

1) Os manipuladores de interrupção são frequentemente escritos em linguagem de montagem por algumas razões:

- Eficiência: A linguagem de montagem permite um controle detalhado sobre o hardware e a execução do código, o que é fundamental para o tratamento de interrupções, que geralmente requerem um tempo de resposta muito rápido. Escrever manipuladores em linguagem de montagem permite otimizar o código para garantir a menor latência possível.

- Acesso direto ao hardware: Os manipuladores de interrupção frequentemente precisam acessar diretamente o hardware, como registradores de CPU, para lidar com eventos de hardware. A linguagem de montagem permite esse nível de acesso.

- Portabilidade: Escrever manipuladores de interrupção em linguagem de montagem garante que eles sejam específicos para a arquitetura do processador. Isso pode ser uma vantagem para otimização e controle, mas também significa que os manipuladores devem ser escritos ou adaptados para cada arquitetura específica.

2) O Process Control Block (PCB) é uma estrutura de dados utilizada pelos sistemas operacionais para representar e controlar processos. Ele contém informações importantes sobre o processo e seu estado atual. O PCB serve para:

- **Gerenciamento de processos**: Armazena informações sobre o processo, como identificação única (PID), estado atual, registros de CPU, prioridade, estado de E/S e informações de gerenciamento de memória. Esses detalhes são essenciais para o sistema operacional manter o controle sobre os processos em execução.

- **Comutação de contexto**: O PCB é usado para salvar e restaurar o contexto de execução do processo. Quando um processo é suspenso e outro é iniciado, o PCB é usado para armazenar o estado do processo em execução, de modo que ele possa ser restaurado mais tarde.

- **Escalonamento de CPU**: O PCB contém informações que ajudam o escalonador a tomar decisões sobre quais processos devem ser executados e em que ordem.

3) Transições de estados possíveis para um processo:

a) Em execução → pronto: Isso é possível quando um processo em execução é interrompido (por exemplo, devido a uma interrupção de tempo ou uma operação de E/S concluída) e precisa aguardar na fila de processos prontos para ser retomado.

b) Em execução → bloqueado: Isso ocorre quando um processo em execução realiza uma operação de E/S que levará algum tempo para ser concluída. O processo é colocado no estado bloqueado até que a E/S seja finalizada.

c) Pronto → em execução: O escalonador seleciona um processo pronto para ser executado e o move para o estado de execução.

d) Pronto → bloqueado: Um processo que está pronto para ser executado realiza uma operação de E/S que não pode ser concluída imediatamente, então ele é movido para o estado bloqueado.

e) Bloqueado → pronto: Um processo bloqueado termina uma operação de E/S e é movido de volta para o estado pronto para ser retomado pela CPU.

f) Bloqueado → em execução: Isso só é possível se um processo em execução for desbloqueado imediatamente após uma operação de E/S (por exemplo, se a E/S for concluída instantaneamente). Nesse caso, o processo bloqueado é movido para o estado de execução.