

## DISCIPLINA: Sistemas Operacionais

### CAPÍTULO 02 – GERÊNCIA DE ATIVIDADES

#### \_\_Questão 01\_\_

O núcleo do sistema operacional mantém descritores de processos, denominados PCBs (Process Control Blocks), para armazenar as informações referentes aos processos ativos.

#### \_\_Questão 02\_\_

O Time Sharing é o tipo de processamento onde se alterna entre diferentes processos de forma que o usuário tenha a impressão de que todos os processos estão sendo executados simultaneamente, permitindo a interação simultânea com múltiplos processos em execução.

#### \_\_Questão 03\_\_

A cada processo é atribuído um intervalo de tempo, o quantum, no qual ele é permitido executar; Se no final do quantum o processo não terminou, a CPU sofre uma preempção e outro processo entra para executar; quando um processo termina o seu quantum, ele é colocado no final da fila.

#### \_\_Questão 04\_\_

#### \_\_Questão 05\_\_

. E -> P

. E -> S

. S -> E

. P -> N

. S -> T

. E -> T

. N -> S

. P -> S

\_\_Questão 06\_\_

[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.

\_\_Questão 07\_\_

Valor de x: O tempo de execução do código aproximadamente foi entre 24 e 25 segundos. O resultado da saída está na coluna ao lado.

\_\_Questão 08\_\_

\_\_Questão 09\_\_

Thread é um pequeno programa que trabalha como um subsistema, sendo uma forma de um processo se auto dividir em duas ou mais tarefas. T hreads servem para executar mais de um processo ao mesmo tempo.

\_\_Questão 10\_\_

Uma das vantagens é que isso facilita o desenvolvimento, visto que torna possível elaborar e criar o programa em módulos, experimentando-os isoladamente no lugar de escrever em um único bloco de código. Outro benefício dos threads é que eles não deixam o processo parado, pois quando um deles está aguardando um determinado dispositivo de entrada ou saída, ou ainda outro recurso do sistema, outro thread pode estar trabalhando.

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. Assim, uma aplicação com 100 threads de usuário irá receber o mesmo tempo de processador que outra aplicação com apenas um thread (considerando que ambas as aplicações têm a mesma prioridade). Cada thread da primeira aplicação irá, portanto, receber 1/100 do tempo que recebe o thread único da segunda aplicação, o que não pode ser considerado uma divisão justa desse recurso.

#### \_\_Questão 11\_\_

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).

#### \_\_Questão 12\_\_

- [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.
- [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.