

Relatório de Análise do Escalonador de Processos

1. Justificativa de Design

A implementação do escalonador de processos foi baseada na utilização de uma lista encadeada simples como estrutura de dados para as filas de prioridade (alta, média e baixa) e a fila de processos bloqueados. Essa escolha foi motivada pela simplicidade de implementação e pela eficiência na operação de remoção de elementos no início da fila, característica fundamental para o comportamento de um escalonador.

2. Análise de Complexidade (Notação Big-O)

A complexidade das principais operações do escalonador é a seguinte:

Remoção do Início ($O(1)$): A operação de remoção de um processo do início de uma fila possui complexidade de tempo constante. Isso se deve ao fato de que a operação exige apenas a atualização do ponteiro head, independentemente do número de elementos na lista.

Inserção no Fim ($O(n)$): A inserção de um novo processo no final da fila tem uma complexidade de tempo linear. Essa complexidade é resultado da necessidade de percorrer todos os n elementos da lista, do início ao fim, para encontrar o último nó onde a nova inserção será realizada.

Reinserção no Fim ($O(n)$): De forma análoga à inserção, a reinserção de um processo que ainda não concluiu sua execução também possui complexidade de tempo linear, pois utiliza a mesma lógica de percorrer a fila até o final para adicionar o processo.

3. Mecanismo Anti-Inanição

Para evitar a inanição (starvation) de processos de baixa prioridade, foi implementado um mecanismo de controle. O sistema mantém um contador que monitora o número de execuções consecutivas de processos de alta prioridade. Após cinco execuções seguidas de processos de alta prioridade, o escalonador é obrigado a selecionar um processo da fila de prioridade média ou, se esta estiver vazia, um da fila de baixa prioridade. Esse procedimento garante que processos de menor prioridade tenham a oportunidade de serem executados, promovendo a equidade no sistema.

4. Gerenciamento de Processos Bloqueados (E/S)

Quando um processo em execução solicita uma operação de entrada/saída, como o acesso a um recurso de disco, ele não é executado e é imediatamente movido para uma fila de processos bloqueados. Para gerenciar essa fila, o escalonador implementa um ciclo simples: no início de cada novo ciclo de execução, o processo mais antigo da fila de bloqueados é 'desbloqueado' e retornado à sua fila de prioridade original, de onde poderá ser selecionado para execução em um ciclo futuro.

5. Pontos de Melhoria

O principal gargalo de desempenho do escalonador é a operação de inserção de processos no final da fila, que possui uma complexidade de tempo de $O(n)$. Para otimizar essa operação e reduzir sua complexidade para $O(1)$, uma melhoria significativa seria a adoção de uma estrutura de dados de lista duplamente encadeada ou a simples manutenção de um ponteiro adicional que sempre aponte para o último nó da lista. Essa modificação eliminaria a necessidade de percorrer toda a lista, resultando em um escalonador mais eficiente, especialmente em cenários com grande volume de processos.