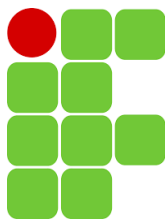


**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA BAIANO**

ESCALONAMENTO (PARTE 2)

Arthur Coquero¹, Emanuel Brito da Silva², Gabriel Silva Oliveira³ e
Lucas Silva de Oliveira⁴

Itapetinga - BA
2023



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAIANO

ESCALONAMENTO (PARTE 2)

Relatório apresentado ao
docente Hudson Barros de
Oliveira, como atividade da
disciplina Sistemas Operacionais,
do Curso de Bacharelado em
Sistemas de Informação, do
Instituto Federal de Educação,
Ciência e Tecnologia Baiano.

Itapetinga - BA
2023

SUMÁRIO

1. ESCALONADOR PREEMPTIVO E COOPERATIVO	3
2. ALGORITMOS DE ESCALONAMENTO PREEMPTIVOS	3
2.1 ROUND-ROBIN (RR)	3
2.2 SHORTEST REMAINING TIME FIRST (SRTF)	4
2.3 ESCALONAMENTO POR PRIORIDADE	4
2.4 ESCALONAMENTO DE FILA MULTINÍVEL	4
2.5 ESCALONAMENTO DE FILA DE FEEDBACK MULTINÍVEL	5
3. VANTAGENS DE DESVANTAGENS DO USO DE ESCALONAMENTO PREEMPTIVO	5
4. EXEMPLOS DE APLICAÇÃO EM SISTEMAS OPERACIONAIS	6
5. REFERÊNCIAS BIBLIOGRÁFICAS	6

1. ESCALONADOR PREEMPTIVO E COOPERATIVO

O escalonador de um sistema operacional pode ser preemptivo ou cooperativo(não-preemptivo):

Sistemas preemptivos: nestes sistemas um processo pode perder a vez no processador caso termine seu tempo, caso ele realize uma chamada de sistema ou uma interrupção seja realizada priorizando outro processo. Para qualquer evento que possa acarretar em troca do direito de uso do processador, o escalonador reavalia todos os processos da fila de processos prontos e decide se mantém ou substitui o processo em execução.

Sistemas cooperativos: o processo em execução permanece no processador o quanto for possível, só liberando o mesmo caso termine sua rotina, solicite uma operação de E/S ou libere explicitamente o processador, retornando para a fila de processos prontos. Esses sistemas são chamados de cooperativos, visto que, exige a cooperação de processos entre si na gestão do processador, para que todos possam executar.

Sistemas preemptivos mais sofisticados são os que permitem interrupções de processos no modo usuário e modo núcleo, fator importante para sistemas em tempo real, núcleos que permitem essa possibilidade são chamados núcleos preemptivos. Enquanto em sistemas preemptivos mais simples as interrupções ocorrem apenas no modo usuário.

2. ALGORITMOS DE ESCALONAMENTO PREEMPTIVOS

2.1 ROUND-ROBIN (RR)

Ao adicionar o fator de preempção no algoritmo First-Come, First-Served (FCFS), se origina o algoritmo conhecido como escalonamento por revezamento. No FCFS, o escalonador atende os processos em sequência, ou seja, na ordem da fila de tarefas prontas, e não há interrupções, sendo sua principal vantagem a simplicidade. Já no RR, existe o conceito de quantum do escalonador, que é basicamente um timer, esse quantum é dividido de forma igualitária para cada processo. Para implementar o algoritmo de escalonamento RR é utilizada a fila de processos pronta, e aplicando o algoritmo FIFO. Dessa forma o escalonador seleciona o primeiro processo da fila de processos pronta, dar-lhe a vez para o mesmo utilizar a CPU e inicia a contagem. O processo pode conseguir cumprir com sua rotina na quantidade de quantum disponível para ele, neste caso, o próprio processo libera voluntariamente a CPU, e o escalonador segue para o próximo processo na fila de processos pronta. Todavia se o processo exceder o seu quantum, o timer irá disparar e interromper o funcionamento do sistema, efetuando uma troca de contexto e colando o processo no fim da fila de processos prontos, em seguida, o escalonador seleciona o próximo processo na fila de processo prontos.

2.2 SHORTEST REMAINING TIME FIRST (SRTF)

O escalonador compara a duração prevista de cada novo processo que ingressa no sistema com o tempo de processamento restante dos demais processos presentes. Todavia, se a nova tarefa possuir um menor tempo restante, então ela recebe o processador, diferente de sua versão cooperativa SJF. Para estimar a duração de cada processo pode ser necessário a junção do SRTF e RR, essa abordagem pode ser de grande valor, principalmente em tarefas de entrada/saída.

2.3 ESCALONAMENTO POR PRIORIDADE

Muitos critérios podem ser utilizados para determinar o comportamento da fila de processos prontos e selecionar o processo a ser executado, os algoritmos anteriores utilizam alguns critérios para tal. Vários outros critérios poderiam ser utilizados, como o comportamento da tarefa(interativa, em lote, real), seu proprietário etc.

No escalonamento por prioridade, cada tarefa é associada a uma prioridade e a CPU é alocada para o processo com a maior prioridade. O escalonador utiliza o algoritmo de FCFS quando há processos de igual prioridade. Importante ressaltar que o algoritmo SJF é um algoritmo de prioridade(p) onde a maior prioridade é o inverso da prioridade prevista. Normalmente, são utilizados números inteiros para indicar os níveis de prioridade. Quando um processo é adicionado à fila de processos prontos, o escalonador compara o processo que está sendo executado com o que acabou de chegar, se o novo processo possuir prioridade mais alta, o escalonador interrompe a tarefa que estava sendo executada, trazendo ela novamente para a fila de prontos e entrega a CPU para o novo processo.

Uma desvantagem de algoritmos de escalonamento por prioridade é o bloqueio indefinido chamado starvation. De modo que, processos com baixa prioridade podem acabar não sendo executados caso os novos processos prontos que chegam à fila tenham prioridade maior, esses processos podem ser considerados bloqueados, pois dificilmente terão a oportunidade de execução, isso é algo que poderia ocorrer nos sistemas atuais que possuem alto tráfego de processos. Uma solução para evitar o starvation é o envelhecimento (aging). O aging envolve aumentar de forma gradual a prioridade dos processos que estão em espera na fila de prontos por muito tempo.

2.4 ESCALONAMENTO DE FILA MULTINÍVEL

Quando processos são facilmente classificados em diferentes grupos, ou seja, possuem diferentes requisitos de tempo de resposta e diferentes necessidades de escalonamento. Nesse tipo de escalonamento, a fila de processos prontos é dividida em várias filas dependendo do seu grupo, este grupo pode ser definido com base em alguma propriedade do processo, como tamanho, prioridade do processo ou tipo de processo.

Cada uma dessas filas possuem o seu próprio algoritmo de escalonamento, uma fila pode usar RR, outra SJF e outra por prioridade. Ademais, deve haver escalonamento entre as

filas, esse escalonamento normalmente é preemptivo de prioridade fixa. Pode ser usado para gerenciar processos de primeiro e segundo plano.

2.5 ESCALONAMENTO DE FILA DE FEEDBACK MULTINÍVEL

Um problema do escalonamento multinível é a impossibilidade de um processo ser realocado para outra fila, visto que, os processos são atribuídos permanentemente a uma fila quando entram no sistema. Para resolver esse problema foi criada a fila de feedback multinível, nela é permitido que os processos mudem de fila, de modo que, a ideia principal é separar os processos de acordo com seu tempo de CPU.

Dessa forma, se um processo usa muito tempo de CPU, ele será realocado para uma fila de prioridade mais baixa. Esse algoritmo deixa entrada/saída e processos interativos em filas de maior prioridade. Além do mais, um processo que espera muito tempo em uma fila de prioridade mais baixa pode ser realocado para uma fila de prioridade mais alta. O escalonador de fila de feedback multinível é o algoritmo mais geral para gerenciar o uso da CPU, visto que, o escalonamento de fila de feedback multinível busca encontrar um equilíbrio entre o tempo de resposta rápido para processos interativos e a justiça na distribuição de recursos do sistema, porém, ele é o algoritmo mais complexo pois é necessário definir valores para todos os parâmetros do mesmo.

3. VANTAGENS DE DESVANTAGENS DO USO DE ESCALONAMENTO PREEMPTIVO

Vantagens:

1. Justiça: ajuda a garantir uma distribuição justa de recursos do sistemas entre processos. Podendo tratar processos de forma adequada a depender de sua prioridade.
2. Multitarefa: é possível executar vários processos simultaneamente.
3. Resposta ágil: garante resposta rápida, visto que, é possível priorizar a execução de processos críticos, como interativos ou de alta prioridade.

Desvantagens:

1. Complexidade: como já dito anteriormente, os algoritmos de escalonamento preemptivos são mais complexos de serem implementados e gerenciados. Sendo necessário levar em conta as prioridades dos processos, garantir que o sistema seja justo, tenha respostas ágeis e gerencie bem as interrupções.
2. Problemas de sincronização: interrupções de processos podem acarretar problemas de sincronização, recursos, inconsistências ou erros no sistema.
3. Overhead: a necessidade de troca de contexto a todo o momento consome recursos significativos da CPU e pode gerar um aumento na latência do sistema.

4. EXEMPLOS DE APLICAÇÃO EM SISTEMAS OPERACIONAIS

1. Sistemas mais antigos, como o Windows 3.*, PalmOS 3 e MacOS 8 e 9 operavam de forma cooperativa. Mas os SOs atuais utilizam o escalonamento preemptivo para gerenciar a execução de processos e threads.
2. Sistemas embarcados também utilizam o escalonamento preemptivo, de modo que, garanta uma maior responsividade na interação com o usuário.
3. Servidores: é usado para otimizar a utilização de recursos da CPU e garantir que as solicitações de serviços sejam processados de forma eficiente.
4. Sistemas em tempo real: nesse tipo de sistema é essencial que haja precisão no tempo das respostas, com o escalonamento preemptivo, processos críticos podem ser executados no momento certo, interrompendo os demais processos com menor prioridade.

5. REFERÊNCIAS BIBLIOGRÁFICAS

TANENBAUM, Andrew S. Sistemas Operacionais Modernos. 4. ed. São Paulo: Pearson Education, 2015.

MAZIERO, Carlos A. Sistemas Operacionais: Conceitos e Mecanismos. Sagra Luzzatto, 2005.

SILBERSCHATZ, Abraham; GALVIN, Peter Baer; GAGNE, Greg. Operating System Concepts. 9. ed. John Wiley & Sons, 2012.