Relatório Trabalho P1 -

João Evangelista, Lucas Ribeiro, Adriel Martim

Curso: Engenharia de Software	Turma: Borg
Disciplina: Algoritmos e Estrutura de Dados I	Professor(a): Dimmy Magalhães
Prova: P1	Data de abertura e entrega: 21/08/2025 - 17/09/2025
Aluno(a): João Evangelista, Lucas Ribeiro Adriel Martim	

1. Justificativa de Design

Adotamos a estrutura de **lista circular encadeada** para cada fila de prioridade e para a fila de processos bloqueados. Esta escolha se justifica pela sua eficiência, que se manifesta em diversos pontos:

- **Otimização de Operações Essenciais:** Permite a remoção e inserção de elementos em tempo constante (O(1)) tanto no início quanto no final da lista, o que é crucial para as operações de escalonamento.
- **Reaproveitamento de Processos:** A natureza circular facilita o retorno de processos não finalizados para o fim da fila, eliminando a necessidade de percorrer a lista inteira para reinseri-los.
- **Gerenciamento Simplificado:** A organização das filas por prioridade distintas simplifica significativamente a lógica para selecionar o próximo processo a ser executado e a implementação de mecanismos anti-inanição.

2. Complexidade (Big-O) das Operações

- **Adicionar processo ao final da fila:** O(1) – Requer apenas a atualização dos ponteiros do último nó.
- **Remover processo do início da fila:** O(1) – Requer apenas a atualização dos ponteiros do primeiro nó.
- **Desbloquear processo:** O(1) – Implica remover o processo do início da fila de bloqueados e inseri-lo no final da fila de origem.
- **Verificar se todas as filas estão vazias:** O(1) – Envolve apenas a checagem dos ponteiros das listas.

3. Análise da Anti-Inanição

Para garantir que todos os processos sejam eventualmente executados, a lógica anti-inanição estabelece que, após cinco execuções consecutivas de processos de alta prioridade, o escalonador deve obrigatoriamente executar um processo de prioridade média ou baixa.

Justiça: Essa medida é crucial para promover a justiça no uso da CPU, impedindo que processos de menor prioridade fiquem em espera indefinida.

Risco sem a regra: A ausência dessa regra poderia levar à inanição (starvation) de processos de baixa prioridade, que nunca seriam executados se houvesse um fluxo constante de processos de alta prioridade.

4. Análise do Bloqueio de "DISCO"

Fluxo de vida de um processo que requer acesso a "DISCO":

1. **Entrada na Fila:** O processo é inserido na fila correspondente à sua prioridade.
2. **Primeira Requisição (Bloqueio Inicial):** Se, ao ser selecionado pela primeira vez, o processo necessitar de acesso a "DISCO", ele é imediatamente movido para a fila de bloqueados sem ser executado.
3. **Desbloqueio e Retorno:** No início de cada ciclo, o processo que está há mais tempo na fila de bloqueados é desbloqueado e reinserido no final da sua fila de prioridade original.
4. **Execução:** Quando novamente selecionado, o processo executa suas tarefas normalmente, decrementando seus ciclos até a conclusão ou até ser recolocado no final da fila.

5. Gargalos de Performance e Propostas de Melhoria

Gargalos Atuais:

- **Simulação Sequencial:** O principal gargalo reside na simulação sequencial dos ciclos, que é particularmente ineficiente em cenários com muitas operações de I/O (impressões no console) ou listas extensas.
- **Escalonamento Limitado:** O escalonador atual não otimiza o desempenho, pois não prioriza desbloqueios em lote nem paraleliza as operações.

Melhorias Propostas:

- **Paralelismo:** Implementar threads para permitir a execução e o desbloqueio de processos de forma paralela.
- **Filas de Desbloqueio Prioritárias:** Adotar estruturas como filas de prioridade para gerenciar os desbloqueios de forma mais inteligente.
- **Otimização de I/O:** Reduzir a quantidade de impressões no console ou utilizar logs assíncronos para evitar interrupções no ciclo principal.