Centralized Distributed Mutual Exclusion
Distributed Systems 2g1509

Ali Ghodsi  aligh@imit.kth.se
**Distributed ME – Centralized [1/4]**

- **Idea:**
  - Let a process, $p_1$, be the coordinator.
  - All processes must ask for $p_1$'s permission before entering the critical section.
  - The coordinator $p_1$ only lets one process at a time enter the critical section. The rest are put in a FIFO queue.


**Coordinator**

- If the critical section is busy, and a new request arrives, the coordinator:
  - queues the request (blocking)
  - queues the request and sends a Reject message to the requesting process
- When the critical section becomes empty, the coordinator:
  - sends an accept message to the next process in the FIFO queue
Distributed ME – Centralized [3/4]

- **Requesting process**
  - A requesting process always sends a request to the coordinator $p$, and:
    - waits for an accept message back (blocking)
    - waits for an accept or reject message (non-blocking)
      - Upon receipt of a reject message the requesting process continues until an accept message is received
Distributed ME – Centralized [4/4]

- **Advantages:**
  - Simple
  - Cheap (2/3 messages per request)
  - No starvation
  - Fair (requests are served in FIFO order)

- **Disadvantages:**
  - Single-point of failure
  - Coordinator becomes a bottleneck