Aula 10: Escalonamento da CPU

O **escalonamento da CPU** é a base dos sistemas operacionais multiprogramados. A partir da redistribuição da CPU entre processos, o sistema operacional pode tornar o computador mais produtivo.

Conceitos Básicos

O objetivo da **multiprogramação** é contar sempre com algum processo em execução para maximizar a utilização da CPU. Em um sistema com um único processador, somente um processo pode ser executado de cada vez; quaisquer outros processos devem esperar até que a CPU esteja livre e possa ser redistribuída.

O **escalonador** é um função fundamental do sistema operacional. Quase todos os recursos do computador são submetidos a um processo de escalonamento antes do uso. A CPU é, naturalmente, um dos recursos primordiais do computador. Assim, seu escalonador é central para o projeto do sistema operacional.

Um outro componente envolvido na função de escalonamento da CPU é o **despachante**. O despachante é o módulo que passa o controle da CPU ao processo selecionado pelo escalonador. O despachante precisa ser o mais veloz possível, pois é invocado durante cada comutação de processo.

Escalonamento da CPU

Sempre que a CPU se torna **ociosa**, o sistema operacional deve selecionar um dos processos existentes na **fila pronta** para que seja executado. O escalonador escolhe dentre os processos na memória que estão **prontos** para execução, e aloca a CPU a um deles.

A fila pronta não é necessariamente uma fila *first-in*, *first-out* (FIFO, primeiro que entra é o primeiro a sair). Como veremos quando considerarmos os diversos algoritmos de escalonamento, uma fila pronta pode ser implementada como uma fila FIFO, uma fila de prioridades, uma árvore ou simplesmente uma lista encadeada desordenada. Os registros na fila são em geral blocos de controle de processos (PCBs, **process control blocks**).

Escalonamento com e sem Preempção

Um algoritmo de escalonamento diz-se não-preemptivo, se, uma vez na posse do CPU, um processo executa até o (ao CPU) “libertar” voluntariamente. Algoritmos não-preemptivos têm

problemas graves:

- certas classes de processos executam durante muito tempo até bloquear;
- um utilizador egoísta pode impedir que o computador execute processos de outros utilizadores.

Praticamente todos os sistemas operativos usam algoritmos preemptivos: O sistema operacional usa as interrupções do relógio para retirar o CPU ao processo em execução.

Algoritmos de Escalonamento

Diferente **algoritmos de escalonamento** da CPU possuem propriedades diferentes e podem favorecer uma classe de processos em detrimento de outras. Ao escolher qual algoritmo usar em uma situação particular, devemos considerar as propriedades dos diversos algoritmos.

O escalonamento da CPU trata do problema de decidir qual dos processos na fila pronta deve ser alocado a CPU. Nesta seção, será descrito um conjunto dos muitos algoritmos de escalonamento a CPU que existem.

First-Come, First-Served (Primeiro que chega, primeiro atendido)

O algoritmo de escalonamento **FCFS** é o mais simples. Com este esquema, o processo que primeiro requisita a CPU é o primeiro a ser alocado à CPU. A implementação da política de FCFS é facilmente gerenciada com uma fila FIFO.

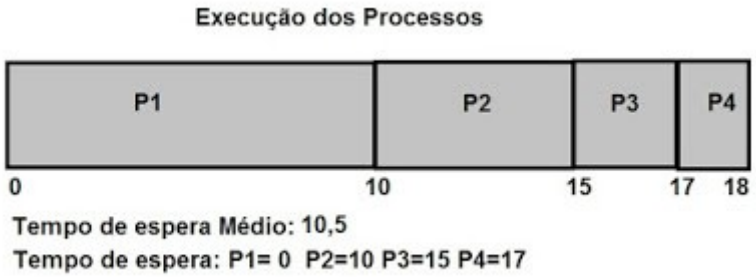

Figura 1: Exemplo da Fila pronta do FCFS.

A Figura 1 demonstra a execução dos processos neste algoritmo. É possível visualizar que este algoritmo não é eficiente em relação a processos com pequenos tempos de execução. Por exemplo, o P4 que tem um tempo pequeno de execução poderia ser executado primeiro, mas é somente executado no final.

Shortest-job-first (Menor Job Primeiro)

O algoritmo de escalonamento **SJF** se baseia no tempo de execução de cada processo. Quando a CPU está disponível, ela é designada para o processo com menor tempo de duração. Se dois ou mais processos tem o mesmo tempo de execução, então o escalonador FCFS é usado para resolver este impasse.

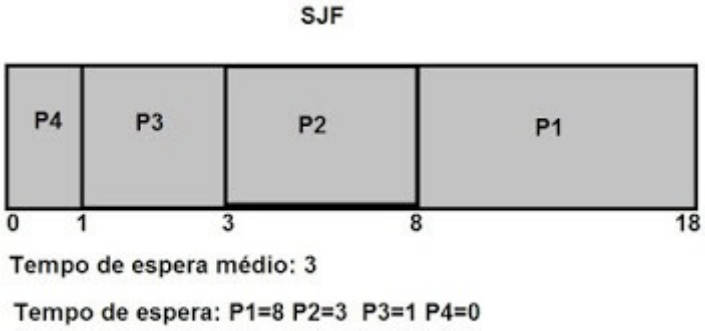

Figura 2: Exemplo da fila pronta do SJF.

A Figura 2 demonstra a execução dos processos neste algoritmo. Repare a diferença entre FCFS e SJF, onde os mesmos processos usado no FCFS tiveram média de espera de 3, por isso o SJF é uma opção melhor que o FCFS.

Shortest Remaining Time (Tempo Remanescente Mais Curto)

SRT é a variante preemptiva do escalonamento SJF. A fila de processos a serem executados pelo SRT é organizada conforme o tempo estimado de execução, ou seja, de foma semelhante ao SJF, sendo processados primeiros os menores jobs. Na entrada de um novo processo, o algoritmo de escalonamento avalia seu tempo de execução incluindo o job em execução, caso a estimativa de seu tempo de execução seja menor que o do processo concorrentemente em execução, ocorre a substituição do processo em execução pelo recém chegado, de duração mais curta, ou seja, ocorre a preempção do processo em execução.

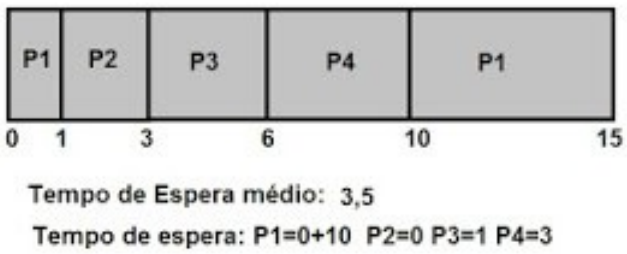

Figura 3: Exemplo da fila pronta do SRT.

A Figura 3 demonstra a execução dos processos neste algoritmo. No SRT o tempo de espera é calculado a partir do tempo de chegada. Neste exemplo, o P2 chegou no momento de execução do P1, entretanto, como P2 tinha um menor tempo de execução, a CPU foi alocada para ele.

Duling (Prioridade)

O algoritmo de escalonamento **Duling** associa uma prioridade a cada processo e a CPU é alocada ao processo com prioridade mais alta. Se dois ou mais processos tem o mesmo tempo de execução, então o escalonador FCFS é usado para resolver este impasse.

Round-Robin (Porção de Tempo em Fila Circular)

No algoritmo de escalonamento **RR**, uma unidade de tempo pequena chamada **porção de tempo** é definida. A fila pronta é tratada como uma fila circular. O escalonador da CPU circula a fila pronta alocando a CPU a cada processo, por um intervalo de tempo de 1 porção de tempo. A Figura 4 demonstra a execução dos processos neste algoritmo.

Figura 4: Exemplo da fila pronta do RR.

Exercícios

1. Defina a diferença entre escalonamento com preempção e sem preempção.

2. Qual a função do Despachante?

3. Considere o seguinte conjunto de processos, com o tempo de duração de execução na CPU dado em milissegundos:

Processo	Tempo de Execução	Prioridade
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

   Considera-se que os processos tenham chegado na ordem P1, P2, P3, P4, P5, todos no tempo 0.

   a) Desenhe cinco gráficos de Gantt ilustrando a execução destes processos utilizando FCFS, SJF, SRT, Duling e RR (porção de tempo = 1).

   b) Que escalonador resulta em um menor tempo de execução?

   c) Que escalonador resulta em um menor tempo médio de espera (soma dos tempos gasto esperando na fila pronta)?

4. Considere que os seguintes processos chegaram para execução nos tempos indicados:

Processo	Tempo de Chegada	Tempo de Execução
P1	0,0	8
P2	0,4	4
P3	1,0	1

   a) Desenhe cinco gráficos de Gantt ilustrando a execução destes processos utilizando FCFS, SJF, SRT, Duling e RR (porção de tempo = 1).

   b) Que escalonador resulta em um menor tempo de execução?

   c) Que escalonador resulta em um menor tempo médio de espera (soma dos tempos gasto esperando na fila pronta)?