

Nome: Luiz Gabriel Zeferino Duarte .

RA: n454cd-8

1) sistema operacional é um software ou conjunto de softwares responsáveis por administrar todos os recursos do computador de forma eficiente. dentre suas principais funções estão o gerenciamento de processos, escalonador de processos, gerenciamento de memória, entrada e saída de dados e gerenciamento de arquivos.

2) multiprogramação é aproveitar o tempo em que um processo está aguardando resposta para colocar outro no lugar de forma concorrente, dando assim a impressão de que o cpu está rodando todos simultaneamente. Duas razões para o uso de multiprogramação seria o maior uso do hardware(reduzindo seu tempo ocioso) e maior número de Jobs finalizados em um mesmo espaço de tempo, aumentando a performance.

3) multiprocessamento é o uso de mais de uma central única de processamento (cpu) no mesmo computador. Multiprocessamento simétrico significa que todas as cpus tem a mesma função, ou seja, nenhuma delas tem acesso exclusivo ou restritivo a determinadas áreas. O assimétrico é justamente o oposto, algumas cpus ficam dedicadas a funções específicas determinadas pelo SO.

4) o SO interpreta o processo como se fosse a instância de um programa. Ou seja, programa é uma série de instruções, mas sem execução. Já o processo são a execuções dessas instruções.

5) PCB é uma estrutura de dados localizada no núcleo do sistema operacional . seu conteúdo é constituído de todas as informações relevantes para a execução de um processo, como identificação, prioridade e estado atual.

6) Imagem do Processo é um nome dado à coleção formada por: Código do programa a ser executado, pilha do sistema para controle de chamadas de procedimentos e de SVCs, Área de dados para armazenamento de variáveis locais e globais e Coleção de atributos do processo (englobados no PCB).

7)

a) 6

b)10

c) 11,2

d)8

8) o problema de se utilizar da solução que desabilita interrupções ao se aplicar a exclusão mútua é que o sistema depender que o próprio programa retire essa desabilitação, o que pode gerar sérios problemas de monopolização de CPU se o programador se esquecer disso.

9) busy waiting é uma forma de exclusão mútua em que o processo fica constantemente chegando se a zona crítica está livre. O problema dessa solução é que o processo usa clocks de processamento apenas para checar uma condição, o que acaba por atrapalhar o andamento dos demais processos.

10)

a) sim, o processo não conseguirá acessar a zona crítica ocupada pois chamará sempre o "yield()" e abandonará o quantum do processador.

b) sim, pois um processo "azarado" pode sempre pegar a região crítica ocupada e ser constantemente colocado no final da fila.

c) sim, a execução dessa chamada em uma thread não impede que ela seja usada por outras, por isso, pode ser usada por várias.

d) não, pois mesmo que esteja na fila de processos prontos, como a "yield()" faz com que ele desista do seu quantum, ele não gasta clocks do processador verificando a zona crítica.