

1) A diferença entre sistemas de multiprogramação e de tempo compartilhado está na maneira como os processos são executados e compartilham recursos.

- Multiprogramação: Nesse tipo de sistema, vários processos são carregados na memória ao mesmo tempo, mas a CPU executa um processo por vez. Quando um processo é interrompido, outro é imediatamente iniciado, o que permite que a CPU seja mantida ocupada. A multiprogramação visa maximizar a utilização da CPU, especialmente em cenários em que os processos frequentemente aguardam E/S. No entanto, um processo não é executado continuamente, e outros processos são intercalados durante a execução.

- Tempo compartilhado (ou multitarefa): Nesse tipo de sistema, a CPU é compartilhada entre vários processos em intervalos muito curtos de tempo. Isso cria a ilusão de que vários processos estão sendo executados simultaneamente, embora a CPU esteja alternando rapidamente entre eles. O tempo compartilhado é comum em sistemas interativos, nos quais os usuários esperam que suas ações sejam respondidas instantaneamente.

2) Antes da introdução do DMA (Direct Memory Access), a CPU era responsável por tratar todas as operações de leitura e escrita de dados diretamente na memória ou dispositivos de E/S. Isso tinha implicações significativas para a multiprogramação, pois a CPU precisava dedicar uma quantidade significativa de tempo e recursos para a transferência de dados entre a memória e os dispositivos de E/S.

Sem o DMA, a CPU precisava executar operações de E/S de maneira síncrona, o que significava que o processo que solicitou a E/S ficaria bloqueado até que a operação fosse concluída. Isso resultava em ineficiência, pois a CPU ficava inativa enquanto aguardava a conclusão das operações de E/S, reduzindo a capacidade de atender a outros processos em execução. Com o DMA, as operações de E/S podem ser realizadas sem a intervenção direta da CPU, permitindo que a CPU execute outras tarefas, melhorando a multiprogramação e a eficiência geral do sistema.

3) As instruções relacionadas ao acesso a dispositivos de E/S são tipicamente instruções privilegiadas porque essas operações têm o potencial de afetar diretamente o funcionamento do hardware e podem ser usadas para contornar as proteções do sistema operacional. Algumas razões para essas instruções serem privilegiadas incluem:

- Controle de recursos: As instruções de E/S controlam o acesso a dispositivos físicos, como discos rígidos, impressoras, rede, etc. Permitir que processos de usuário executem essas instruções sem restrições poderia levar a conflitos, disputas de recursos e impactos negativos no sistema como um todo.

- Segurança: Instruções de E/S podem ser usadas para acessar áreas críticas da memória ou interromper operações normais do sistema. Privilegiar essas instruções ajuda a evitar que programas maliciosos causem danos ao sistema.

- Integridade do sistema: Instruções de E/S podem afetar diretamente a integridade e estabilidade do sistema. Privilegiá-las permite que o sistema operacional mantenha o controle sobre essas operações e assegure que elas ocorram de maneira ordenada e segura.

4) Para calcular o tempo total necessário para completar a execução dos três programas em um sistema com duas CPUs, onde cada CPU tem duas threads, você precisa alocar os programas para as CPUs e threads disponíveis. Assumindo que os programas não trocam de CPU uma vez atribuídos e que são 100% CPU intensivos, a alocação seria a seguinte:

- CPU 1, Thread 1: P0 (5 ms)
- CPU 1, Thread 2: P1 (10 ms)
- CPU 2, Thread 1: P2 (20 ms)
- CPU 2, Thread 2: Ociosa

Nesse cenário, os programas são executados simultaneamente, e o tempo total para a conclusão dos três programas será determinado pelo programa que levar mais tempo para terminar, ou seja, P2 com 20 ms.

Portanto, o tempo total necessário para completar a execução dos três programas é de 20 ms.

5) Para calcular o tempo médio de acesso a um byte em um sistema com memória cache, RAM e disco, levando em consideração as taxas de acerto da cache e da RAM, podemos usar o conceito de hierarquia de memória. O tempo médio de acesso (TMA) pode ser calculado usando a seguinte fórmula:

$$\text{TMA} = \text{Taxa de acerto da cache} * \text{Tempo de acesso à cache} + (1 - \text{Taxa de acerto da cache}) * (\text{Taxa de acerto da RAM} * \text{Tempo de acesso à RAM} + (1 - \text{Taxa de acerto da RAM}) * \text{Tempo de acesso ao disco})$$

Substituindo os valores fornecidos:

- Taxa de acerto da cache = 95% = 0,95

- Tempo de acesso à cache = 1 ns
- Taxa de acerto da RAM = 99% = 0,99
- Tempo de acesso à RAM = 10 ns
- Tempo de acesso ao disco = 10 ms = 10,000,000 ns

Calculando o TMA:

$$\text{TMA} = 0,95 * 1 \text{ ns} + (1 - 0,95) * (0,99 * 10 \text{ ns} + (1 - 0,99) * 10,000,000 \text{ ns})$$

$$\text{TMA} = 0,95 \text{ ns} + (0,05) * (9,9 \text{ ns} + 10,000,000 \text{ ns})$$

$$\text{TMA} = 0,95 \text{ ns} + 0,05 * 10,009,900 \text{ ns}$$

$$\text{TMA} = 0,95 \text{ ns} + 500,495 \text{ ns}$$

$$\text{TMA} \approx 501,445 \text{ ns}$$

Portanto, o tempo médio de acesso a um byte nesse sistema é aproximadamente 501,445 nanossegundos.