

## Tempo lógico e relógios lógicos

Tempo lógico: que não tem relação direta com tempo real.

- A impossibilidade de sincronizar perfeitamente os relógios em um sistema distribuído.

Não se pode ter um relógio físico e máquinas perfeitamente sincronizadas com ele.

- O uso da ordenação causal para ordenar eventos que ocorrem em diferentes processos.

Para haver sincronia sem uso de relógios físicos usa-se a **ordenação causal**, para organizar um conjunto de eventos.

- Premissas básicas:

Da ordenação causal.

- Se dois eventos ocorreram no mesmo processo, então eles ocorreram na ordem em que o processo observou.
- Quando uma mensagem é enviada

entre dois processos, o evento de envio ocorreu antes do evento de recepção.

- Lamport chamou essa ordenação parcial de relação *acontece antes* ( $\rightarrow$ )

---

### Definição da relação *acontece antes*

- **AA1:** Se 3 processo  $p_i$ :  $e \rightarrow e'$ , então  $e$  o  $e'$ .

- **AA2:** Para qualquer mensagem  $m$ ,  $\text{send}(m) \rightarrow \text{receive}(m)$

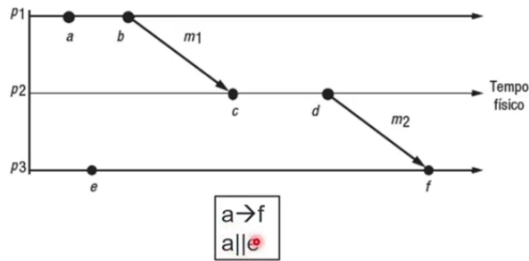
O envio de uma mensagem de acontece antes do seu recebimento.

- onde  $\text{send}(m)$  é o evento de envio da mensagem e  $\text{receive}(m)$  é o evento de sua recepção.

- **AA3:** Se  $e, e'$  e  $e''$  são eventos tais que  $e \rightarrow e'$  e  $e' \rightarrow e''$ , então  $e \rightarrow e''$ .

**Relação de transitividade.**

### Exemplo:



**carimbo de tempo:** valores numéricos associados a um acontecimento.

Onde  $L_i(e)$  é o carimbo de tempo de determinado evento.

- Carimbo de tempo do evento  $e$  no processo em que ocorreu:  $L(e)$

$L(e)$  é o carimbo de tempo global.

### Algoritmo

RL1:  $L_i$  é incrementado antes da ocorrência de um evento no processo  $p_i$ :  
 $L_i := L_i + 1$

RL2: (a) Quando um processo  $p_i$  envia uma mensagem  $m$ ,  $m$  leva "de carona" (piggybacking) o valor  $t = L_i$ .  
 (b) Na recepção ( $m, t$ ), um processo  $p_j$  calcula  $L_j := \max(L_j, t)$  e, então, aplica RL1 antes de registrar o carimbo de tempo do evento  $receive(m)$ .

- Se  $e > e'$ , então  $L(e) < L(e')$
- Limitação do algoritmo:
  - Se  $L(e) < L(e')$ , não se pode inferir que  $e > e'$

Pois não há relação entre os processos.

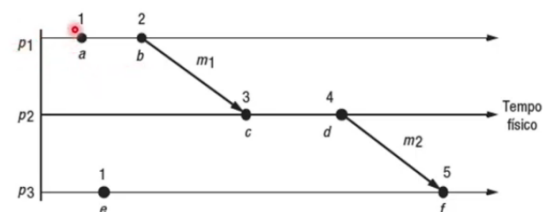
### Relógios lógicos

- O relógio lógico é um contador simples que aumenta monotonicamente, sem qualquer relação com o relógio físico.

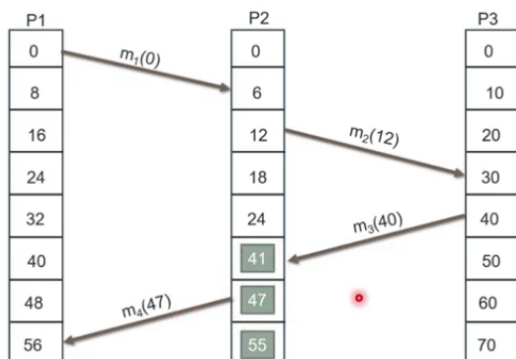
A cada processo é associado um contador que é incrementado positivamente.

- Cada processo  $p_i$  tem um relógio lógico  $L_i$
- Usado para aplicar carimbos de tempo aos eventos.
- Carimbo de tempo do evento  $e$  em  $p_i$ :  $L_i(e)$

### Exemplo



## Exemplo



## Relógios lógicos totalmente ordenados

- Limitação de Lamport:

Eventos distintos, de processos diferentes, podem ter carimbos de tempo idênticos.

- Solução: Utilizar os identificadores dos processos na composição do carimbo de tempo.

- Ex:
  - Evento  $e$  em  $p_i$  com carimbo  $T_i$  terá carimbo global  $(T_i, i)$
  - Evento  $e'$  em  $p_j$  com carimbo  $T_j$  terá carimbo global  $(T_j, j)$

- Define-se  $(T_i, i) < (T_j, j)$ , se e somente se  $T_i < T_j$  ou  $T_i = T_j$  e  $i < j$

## Relógios vetoriais

A ideia é que cada evento tem um vetor com uma posição para cada um dos processos do sistema. Dentro desse vetor cada processo mantém seu tempo atualizado.

- Supera a limitação dos relógios de Lamport:

- A partir de  $L(e) < L(e')$ , não se pode inferir que  $e > e'$
- Um relógio vetorial para um sistema com  $N$  processos é um vetor com  $N$  inteiros.
- Cada processo mantém seu relógio  $V_i$

## Relógios vetoriais – Regras de atualização

RV1: Inicialmente,  $V_i[j] = 0$ , para  $i, j = 1, 2, \dots, N$ .

RV2: Imediatamente antes de  $p_i$  gerar o carimbo de tempo de um evento, ele configura  $V_i[i] := V_i[i] + 1$ .

RV3:  $p_i$  inclui o valor  $t = V_i$  em cada mensagem que envia.

RV4: Quando  $p_i$  recebe um carimbo de tempo  $t$  em uma mensagem, ele configura  $V_i[j] := \max(V_i[j], t[j])$ , para  $j = 1, 2, \dots, N$ . Considerar o máximo de duas componentes de carimbos de tempo vetoriais dessa maneira é conhecido como operação de *merge* (integração).

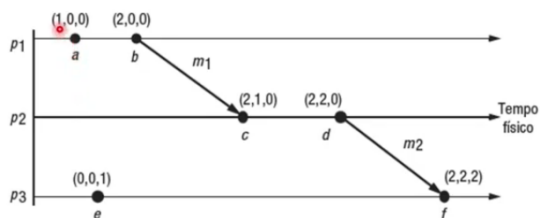
O carimbo de tempo agora é um vetor.

**Merge:** operação de verificar carimbos de tempo dentro de um vetor.

Sua desvantagem é o overhead adicional pela quantidade de dados maior enviada (pelo uso de um vetor).

---

### Exemplo



---

### Regras de comparação

$V = V' \text{ sse } V[j] = V'[j] \text{ para } j = 1, 2, \dots, N$

$V \leq V' \text{ sse } V[j] \leq V'[j] \text{ para } j = 1, 2, \dots, N$

$V < V' \text{ sse } V \leq V' \wedge V \neq V'$

---