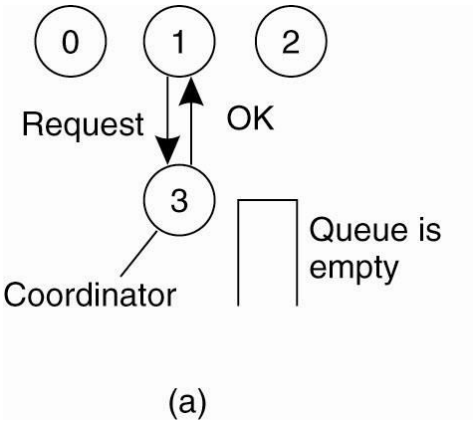
Mutual Exclusion

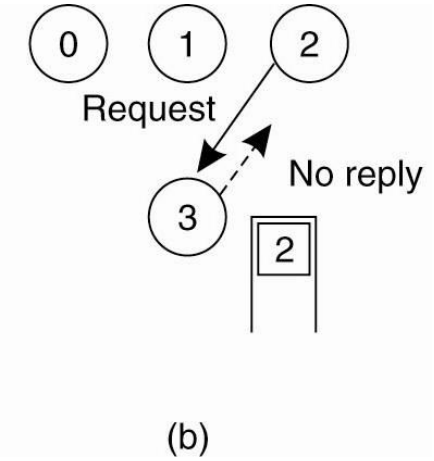
Mutual Exclusion
A Centralized Algorithm

Figure 6-14, Tanenbaum: Process 1 asks the coordinator for permission to access a shared resource. Permission is granted.



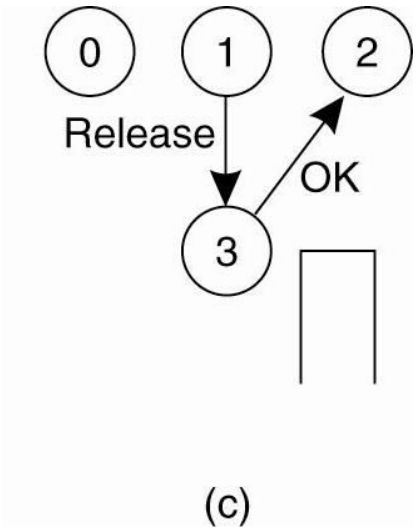
Mutual Exclusion
A Centralized Algorithm

- Figure 6-14, Tanenbaum: Process 2 then asks permission to access the same resource. The coordinator does not reply.



Mutual Exclusion
A Centralized Algorithm

- Figure 6-14, Tanernbaum:
- When process 1 releases the resource, it tells the coordinator, which then replies to 2.



Mutual Exclusion
Richard, Agrawala Algorithm

- Logical timestamps for events are generated using *Lamport's* algorithm.
- A logical timestamp is a pair (c, i), where c c is an integer and i is a process index
  - Timestamps are ordered lexicographically
- A process p_i that wants to enter the critical section, broadcasts a message of type *enter*.
- It goes into C after it receives subsequent *ok* messages from all the other processes.

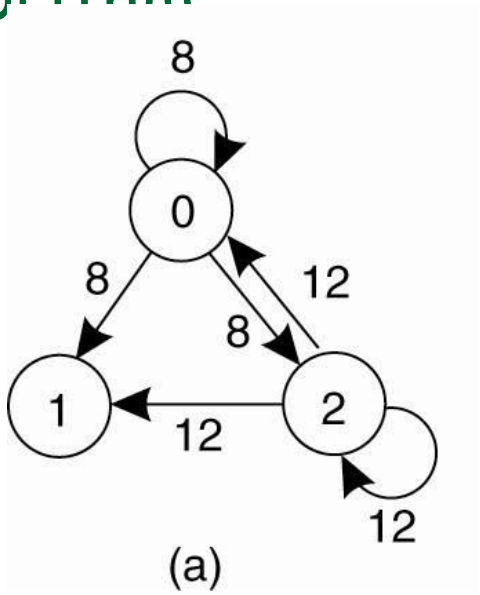
Mutual Exclusion
Richard, Agrawala Algorithm

Three different cases:

1. If the receiver is not accessing the resource and does not want to access it, it sends back an OK message to the sender.
2. If the receiver already has access to the resource, it simply does not reply. Instead, it queues the request.
3. If the receiver wants to access the resource as well but has not yet done so, it compares the timestamp of the incoming message with the one contained in the message that it has sent everyone. The lowest one wins.

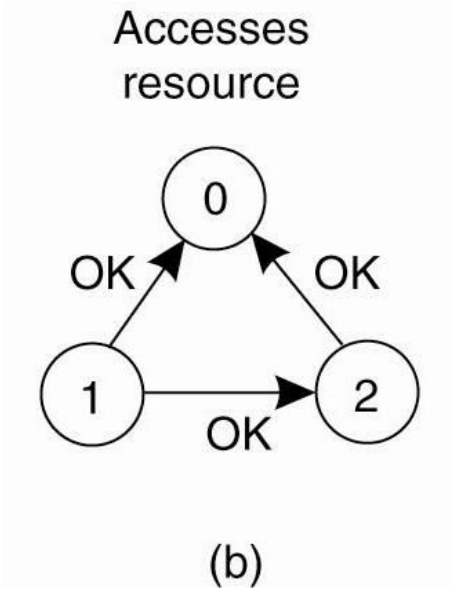
Mutual Exclusion
Richard, Agrawala Algorithm

- Figure 6-15, Tanenbaum: (a) Two processes want to access a shared resource at the same moment.



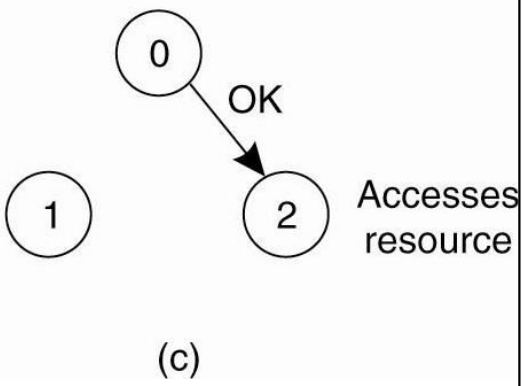
Mutual Exclusion
Richard, Agrawala Algo

- Figure 6-15, Tanenbaum: (b) Process 0 has the lowest timestamp, so it wins.

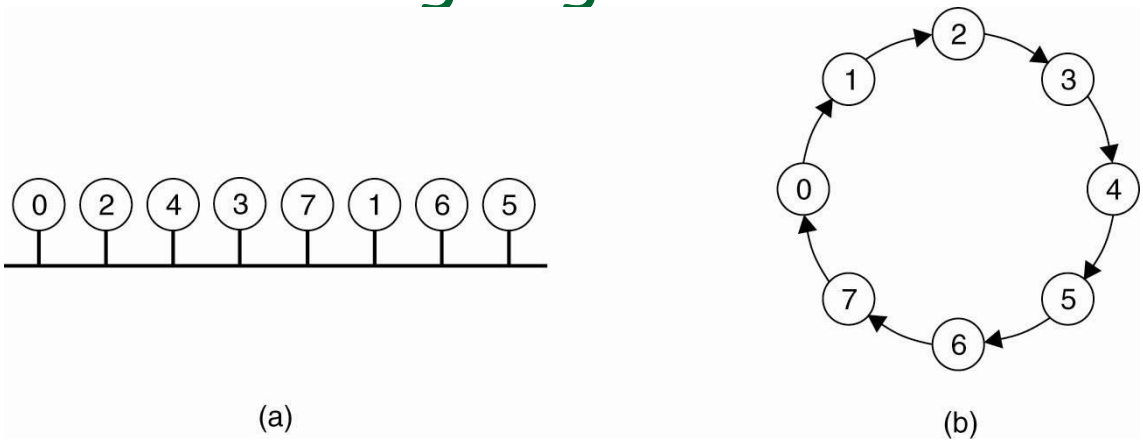


Mutual Exclusion
Richard, Agrawala Al

- Figure 6-15, Tanenbaum:
  (c) When process 0 is done, it sends an OK also so 2 can now go ahead.



A Token Ring Algorithm



- Figure 6-16, Tanenbaum:
  (a) An unordered group of processes on a network.
  (b) A logical ring constructed in software.

Resource Allocation

Resource Allocation: The Problem

- Dfs
- Dfs