

Questões do capítulo 2

Juciane de Azevedo Chianca

Q1 – PCB é a nomenclatura dada à estrutura de dados responsável por armazenar as informações relativas ao contexto e os demais dados necessários à gerência de uma determinada tarefa presente no sistema.

Um PCB contém:

- Identificador da Tarefa;
- Estado da tarefa;
- Informações de contexto do processador;
- Lista de áreas de memória usadas pela tarefa;
- Listas de arquivos abertos, conexões de redes e outros recursos usados pela tarefa;
- Informações de gerência e contabilização;

Q2 - Time-sharing é o termo para um tipo de multiprocessamento quando o sistema operacional implementa uma técnica de partilhar o uso da cpu por intervalo de tempo. Cada processo que recebe o controle tem um intervalo de tempo para processar e depois devolver o controle, que será repassado para outro processo e assim sucessivamente. Graças ao time-sharing pode-se ter dezenas de aplicações abertas em um SO, por exemplo.

Q3 - A duração do quantum depende muito do tipo de sistema operacional; no Linux ela varia de 10 a 200 milissegundos, dependendo do tipo e prioridade da tarefa.

Q4 -

Q5 –

- $E \rightarrow P$: Possível (Quando o quantum de uma tarefa é esgotado);
- $E \rightarrow S$: Possível (Esperando por um dado externo);
- $S \rightarrow E$: não
- $P \rightarrow N$: não;
- $S \rightarrow T$: não;
- $E \rightarrow T$: Possível (Processo terminado);
- $N \rightarrow S$: não;
- $P \rightarrow S$: não.

Q6 – [N] O código da tarefa está sendo carregado.

[P] A tarefas são ordenadas por prioridades.

[E] A tarefa sai deste estado ao solicitar uma operação de entrada/saída.

[T] Os recursos usados pela tarefa são devolvidos ao sistema.

[P] A tarefa vai a este estado ao terminar seu quantum.

- [P] A tarefa só precisa do processador para poder executar.
- [S] O acesso a um semáforo em uso pode levar a tarefa a este estado.
- [E] A tarefa pode criar novas tarefas.
- [E] Há uma tarefa neste estado para cada processador do sistema.
- [S] A tarefa aguarda a ocorrência de um evento externo.

Q7 –

Q8 –

Q9 - Cada fluxo de execução do sistema, seja associado a um processo ou no interior do núcleo, é denominado thread. Servem para que mais de uma tarefa possa ser executada em um processo.

Q10 – Vantagens – permite que várias ações sejam executadas em paralelo por um mesmo processo;

A comunicação entre threads é mais rápida do que a comunicação entre processos - porque as threads compartilham tudo: espaço de endereçamento, variáveis globais etc.

Desvantagens – Complexidade de implementação.

Q11 – A implementação multi-thread é pouco escalável, inviabilizando aplicações com muitas tarefas, como grandes servidores web e simulações de grande porte.

Q12 - [A] Tem a implementação mais simples, leve e eficiente.

[C] Multiplexa os threads de usuário em um pool de threads de núcleo.

[B] Pode impor uma carga muito pesada ao núcleo.

[A] Não permite explorar a presença de várias CPUs pelo mesmo processo.

[C] Permite uma maior concorrência sem impor muita carga ao núcleo.

[A] Geralmente implementado por bibliotecas.

[B] É o modelo implementado no Windows NT e seus sucessores.

[A] Se um thread bloquear, todos os demais têm de esperar por ele.

[B] Cada thread no nível do usuário tem sua correspondente dentro do núcleo.

[C] É o modelo com implementação mais complexa.

