

1. PCB é uma estrutura de dados que serve para armazenar as informações relativas ao contexto e os demais dados necessários à gerência de uma tarefa presente no sistema. Ele serve também para que seja efetuada a Troca de Contexto o que é: interromper a execução de uma tarefa e retornar a ela mais tarde, sem corromper seu estado interno.
2. Time sharing (compartilhamento de dados), foi um conceito introduzido nos anos 60 para resolver a inviabilização do sistema devido a uma tarefa em execução que nunca termina e nem solicita operações de entrada/saída, monopolizando o processador e impedindo a execução das demais tarefas. 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).

#### 6. Sequência – N P E T P E S E E S

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

10. Vantagens: Leve e fácil implementação. Como o núcleo somente considera uma thread, a carga de gerência imposta ao núcleo é pequena e não depende do núcleo 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) o thread de núcleo correspondente será suspenso até a conclusão da operação, fazendo com todos os threads de usuário associados ao processo parem de executar enquanto a operação não for concluída

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 grande servidores Web e simulações de grande porte).

#### 12. Sequência – a b b a c b a c c

14. R: Round-robin (RR) é um dos algoritmos mais simples de agendamento de processos em um sistema operacional, que atribui frações de tempo para cada processo em partes

iguais e de forma circular, manipulando todos os processos sem prioridades.  
Escalonamento Round-Robin é simples e fácil de implementar

15. :  $E = tq / tq + ttc$

16. Para evitar a inanição (quando um processo nunca assegura um recurso) e garantir a proporcionalidade expressa através das prioridades estáticas, um fator interno denominado envelhecimento (task aging) deve ser definido. O envelhecimento indica há quanto tempo uma tarefa está aguardando o processador e aumenta sua prioridade proporcionalmente. Dessa forma, o envelhecimento evita a inanição dos processos de baixa prioridade, permitindo a eles obter o processador periodicamente.