RELATÓRIO DE ANÁLISE E JUSTIFICATIVA DE DESIGN

Justificativa de Design:

Para implementar o escalonador eu optei por utilizar listas encadeadas circulares, separadas por três níveis de prioridade (alta, média e baixa), além do uso de uma lista auxiliar para os processos bloqueados. A escolha desse modelo foi feita por ser mais simples de se implementar e pela eficiência nas operações básicas como a inserção e remoção, que são realizadas constantemente. Além disso, as listas circulares permitem simular o "rodízio" entre processos que é essencial paro o escalonador.

Análise da Complexidade (Big-O):

Complexidade O(1):

- Inserção em uma fila
- Remoção do processo finalizado
- Desbloqueio de processos
- Escolha do próximo processo

Complexidade O(n):

- Troca de prioridade (anti inanição)

Análise da Anti-Inanição:

Ela evita o problema da inanição, pois se um processo permanece muito tempo em uma fila com baixa prioridade e não está recebendo sua vez, ele pode ser escalado para uma fila de prioridade superior temporariamente. Dessa forma é possível garantir um processo justo no escalonamento, pois mesmo os processos com menor prioridade irão ter sua chance de executar. Sem essa regra, um processo de baixa prioridade pode ficar pendente sem nunca ser executado.

Análise do Bloqueio:

Primeiramente ele entra na fila de prioridade conforme especificado. Ao chegar a sua vez de ser executado, o escalonador verifica se o recurso está disponível, se o recurso não estiver disponível, o processo é movido para a

lista de bloqueados, já se o recurso estiver livre, ele executa normalmente. Assim que o recurso for liberado, o processo retorna à sua fila de prioridade original, sendo executado até terminar seus ciclos e então é removido do sistema.

Ponto fraco:

O principal problema está no sistema de anti-inanição, pois essa operação pode pedir para percorrer toda a lista de processos, isso a torna O(n).

Melhoria teórica:

Uma forma de melhorar seria substituindo as listas por uma Heap de prioridades ou uma árvore balanceada. Assim, ao escolher o próximo processo e a promoção, poderiam ser feitas em notação O(log n), melhorando a eficácia quando se tem muitos processos. Outra melhoria seria implementar um contador de ciclos de espera para cada processo, de forma que promova os processos automaticamente quando chegassem em um determinado limite, sem ser necessário que percorra toda a lista.

Nota:

Foi feito o uso de ferramentas de IA como apoio para a melhoria da redação do relatório, na organização do arquivo "README" para formatação mais profissional e na interpretação de mensagens de erro de execução e compilação.