

Respostas lista 2

1- PCB é uma estrutura de dados que serve para armazenar as informações relativas ao contexto e os dados necessários à gerência de uma tarefa presente no sistema. Serve também para que seja efetuado a troca de contexto, que é: Interromper a execução de uma tarefa e retornar a ela mais tarde.

Um PCB contém: Identificador de tarefa, estado de tarefa, informações de contexto de processador, listas de arquivos abertos, conexões de rede e outros recursos usados pela tarefa, informações de gerência e contabilização.

2- Foi introduzido no início dos anos 60 um novo conceito: o compartilhamento de dados, ou *time-sharing*, através do sistema CTSS- *compartible time sharing system*[Corbató, 1963]. Nessa solução, cada atividade que detém o processador recebe um limite de tempo de processamento, denominado quantum. Esgotado seu quantum, a tarefa em execução perde o processador e volta para uma fila de tarefas “prontas”, que estão na memória aguardando sua oportunidade de executar.

3- A duração atual 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. Vários critérios podem ser definidos para a avaliação de escalonadores (quem define a duração dos quanta). Os mais frequentemente utilizados são:

- Tempo de execução ou de vida: Esse é o tempo total da “vida” de cada tarefa, ou seja, o tempo decorrido entre a criação da tarefa e seu encerramento, computando todos os tempos de processamento e de espera.
- Tempo de espera: É o tempo total perdido pela tarefa na fila de tarefas prontas, aguardando o processador.
- Tempo de resposta: É o tempo decorrido entre a chegada de um evento ao sistema e o resultado imediato de seu processamento.
- Eficiência: Indica o grau de utilização do processador na execução das tarefas do usuário.

6- (N: Nova, P: Pronta, E: Executando, S: Suspensa, T: Terminada)

[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.

[E] 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.

9- De forma geral, cada fluxo de execução do sistema, seja associado a uma processo ou no interior do núcleo, é denominado thread. Threads servem para se executar mais um processo ao mesmo tempo.

10- Vantagens: Leve e de fácil implementação. Como o núcleo somente considera uma thread, a carga de gerência imposta ao núcleo é pequena e não depende do número de threads dentro da aplicação.

Desvantagens: Como essas operações são intermediadas pelo núcleo, se um thread de usuário solicitar uma operação de E/S (recepção de um pacote de rede, por exemplo) o thread de núcleo correspondente será suspenso até a conclusão da operação, fazendo com que todos os threads de usuário associados ao processo parem de executar enquanto a operação não for concluída. Outro problema desse modelo diz respeito à divisão de recursos entre as tarefas. O núcleo do sistema divide o tempo do processador entre os fluxos de execução que ele conhece e gerencia: as threads de núcleo.

11- O modelo de threads 1:1(multi-thread) é adequado para a maioria das situações e atende bem às necessidades das aplicações interativas e servidores de rede. No entanto, é pouco escalável: a criação de um grande número de threads impõe uma carga significativa ao núcleo do sistema, inviabilizando aplicações com muitas tarefas (como grandes servidores web e simulações de grande porte).

12- a) many-to-one, b) one-to-one, c) many-to-many

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

[b] 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.

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

[c] É o modelo com implementação mais complexa.