
- wait(cheio), wait(mutex), signal(mutex), signal(vazio), wait(vazio), signal(mutex), signal(mutex) e wait(cheio).

- c. wait(mutex), wait(vazio), signal(cheio), signal(mutex), wait(mutex), wait(vazio), signal(cheio) e signal(mutex).
- d. wait(mutex), wait(vazio), signal(cheio), signal(mutex), wait(mutex), wait(cheio), signal(vazio) e signal(mutex).
- e. wait(vazio), signal(mutex), signal(cheio), wait(mutex), wait(cheio), signal(mutex), signal(vazio) e signal(mutex).

4. Uma antiga empresa de desenvolvimento de software resolveu atualizar toda sua infraestrutura computacional adquirindo um sistema operacional multitarefa, processadores multicore (múltiplos núcleos) e o uso de uma linguagem de programação com suporte a threads. O sistema operacional multitarefa de um computador é capaz de executar vários processos (programas) em paralelo. Considerando esses processos implementados com mais de uma thread (multithreads), analise as afirmações abaixo.

- I. Os ciclos de vida de processos e threads são idênticos.
- II. Threads de diferentes processos compartilham memória.
- III. Somente processadores multicore são capazes de executar programas multithreads.
- IV. Em sistemas operacionais multitarefa, threads podem migrar de um processo para outro.

É correto apenas o que se afirma em

- a. I.
- b. II.
- c. I e III.
- d. I e IV.
- e. II e IV.

5. Calcule waiting time, turnaround time e média de turnaround time utilizando os algoritmos FCFS, SJF, SJF com preempção e round-robin (quantum=4) para os seguintes casos

a)

P	PT	AT
P1	7	0
P2	3	2
P3	5	8

b)

P	PT	AT
P1	12	5
P2	8	10
P3	5	14
P4	8	14

c)

P	PT	AT
P1	5	0
P2	10	7
P3	7	10
P4	4	12
P5	8	12
P6	5	20
P7	5	27