

Sistemas Operacionais

Primeira Prova - 02/06/2022

NOME: _____

1. O escalonamento de processos depende de estruturas eficientes para reduzir o tempo gasto com a troca de contexto. Nesse sentido, considerando que processos podem estar no estado de prontos ou em espera, é melhor ter estruturas diferentes para cada estado ou apenas uma estrutura com campo para identificar o estado? Justifique. (VALOR 1,5 pontos)

A melhor solução é usar apenas uma lista, pois se usarmos duas listas teremos que mover processos de uma lista para a outra toda vez que ele mudar de estado (pronto/espera), o que é mais lento do que se verificar o valor de um campo de cada elemento da lista.

2. Um empresário do setor industrial diz que seus analistas examinaram dois sistemas operacionais, A e B, que fazem escalonamento por prioridade baseado no tempo médio de cpu consumido entre duas operações de E/S. Nos testes efetuados com os mesmos conjuntos de processos, os valores de tempos de espera do sistema A foram bem maiores do que os do sistema B. Como isso pode ser explicado? (VALOR 2 pontos)

Uma explicação possível é de que a prioridade para o sistema A é o inverso da prioridade do sistema B, fazendo com que tenham resultados contrários. Por exemplo, o sistema A iria priorizar processos mais longos entre cada E/S, aumentando a espera dos demais.

Outra explicação poderia estar ligada a como cada SO trata a execução da operação de E/S, fazendo com que no sistema A isso seja feito de modo mais lento, por exemplo.

3. Os mecanismos de endereçamento fazem a tradução entre endereço lógico e endereço físico, normalmente com o uso de hardware especial. Caso esse hardware não exista, como esse endereçamento poderia ser feito? (VALOR 2 pontos)

O endereçamento teria que ser feito por software (provavelmente microprogramado na própria CPU). Isso é bastante ineficiente, sendo aceitável apenas com o uso de relocação estática, quando os endereços teriam que ser calculados apenas uma vez, durante o carregamento.

4. Algoritmos de paginação dependem, em parte, de os processos usarem algoritmos e estruturas de dados que atendam o princípio da localidade. Como seria essa dependência caso usássemos *swapping* de segmentos inteiros em vez de páginas? (VALOR 1.5 pontos)

A localidade diz respeito a acessar, nas próximas instruções, endereços próximos do atual (nas mesmas páginas). Como segmentos contém mais instruções/dados, então é mais fácil manter a localidade, ainda mais considerando que ter mudanças de segmentos são menos frequentes do que mudanças de páginas.

5. Considerando um sistema com limite no número de processos simultâneos, comente qual o impacto em aumentar esse número para os casos em que o sistema tenha: (VALOR 3 pontos)
- a) Ocupação da CPU = 20% e uso de paginação = 90%
 - b) Ocupação da CPU = 80% e uso de paginação = 5%
 - c) Ocupação da CPU = 20% e uso de paginação = 5%

a) O desempenho **DEVE PIORAR** pois o nível de paginação indica que já falta espaço para os processos atualmente executando.

b) O desempenho **PODE PIORAR** pois a alta ocupação da CPU indica que deve haver pouco espaço para novos processos.

c) O desempenho **DEVE MELHORAR** pois os atuais níveis de ocupação de CPU e memória estão baixos e podem ser aumentados.

O problema de thrashing **NÃO** está relacionado ao intenso uso da CPU e sim à grande necessidade de paginação. Então, ter 100% de uso da CPU é objetivo de se fazer compartilhamento entre processos.

Boa prova,