Cap 05

**Scheduling de CPU**

# Parte 1

1. Qual a função de um **Scheduler de Curto Prazo**? E um **Dispather**?

|  |
| --- |
| Scheduler de Curto Prazo decide qual processo será executado a seguir, enquanto o Dispatcher é responsável pela implementação prática dessa decisão, realizando as operações necessárias para fazer a troca de contexto e transferir o controle da CPU de um processo para outro. Essas duas componentes trabalham em conjunto para garantir uma execução eficiente e equitativa dos processos no sistema operacional. |

1. O que você entendeu sobre **Latência de Despacho**?

|  |
| --- |
| A "Latência de Despacho" refere-se ao intervalo de tempo entre a decisão de executar um processo pelo escalonador e o início efetivo da execução desse processo pelo dispatcher. É o tempo que leva para realizar a transição entre processos, minimizando essa latência é crucial para otimizar o desempenho do sistema |

1. Geralmente, quais são os critérios utilizados para comparação de algoritmos? Cite três destes critérios e descreva sucintamente o que eles avaliam;

|  |
| --- |
| Utilização da CPU: Indica quando a CPU permanece ocupada  Throughput: Número de processos que são completados por unidade de tempo  Tempo de Turnaround: Intervalo de tempo entre submissão e o término de um processo |

1. Quais são os principais algoritmos utilizados por sistemas operacionais para Scheduling de CPU? Escreva de forma sucinta como funciona cada um deles;

|  |
| --- |
| First Come, First Served (FCFS):  Executa processos na ordem de chegada à fila de prontos. Pode causar "efeito de inversão de prioridade".  Shortest Job First (SJF):  Executa o processo com o menor tempo de execução primeiro. Pode ser preemptivo ou não preemptivo.  Round Robin (RR):  Atribui um quantum de tempo a cada processo. Após o quantum, o processo vai para o final da fila.  Priority Scheduling:  Executa o processo com a maior prioridade primeiro. Pode ser preemptivo ou não preemptivo.  Multilevel Queue Scheduling (MQS): |

1. O que você entende por Processos **CPU-Bound** e **IO-Bound**?

|  |
| --- |
| Processos CPU-Bound:  Descrição: Demandam principalmente a capacidade de processamento da CPU.  Características:  Uso intensivo da CPU.  Tempos de CPU prolongados.  Processos I/O-Bound:  Descrição: Frequentemente interrompidos por operações de entrada/saída.  Características:  Realizam operações de entrada/saída com frequência.  Tempos de espera mais longos em comparação com o tempo de CPU. |

# Parte 2

1. Considere o conjunto de Processos a seguir, com duração do pico de CPU dada em milissegundos:

|  |  |  |
| --- | --- | --- |
| Process | Burst Time | Priority |
| A | 10 | 3 |
| B | 1 | 1 |
| C | 2 | 3 |
| D | 1 | 4 |
| E | 5 | 2 |

Presume-se que os processos tenham chegado na ordem **A, B, C, D, E**, todos no momento **t0**

* 1. Desenhe quatro gráficos de Gantt que ilustrem a execução desses processos usando os Algoritmos de Scheduling a seguir: **FCFS**, **SJF**, **PS sem preempção** e **RR (Quantum = 1)**

|  |
| --- |
|  |

* 1. Qual é o **tempo médio de espera** de cada processo para cada um desses algoritmos de Scheduling?

|  |
| --- |
| TME no print do exercício anterior. |

* 1. Qual Algoritmo resulta no menor **tempo médio de espera** (considerando todos os processos)?

|  |
| --- |
| RR |

# Parte 3

1. No algoritmo **Priority Scheduling**, porque usar um **Quantum Time** pequeno pode degradar a performance?

|  |
| --- |
| O uso de um Quantum Time pequeno no algoritmo Priority Scheduling pode degradar o desempenho devido a preempção excessiva, aumento do tempo de espera para processos de baixa prioridade, aumento do overhead do sistema devido à troca de contexto frequente, e ineficiência em tarefas CPU-bound. O tamanho do quantum deve ser escolhido considerando um equilíbrio entre respeitar as prioridades e evitar overhead desnecessário. |

1. Qual é a diferença entre sistemas com **Multiprocessamento Assimétrico** e com **Multiprocessamento Simétrico**?

|  |
| --- |
| Multiprocessamento Assimétrico:  Descrição: Um processador é designado como principal, responsável pelo gerenciamento do sistema, enquanto os processadores secundários executam tarefas de aplicativos específicos.  Multiprocessamento Simétrico:  Descrição: Todos os processadores são tratados como equivalentes, sem distinção de funções. O sistema visa distribuir tarefas de maneira equitativa para otimizar o uso global dos recursos. |

1. O que significa um processo ter **afinidade** com o Processador?

|  |
| --- |
| A afinidade de processador significa associar um processo a um processador específico ou a um conjunto específico de processadores em um sistema multiprocessador. Isso pode ser feito para otimizar o uso de cache, melhorar o desempenho, controlar recursos ou reduzir a migração de processos entre processadores. Essa configuração pode ser feita pelo sistema operacional ou pelos desenvolvedores, visando atender a requisitos específicos do sistema e da carga de trabalho. |

1. O que você entende por **Queda de Memória** (***Memory Stall***)? O que pode ser feito/utilizado em uma CPU para evitar que esta etapa atrase o processamento de dados?

|  |
| --- |
| Para evitar esse atraso, podem ser adotadas as seguintes estratégias:  Memória Cache: Utilizar caches para armazenar cópias de dados frequentemente acessados, reduzindo a necessidade de acessos à memória principal.  Pré-Busca (Prefetching): Algoritmos de pré-busca antecipam e buscam dados antes de serem estritamente necessários, minimizando a latência de acesso à memória.  Pipelines e Superscalaridade: Permitir que a CPU execute instruções não relacionadas à memória enquanto aguarda o resultado de operações de acesso à memória.  Execução Fora de Ordem (Out-of-Order Execution): Permite que a CPU execute instruções não dependentes da memória durante operações de acesso à memória, reorganizando a ordem das instruções para otimizar o tempo ocioso.  Memória de Acesso Rápido (Fast Memory): Utilizar memórias mais rápidas, como memória cache de alta velocidade, para reduzir a latência de acesso à memória principal.  Algoritmos de Escalonamento Inteligente: Priorizar a execução de tarefas com menor dependência de memória para minimizar os atrasos causados pela queda de memória. |

1. Explique o que você entendeu sobre **balanceamento de carga** em sistema com múltiplos processadores:

|  |
| --- |
| O balanceamento de carga em sistemas com múltiplos processadores refere-se à distribuição eficiente e equitativa de tarefas entre os processadores disponíveis. Isso é alcançado por meio de estratégias como a distribuição uniforme de tarefas, a detecção dinâmica de carga, a migração de tarefas quando necessário, políticas de escalonamento apropriadas e uma comunicação eficiente entre os processadores. O objetivo é otimizar o desempenho do sistema, evitando sobrecarga em alguns processadores enquanto outros permanecem subutilizados. |

1. Explique o que você entendeu sobre ***Hyper-Threading***:

|  |
| --- |
| Hyper-Threading é uma tecnologia que visa melhorar a eficiência do processador, permitindo a execução simultânea de múltiplos threads em um único núcleo físico. Isso é alcançado através do compartilhamento inteligente de recursos e é projetado para otimizar o desempenho em tarefas paralelas e em situações em que há esperas por I/O ou acesso à memória. |