



Relógios Lógicos



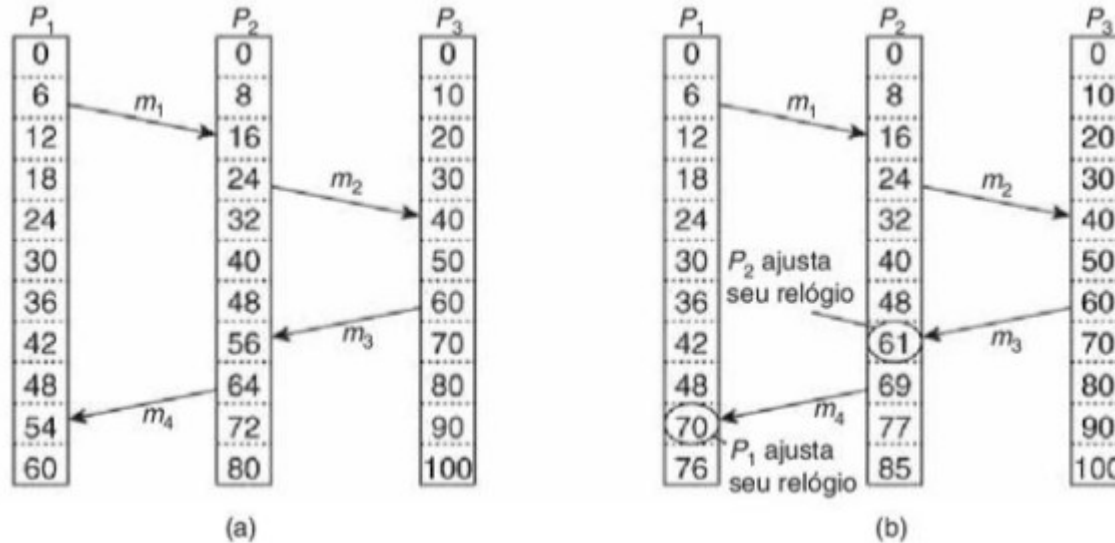
Universidade Estadual de Santa Cruz



Departamento de Engenharias e Computação
Área de Computação
Prof. Dr. Paulo André S. Giacomini



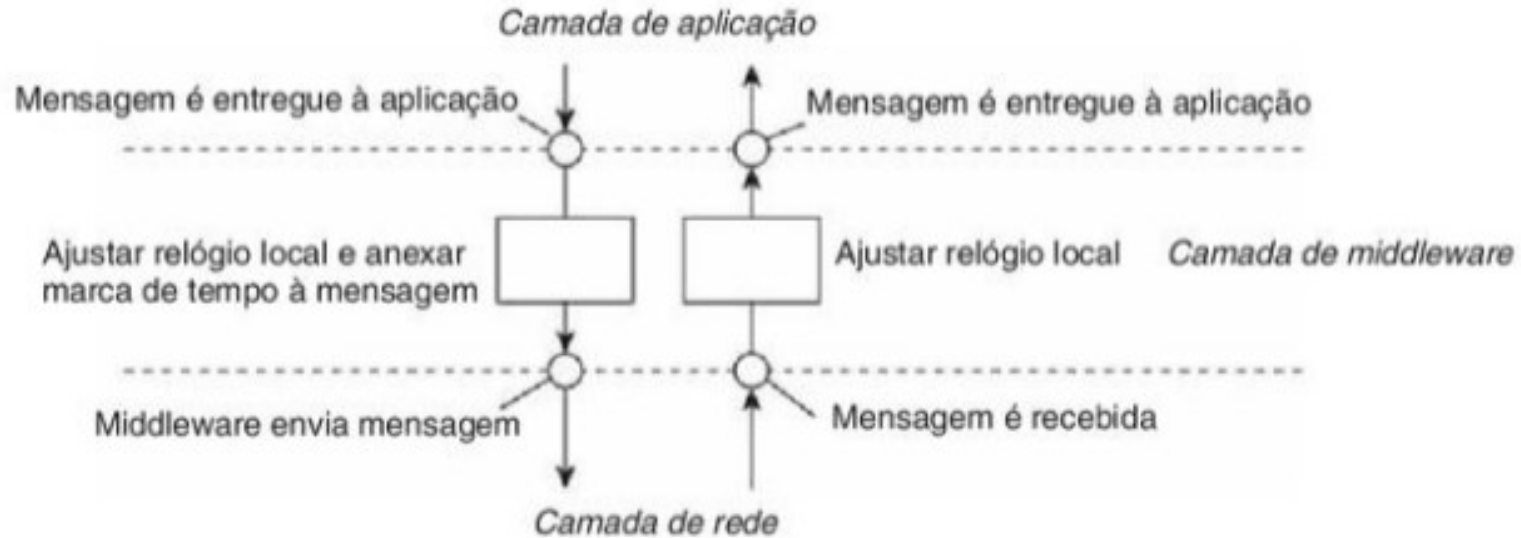
RELÓGIOS LÓGICOS



Se a é uma mensagem enviada pela rede, e b é uma mensagem recebida, então $a \rightarrow b$ (a precede b). Isso pode ser usado para sincronizar relógios, mesmo quando a hora não é a real (relógios lógicos de Lamport). **Fonte:** Tanenbaum e Steen (2007)



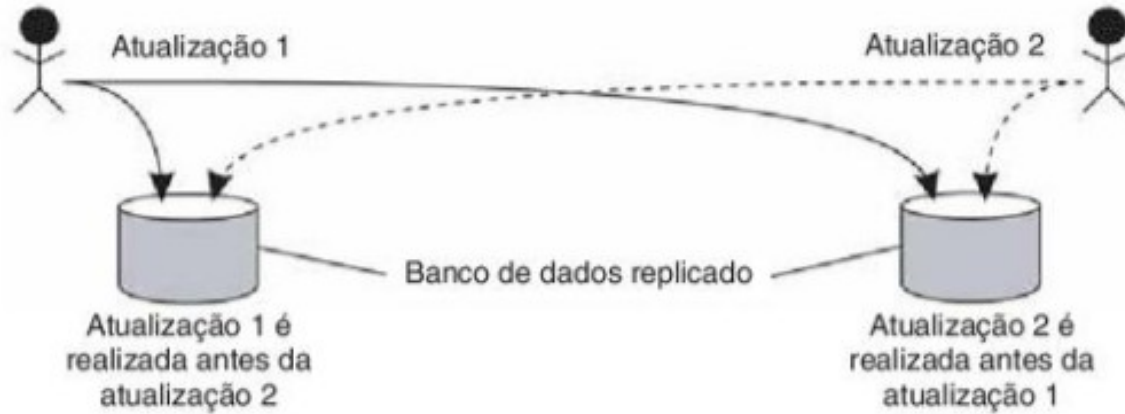
CAMADAS DE SOFTWARE



Os relógios são ajustados no envio e no recebimento das mensagens. O tempo no recebimento não pode ser menor do que no envio. **Fonte:** Tanenbaum e Steen (2007)



REPLICAÇÃO DE DADOS

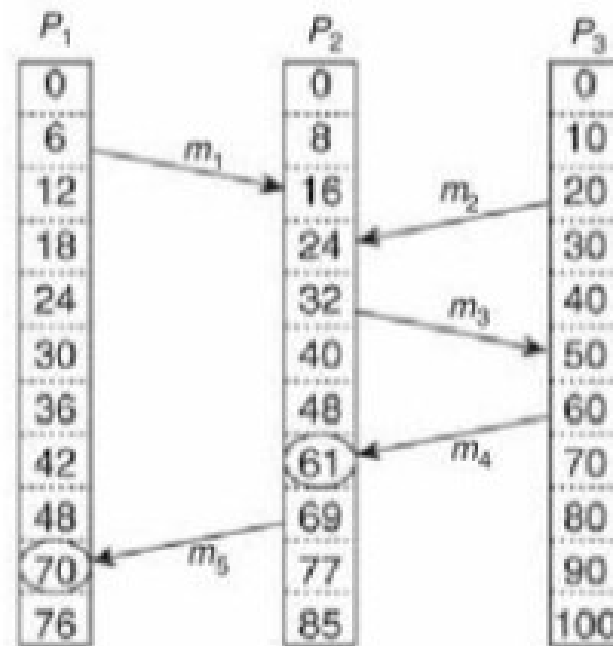


Replicações de dados podem causar problemas de consistência, pois os banco de dados precisam ser idênticos (conteúdo e ordem). **Fonte:** Tanenbaum e Steen (2007)



MENSAGENS CONCORRENTES

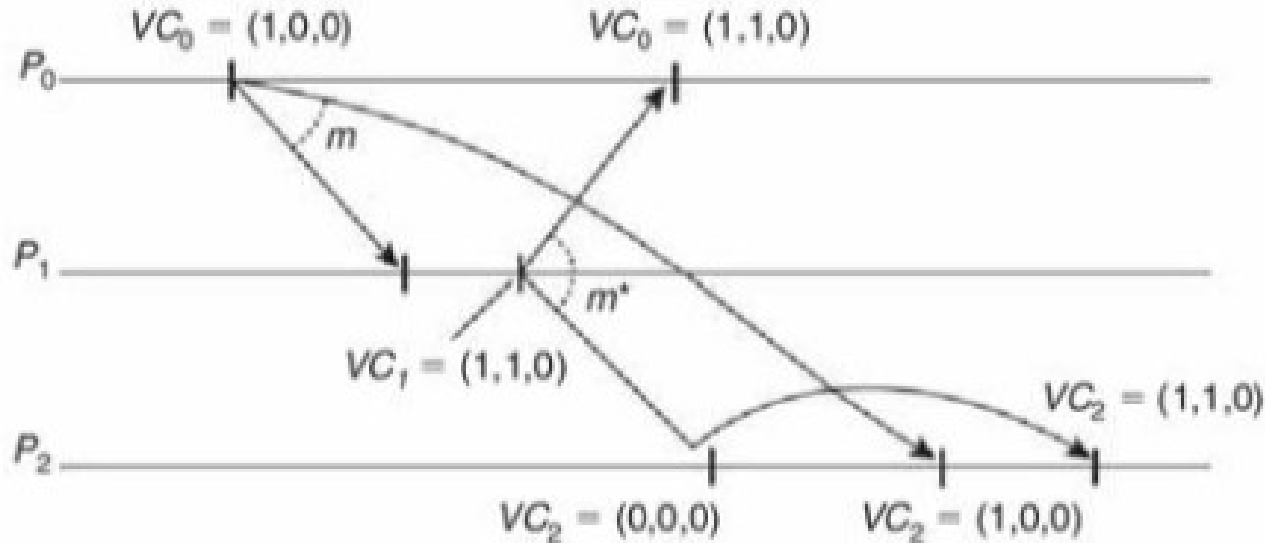
- O processo P_1 , no tempo 15, nada conhece sobre o tempo no processo P_3 ;
- Nesse ponto o relógio lógico de Lamport pode não ser suficiente;
- Relógios vetoriais podem ser usados para resolver o problema;



Fonte: Tanenbaum e Steen (2007)



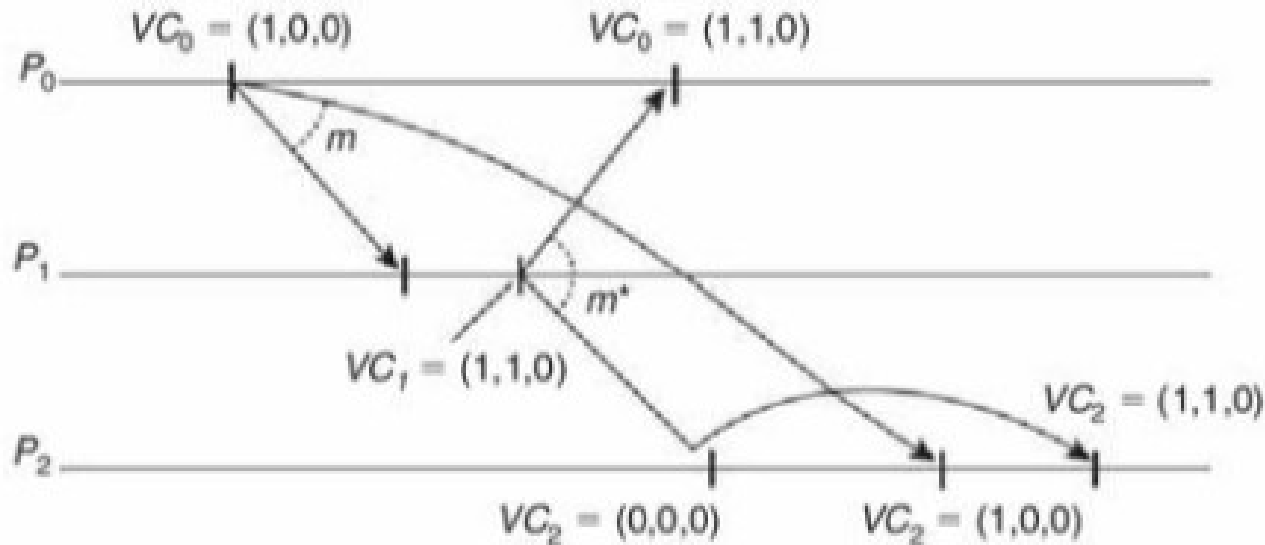
COMUNICAÇÃO CAUSAL



1 – A cada criação de um novo evento por parte do processo P_i , $VC_i[i] = VC_i[i] + 1$; 2 – Este valor é repassado em novas mensagens ($ts[m]$) saindo de P_i ; 3 - No recebimento de uma nova mensagem m , $VC_j[k] = \max\{VC_j[k], ts(m)[k]\}$. **Fonte:** Tanenbaum e Steen (2007)



COMUNICAÇÃO CAUSAL



Uma mensagem m atrasa até que:

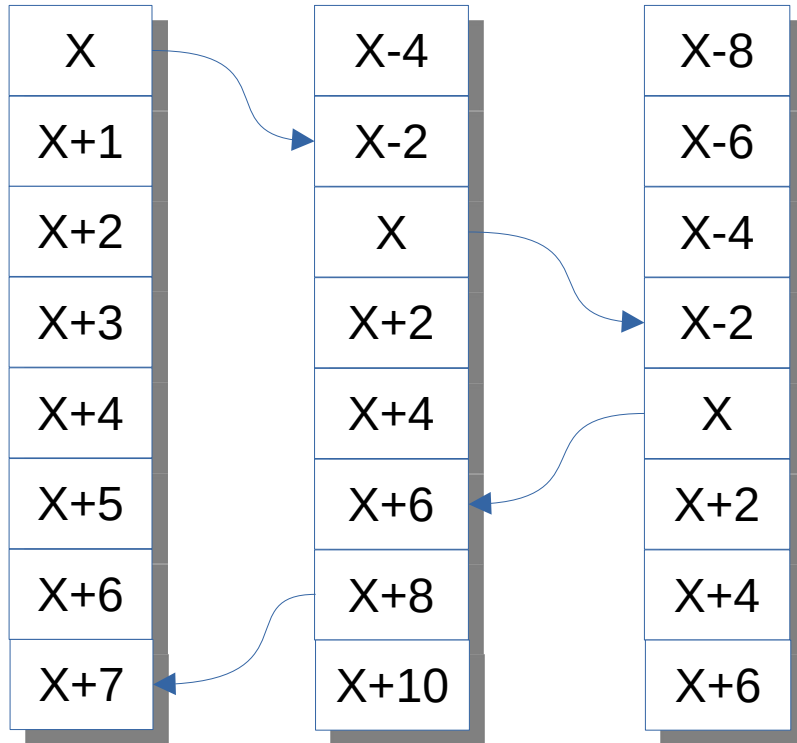
- $m[i] = VC_j[i] + 1$;
- $m[k] \leq VC_j[k] \forall i \neq k$

i representa o processo que enviou m , e j o processo receptor.

O processo dois recebe informações de causalidade de todos os processos (zero e um). Assim, a mensagem m^* é atrasada. Cada vetor armazena a quantidade de eventos locais de todos os processos. **Fonte:** Tanenbaum e Steen (2007)



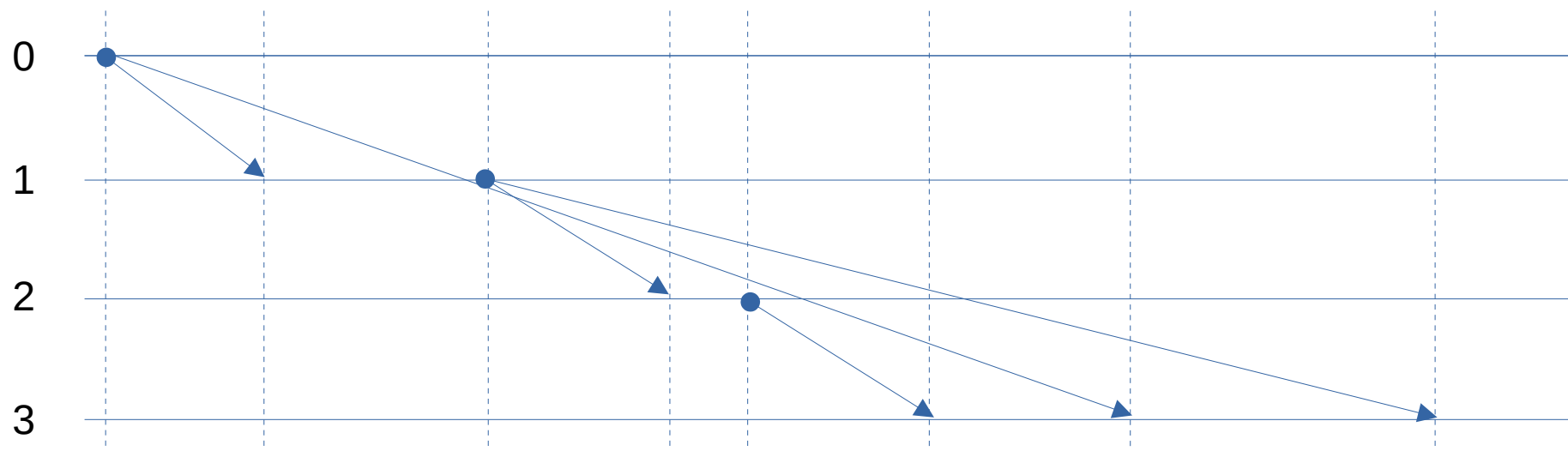
EXERCÍCIO



- Seja X o resto mais 2 da divisão do seu número de matrícula por 20;
- Mostre como os horários devem ser ajustados se $t(a) < t(b)$ quando $a \rightarrow b$;



EXERCÍCIO



Diga, e se necessário, ilustre: a) (40%) Quais são os vetores de causalidade de cada um dos processos de 1 a 4; b) (30%) Quais mensagens devem ser atrasadas para preservação da causalidade; c) (30%) Até quando cada mensagem atrasada deve atrasar-se;



TRABALHO

Programar o algoritmo de relógio de Lamport para sincronização de relógios lógicos com comunicação causal (via relógios vetoriais) utilizando MPI para o caso do exercício anterior. Qualquer linguagem de programação pode ser utilizada.



BIBLIOGRAFIA

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.; **Distributed Systems – Concepts and Design**. Ed. ADDISON WESLEY PUB. 2001. 772 p.

KENDALL, Wes; MPI Hello World. Disponível em: <https://mpitutorial.com/tutorials/mpi-hello-world/>. Acesso em: 22 de Agosto de 2022.

KENDALL, Wes; MPI Send and Receive. Disponível em: <https://mpitutorial.com/tutorials/mpi-send-and-receive/>. Acesso em 22 de Agosto de 2022.

ORACLE. The Java Tutorials - Trail: RMI. Disponível em: <https://docs.oracle.com/javase/tutorial/rmi/>. Acesso em: 22 de Agosto de 2022.

PETERSON, L. L.; DAVIE, B. S.; **Redes de computadores – uma abordagem de sistemas**. Editora Elsevier. 5º Edição. 2013. P. 545. ISBN: 978-85-352-4897-5.

STEEN, M. V.; TANENBAUM, A. S.; **Distributed Systems – Principles and Paradigms** . Ed. PRENTICE HALL. 2002. 803p.

TANENBAUM, Andrew S.; STEEN, Maarten van. **Sistemas Distribuídos: princípios e paradigmas - 2ª edição**. Editora Pearson. 2007. P. 416. ISBN 9788576051428.

TANENBAUM, Andrew S. **Sistemas Operacionais Modernos - 3ª edição**. Editora Pearson. 2009. P. 674. ISBN 9788576052371.

THE OPEN MPI TEAM. Open MPI Documentation. **v4.1**. Disponível em: <https://www.open-mpi.org/doc/>. Acesso em: 22 de Agosto de 2022.