



RAMA UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF ENGINEERING AND TECHNOLOGY

Distributed Systems(BCS-701) LECTURE-08

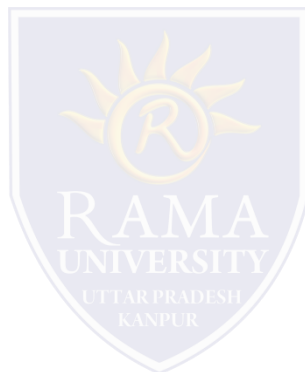
Dr. Hariom Sharan

Professor & Dean

Computer Science & Engineering

OUTLINE

- ❖ Logical clocks
- ❖ Happened-before
- ❖ Logical clocks & concurrency
- ❖ Event counting example
- ❖ Lamport's algorithm
- ❖ Summary
- ❖ References



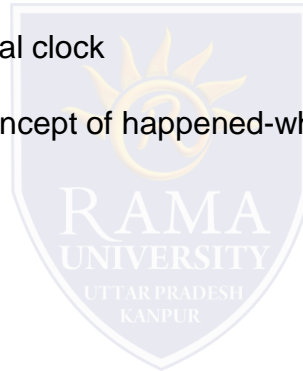
Logical clocks

Assign sequence numbers to messages

- All cooperating processes can agree on order of events
- vs. physical clocks: time of day

Assume no central time source

- Each system maintains its own local clock
- No total ordering of events • No concept of happened-when



Happened-before

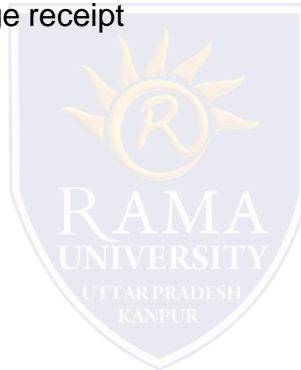
➤ Lamport's "happened-before" notation

➤ $a \rightarrow b$ event a happened before event b

➤ e.g.: a: message being sent, b: message receipt

➤ Transitive:

➤ if $a \rightarrow b$ and $b \rightarrow c$ then $a \rightarrow c$



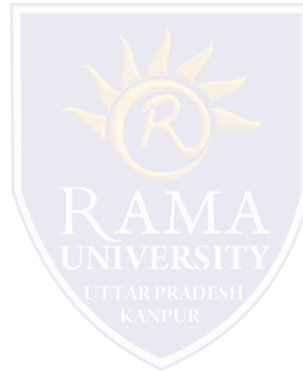
Logical clocks & concurrency

Assign “clock” value to each event

- if $a \rightarrow b$ then $\text{clock}(a) < \text{clock}(b)$
- since time cannot run backwards

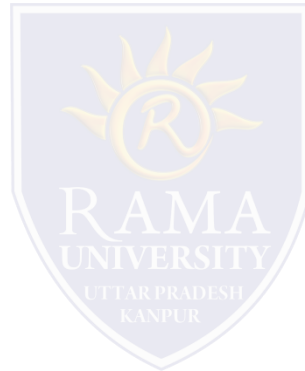
If a and b occur on different processes that do not exchange messages, then neither $a \rightarrow b$ nor $b \rightarrow a$ are true

- These events are concurrent



Event counting example

- Three systems: P0 , P1 , P2
- Events a, b, c, ...
- Local event counter on each system
- Systems occasionally communicate



Lamport's algorithm

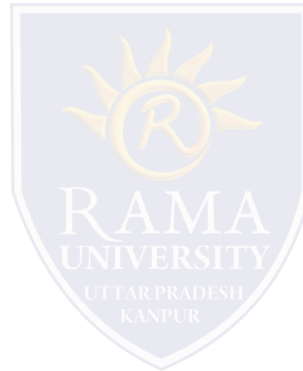
- Each message carries a timestamp of the sender's clock
- When a message arrives:
 - if receiver's clock < message timestamp set system clock to (message timestamp + 1)
 - else do nothing
- Clock must be advanced between any two events in the same process



Summary

- Algorithm needs monotonically increasing software counter
- Incremented at least when events that need to be timestamped occur
- Each event has a Lamport timestamp attached to it
- For any two events, where $a \rightarrow b$:

$$L(a) < L(b)$$



References

- <https://www.javatpoint.com/digital-image-processing-tutorial>
- <https://www.tutorialpoint.com/>
- <https://www.geeksforgeeks.org/limitation-of-distributed-system/>
- Distributed Systems, Concepts and Design, George Coulouris, J Dollimore and Tim Kindberg, Pearson Education, 4th Edition, 2009.
- Distributed Systems, Principles and paradigms, Andrew S. Tanenbaum, Maarten Van Steen, Second Edition, PHI.
- Distributed Systems, An Algorithm Approach, Sikumar Ghosh, Chapman & Hall/CRC, Taylor & Francis Group, 2007.

