

01. PCB é uma estrutura de dados alocada no núcleo que serve para descrever uma tarefa associada, essa estrutura armazena informações sobre o contexto da tarefa e a sua gerência. Uma PCB contém: identificador da tarefa, estado da tarefa, informações de contexto do processador, lista de áreas de memória utilizadas pela tarefa, lista de arquivos abertos e informações de gerência e contabilização.

02. Significa compartilhamento de tempo esse conceito é muito importante pois ele faz com que cada tarefa tenha um tempo limite de execução (quantum), dessa forma o processador pode realizar outras tarefas mesma que uma das tarefas realizadas seja um loop infinito.

03. No hardware há um temporizador que gera interrupções em intervalos regulares no sistemas, essas interrupções são chamadas de tick. Quando o processador inicia uma tarefa o núcleo determina uma quantidade de tick's que essa tarefa vai ser processada, essa quantidade de tick's é o quantum da tarefa, ou seja, o tempo que a mesma tem para ser executada antes de voltar para o pilha de tarefas.

04.

05. $E \rightarrow P$. Essa transição de estado é possível e ocorre quando o quantum da tarefa acaba e a mesma não precisa de nenhum recurso para ser executada além do processador, dessa forma a tarefa volta para a fila de tarefas prontas.

$E \rightarrow S$. Essa transição de estado é possível e ocorre quando a tarefa solicita acesso a um recurso não disponível, então a tarefa ficará suspensa até que o recurso solicitado fique disponível.

$S \rightarrow E$. Essa transição de estado não ocorre.

$P \rightarrow N$. Essa transição de estado não ocorre.

$S \rightarrow T$. Essa transição de estado não ocorre.

$E \rightarrow T$. Essa transição de estado é possível e ocorre quando a tarefa encerra sua execução ou é abortada em consequência de algum erro.

$N \rightarrow S$. Essa transição de estado não ocorre.

$P \rightarrow S$. Essa transição de estado não ocorre.

06. [N] O código da tarefa está sendo carregado.

[P] As 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.

07.

08. Ao executar a linha de comando 'a.out 4 3 2 1' são impressos 5 "X" no terminal("XXXXX")

09. O thread é um fluxo de execução do sistema e serve para que as aplicações possam executar varias tarefas ao mesmo tempo.

11. No sistema de threads N:1, caso um thread de usuário solicitar uma operação de entrada saída o thread do núcleo correspondente será suspenso até a conclusão da operação. Outro exemplo e no caso da divisão do tempo de processador entre os fluxos, caso uma aplicação com 100 threads seja executadaele vai receber o mesmo quantum de uma aplica com apenas um thread, dessa forma, os threads vão receber 1/100 do tempo de execução.

12. [N:1] Tem a implementação mais simples, leve e eficiente.

[N:M] Multiplexa os threads de usuário em um pool de threads de núcleo.

[1:1] Pode impor uma carga muito pesada ao núcleo.

[N:1]] Não permite explorar a presença de várias CPUs pelo mesmo processo.

[N:M] Permite uma maior concorrência sem impor muita carga ao núcleo.

[N:1] Geralmente implementado por bibliotecas.

[1:1] É o modelo implementado no Windows NT e seus sucessores.

[N:1] Se um thread bloquear, todos os demais têm de esperar por ele.

[1:1] Cada thread no nível do usuário tem sua correspondente dentro do núcleo.

[N:M] É o modelo com implementação mais complexa