
Lista de Exercícios

Antes de codificar, esboce em um papel a sequência de passos necessários para criar o seu programa. Isso ajuda a programar a solução;

Exercício 1. Quais são as duas principais funções de um sistema operacional?

Exercício 2: Qual é a diferença entre modo núcleo e modo usuário? Explique como ter dois modos distintos ajuda no projeto de um sistema operacional.

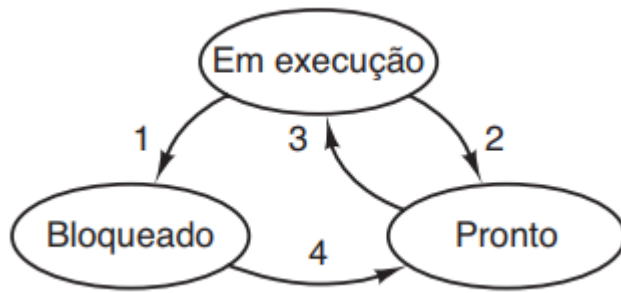
Exercício 3: Instruções relacionadas ao acesso a dispositivos de E/S são tipicamente instruções privilegiadas, isto é, podem ser executadas em modo núcleo, mas não em modo usuário. Dê uma razão de por que essas instruções são privilegiadas.

Exercício 4: Quais das instruções a seguir devem ser deixadas somente em modo núcleo?

- (a) Desabilitar todas as interrupções.
- (b) Ler o relógio da hora do dia.
- (c) Configurar o relógio da hora do dia.
- (d) Mudar o mapa de memória.

Exercício 5: Defina o que são chamadas de sistema e interrupções.

Exercício 6: Na Figura abaixo, são mostrados três estados de processos. Na teoria, com três estados, poderia haver seis transições, duas para cada. No entanto, apenas quatro transições são mostradas. Existe alguma circunstância na qual uma delas ou ambas as transições perdidas possam ocorrer?



Exercício 7: Em todos os computadores atuais, pelo menos parte dos tratadores de interrupções é escrita em linguagem de montagem. Por quê?

Exercício 8: Quando uma interrupção ou uma chamada de sistema transfere controle para o sistema operacional, geralmente uma área da pilha do núcleo separada da pilha do processo interrompido é usada. Por quê?

Exercício 9: Explique como funciona uma tabela de processos.

Exercício 10: Explique as diferenças e implicações entre sistemas monoprogramados e sistemas multiprogramados.

Exercício 11: Explique as razões para existirem as threads.

Exercício 12: Explique como funcionaria dois projetos possíveis: um servidor web multithread e um servidor web com um único thread.

Exercício 13: Normalmente as criações dos threads e mensagens impressas pelos threads são intercaladas ao acaso. Existe alguma maneira de se forçar que a ordem seja estritamente thread 1 criado, thread 1 imprime mensagem, thread 1 sai, thread 2 criado, thread 2 imprime mensagem, thread 2 sai e assim por diante? Se a resposta for afirmativa, como? Se não, por que não?

Exercício 14: A solução da espera ocupada usando a variável *turn* funciona quando os dois processos estão executando em um multiprocessador de memória compartilhada, isto é, duas CPUs compartilhando uma memória comum?

Exercício 15: A solução de Peterson para o problema da exclusão mútua mostrado na Figura 2.24 funciona quando o escalonamento de processos é preemptivo? E quando ele é não preemptivo?

Exercício 16: Quando um computador está sendo desenvolvido, normalmente ele é primeiro simulado por um programa que executa uma instrução de cada vez. Mesmo multiprocessadores são simulados de maneira estritamente sequencial. É possível que uma condição de corrida ocorra quando não há eventos simultâneos como nesses casos?

Exercício 17: Qual é a diferença entre um endereço físico e um endereço virtual?

Exercício 18: Que tipo de suporte de hardware é necessário para uma memória virtual paginada funcionar?

Exercício 19: Um estudante afirmou que “no abstrato, os algoritmos de substituição de páginas básicos (FIFO, LRU, ótimo) são idênticos, exceto pelo atributo usado para selecionar a página a ser substituída”.

(a) Qual é o atributo para o algoritmo FIFO? Algoritmo LRU? Algoritmo ótimo?

(b) Dê o algoritmo genérico para esses algoritmos de substituição de páginas.

Exercício 20: Um estudante em um curso de design de compiladores propõe ao professor um projeto de escrever um compilador que produzirá uma lista de referências de páginas que podem ser usadas para implementar algoritmo ótimo de substituição de página. Isso é possível? Por quê? Existe algo que poderia ser feito para melhorar a eficiência da paginação no tempo de execução?