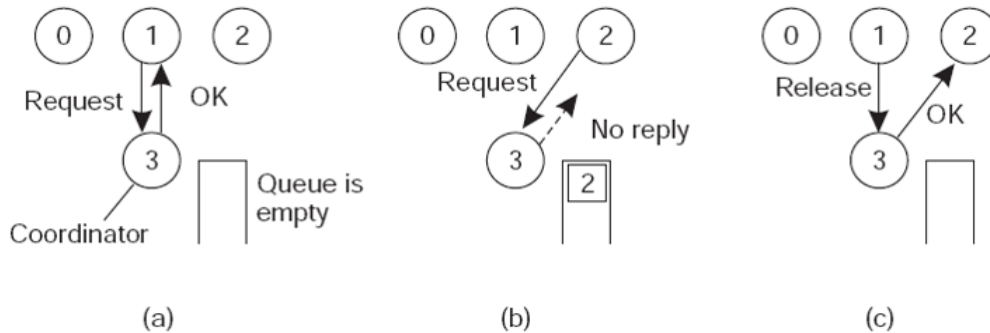


# CENTRALIZED ALGORITHM

- In centralized algorithm one process is elected as the coordinator which may be the machine with the highest network address.



- Whenever a process wants to enter a critical region, it sends a request message to the coordinator stating which critical region it wants to enter and asking for permission. If no other process is currently in that critical region, the coordinator sends back a reply granting permission, as shown in Fig (a). When the reply arrives, the requesting process enters the critical region.
- Suppose another process 2 shown in Fig (b), asks for permission to enter the same critical region. Now the coordinator knows that a different process is already in the critical region, so it cannot grant permission. The coordinator just refrains from replying, thus blocking process 2, which is waiting for a reply or it could send a reply saying 'permission denied.'
- When process 1 exits the critical region, it sends a message to the coordinator releasing its exclusive access as shown in Fig (c).
- The coordinator takes the first item off the queue of deferred requests and sends that process a grant message. If the process was still blocked it unblocks and enters the critical region.
- If an explicit message has already been sent denying permission, the process will have to poll for incoming traffic or block later. When it sees the grant, it can enter the critical region.
- **Advantages:**
  - Algorithm guarantees **mutual exclusion by letting one process at a time** into each critical region.
  - It is also fair as requests are **granted in the order in which they are received.**
  - **No process ever waits** forever so no starvation.
  - Easy to implement so it requires only three messages per use of a critical region (**request, grant, release**).
  - Used for more general resource allocation rather than just managing critical regions.
- **Disadvantages:**
  - The **coordinator is a single point of failure**, the entire system may go down if it crashes.
  - If processes **normally block after making a request, they cannot distinguish a dead coordinator from "permission denied"** since no message comes back.
  - In a **large system a single coordinator can become a performance bottleneck.**