|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nome: Pedro Gabriel Garcia Ribeiro Balestra | | |  |  | Matrícula: 1551 |  |
| Curso: GEC |  |  | Período: P8 |  | Matéria: C012 |  |

**Cap.5 – Scheduling de Cpu**

1. Scheduler de Curto Prazo são algoritmos que realizam o processo de selecionar os processos presentes na Fila de Prontos para ser executado, podendo ser algoritmos preemptivos (Podem ser frustados pelo SO) e não-preemptivos (não podem ser frustados pelo SO). E o despachante é o modulo que passa o controle da Cpu para o processo selecionado pelo Scheduler de Curto Prazo, sendo chamado a cada mudança de contexto
2. Pode ser mantido utilizando a abordagem de **migração por expulsão**, verificando constantemente a carga de cada processador, e a abordagem de **migração por absorção**, que é quando um processador ocioso extrai uma tarefa que está esperando em um processador ocupado
3. Podemos comparar algoritmos por 3 maneiras: a **utilização da CPU**, indicando o quanto a CPU permanece ocupara, **throughput,** quantidade de processos completados por unidade de tempo, **tempo de turnaround,** intervalo de tempo entre a submissão e término do processo, dentre outros.
4. Existem vários algoritmos de Scheduling de Processos, dentre eles temos
   * **FIFO/FCFS** – Onde o primeiro processo a entrar na fila será “servido”, algoritmos Não-Preemptivo.
   * **SJF** – O Processo que tem o pico de CPU mais curto será “servido”, pode oferecer o menor tempo médio de espera, algoritmo Não-Preemptivo e Preemptivo.
   * **PS** – Uma prioridade é associa a cada processo, executando do que tem maior prioridade ao que tem menos, algoritmo Não-Preemptivo e Preemptivo.
   * **RR** – É dado um Quantum de Tempo ao algoritmo, onde a cada Quantum de Tempo o algoritmo verifica qual processo irá executar, semelhante ao FIFO, porém com Preempção.
5. **CPU-Bound:** Processos com longos picos de CPU

**IO-Bound:** Processos com curtos picos de CPU

1. A)

Gráfico, Gráfico de caixa estreita

Descrição gerada automaticamente

B) Tabela

Descrição gerada automaticamente

C) O algoritmo SJF resulta no menor tempo de espera com **3.2 ms**

Gráfico

Descrição gerada automaticamente

1. Afinidade com o processador é quando o processo busca sempre ser executado no mesmo processador, podemos ter 2 tipo de afinidades, **leve** quando o SO tenta manter o o processo executando no mesmo processador, porem não garante, já o **forte** o SO obriga o processo a executar no mesmo processador
2. **Memory Stall** é o tempo de espera do processador para receber os dados da memória acessada, podendo gastar ate 50% do tempo esprando. Para resolver esse problema atribuimos 2 Threads de hardware para cada nucleo, assim quando um for interrompido, o nucleo passa para a outra Thread, chamamos isso de Hyper-Threading