

<p style="text-align: center;">MATERIAL DE APOIO</p> <p style="text-align: center;">RESUMO 6</p>	
DISCIPLINA: Sistemas Operacionais	PROFESSOR: Lincon M. Peretto
BIBLIOGRAFIA: MACHADO, Francis Berenger; MAIA, Luiz Paulo. Arquitetura de Sistemas Operacionais . Capítulo 8. 4ª ed. São Paulo: LTC, 2007.	

Gerência do Processador

Algoritmos de Escalonamento

- Possui a tarefa de escolher qual processo deve ocupar o processador
- Tem como objetivo aumentar a produção do sistema e ao mesmo tempo, diminuir o tempo de resposta percebido pelos usuários.
- throughput -> produção do sistema (manter o processador ocupado todo o tempo, produzindo mais em menos tempo).
- turnaround time -> Oferecer baixo tempo de resposta ao usuário (obtido com um baixo tempo médio de espera na fila do processador).

Às vezes é melhor sacrificar o tempo médio de resposta a ter uma variância muito grande, ou seja, enquanto alguns processos recebem tempo de processador satisfatório, outros são bastante prejudicados.

Ordem de chegada (FIFO – First-in first-out)

- Implementação mais simples, pois a fila do processador é uma fila simples.
- Os processos são executados na mesma ordem em que chegaram à fila.
- Um processo só desocupa o processador quando efetua uma chamada de sistema ou ocorre algum erro.
- O problema é o seu desempenho:
- Quando se trata de um processo cpu-bound é necessário aguardar seu ciclo de processador.

Exemplo:

Processo	Duração do próximo ciclo de processador
A	12 unidades de tempo
B	8
C	15
D	5

Tempo médio de espera:

$$(0+12+20+35)=67/4=16.75$$

Ciclo de processador menor antes (SJF – Shortest job first)

- Seleciona antes o processo cujo ciclo de processador é o menor entre os processos que estão na fila.
- Os processos são inseridos na lista de maneira a manter a ordem proposta, assim é só pegar o primeiro da lista e executar.
- É útil porque oferece um limite teórico para o tempo médio de espera e serve como comparativo em outros implementáveis.
- Pode-se implementar aproximações do SJF levando em consideração a duração dos ciclos de processador mais recentes para utilizar como aproximação nos próximos ciclos.
- Processos i/o-bound são favorecidos, pois liberam o processador mais rapidamente, minimizando o tempo de espera.

Exemplo:

Processo	Duração do próximo ciclo de processador
D	5 unidades de tempo
B	8
A	12
C	15

Tempo médio de espera:

$$(0+5+13+25)=43/4=10.75$$

Prioridade

- Quando os processos possuem prioridades, essa prioridade pode ser utilizada para decidir qual processo é executado a seguir.
- Pode ser definido como uma lista onde ordenada conforme a prioridade, sendo que o processo a ser executado é sempre o primeiro da fila.
- Quando há empate posso, por exemplo, levar em consideração a ordem de chegada (FIFO).
- Por exemplo, se os processos i/o-bound possuírem maior prioridade, o algoritmo irá se aproximar de SJF.

Exemplo:

Processo	Prioridade	Duração do próximo ciclo de processador
D	3	5 unidades de tempo
B	4	8
A	2	12
C	1	15

Postergação indefinida – um processo de baixa prioridade pode nunca executar porque sempre existe outro de maior prioridade na fila.

- É impossível ocorrer isso quando utilizar FIFO, pois uma vez na fila sua vez chegará.
- Com SJF é possível ocorrer isso também, pois com a perspectiva de um grande ciclo de processador ele pode ser eternamente preterido em favor de processos com ciclos de processador menores.
- Para isso podemos introduzir um algoritmo de envelhecimento (aging) ao método básico. Lentamente os processos têm sua prioridade elevada. O objetivo deste procedimento não é elevar sua prioridade e sim evitar que ele fique para sempre na fila.
- Afinal se ele tem prioridade baixa é natural que execute lentamente.
- Preemptivo – processo em execução pode perder o processador para outro processo por algum motivo que não seja o término do seu ciclo.
- Não-preemptivo – processo só liberar o processador por vontade própria (chamada de sistema).
- FIFO é exclusivamente não-preemptivo, uma vez que não existe a possibilidade de um processo tirar o processador de outro.
- SJF já admite as duas possibilidades.
- Tipicamente, soluções baseadas em prioridade utilizam preempção, pois não faz sentido um processo com baixa prioridade ficar ocupando o processador. Permite implementar o conceito de prioridades de forma mais completa no sistema.

Fatia de Tempo (Round-Robin)

- Cada processo recebe uma fatia de tempo do processador (quantum).
- Pode ser implementado através de uma fila simples, sendo que o primeiro da fila executa primeiro.
- O processo fica até o término da sua fatia de tempo, fim da sua execução ou até realizar uma chamada de sistema.
- Problema é definir o tamanho da fatia de tempo. Chaveamento entre processos não são instantâneos, envolve um custo.

Ex:

1 microssegundo para cada instrução de máquina.

200 instruções para realizar chaveamento.

Fatia de tempo = 1 milissegundo -> 20% do tempo chaveando. O ideal é 1%.

Fatias de tempo muito grandes também tiram a aparência de paralelismo na execução dos processos.

Ex: Se o sistema possuir 20 processos em execução e a fatia de tempo for de 1 segundo, cada um vai receber 1s a cada 20s. Vai executar aos pulinhos.

Uma fatia de tempo de 100ms vai resultar em 0.1s a cada 2s.

Vai depender da sua necessidade.

Múltiplas Filas

- Existência de diversas filas de processos, onde a prioridade é atribuída à fila e não ao processo. A vantagem é tratar os processos de forma mais específica e poder aplicar regras diferentes para cada fila.

- Desvantagem – um processo não pode alterar de fila, mesmo que ele devesse ter uma prioridade maior durante sua execução, para isso é necessário utilizar um algoritmo chamado de múltiplas filas com realimentação.