

Relatório: Controle de Recursos com Semaphore para Múltiplas Impressoras

1. Objetivo

Este exemplo demonstra o uso de **Semaphore** em Java para controlar o acesso concorrente a múltiplas impressoras. O sistema permite que até três impressoras sejam utilizadas simultaneamente, distribuindo os trabalhos de impressão de forma ordenada entre elas.

2. Estrutura do Código

- **Trabalho:**
Representa uma tarefa de impressão. Cada instância de **Trabalho** solicita a impressão de um documento através da **FilaImpressao**.
- **FilaImpressao:**
Gerencia a impressão de trabalhos utilizando um **Semaphore** com 3 permissões. Um array de booleanos (**impressorasLivres**) é utilizado para indicar quais impressoras estão disponíveis, e um **Lock** (**lockImpressoras**) protege o acesso a esse array. O método **imprimirTrabalho** simula a impressão, exibindo os horários de início e término da operação.
- **Principal:**
Inicializa o sistema criando uma instância de **FilaImpressao** e 12 threads, cada uma executando um **Trabalho**. As threads demonstram o acesso concorrente às impressoras.

3. Fluxo de Execução

1. A classe **Principal** cria uma instância de **FilaImpressao** e 12 threads, cada uma executando um objeto **Trabalho**.
2. Cada **Trabalho** chama o método **imprimirTrabalho** da **FilaImpressao**, que adquire uma permissão do **Semaphore** para acessar uma impressora.
3. O método **obterImpressora** é chamado para selecionar uma impressora livre, marcando-a como ocupada.
4. A impressão é simulada por um tempo aleatório (em segundos). Após a impressão, a impressora é liberada e o **Semaphore** é atualizado.
5. O uso do **Semaphore** com 3 permissões garante que, no máximo, três trabalhos sejam processados simultaneamente, controlando o acesso ao recurso compartilhado.

4. Exemplo de Execução

Saída do Console:

Thread 0 em: 12:34.567

Thread 0: FilaImpressao: Imprimindo um trabalho na Impressora 0 durante 3 segundos

Thread 0 libera semáforo em: 12:34.570

Thread 1 em: 12:34.567

Thread 1: FilaImpressao: Imprimindo um trabalho na Impressora 1 durante 5 segundos

Thread 1 libera semáforo em: 12:34.572

...

5. Conclusão

O exemplo evidencia como o uso de Semaphore permite o controle eficiente do acesso a múltiplos recursos, neste caso, impressoras. Ao limitar o número de trabalhos simultâneos a 3, o sistema previne conflitos e garante a execução ordenada dos trabalhos de impressão. A implementação do método obterImpressora, protegida por um lock, assegura que a distribuição dos trabalhos entre as impressoras seja feita de forma segura e eficiente, mantendo a integridade do sistema em ambientes concorrentes.