

Relatório de Análise e Justificativa de Design

1. Justificativa de Design

A estrutura de dados escolhida para representar as listas de processos foi a Lista Encadeada implementada manualmente. Essa escolha se justifica porque o escalonador precisa realizar operações frequentes de inserção no final e remoção no início da lista, o que em uma lista encadeada possui custo constante ($O(1)$), desde que se mantenha referências para o primeiro e último nó. Além disso, a implementação manual garante maior controle sobre a lógica de manipulação dos processos e está de acordo com a regra do projeto, que proíbe o uso de estruturas prontas da linguagem.

2. Análise de Complexidade (Big-O)

• Inserção no final da lista: $O(1)$ • Remoção no início da lista: $O(1)$ • Impressão ou varredura completa: $O(n)$, onde n é o número de processos na lista. Essas operações são eficientes para o cenário do escalonador, que trabalha com múltiplas listas de prioridade e bloqueio. O custo linear só ocorre em casos de varredura, como exibição do estado do sistema, o que é aceitável.

3. Análise da Anti-Inanição

A regra de anti-inanição foi implementada de modo que, após cinco execuções consecutivas de processos de alta prioridade, o escalonador obrigatoriamente execute um processo de prioridade média ou baixa. Isso garante que processos de menor prioridade não fiquem indefinidamente sem CPU. Sem essa regra, haveria o risco de que um fluxo contínuo de processos de alta prioridade nunca permitisse a execução de processos menos urgentes, caracterizando inanição.

4. Análise do Bloqueio

Um processo que necessita do recurso 'DISCO' segue o seguinte ciclo de vida: 1. Quando chega para execução pela primeira vez, ele é detectado como dependente de DISCO. 2. Em vez de executar, é removido da lista de sua prioridade original e colocado no final da lista de bloqueados. 3. No início de cada ciclo do escalonador, o processo mais antigo da lista de bloqueados é desbloqueado, retornando ao final de sua lista original. 4. Ao voltar à lista de prioridade, ele retoma a execução normal, com seus ciclos necessários sendo decrementados até finalizar.

5. Ponto Fraco e Possível Melhoria

O principal gargalo de performance está na exibição do estado do sistema, que exige percorrer todas as listas de processos, resultando em custo $O(n)$. Em casos de listas muito grandes, isso pode impactar a eficiência da simulação. Uma melhoria teórica seria implementar

estruturas auxiliares de indexação ou contadores que armazenem informações resumidas das listas, reduzindo a necessidade de varreduras completas para operações de monitoramento.