



INSTITUTO FEDERAL DO ACRE
CURSO DE TECNOLOGIA DE SISTEMAS PARA INTERNET

RELATÓRIO FINAL: ESTRUTURAS DE DADOS

RIO BRANCO
2025

**MARCUS VINICIUS DE SOUZA MALVEIRA
JOÃO VICTOR DE SOUZA MALVEIRA
EDWILLER DE LIMA ARAUJO
MYRELLA DA SILVA SOUZA**

RELATÓRIO FINAL: ESTRUTURAS DE DADOS

Relatório apresentado como requisito de
obtenção da nota parcial da N2 na
disciplina de Interação Homem-
Computador do 1º período do curso de
Tecnologia de Sistemas para Internet

Prof. Orientador: Me. Alvaro Lopes Rios

RIO BRANCO

2025

SUMARIO

INTRODUÇÃO	6
ALGORITMO IMPLEMENTADO	7
ESTRUTURAS DE DADOS UTILIZADAS	7
ESTRUTURA DO PROJETO	8
COMO COMPILAR E EXECUTAR.....	8
ENTRADA DO PROGRAMA.....	8
SAIDA DO PROGRAMA.....	9
MÉTRICAS CALCULADAS	9
CRITERIO DE DESEMPATE	9
EXEMPLO DE EXECUÇÃO	10
DECISÕES DE PROJETO	10
REFERÊNCIAS	11

INTRODUÇÃO

Este trabalho apresenta o desenvolvimento de um simulador de escalonamento de processos, elaborado para a disciplina de ESTRUTURAS DE DADOS, com o foco no algoritmo Round-Robin. Com o objetivo de entender implementar e estudar o funcionamento do algoritmos de escalonamento na CPU.

O simulador permite a inserção manual do processo, a definição do quantum(duração) e a execução do algoritmo de forma preventiva, respeitando as características do Round-Robin. Cada processo é representado por uma estrutura de dados contendo informações como o identificador(id), tempo de chegada, duração de CPU e prioridade.

Como diferencial foi utilizada uma fila ordenada para gerenciar os processos prontos para a execução, permitindo maior controle sobre a ordem de atendimento e possibilitando a aplicação de critérios de desempate, como prioridade e tempo de chegada. Essa abordagem reforça a aplicação prática de conceitos de estruturas de dados no contexto de sistemas operacionais.

Ao final da simulação, o programa apresenta uma tabela com os resultados individuais dos processos, bem como métricas de desempenho, como tempo médio de espera e tempo médio de turnaround, possibilitando a análise e comparação do comportamento do algoritmo em diferentes cenários.

ALGORITMO IMPLEMENTADO

O Round Robin é um algoritmo de escalonamento preemptivo que distribui o tempo de CPU de forma justa entre os processos. Cada processo recebe uma fatia de tempo (quantum) para execução. Caso não finalize dentro desse intervalo, o processo é interrompido e reinserido na fila de prontos, aguardando uma nova oportunidade de execução.

Neste projeto, o Round Robin foi implementado utilizando uma fila ordenada, mantendo as características principais do algoritmo (quantum e preempção), porém com controle adicional da ordem de inserção dos processos. Essa abordagem permite considerar atributos como prioridade e tempo de chegada sem descaracterizar o algoritmo.

ESTRUTURAS DE DADOS UTILIZADAS

A principal estrutura de dados utilizada é uma fila encadeada ordenada, composta por nós que armazenam os processos e um ponteiro para o próximo elemento.

A ordenação da fila é definida pelos seguintes critérios:

1. **Prioridade** (menor valor indica maior prioridade);
2. **Tempo de chegada** (processos que chegam antes têm preferência);
3. **Identificador do processo (ID)**, como critério final de desempate.

Essa estrutura foi escolhida por permitir:

- Inserções ordenadas de forma eficiente;
- Remoção do primeiro elemento da fila em tempo constante;
- Adequação natural ao comportamento do Round Robin.

ESTRUTURA DO PROJETO

O projeto foi implementado em um unico arquivo em linguegem C, contendo todas as estruturas de dados, funções auxiliares e lógica principal do escalonador.

A organização interna do código foi realizada de forma logica, separando:

- Definições das estruturas (processos, fila e No);
- As funções responsaveis pelo gerenciamento da fila;
- A função principal (main), que controla a simulação do escalonamento;
- O cálculo das métricas de desempenho ao final da execução.

Essa abordagem foi adotada para facilitar a compreensão do fluxo do programa, mantendo o código compacto e organizado.

COMO COMPILAR E EXECUTAR

Compilação: utilize o Compilador GCC:

`Gcc fila.c -o simulador`

Execução:

`./simulador`

ENTRADA DO PROGRAMA

O simulador permite a inserção manual de processos, solicitando os seguintes dados:

ID: identificador único do processo;

Tempo de chegada: instante em que o processo entra no sistema;

Duração (burst time): tempo total de CPU necessário;

Prioridade: número inteiro onde menor valor indica maior prioridade (desconsiderado em Round-Robin).

Além disso, o usuário define o quantum, que representa a fatia de tempo máxima que cada processo pode utilizar a CPU em uma execução.

SAIDA DO PROGRAMA

Ao final da simulação, o programa exibe uma tabela contendo, para cada processo:

- Identificador do processo;
- Tempo de espera;
- Tempo de turnaround.

Também são exibidas as médias das métricas, permitindo uma análise geral do desempenho do algoritmo.

MÉTRICAS CALCULADAS

Tempo de espera: representa o tempo total que um processo permaneceu na fila de prontos aguardando execução

Calculo: $\text{Tempo de espera} = \text{turnaround} - \text{duração}$

Turnaround: representa o tempo total desde a chegada do processo ao sistema até sua finalização

Calculo: $\text{turnaround} = \text{Tempo de finalização} - \text{Tempo de chegada}$

As métricas médias são obtidas pela soma dos valores individuais dividida pelo número total de processos.

CRITERIO DE DESEMPATE

Em situações onde dois ou mais processos apresentam valores iguais para determinado critério, foram adotadas as seguintes regras de desempates:

1. Menor prioridade;
2. Menor tempo de chegada;
3. Menor ID;

Esses criterios garatem previsibilidade e consistência no comportamento do algoritmo.

EXEMPLO DE EXECUÇÃO

Processos	Chegada	Duração	Prioridade
P1	0	5	1
P2	1	3	2
P3	2	1	1

Quantum: 2

EXECUÇÃO (linha do tempo)

Tempo 0: P1

Tempo 2: P1

Tempo 4: P2

Tempo 6: P3

Tempo 7: P1

Tempo 8: P2

Ao final, a tabela de métricas é apresentada com os tempos de espera e turnaround de cda processo.

DECISÕES DE PROJETO

- Utilização de fila ordenada para valorizar os campos de prioridade e tempo de chegada;
- Implementação modular para facilitar manutenção e expansão;
- Critérios de desempate definidos explicitamente para evitar comportamentos ambíguos;
- Cálculo das métricas seguindo definições clássicas da literatura.

REFERÊNCIAS

- Material didático da disciplina de Estrutura de Dados.