UNIVERSIDADE FEDERAL DO PIAUÍ - UFPI CURSO: BACHARELADO EM SISTEMAS DE INFORMAÇÃO

PROFESSOR(A): DEBORAH MAGALHÃES

C.H.: 60h CRÉDITOS: 2.2.0 PERÍODO: 2023.1

DISCIPLINA: SISTEMAS OPERACIONAIS

ALUNO: LUIS EDUARDO SILVA BRITO

1) Qual a motivação por trás do conceito de threads?

R= é dividir um processo em várias threads que podem ser executadas de forma independente, aproveitando ao máximo os recursos do processador e evitando bloqueios e a espera desnecessária. A execução de várias tarefas simultaneamente pode melhorar significativamente o desempenho do sistema.

2) Explique a diferença entre um ambiente multithread e monothread?

R= No ambiente monothread apenas uma thread pode ser executada por vez, onde as instruções são executadas sequencialmente, uma após a outra. Já no ambiente multithread várias threads podem ser executadas simultaneamente ou em paralelo, onde instruções que podem ser executadas independentemente.

3) Por que o gerenciamento de threads é mais eficiente que o de processos?

R= O gerenciamento de threads é mais eficiente porque as threads podem se comunicar entre si, compartilham o mesmo espaço do processo pai, ou seja não precisa alocar recursos adicionais, elas têm uma melhor utilização de múltiplos núcleos de CPU, além da criação e término de threads ser mais rápido.

4) Quais informações são compartilhadas entre as threads de um processo? Quais informações são específicas de cada thread?

R= espaço de endereçamento, variáveis globais e estáticas, recursos do processo. As informações específicas de cada thread são: ID da thread, contador de Programa, registradores da CPU, pilha de execução, estado da CPU.

5) Em que consiste uma thread?

R= É a unidade básica para a qual o sistema operacional aloca o tempo do processador. Um thread pode executar qualquer parte do código do processo, incluindo partes que estão sendo executadas por outro thread.

6) O que é um pacote de threads e quais são seus diferentes modos?

R= Um pacote de threads é uma técnica usada em alguns processadores com capacidade de execução multithread, onde cada núcleo físico do processador pode suportar a execução simultânea de várias threads. O objetivo é melhorar o desempenho e a eficiência do processador, permitindo que ele execute múltiplas threads de maneira mais eficiente. Os modos de pacote de threads são: SMT (Simultaneous MultiThreading) e CMT (Chip-level Multi-Threading).

7) Quais as vantagens e desvantagens de implementar threads em modo kernel?

R= As vantagens são: escalonamento justo, trocas de contexto rápidas, prioridades de threads, bloqueio de threads, suporte a sistemas de multiprocessamento. E as desvantagens são: custo de trocas de contexto, sobrecarga de chamadas de sistema, complexidade, dependência do kernel, menor flexibilidade.

8) Quais as vantagens e desvantagens de implementar threads em modo usuário?

R= As vantagens são: maior controle, flexibilidade, independência do kernel, menor sobrecarga de trocas de contexto, escalonamento cooperativo. E as desvantagens são: bloqueio de todo o processo, escalonamento limitado, sincronização mais complexa, falta de suporte a E/S bloqueante.

9) O conceito a seguir, se refere a qual das alternativas abaixo: 2 ou mais processos estão lendo ou escrevendo em uma região de memória compartilhada ao mesmo tempo.

( ) Exclusão mútua
( ) Região crítica
( ) Comunicação entre processos
(x) Condição de corrida

10) Qual das opções abaixo não corresponde a uma boa solução de exclusão mútua:

( ) Nenhuma suposição pode ser feita a respeito de velocidades ou do número de CPUs
( ) Nenhum processo deve ser obrigado a esperar eternamente para entrar em sua região crítica
(x) Nenhum processo deve ser impedido de entrar em sua região crítica
( ) Nenhum processo executando fora de sua região crítica pode bloquear qualquer processo

11) Quais das alternativas abaixo corresponde ao conceito de preempção :

( ) o processo é executado sem tempo determinado até acabar
( ) o processo executado é aquele que possui menor tempo de execução
(x) o processo é executado por um intervalo de tempo pré-definido
( ) o processo executado é aquele que chegou primeiro

12) Defina o que é exclusão mútua com espera ocupada.

R= É uma técnica de sincronização utilizada em programação concorrente para garantir que apenas uma thread (ou processo) possa acessar um recurso compartilhado por vez.

13) Explique como a solução de Peterson garante a exclusão mútua.

R= porque quando dois processos tentam acessar a região crítica simultaneamente, somente um deles será capaz de entrar enquanto o outro ficará bloqueado no loop de espera até que seja sua vez. Isso evita condições de corrida e garante que apenas um processo possa ser executado na região crítica a qualquer momento.

14) Explique o problema do Produtor/Consumidor.

R= envolve a coordenação entre dois tipos de threads ou processos: os produtores, que produzem dados ou itens, e os consumidores, que consomem esses itens. O objetivo é garantir que a comunicação entre produtores e consumidores seja feita de forma correta e eficiente, evitando condições de corrida e uso indevido dos recursos compartilhados.

15) Qual a diferença entre as abordagens de escalonamento de sistemas interativos e sistemas em lote, incluindo seus objetivos específicos?

R= O Escalonamento de Sistemas Interativos o objetivo principal é fornecer uma resposta rápida e interativa ao usuário, o escalonador em sistemas interativos geralmente prioriza processos ou threads que estão relacionados à interação com o usuário. Já o escalonamento de Sistemas em Lote o principal objetivo é maximizar a eficiência do sistema, buscando a melhor utilização possível dos recursos disponíveis, o escalonador prioriza a eficiência e a utilização equilibrada dos recursos.

16) Explique a solução para o problema produtor/consumidor baseada em semáforos.

R= permite que os produtores e consumidores compartilhem um buffer (ou fila) de forma segura e coordenada. Os semáforos são usados para controlar o acesso ao buffer e

garantir que os produtores e consumidores não interfiram um no outro durante a produção e consumo dos itens. Essa solução envolve o uso de três semáforos: Semáforo do Buffer Vazio (vazio): Representa o número de espaços vazios no buffer.Semáforo do Buffer Cheio (cheio): Representa o número de espaços ocupados no buffer. Semáforo de Exclusão Mútua (mutex): Garante a exclusão mútua na manipulação do buffer, permitindo que apenas um produtor ou consumidor acesse o buffer por vez.
Produtor:
- Verifica se há um espaço vazio no buffer.
- Se houver um espaço vazio, adquire o semáforo de exclusão mútua (mutex) para evitar interferência de outros produtores.
- Escreve o item no buffer.
- Incrementa o semáforo do buffer cheio (cheio) para indicar que há um espaço ocupado no buffer.
- Libera o semáforo de exclusão mútua (mutex) para permitir que outros produtores possam escrever.

Consumidor:
- Verifica se há um item disponível para consumir no buffer.
- Se houver um item disponível, adquire o semáforo de exclusão mútua (mutex) para evitar interferência de outros consumidores.
- Lê o item do buffer.
- Decrementa o semáforo do buffer cheio (cheio) para indicar que há um espaço vazio no buffer.
- Libera o semáforo de exclusão mútua (mutex) para permitir que outros consumidores possam ler.

17) Quais as primitivas de troca de mensagem e quais os desafios dessa abordagem?

R= As primitivas de troca de mensagem são as operações de inserção (produção) e remoção (consumo) de itens no buffer compartilhado pelos produtores e consumidores. Já os desafios são: exclusão mútua, sincronização, deadlock, tamanho do Buffer, coordenação entre produtores e consumidores.

18) Qual a diferença entre um algoritmo de escalonamento preemptivo e não preemptivo?

R= No escalonamento preemptivo, o sistema operacional pode interromper um processo em execução e passá-lo para o estado de pronto, com o objetivo de alocar outro processo na UCP. No escalonamento não-preemptivo, quando um processo está em execução, nenhum evento externo pode ocasionar a perda do uso do processador.

19) Por que a preempção é fundamental para sistemas interativos?

R= Porque esses sistemas têm como característica principal a necessidade de fornecer uma resposta rápida e interativa aos usuários. E permite que o sistema operacional gerencie de forma eficaz a execução de tarefas, priorizando aquelas que são mais importantes para a interação com o usuário e respondendo prontamente a eventos externos.

20) Quais os objetivos gerais de um algoritmo de escalonamento?

R= Eles têm por objetivo realizar o ordenamento temporal de um conjunto de tarefas, gerenciando a atualização da fila de execução, dinamicamente ou não, dependendo da técnica utilizada. Alguns dos objetivos do algoritmo de escalonamento incluem minimizar o

tempo de resposta em sistemas interativos, principalmente servidores, e decidir qual dos processos prontos para execução deve ser alocado à CPU.

21) Explique o funcionamento dos seguintes algoritmos de escalonamento: (i) Escalonamento Circular (Round-Robin); (ii) Escalonamento por prioridades; (iii) Escalonamento por loteria.

R= (i) Escalonamento Circular (Round-Robin) é um algoritmo de escalonamento preemptivo que consiste em repartir uniformemente o tempo da CPU entre todos os processos prontos para a execução. Os processos são organizados numa fila circular, alocando-se a cada um uma fatia de tempo da CPU, igual a um número inteiro de quanta. Caso um processo não termine dentro de sua fatia de tempo, ele é colocado no fim da fila e uma nova fatia de tempo é alocada para o processo no começo da fila 1.

(ii) Escalonamento por prioridades é um algoritmo de escalonamento que atribui prioridades aos processos e os executa com base nessas prioridades. Processos com prioridades mais altas são executados antes dos processos com prioridades mais baixas. As prioridades podem ser estáticas (definidas pelo sistema ou pelo usuário) ou dinâmicas (alteradas pelo sistema durante a execução).

(iii) Escalonamento por loteria é um algoritmo de escalonamento que atribui bilhetes de loteria aos processos e seleciona aleatoriamente um bilhete vencedor para determinar qual processo será executado a seguir. Processos com mais bilhetes têm uma chance maior de serem selecionados. Os bilhetes podem ser atribuídos com base em vários critérios, como prioridade ou uso de recursos.

22) O que diferencia o escalonamento por fração justa dos descritos na questão anterior.

R= O Escalonamento por Fração Justa (Fair-Share) é um algoritmo de escalonamento que garante que cada usuário tenha sua fração da CPU, independentemente do número de processos que cada usuário tenha. A fração alocada ao usuário é garantida pelo escalonador conforme a “noção” de justiça sobre a prioridade do usuário. Isso difere dos outros algoritmos mencionados anteriormente, pois eles se concentram em atribuir tempo de CPU aos processos individuais, enquanto o Escalonamento por Fração Justa se concentra em garantir que cada usuário tenha uma fração justa do tempo de CPU.