Escalonamento de Processos do Sistema Operacional

Aluno: Perilo Oliveira

Curso e Período: CCP2

O Escalonamento de Processos é um subsistema do Sistema Operacional cuja função é gerenciar a utilização de recursos do computador, como por exemplo os recursos da CPU (Unidade Central de Processamento), entre os processos que estão sendo executados. Para isso, esse subsistema utiliza uma lógica de escalonamento e os critérios para estabelecer a ordem, o tempo de utilização e a divisão dos recursos, de modo que cabe a esse sistema avaliar os possíveis cenários e aplicar o escalonador mais bem aplicado nesse cenário. Dentre as funções e prioridades do escalonamento, pode-se afirmar que os objetivos do escalonamento de processos são: maximizar a utilização da CPU de forma eficiente, minimizar o tempo gasto para execução de processos, garantir a alocação de recursos de forma justa entre os processos e por fim, priorizar o acesso a recursos disponíveis, de maneira que otimize o rendimento dos recursos do SO.

Diante disso, é importante destacar a relação entre a Multiprogramação e o Escalonamento de Processos, visto que ambos os conceitos estão relacionados com a gestão de recursos, em especial a CPU, entre processos/usuários e no aumento da eficiência do uso desses mesmos recursos. A Multiprogramação é uma técnica que permite a execução simultânea de vários processos no SO, sendo carregados na memória e competindo pelo uso da CPU. Enquanto o Escalonamento gerencia e define a ordem da alocação de recursos como a CPU. Logo, a primeira permite a execução de processos simultaneamente e o segundo define as ordens do uso de recursos. E juntos garantem o funcionamento eficiente do que já foi explicado no parágrafo anterior.

Ademais, os escalonadores são responsáveis por selecionar o processo a ser executado, baseado em um critério. Existem 3 tipos de escalonadores, sendo eles:

**Escalonador de Longo Prazo:** Escalonador responsável por controlar o grau de Multiprogramação do sistema, ou seja, controlar o número de processo que serão executados e admitir novos processos a serem executados.

**Escalonador de Médio Prazo:** Escalonador responsável por escolher os processos a serem suspensos/removidos da fila para garantir rendimento e eficiência do sistema.

**Escalonador de Curto Prazo:** Escalonador responsável por selecionar os processos prontos para utilizar a CPU e alocar à CPU.

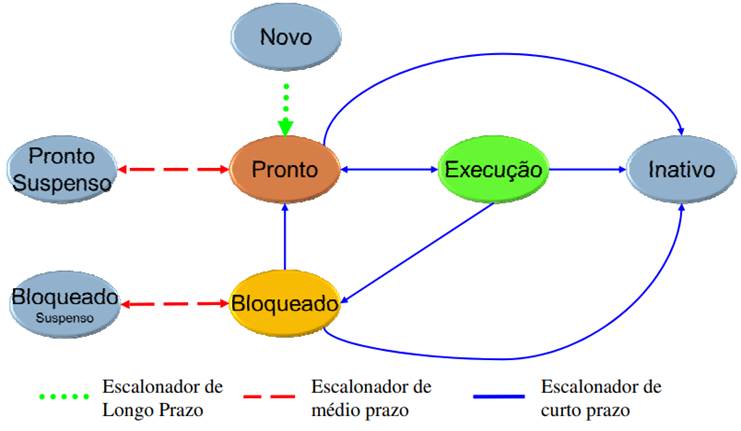


Figura 1 – Escalonamento e os Escalonadores

Como pode-se observar, o Escalonador de longo prazo admite novos processos na fila, o Escalonador de médio prazo garante a eficiência do sistema bloqueando e suspendendo processos enquanto outros tomam o uso dos recursos para si e por fim, o Escalonador de curto prazo realiza as movimentações entre pronto, em execução e inativo, sendo responsável por posicionar os processos e os recursos que esses processos podem utilizar.

Outrossim, os Algoritmos de Escalonamento são responsáveis por determinar os critérios que serão utilizados para definir uma ordem lógica da fila, eles podem ser classificados em Preemptivos e Não Preemptivos.

Os Preemptivos compreendem os algoritmos capazes de tomar a CPU de um processo em execução, mesmo que o processo não tenha sido concluído. Assim, garantindo a prioridade de processos ou respostas a eventos críticos. Esses algoritmos são utilizados em situações com eventos essenciais e que demandem reações rápidas. Eles são divididos em 4 tipos, sendo eles:

**1 - Algoritmo Por Prioridade**

- Cada processo é associado a um valor INT que define sua prioridade de execução.

- O escalonador define a fila baseado na prioridade de cada processo.

Obs.: As prioridades podem ser definidas pelo SO ou pelo usuário.

Exemplo: 1 a 9, onde o 1 é mais prioritário e o 9 é de menor prioridade. Onde os processos A, B e C receberam, respectivamente os valores 2, 6, 1. Nesse caso, a ordem de execução será C > A > B.

**2 – Algoritmo Turno Rotativo (Round-Robin)**

- Associa-se um intervalo de tempo a cada processo denominado quantum.

- Rotaciona entre cada processo durante o tempo de quantum de cada processo.

Obs.: Permite o aproveitamento máximo de tempo e serve muito bem para sistemas de compartilhamento de tempo.

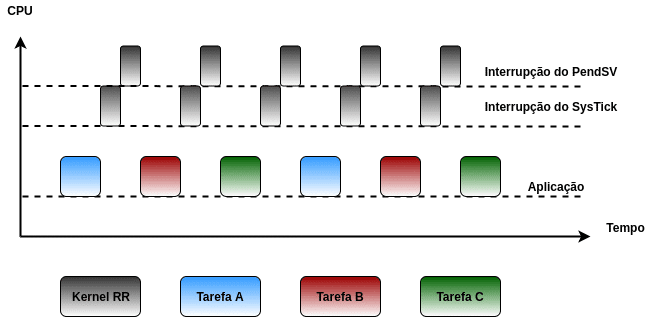


Figura 2 - Algoritmo Round-Robin

Como pode-se observar acima, cada processo é executado durante um intervalo de tempo e substituído pelo próximo quando esse intervalo acabar. E continua rotacionando até todos serem finalizados.

**3 – Algoritmo de Filas Multiníveis**

- Os processos prontos são divididos em várias filas.

- Cada fila possui seu próprio algoritmo de escalonamento.

- Definição de filas em primeiro plano e segundo plano, para intercalar entre elas.

- Filas prioritárias ou divisão de tempo entre execução de cada fila.

- Os processos podem ser trocados de fila, chamado de com Retroalimentação, e sem esse recurso, é chamado de sem Retroalimentação.

**4 – Algoritmo de Tempo Real**

- Dividido em sistemas críticos (*Hard Real-Time*) e não críticos (*Soft Real-Time*).

- Aplicados em Indústrias, Automóveis e Multimídia.

- Estabelecimento de metas temporais que devem ser cumpridas no tempo estabelecido se for um sistema de tempo real rígido. Se for de tempo real flexível, prioriza tarefas com maior prioridade, baseado no nível de criticidade.

- Possuem pontos de verificação de processos de tempo real que devem ser precisos.

***Hard Real-Time:***

- Necessita cumprir as metas temporais dos processos, antes que o tempo acabe, pois há deadline e isso pode resultar em danos graves para o sistema.

- Exemplos: ABS de um carro, tem que cumprir sua função de forma precisa para não gerar acidentes. Nesse caso, se houver descumprimento da meta temporal, o resultado grave é a morte de pessoas ou acidentes.

***Soft Real-Time***

- Possui metas temporais, mas caso houver deadline, as consequências são literalmente mais *soft* e não gera graves erros no sistema.

- Presente em Jogos de Computador e Aplicações Audiovisuais.

Já os Algoritmos Não Preemptivos compreendem algoritmos que não tem o poder de interromper voluntariamente os processos, mas somente quando o processo for finalizado, finalizado por bugs, morto por outros processos ou morto pelo usuário. De modo que a finalização libere a CPU voluntariamente. Esse tipo de escalonamento é mais útil em sistemas/eventos que não necessitem de respostas rápidas ou críticas, onde é mais útil a finalização natural do processo. Eles são divididos em 2 tipos:

**1 – Algoritmo FCFS (*First-Come First-Served*)**

- Funciona Baseado na ordem de chegada dos processos à fila de prontos.

- Um processo só recebe acesso aos recursos, quando o anterior for concluído ou bloqueado.

- Implementado baseado em uma fila FIFO (*First-In First-Out*).

- Simples de implementar.

- Altos tempos de espera.

- Favorece os processos mais lentos e que demandam mais tempo.

**2 – Algoritmo SJF (*Shortest Job First*)**

- A CPU é destinada ao processo com menor tempo de execução.

- A fila de prioridade baseia-se do tempo decrescente para o tempo crescente.

- Em casos de intervalos de tempos iguais, utiliza-se o método FIFO (*First-In First-Out*).

- Pode possuir preempção ou não, tendo em vista que esse tipo escalonamento é naturalmente não preemptivo, mas se um processo com tempo menor de execução do que o em execução for alocado na fila, o algoritmo força o atual a ceder a CPU ao de menor intervalo de tempo. Assim alterando sua preempção e recebendo o nome de SRTF (*Shortest Remaining Time First*).

- Reduz o tempo de espera.

- Difícil de prever o tempo de utilização da CPU.

Referências:

- <http://www.univasf.edu.br/~andreza.leite/aulas/SO/ProcessosEscalonamento.pdf>

- <https://www.oficinadanet.com.br/post/12781-sistemas-operacionais-o-que-e-escalonamento-de-processos>

- <https://www.dcce.ibilce.unesp.br/~aleardo/cursos/fsc/cap03.php>

Imagens:

- <https://embarcados.com.br/wp-content/uploads/2018/05/Kernel_RR.png>

- <https://deinfo.uepg.br/~alunoso/2020/SO/EscalonamentocomPrioridades/EscalonamentoPrioridades_arquivos/image002.jpg>