

Relatório de Análise e Justificativa de Design

1) Justificativa de Design

O escalonador utiliza filas encadeadas simples para cada prioridade (Alta, Média, Baixa) e uma fila de Bloqueados. As operações de inserção e remoção são $O(1)$, garantindo eficiência. A lógica de prioridade com contador de 5 ciclos impede inanição, e o fluxo de processos com bloqueio em DISCO é modelado de forma simples e previsível.

2) Análise de Complexidade (Big-O)

- Enfileirar e desenfileirar: $O(1)$ - Carregamento de processos: $O(P)$ - Executar ciclo: $O(1)$ - Loop total: $\Theta(\sum \text{ciclos dos processos})$ - Impressão: $O(N)$ no número de processos

3) Análise da Anti-Inanição (Justiça)

A regra de limitar 5 execuções consecutivas da fila de Alta prioridade e forçar execuções das filas de Média ou Baixa evita starvation. Sem isso, processos de baixa prioridade poderiam nunca ser executados.

4) Análise do Bloqueio (Fluxo DISCO)

Um processo que precisa de DISCO é movido para a fila de Bloqueados sem consumir CPU. No ciclo seguinte, é desbloqueado e retornado à fila de sua prioridade original, retomando a execução normalmente.

5) Ponto Fraco e Melhoria Proposta

O maior gargalo é o I/O de impressão quando se usa `rodarEscalonadorImp`, que imprime todas as filas a cada ciclo. Melhorias teóricas incluem bufferização de logs, impressão incremental ou apenas por eventos importantes.