

## Sistemas Distribuídos

### 8 Relógios Lógicos

- **Introdução**
- **Tempo Lógico**
- **Relógios Lógicos de Lamport**
- **Relógios Lógicos Vetoriais**

Profª Ana Cristina B. Kochem Vendramin  
DAINF / UTFPR

## Introdução

### • Princípios:

- Vistos de um processo, os eventos são ordenados pelo tempo do relógio local.
- Somente processos que interagem precisam sincronizar seus relógios.
- Não conseguimos sincronizar os relógios físicos perfeitamente. Logo, não podemos utilizá-los para determinar a ordem de ocorrência de dois eventos quaisquer num SD (Lamport, 78).

Profª Ana Cristina B. Kochem Vendramin.  
DAINF/UTFPR

2

## Relação de causalidade ou *happened-before*

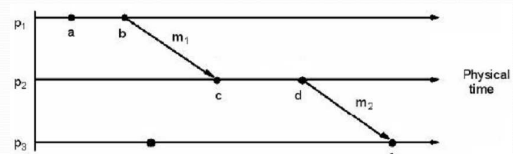
### • Relação acontece-antes ( $\rightarrow$ ):

1. Se dois eventos  $x$  e  $y$  ocorrem em um mesmo processo  $p$  então esses dois eventos ocorrem na mesma ordem observada por  $p$ . Se  $p: x \rightarrow y$  então  $x \rightarrow y$  é verdadeiro.
2. Seja  $x$  o evento de uma mensagem  $m$  enviada por um processo, e  $y$  o evento do recebimento dessa mensagem por outro processo.  
 $\forall m, x \rightarrow y$
3. **Relação transitiva:**  
Sejam  $x, y$  e  $z$  eventos tal que  $x \rightarrow y$  e  $y \rightarrow z$ .  
Então,  $x \rightarrow z$  é verdadeiro.

Profª Ana Cristina B. Kochem Vendramin.  
DAINF/UTFPR

3

## Relação de causalidade ou *happened-before*



- Localmente no processo:  $a \rightarrow b, c \rightarrow d, e \rightarrow f$
- *send*  $\rightarrow$  *receive*:  $b \rightarrow c, d \rightarrow f$
- Relação transitiva:  $a \rightarrow f$
- $a \parallel e$  = eventos "a" e "e" são concorrentes

Profª Ana Cristina B. Kochem Vendramin.  
DAINF/UTFPR

4

## Relógio Lógico de Lamport

- Mecanismo numérico para representar a relação lógica entre eventos de processos;
- Não tem relação com relógios físicos;
- Cada processo ( $p_i$ ) tem seu relógio lógico ( $L_i$ ) usado para fazer *timestamp* de eventos;

Profª Ana Cristina B. Kochem Vendramin.  
DAINF/UTFPR

5

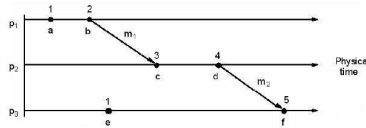
## Relógio Lógico de Lamport

- Cada relógio é incrementado antes de colocá-lo como *timestamp*  $t$  em um evento.
- Quando o processo  $p_i$  envia uma mensagem  $m$ , essa mensagem leva junto o valor do seu relógio  $L_i$ ;
- Quando o processo  $p_i$  recebe uma mensagem  $m$  com *timestamp*  $t$ ,  $p_i$  faz  $L_i := \max(L_i, t) + 1$

Profª Ana Cristina B. Kochem Vendramin.  
DAINF/UTFPR

6

## Relógio Lógico de Lamport



- L1, L2, L3 são inicializados com zero
- No envio de m1, p1 manda junto L1=2
- P2 recebe m1 e faz L2=max(0, 2) + 1

### Importante:

- $e \rightarrow e'$  implica  $L(e) < L(e')$
- O contrário não é verdadeiro:  $L(e) < L(e')$  não implica  $e \rightarrow e'$
- Exemplo:  $L(e) < L(b)$  porém  $e \parallel b$

## Relógio de Lamport x Relógio Vetorial

- Relógios Lógicos de *Lamport*
  - Ordem parcial dos eventos - relação causal HB
    - $x \rightarrow y \Leftrightarrow L(x) \rightarrow L(y)$ , porém
    - Se  $L(x) \rightarrow L(y)$  não se pode deduzir que  $x \rightarrow y$ .
- Relógios Lógicos Vetoriais
  - Surgem da necessidade de satisfazer completamente a relação acima
  - $x \rightarrow y \Leftrightarrow L(x) \rightarrow L(y)$

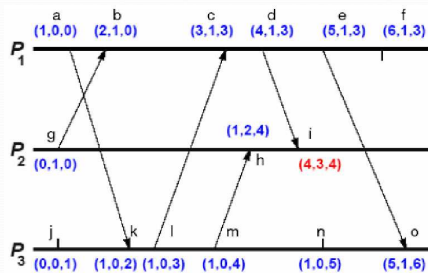
## Relógio Lógico Vetorial

- Considere um SD com  $N$  processos;
- Cada processo possui seu relógio vetorial  $V_i$ ;
- Cada relógio vetorial terá  $N$  posições  $V_i[N]$ ;
- A posição  $V_i[j]$  contém o número de eventos que ocorreram no processo  $P_j$ ;
- Uma posição  $V_i[j]$  com  $i \neq j$  contém o número de eventos que ocorreram em  $P_j$  e afetaram  $P_i$ .

## Relógio Lógico Vetorial

- Regras de atualização:
  - **Inicialmente:**  $V_i[j] = 0$  para  $j = 1$  até  $N$ ;
  - **Evento Local:**
    - Antes de colocar o *timestamp* em um evento, o processo faz:
      - $V_i[i] := V_i[i] + 1$ ;
  - **Envio de mensagem:**
    - $P_i$  anexa o valor de seu relógio  $V_i$  na mensagem;
  - **Recebimento de mensagem:**
    - Quando  $P_i$  recebe uma mensagem com *timestamp*  $t$ , ele faz:
      - $V_i[j] := \max(V_i[j], t[j])$

## Relógio Lógico Vetorial - Exemplo



## Relógio Lógico Vetorial

- Comparação de relógios vetoriais
  - **Precedência:**  $g \rightarrow c \Leftrightarrow V(g) < V(c)$ 
    - $[0, 1, 0] \leq [3, 1, 3]$ ?
      - $[0 \leq 3, 1 \leq 1, 0 \leq 3] = [V, V, V] = \text{Verdadeiro}$
  - **Não precedência:**  $\sim(V(c) \leq V(g))$ 
    - $[3, 1, 3] < [0, 1, 0]$ ?  $[F, V, F] = \sim(F) = V$
  - **Concorrência:**  $c \parallel h \Leftrightarrow \sim(V(c) \leq V(h)) \wedge \sim(V(c) \geq V(h))$ 
    - $\sim([3, 1, 3] \leq [1, 2, 4]) \wedge \sim([3, 1, 3] \geq [1, 2, 4]) = \sim(F) \wedge \sim(F) = V$

## Referências Bibliográficas

- Coulouris, George; Dollimore, Jean; Kindberg, Tim.  
Distributed Systems: concepts and design. Third Edition.  
Addison-Wesley 2001.
- Coulouris, George; Dollimore, Jean; Kindberg, Tim; tradução  
João Tortello. Sistemas Distribuídos: conceitos e projeto. 4. ed.  
Bookman 2007.