Trabalho M2 – Escalonadores

Universidade do Vale do Itajaí
Disciplina: Sistemas Operacionais
Professor: Michael Douglas C A
Trabalho M2 – Escalonadores

Instruções bases:

1. Esta avaliação deve ser feita em trio. Trabalhos individuais não serão aceitos.

2. Data de entrega final: 16/10/2025 até 23:59. Não serão aceitos trabalhos entregues em atraso.

3. Esta avaliação tem por objetivo consolidar o aprendizado sobre escalonadores.

4. A implementação deverá ser desenvolvida utilizando c++ ou Java . O uso de bibliotecas de terceiros deverão ser explicado com uma justificativa válida que ficará ao critério do professor aceitar ou não.

5. O sistema deve ser entregue funcionando corretamente.

6. Deve ser apresentado um relatório eletrônico em formato PDF (em outro formato é descontado 2,0 pontos) que contenha:

    a. Identificação dos autores e do trabalho.

    b. Enunciado do projeto.

    c. Explicação e contexto da aplicação para compreensão do problema tratado pela solução.

    d. Resultados/simulações.

    e. Trechos de códigos pertinetes dasolução.

    f. Análise e discussão sobre os resultados finais.

7. Seguindo a mesma diretriz do trabalho anterior, deverá ser disponibilizado os códigos da implementação juntamente com o relatório (salvo o caso da disponibilidade em repositório aberto do aluno, que deve ser fornecido o link). Também deve ser apresentado o trabalho em aula para o professor. Na apresentação do trabalho, não é necessário utilizar slides, apenas apresentar o código e sua execução (compilar na apresentação).

Mais detalhes no decorrer do documento.

Descrição do projeto a ser desenvolvido

Projeto

1. Objetivo

Desenvolver um simulador completo do algoritmo de escalonamento Round Robin (RR), capaz de representar de forma clara o funcionamento de um sistema operacional multiprocessado. O trabalho visa consolidar os conceitos de escalonamento, controle de processos, bloqueios de E/S, preempção, uso de quantum e sincronização.

2. Descrição do Projeto

O simulador deverá reproduzir o funcionamento de um escalonador Round Robin com as seguintes características:

1. Quantum fixo e dinâmico:
   a) O valor do quantum deve poder ser definido pelo usuário.
   b) O modo dinâmico ajusta o quantum conforme a carga atual do sistema.
2. Execução em múltiplos núcleos de CPU (mínimo 2 núcleos).
3. Gerenciamento de bloqueios (E/S): processos podem entrar em bloqueio e retornar à fila.
4. Inserção dinâmica de processos durante a execução (entrada manual ou via arquivo).
5. Controle de tempo global, mantendo coerência entre execuções e desbloqueios.

3. Formato de Entrada e Saída

A entrada pode ser feita por arquivo texto (.txt) ou manualmente durante a execução.

Formato de arquivo:

ID | TempoChegada | Execucao1 | Bloqueio? | Espera | Execucao2

Exemplo:

P1 | 0 | 4 | S | 3 | 2

P2 | 1 | 5 | N | 0 | 0

P3 | 2 | 3 | S | 4 | 1

Saída esperada:

1. Tabela com tempo de espera, turnaround, número de trocas de contexto e uso da CPU.
2. Gráfico temporal (estilo Gantt) mostrando a execução dos processos nos núcleos.
3. Log textual dos eventos, indicando chegadas, execuções e bloqueios.

Exemplo de gráfico textual:

Núcleo 1: | P1 | P1 | P2 | P3 | P3 | P4 |

Núcleo 2: | P2 | P2 | P1 | P1 | P3 | - |

Tempo: 0 1 2 3 4 5 6

4. Implementação Técnica

A implementação pode ser feita em C++, Java ou outra linguagem aprovada pelo professor.
Faça o uso de threads para simular os núcleos de CPU e a chegada dinâmica de processos. As filas devem ser utilizadas para gerenciar processos prontos, bloqueados e em execução.

Estrutura de processo:

```
struct Processo {
  string id;
  int tempoChegada;
  int tempoExecucao;
  int tempoRestante;
  int quantumRestante;
  string estado; // PRONTO, EXECUTANDO, BLOQUEADO, FINALIZADO
};
```

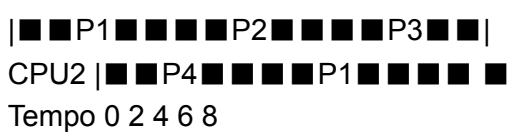
Lógica de execução mínima:

1. Incrementar o tempo global a cada passo.
2. Verificar desbloqueios e novas chegadas.
3. Alocar processos em núcleos disponíveis.
4. Atualizar o quantum restante e decidir preempções.
5. Registrar os eventos para o gráfico e relatório.

5. Parte Gráfica

O simulador deve exibir uma representação gráfica (linha do tempo) com o comportamento dos processos. Essa visualização pode ser feita em formato textual, gráfico simples ou interface visual. Cada núcleo deve ter sua própria linha no gráfico, representando os períodos de execução e espera.

Exemplo de saída visual esperada:



6. Relatório Técnico

O relatório final deve ser entregue em formato PDF e conter as seguintes seções:

1. Identificação dos autores (nomes e RA).
2. Enunciado resumido da atividade.
3. Descrição detalhada do sistema implementado, incluindo estruturas de dados e fluxos de execução.
4. Diagramas de sequência e de fluxo da aplicação.
5. Resultados obtidos: tabelas, gráficos e registros de eventos.
6. Discussão e análise dos resultados, incluindo variação de desempenho conforme o quantum e número de núcleos.

7. Entregas e Prazos

O trabalho deverá ser realizado em trio.
Primeira entrega parcial: até o dia **08/10/2025 às 21:45**, com apresentação do que foi desenvolvido até o momento diretamente ao professor.
Entrega final: até o dia **16/10/2025 às 23:59**, contendo o simulador completo e o relatório técnico final.
Trabalhos entregues após o prazo serão desconsiderados.
Forma de entrega:
- Enviar os arquivos de código e o relatório em formato PDF.
- Caso o código esteja em repositório online, fornecer o link de acesso.
- A apresentação final será prática, com compilação e execução do programa.

8. Recomendações Gerais

- Estruture o código em módulos e utilize boas práticas de programação.
- Comente as seções principais do código.
- Teste a aplicação com diferentes tamanhos de quantum e número de núcleos.
- Utilize diagramas claros e legíveis para representar o fluxo e a sequência do sistema.