**Universidade Feevale** | **Curso de Sistemas Operacionais** –**Grupo 03**

Nome: Herik Rafael Peter da Silva, Mateus Piva e Augusto...

1. **Objetivos do Trabalho**

O objetivo deste trabalho é projetar e desenvolver um simulador de escalonamento de processos baseado no algoritmo Round Robin com Feedback, utilizando a linguagem Java, e sem o uso de orientação a objetos. O simulador deve gerenciar diferentes filas de prioridade, dispositivos de I/O e simular a execução de processos com diferentes características de tempo de CPU e entrada/saída.

**Este projeto visa:**

* Aplicar na prática os conceitos estudados sobre escalonamento e gerência de processos.
* Estimular a organização do trabalho em grupo.
* Desenvolver uma solução funcional com base em premissas definidas pelo grupo.
* Compreender a dinâmica de preempção, filas e bloqueios em sistemas operacionais reais.

1. **Premissas Adotadas para o Simulador**

As seguintes premissas foram definidas pelo grupo para a simulação:

| **Item** | **Valor / Descrição** |
| --- | --- |
| Limite máximo de processos | 5 processos |
| Quantum (fatia de tempo) | 4 unidades de tempo |
| Tempo de CPU por processo | Aleatório entre 5 e 10 unidades |
| Tipo de I/O por processo | Aleatório entre disco, fita magnética e impressora |
| Tempo de I/O (por tipo) | Disco: 6, Fita: 8, Impressora: 10 |
| Gerência de processo (PCB) | PID, Prioridade, Tempo restante de CPU, Estado, Tipo de I/O, Tempo de I/O |
| Filas de escalonamento | Alta prioridade, Baixa prioridade, Fila de I/O |
| Destino após eventos | Novo processo → fila alta Preempção → fila baixa I/O Disco → fila baixa I/O Fita e Impressora → fila alta |

1. **Estrutura do Simulador**
   1. **Linguagem utilizada**

* Java (sem uso de orientação a objetos).
  1. **Estrutura de dados**
* Os processos foram representados por uma estrutura interna (PCB) contendo os dados do processo.
* As filas foram implementadas com as estruturas padrão do Java:
  + *Queue<PCB>* para as filas de alta e baixa prioridade.
  + *Queue<PCB>* para a fila de I/O.
* O simulador é iterativo, executando um ciclo por vez até que todos os processos sejam finalizados.

1. **Funcionamento do Simulador**
   1. **Fluxo do Ciclo**
2. Verifica e atualiza os tempos dos processos na fila de I/O.
3. Escolhe o processo com maior prioridade para execução:
   * Executa por no máximo o quantum (4 unidades de tempo).
   * Se finalizar: é removido.
   * Se requisitar I/O: é bloqueado e enviado para fila de I/O.
   * Se for preemptado: volta para fila de baixa prioridade.
4. Movimenta processos da fila de I/O para as filas de prontidão de acordo com o tipo do dispositivo.

**4.2 Finalização**

* O simulador termina quando todos os processos estão finalizados.

1. **Captura de Saída (Exemplo)**:

[CICLO 0]

Processo PID 1 criado com 9 unidades de CPU. I/O: disco

Processo PID 2 criado com 6 unidades de CPU. I/O: fita

Processo PID 3 criado com 7 unidades de CPU. I/O: impressora

Processo PID 4 criado com 8 unidades de CPU. I/O: disco

Processo PID 5 criado com 10 unidades de CPU. I/O: fita

[CICLO 1]

Executando processo PID 1 (prioridade: alta)

Processo PID 1 requisitou I/O (disco)

[CICLO 2]

Executando processo PID 2 (prioridade: alta)

Processo PID 2 preemptado e movido para fila de baixa prioridade.

[CICLO 3]

Processo PID 1 concluiu I/O (disco) e foi movido para fila baixa

[CICLO 4]

Executando processo PID 3 (prioridade: alta)

Processo PID 3 requisitou I/O (impressora)

[CICLO 5]

Executando processo PID 4 (prioridade: alta)

Processo PID 4 preemptado e movido para fila de baixa prioridade.

[CICLO 6]

Processo PID 3 concluiu I/O (impressora) e foi movido para fila alta

[CICLO 7]

Executando processo PID 5 (prioridade: alta)

Processo PID 5 preemptado e movido para fila de baixa prioridade.

[CICLO 8]

Executando processo PID 3 (prioridade: alta)

Processo PID 3 preemptado e movido para fila de baixa prioridade.

[CICLO 9]

Executando processo PID 1 (prioridade: baixa)

Processo PID 1 preemptado e movido para fila de baixa prioridade.

[CICLO 10]

Executando processo PID 2 (prioridade: baixa)

Processo PID 2 requisitou I/O (fita)

[CICLO 11]

Executando processo PID 4 (prioridade: baixa)

Processo PID 4 requisitou I/O (disco)

[CICLO 12]

Processo PID 2 concluiu I/O (fita) e foi movido para fila alta

[CICLO 13]

Processo PID 4 concluiu I/O (disco) e foi movido para fila baixa

[CICLO 14]

Executando processo PID 2 (prioridade: alta)

Processo PID 2 finalizado

[CICLO 15]

Executando processo PID 5 (prioridade: baixa)

Processo PID 5 preemptado e movido para fila de baixa prioridade.

[CICLO 16]

Executando processo PID 3 (prioridade: baixa)

Processo PID 3 finalizado

[CICLO 17]

Executando processo PID 1 (prioridade: baixa)

Processo PID 1 finalizado

[CICLO 18]

Executando processo PID 4 (prioridade: baixa)

Processo PID 4 finalizado

[CICLO 19]

Executando processo PID 5 (prioridade: baixa)

Processo PID 5 finalizado

[FIM]

Todos os processos finalizados em 20 ciclos.

1. **Conclusão**

O simulador desenvolvido atende todos os requisitos do trabalho, com uma implementação funcional do algoritmo Round Robin com Feedback, respeitando o tempo de quantum, filas por prioridade e diferentes comportamentos conforme o tipo de entrada/saída.

A prática foi essencial para entender:

* A dinâmica da preempção e realimentação entre filas.
* O impacto dos tempos de I/O no escalonamento.
* A importância de decisões de design no comportamento do sistema.

Além disso, permitiu aplicar conceitos fundamentais de sistemas operacionais de forma prática e colaborativa.