

Relatório Laboratório Tempo e estado global

Josué Sucupira de Oliveira - 540630

- **Processos:** Três processos distintos foram simulados como Threads independentes do Java, permitindo a execução concorrente e não determinística.
- **Comunicação:** A troca de mensagens entre os processos foi modelada utilizando filas de bloqueio concorrentes (`java.util.concurrent.BlockingQueue`). Essa abordagem simula de forma eficaz um canal de comunicação assíncrono, onde um processo pode enviar uma mensagem e continuar sua execução sem esperar pelo recebimento.
- **Simulação de Eventos e Latência:** Em seu laço de execução principal, cada processo decide aleatoriamente entre realizar um evento interno ou enviar uma mensagem a outro processo. Uma pausa artificial (`Thread.sleep()`) com duração variável foi introduzida para simular o tempo de processamento e a latência da rede, um fator crucial para expor as inconsistências dos relógios físicos.

Implementação e Resultados

A prática foi dividida em três partes, cada uma implementando um tipo de relógio diferente.

Parte 1: Relógios Físicos

- **Objetivo Específico:** Demonstrar a falta de confiabilidade do relógio físico do sistema (`System.currentTimeMillis()`) para estabelecer uma ordem global e causal de eventos.
- **Lógica de Implementação:** Cada evento (interno, envio ou recebimento) foi registrado no log com seu respectivo timestamp físico local, sem nenhum outro mecanismo de sincronização.

Resultados (Log de Execução):

```
/home/josue/documentos/faculdade/7- semestre/sistemas-distribuidos/tempo_e
--- INICIANDO SIMULAÇÃO: PARTE 1 - RELÓGIOS FÍSICOS ---
[P0] [Físico: 1753387275705] -> enviou mensagem para P2
[P2] [Físico: 1753387276304] -> enviou mensagem para P1
[P2] [Físico: 1753387276315] -> recebeu a mensagem: 'Olá de P0'
[P1] [Físico: 1753387276503] -> realizou um evento interno.
[P1] [Físico: 1753387276503] -> recebeu a mensagem: 'Olá de P2'
[P0] [Físico: 1753387276948] -> enviou mensagem para P1
[P0] [Físico: 1753387277474] -> enviou mensagem para P2
[P2] [Físico: 1753387277547] -> realizou um evento interno.
[P2] [Físico: 1753387277548] -> recebeu a mensagem: 'Olá de P0'
[P2] [Físico: 1753387277769] -> realizou um evento interno.
[P0] [Físico: 1753387277781] -> enviou mensagem para P2
[P1] [Físico: 1753387277878] -> enviou mensagem para P0
[P1] [Físico: 1753387277879] -> recebeu a mensagem: 'Olá de P0'
[P0] [Físico: 1753387278035] -> realizou um evento interno.
[P0] [Físico: 1753387278036] -> recebeu a mensagem: 'Olá de P1'
[P1] [Físico: 1753387278495] -> realizou um evento interno.
[P2] [Físico: 1753387279124] -> enviou mensagem para P1
[P2] [Físico: 1753387279125] -> recebeu a mensagem: 'Olá de P0'
[P1] [Físico: 1753387279677] -> enviou mensagem para P2
[P1] [Físico: 1753387279678] -> recebeu a mensagem: 'Olá de P2'
[P2] [Físico: 1753387280134] -> realizou um evento interno.
[P2] [Físico: 1753387280134] -> recebeu a mensagem: 'Olá de P1'
[P1] [Físico: 1753387280361] -> realizou um evento interno.
```

Análise dos Resultados:

O log de execução da Parte 1 expõe claramente a problemática dos relógios físicos. Embora no print específico possa não ocorrer o paradoxo de um recebimento ter um timestamp anterior ao do envio, a ordem dos eventos registrados no console é confusa e não reflete as verdadeiras dependências causais. Observando a sequência de eventos, é impossível para um observador externo determinar a ordem exata em que os eventos ocorreram no sistema como um todo. As pequenas diferenças de tempo entre eventos em processos distintos e o entrelaçamento dos logs mostram que qualquer tentativa de ordenação baseada nesses timestamps seria frágil e propensa a erros, justificando a necessidade de relógios lógicos.

Parte 2: Relógio Lógico de Lamport

- **Objetivo Específico:** Implementar o algoritmo de relógio lógico de Lamport para garantir a propriedade da ordem causal (se $A \rightarrow B$, então $L(A) < L(B)$).
- **Lógica de Implementação:** Cada processo manteve um contador inteiro. O contador era incrementado a cada evento. Ao enviar uma mensagem, o valor do contador era anexado. Ao receber, o processo atualizava seu contador local para $\max(L_{\text{local}}, L_{\text{recebido}}) + 1$.

Resultados (Log de Execução):

```
7/51/bin/env 7/51/cib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages
Simulador
--- INICIANDO SIMULAÇÃO: PARTE 2 - RELÓGIOS FÍSICOS ---
[P2] [Lamport: 1] [Físico: 1753387453362] -> realizou um evento interno.
[P1] [Lamport: 1] [Físico: 1753387453397] -> realizou um evento interno.
[P1] [Lamport: 2] [Físico: 1753387453474] -> enviou mensagem para P0
[P0] [Lamport: 1] [Físico: 1753387454241] -> enviou mensagem para P1
[P0] [Lamport: 3] [Físico: 1753387454243] -> recebeu mensagem. Timestamp da msg: 2
[P1] [Lamport: 3] [Físico: 1753387454465] -> enviou mensagem para P0
[P1] [Lamport: 4] [Físico: 1753387454465] -> recebeu mensagem. Timestamp da msg: 1
[P0] [Lamport: 4] [Físico: 1753387454759] -> realizou um evento interno.
[P0] [Lamport: 5] [Físico: 1753387454760] -> recebeu mensagem. Timestamp da msg: 3
[P2] [Lamport: 2] [Físico: 1753387454777] -> realizou um evento interno.
[P2] [Lamport: 3] [Físico: 1753387454911] -> enviou mensagem para P0
[P2] [Lamport: 4] [Físico: 1753387454954] -> realizou um evento interno.
[P1] [Lamport: 5] [Físico: 1753387455003] -> realizou um evento interno.
[P0] [Lamport: 6] [Físico: 1753387455627] -> enviou mensagem para P2
[P0] [Lamport: 7] [Físico: 1753387455628] -> recebeu mensagem. Timestamp da msg: 3
[P2] [Lamport: 5] [Físico: 1753387455636] -> enviou mensagem para P0
[P2] [Lamport: 7] [Físico: 1753387455636] -> recebeu mensagem. Timestamp da msg: 6
[P1] [Lamport: 6] [Físico: 1753387455828] -> enviou mensagem para P0
[P0] [Lamport: 8] [Físico: 1753387456353] -> enviou mensagem para P1
[P0] [Lamport: 9] [Físico: 1753387456354] -> recebeu mensagem. Timestamp da msg: 5
[P0] [Lamport: 10] [Físico: 1753387457088] -> realizou um evento interno.
[P0] [Lamport: 11] [Físico: 1753387457088] -> recebeu mensagem. Timestamp da msg: 6
```

Análise dos Resultados:

O resultado da Parte 2 demonstra um avanço significativo. Analisando qualquer par de envio e recebimento de mensagem no log, a regra da ordem causal é consistentemente satisfeita. Por exemplo, ao rastrear uma mensagem, o valor [Lamport: X] do evento de envio é sempre menor que o valor [Lamport: Y] do evento de recebimento correspondente.

No entanto, o log também revela a principal limitação deste algoritmo. Considere dois eventos, A e B, em processos distintos que não possuem uma ligação causal direta. Mesmo que $L(A) < L(B)$, não podemos afirmar que A "aconteceu antes" de B no sentido causal; eles podem ser **concorrentes**. O algoritmo de Lamport impõe uma ordem total sobre os eventos, mas não nos permite distinguir entre uma ordem que surge da causalidade e uma que surge por acaso.

Parte 3: Relógios Vetoriais

- **Objetivo Específico:** Implementar relógios vetoriais para obter uma ordenação causal parcial que permita não apenas verificar a causalidade, mas também identificar eventos concorrentes.
- **Lógica de Implementação:** Cada processo manteve um vetor de inteiros de tamanho N (número de processos). Ao ocorrer um evento local, sua própria posição no vetor era incrementada. Ao enviar uma mensagem, uma cópia do vetor era enviada junto. Ao receber, o processo primeiro mesclava seu vetor com o vetor recebido (através da operação de max em cada elemento) e, em seguida, incrementava sua própria posição.

Resultados (Log de Execução):

```

--- INICIANDO SIMULAÇÃO: PARTE 3 - RELÓGIOS FÍSICOS ---
[P1] [Vetor: [0, 1, 0]] [Físico: 1753392065018] -> enviou mensagem para P0
[P0] [Vetor: [1, 0, 0]] [Físico: 1753392065525] -> enviou mensagem para P1
[P0] [Vetor: [2, 1, 0]] [Físico: 1753392065526] -> recebeu mensagem com vetor [0, 1, 0]
[P2] [Vetor: [0, 0, 1]] [Físico: 1753392065554] -> realizou um evento interno.
[P1] [Vetor: [0, 2, 0]] [Físico: 1753392065635] -> enviou mensagem para P2
[P1] [Vetor: [1, 3, 0]] [Físico: 1753392065635] -> recebeu mensagem com vetor [1, 0, 0]
[P2] [Vetor: [0, 0, 2]] [Físico: 1753392066612] -> realizou um evento interno.
[P2] [Vetor: [0, 2, 3]] [Físico: 1753392066612] -> recebeu mensagem com vetor [0, 2, 0]
[P0] [Vetor: [3, 1, 0]] [Físico: 1753392066911] -> enviou mensagem para P2
[P1] [Vetor: [1, 4, 0]] [Físico: 1753392067187] -> enviou mensagem para P2
[P2] [Vetor: [0, 2, 4]] [Físico: 1753392068068] -> enviou mensagem para P1
[P2] [Vetor: [3, 2, 5]] [Físico: 1753392068068] -> recebeu mensagem com vetor [3, 1, 0]
[P1] [Vetor: [1, 5, 0]] [Físico: 1753392068280] -> enviou mensagem para P2
[P1] [Vetor: [1, 6, 4]] [Físico: 1753392068280] -> recebeu mensagem com vetor [0, 2, 4]
[P1] [Vetor: [1, 7, 4]] [Físico: 1753392068373] -> enviou mensagem para P2
[P0] [Vetor: [4, 1, 0]] [Físico: 1753392068537] -> realizou um evento interno.
[P2] [Vetor: [3, 2, 6]] [Físico: 1753392068596] -> realizou um evento interno.
[P2] [Vetor: [3, 4, 7]] [Físico: 1753392068600] -> recebeu mensagem com vetor [1, 4, 0]
[P1] [Vetor: [1, 8, 4]] [Físico: 1753392068640] -> realizou um evento interno.
[P0] [Vetor: [5, 1, 0]] [Físico: 1753392068716] -> enviou mensagem para P1
[P1] [Vetor: [1, 9, 4]] [Físico: 1753392068894] -> enviou mensagem para P0
[P1] [Vetor: [5, 10, 4]] [Físico: 1753392068895] -> recebeu mensagem com vetor [5, 1, 0]
[P0] [Vetor: [6, 1, 0]] [Físico: 1753392069776] -> realizou um evento interno.
[P0] [Vetor: [7, 9, 4]] [Físico: 1753392069776] -> recebeu mensagem com vetor [1, 9, 4]
[P2] [Vetor: [3, 4, 8]] [Físico: 1753392069980] -> realizou um evento interno.
[P2] [Vetor: [3, 5, 9]] [Físico: 1753392069981] -> recebeu mensagem com vetor [1, 5, 0]
[P0] [Vetor: [8, 9, 4]] [Físico: 1753392070264] -> enviou mensagem para P2
[P2] [Vetor: [3, 5, 10]] [Físico: 1753392070391] -> enviou mensagem para P1
[P2] [Vetor: [3, 7, 11]] [Físico: 1753392070392] -> recebeu mensagem com vetor [1, 7, 4]
[P2] [Vetor: [3, 7, 12]] [Físico: 1753392071435] -> realizou um evento interno.
[P2] [Vetor: [8, 9, 13]] [Físico: 1753392071435] -> recebeu mensagem com vetor [8, 9, 4]
[P0] [Vetor: [9, 9, 4]] [Físico: 1753392072240] -> enviou mensagem para P1

```

Análise dos Resultados:

Os resultados da Parte 3 são os mais completos. A análise do log vetorial permite uma compreensão exata da relação entre quaisquer dois eventos:

1. **Verificação de Causalidade:** Rastreando um par de envio (evento A) e recebimento (evento B), observa-se que o vetor do envio, $V(A)$, é estritamente menor que o vetor do recebimento, $V(B)$. Isso significa que cada componente de $V(A)$ é menor ou igual ao componente correspondente em $V(B)$, e pelo menos um componente é estritamente menor. Isso confirma a relação $A \rightarrow B$.
2. **Identificação de Concorrência:** Ao selecionar dois eventos C e D em processos distintos que não estão em uma mesma cadeia causal, a comparação de seus vetores revela a concorrência. Nem $V(C) \leq V(D)$ nem $V(D) \leq V(C)$ são verdadeiros, pois cada vetor possui pelo menos um componente maior que o do outro. Isso prova matematicamente que os eventos C e D são concorrentes (escrito como $C \parallel D$).

Conclusão

A atividade prática permitiu observar com sucesso as nuances e os trade-offs dos principais algoritmos de relógios em sistemas distribuídos. Foi demonstrado que relógios físicos são insuficientes para a tarefa de ordenação. Os relógios de Lamport surgem como uma solução elegante e de baixo custo para garantir a ordem causal, enquanto os relógios vetoriais, apesar de sua maior complexidade, oferecem a solução mais robusta ao fornecer um panorama completo das relações de causalidade e concorrência entre todos os eventos do sistema. O cumprimento dos objetivos desta prática solidifica a compreensão teórica e prática sobre a importância fundamental do tempo e da ordem em ambientes distribuídos.