

Sistemas Operacionais II

Lista de exercícios:

1. Durante a execução dos processos, que estados eles podem atingir?

- Novo
- Executando
- Esperando
- Pronto
- Terminado

2. O que é PCB? Cite informações contidas em um PCB.

O PCB funciona como uma ficha de atendimento para um determinado processo, contendo:

- Estado do processo
- Contador do programa
- Registradores de CPU
- Informações de escalonamento do processo
- Informações de gerenciamento de memória
- Informações de contabilização
- Informações de status de I/O

3. Qual o objetivo da multiprogramação? Cite exemplos de utilização da multiprogramação no dia a dia.

Maximizar a utilização da CPU executando, em paralelo, múltiplos programas na mesma máquina.

Um exemplo de multiprocessamento seriam os processos de I/O sendo executados paralelamente aos diversos outros processos do computador, interrompendo-os e executando-os conforme a necessidade, em velocidade tão grande, que se torna imperceptível para o usuário.

4. O que é escalonamento de processos?

Consiste em organizar os processos para serem executados pela CPU, sendo o SO responsável por selecionar aqueles que já estão prontos para execução, interromper um processo e salvar seus dados em um PCB, e ler o PCB do processo escolhido para execução. O tempo de espera de I/O que um determinado processo gasta, é utilizado para executar um outro processo.

5. Descreva os tipos de filas de escalonamento de processos.

- Fila de processos prontos: conjunto de todos os processos residentes em memória principal, prontos e esperando para serem executados.
- Fila de dispositivo: conjunto de processos esperando para acessar um dispositivo, cada dispositivo tendo sua própria fila.

6. O que são escalonadores?

É um componente do sistema operacional responsável por decidir quais processos serão executados pela CPU e em qual momento.

7. Descreva o processo de "troca de contexto".

1. O processo atual é interrompido.
2. Seu estado é registrado no PCB.
3. O estado do processo a ser executado é lido do seu PCB.
4. Este é então executado até a necessidade de ser interrompido.
5. Segue-se então interrompendo, gravando, lendo e executando os processos nas filas.

8. Descreva o problema do Produtor e do Consumidor.

Em uma memória compartilhada por dois processos, um deles registra dados na memória (Produtor) e o outro retira dados da memória (Consumidor). Conforme o espaço da memória for preenchido, um contador registra os espaços não livres. Ao chegar ao valor máximo, o contador coloca o processo Produtor para dormir e chama o processo Consumidor.

9. Existem duas abordagens para a comunicação entre processos: a comunicação direta e a indireta. Descreva cada uma delas.

- * Direta: O emissor identifica claramente o receptor e vice-versa
- * Indireta: O emissor e o receptor não precisam se conhecer pois não interagem diretamente um canal de comunicação é criado pelo SO para interligar ambos.

10. Qual a diferença entre escalonamento preemptivo e não-preemptivo?

- * Preemptivo: Um processo pode ser interrompido e retomado em qualquer estado, permitindo a execução de outro processo antes mesmo deste ter sido terminado.
- * Não-preemptivo: Um novo processo só é executado após o término do anterior.

11. Considere a fila de processos prontos a seguir

Processos	Surto de CPU (ms)	Prioridade
P1	54	1
P2	7	5
P3	61	3
P4	22	2
P5	18	1

Mostre a execução dos processos pela CPU, em uma linha de tempo iniciada em 0, com os algoritmos: FCFS, SJF, por prioridade e RR (quantum = 20ms).

- * FCFS: 0 => p1 => 54 ms => p2 => 61 ms => p3 => 122 ms => p4 => 144 ms => p5 => 162 ms
- * SJF: 0 => p2 => 7 ms => p5 => 25 ms => p4 => 47 ms => p1 => 101 ms => p3 => 162 ms
- * Por Prioridade: 0 => p5 => 18 ms => p1 => 72 ms => p4 => 94 ms => p3 => 155 ms => 162 ms
- * RR: 0 => p1 => 20 ms => p2 => 27 ms => p3 => 47 ms => p4 => 67 ms => p5 => 85 ms => p1 => 105 ms => p3 => 125 ms => p4 => 127 ms => p1 => 141 ms => p3 => 162 ms

12. O que são processos cooperativos? Cite uma situação em que é melhor ter uma aplicação implementada por vários processos cooperativos ao invés de um único processo.

São processos influenciados por outros processos pois compartilham recursos (variáveis, segmento de memória, endereçamento, estruturas de dados, etc.)

Estes processos podem transformar uma grande tarefa em diversas subtarefas que serão processadas simultaneamente, aumentando a velocidade do sistema.

Ao programar, um usuário pode estar editando, imprimindo e compilando seu código em paralelo. O acesso de um mesmo arquivo em diversos programas também necessita de processos cooperativos, como editar um mesmo texto pelo vim e emacs em paralelo, ao ser modificado o arquivo, se atualizado o outro editor de texto, o arquivo atualizado constará na aplicação.