

Aula 11 - Ordenação de eventos e relógios lógicos

Monday, April 11, 2016

14:33

Neste contexto: Evento é a execução de uma ação ou instrução

Problema: considerando os eventos dos múltiplos nodos independentes de um sistema distribuído, como dizer se um evento aconteceu antes de outro?

Se todos os nodos têm GPS local (relógio global), problema resolvido!

Caso contrário, como fazer?

Um evento a aconteceu antes de um evento b de acordo com alguma referência, por exemplo, o tempo físico.

Usando um único "relógio", conseguimos determinar qual evento aconteceu antes de qual outro evento.

Ordenação parcial de eventos

Considerando um único processo, com acesso a 1 relógio local, é fácil identificar qual evento aconteceu antes de qual outro.

Vamos definir a relação "aconteceu-antes-de" (*happened before*) como $a \rightarrow b$

Além dos eventos de um único processo, se considerarmos 2 processos quais quer e eles se comunicam...

Para uma mensagem ser recebida, ela **tem** que ter sido transmitida **ANTES**:

$send\ msg \rightarrow receive\ msg$

Para definir a relação \rightarrow sobre um conjunto de eventos de um sistema distribuído:

Esta relação satisfaz as seguintes condições:

1. Se a & b são eventos de um mesmo processo e, usando o relógio local, determinamos *que* a aconteceu antes de b , então $a \rightarrow b$.
2. Se a é o envio de uma mensagem e b é o recebimento daquela mensagem, então $a \rightarrow b$.
3. Se $a \rightarrow b$ e $b \rightarrow c$, então $a \rightarrow c$.

Dois eventos distintos a & b tais que $a \not\rightarrow b$ e $b \not\rightarrow a$ são ditos concorrentes.

No exemplo, a e x são concorrentes.

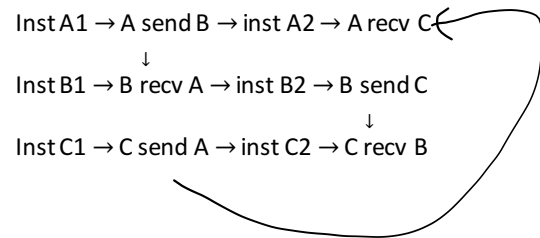
A relação \rightarrow define uma ordem parcial de execução de eventos de um sistema distribuído.

Exercício: usando a relação \rightarrow , ordene parcialmente os seguintes eventos dos processos A, B, C.

A: inst A1, send B, inst A2, recv C

B: inst B1, recv A, inst B2, send C

C: inst C1, send A, inst C2, recv B



Com relógios lógicos conseguimos a ordenação total!