Fair-Share - Escalonamento de Fração Justa

O que é escalonamento?

O escalonamento é responsável pela “organização” de processos. Para explicar um escalonador, precisa entender o que é processo. Esse mesmo pode ser resumidamente definido como sendo o ambiente onde um programa é executado. Como dito, existem dois conceitos que em muitas vezes são confundidos, o de programa e o de processo. Basicamente, o que difere os dois termos, é que o programa é uma entidade estática e permanente, enquanto o processo é uma entidade dinâmica.

Os atuais sistemas operacionais têm muitos processos acontecendo simultaneamente, e muitas vezes, necessitam escolher qual processo ocupará a CPU. O escalonador nativo do sistema operacional é responsável por essa escolha (*scheduler*), o algoritmo usado para escalonar é o *scheduling algorithm*. Existem vários tipos de algoritmos escalonadores, com diferentes características compatíveis com seu desenvolvimento.

Um bom algoritmo de escalonador deve escolher o processo certo na hora certa, sabendo dar prioridade para o processo correto, e utilizar a CPU de forma eficiente, envolvendo alternar de modo usuário para núcleo; salvar o status do processo atual; salvar o mapa de memória do processo atual; selecionar um novo processo, executando o algoritmo de escalonamento; carregar o mapa de memória do novo processo; carregar o estado do novo processo; alternar de modo núcleo para usuário e reiniciar a execução de forma eficiente.

Quando escalonar

Apesar de haver vários processos sendo executados, alguns computadores não necessitam escalonamento, como os computadores pessoais. Para haver escalonamento, existem alguns eventos que incitam o escalonamento, como:

- A criação de um novo processo, quando a CPU precisa decidir entre continuar executando o processo pai ou iniciar a execução do processo filho.
- Com a finalização de um processo que “libera” espaço na CPU, outro processo deve ser escolhido para ocupá-lo.
- Com o bloqueio causado por um processo devido a E/S, ocasionado pela entrada de uma região crítica ou por qualquer outro motivo, outro processo precisa ser escolhido para executar, o motivo do bloqueio pode ser relevante na escolha do próximo a ser executado.

Critérios para um bom algoritmo escalonador

Para um algoritmo ser considerado eficiente, deve corresponder à 5 critérios definidos:

1 - Justiça: cada processo, durante a execução, deve receber uma parcela justa de tempo da CPU.

2 - Eficiência: Executando processos, a CPU deve estar 100% ocupada;

3 - Tempo de execução em bath: mínimo de tempo esperando saídas;

4 - Taxa de execução (throughput): maior número de trabalhos executados por hora;

5 - Tempo de resposta: mínimo tempo para usuários interativos.

Algoritmos de escalonamento

Conforme citado anteriormente, existem vários tipos de algoritmos de escalonamento, mas estão classificados em duas categorias:

- Não preemptivos: Seleciona um processo e o deixa em execução por um tempo indefinido, até o mesmo bloquear ou deixar a CPU voluntariamente.
- Preemptivos: Seleciona um processo e o deixa em execução até o mesmo bloquear ou atingir a próxima interrupção de relógio.

Existem diversos tipos de algoritmos de escalonamento, assim, os sistemas operacionais podem utilizar mais de um algoritmo ao mesmo tempo, não somente limitando-se à um algoritmo.

Fair Share

O escalonamento em sistema interativos tem como objetivo responder rapidamente às requisições do usuário e satisfazer as expectativa do usuário. O escalonamento por fração justa, como o nome já sugere, é responsável por dividir a porcentagem de processamento da CPU de forma igualitária. Por exemplo: se dois usuários utilizam a mesma máquina, um executa 9 processos, e o outro executa apenas um. Não é justo alocar 90% da CPU para um usuário apenas.

De forma prática, o algoritmo prioriza uma divisão do tempo justa de CPU entre todos os usuários. Por exemplo:

Usuários	**Porcentagem de Tempo em CPU**
Usuário 1	50 %
Usuário 2	50 %

Após a divisão de tempo entre usuários, cada usuário tem a sua porcentagem de CPU dividida entre os processos que são atribuídos a ele. Por exemplo:

Usuários	**Processos**	**Porcentagem de Tempo em CPU**
Usuário 1	Processo 1	25 %
	Processo 2	25 %
Usuário 2	Processo 3	25 %
	Processo 4	25 %

Assim o critério de justiça é aplicado e trabalha de forma funcional neste algoritmo de escalonamento.