

Leonardo Vecchi Meirelles - 12011 ECP002

Questão 1) O escalonamento por loteria é simples de implementar, necessitando basicamente de um gerador de números randômicos. Primeiramente, cada processo recebe uma determinada quantidade de tickets, em seguida, um contador simples determina o vencedor.

O contador verifica se o valor "vencedor" excede o valor de cada processo na lista. Se exceder, o último processo é o selecionado.

Questão 2) O CFS mantém um tempo de execução virtual para cada processo, que é proporcional à sua participação na CPU. Processos com uma porção maior da CPU têm um tempo de execução virtual maior e, portanto, são agendados com mais frequência. O CFS também usa círculos red-black para acompanhar os processos em uma estrutura de dados balanceada para agendamento eficiente.

Questão 3) A localidade temporal refere-se à tendência de um programa acessar os mesmos locais de memória várias vezes em um curto período de tempo. Ao armazenar esses dados em um cache local, o programa pode reduzir o número de vezes que precisa acessar a memória principal, melhorando o desempenho. A localidade espacial refere-se à tendência de um programa acessar locais de memória próximos uns dos outros. Ao armazenar um bloco de memória em cache, o programa pode reduzir o acesso à memória, melhorando o desempenho.

Questão 4) O problema com a sincronização no escalonamento multiprocessador é que ele pode introduzir sobrecarga e complexidades adicionais. Os algoritmos de escalonamento devem considerar não apenas o desempenho de tarefas individuais, mas também as interações entre as tarefas e seu impacto nos recursos compartilhados e na coerência do cache.

Questão 5) A ideia básica por trás do SQMS é manter uma fila global de processos executáveis que seja acessível a todos os processadores do sistema. Cada processador pode selecionar o próximo processo a ser executado a partir dessa fila, com base num algoritmo de escalonamento como round-robin ou escalonamento prioritário.

Um problema que ele pode sofrer de balanceamento de carga ruim, principalmente em sistemas com processadores heterogêneos. Se algumas tarefas exigirem mais recursos do que outras, elas podem monopolizar um processador por um longo período de tempo deixando outros processos ociosos.



Questão 6) A ideia por trás do MANS é distribuir a carga de trabalho uniformemente entre os processadores disponíveis, atribuindo a cada processador sua própria fila de processos executáveis.

Um problema é que pode ser difícil manter o equilíbrio das filas, se alguns processadores tiverem mais tarefas do que os outros, ou se tarefas exigirem mais recursos do que outros, as filas podem ficar desbalanceadas e alguns processadores podem ficar ociosos.

Questão 7) Um espaço de endereçamento é um intervalo de endereços de memória que um programa ou processo pode acessar. É o conjunto de todos os endereços possíveis que um programa pode usar para referenciar locais de memória, incluindo memória física e virtual.

Questão 8) Nesse modelo, o heap cresce no sentido de aumentar os endereços de memória, enquanto a stack cresce no sentido contrário. A razão para isso está relacionada à maneira como a memória é alocada e desalocada em um programa.

Quando a memória é alocada no heap, o SO atribui um bloco de memória ao programa incrementando o program's memory pointer permitindo que o programa acesse memória recém-alocada sem sobrescrever dados existentes.

Por outro lado, na stack, o program's memory pointer é decrementado. Isso permite que quando uma chamada de função retorna, a memória na pilha é desalocada incrementando o ponteiro.

Questão 9) A transparência envolve o gerenciamento da memória sem que o usuário ou o programa percebam, enquanto a eficiência envolve minimizar a sobrecarga e maximizar o desempenho. A proteção envolve proteger a memória de outros processos do Sistema Operacional.

