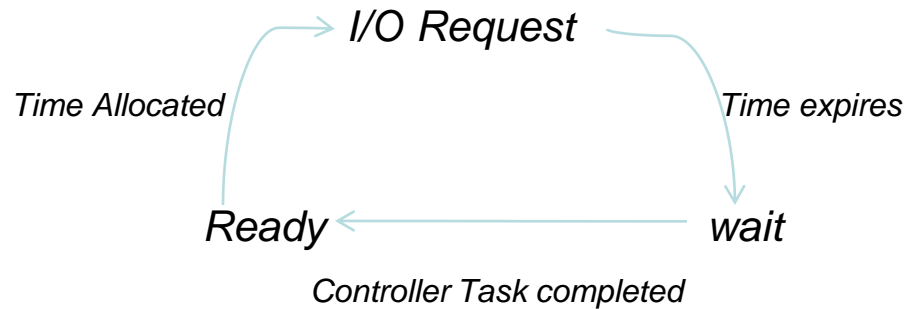


# Process Administration

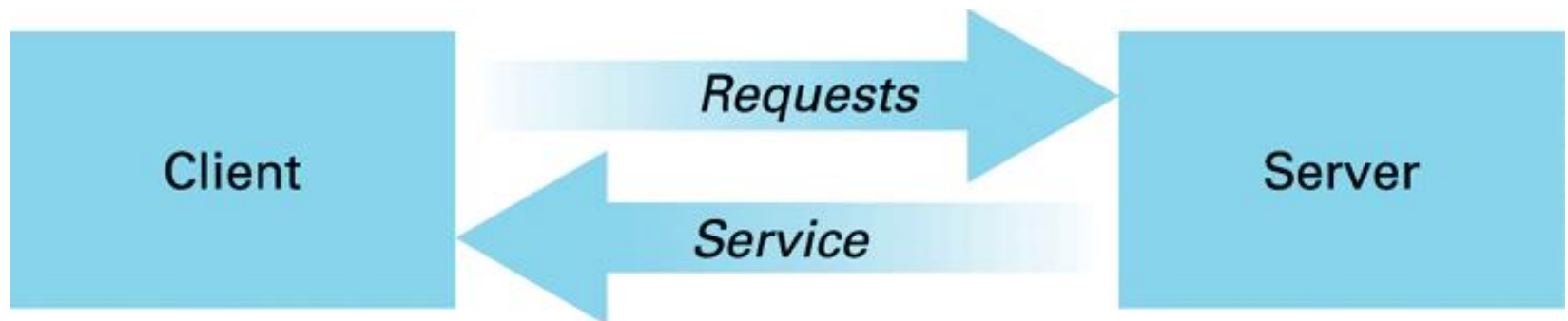
- *Coordination of Dispatcher and Scheduler : Adjusting to Priorities*
- *Example:*



- *Restore at the point of interrupt*
- *Immediate Environment prior to interrupt-> Process States*

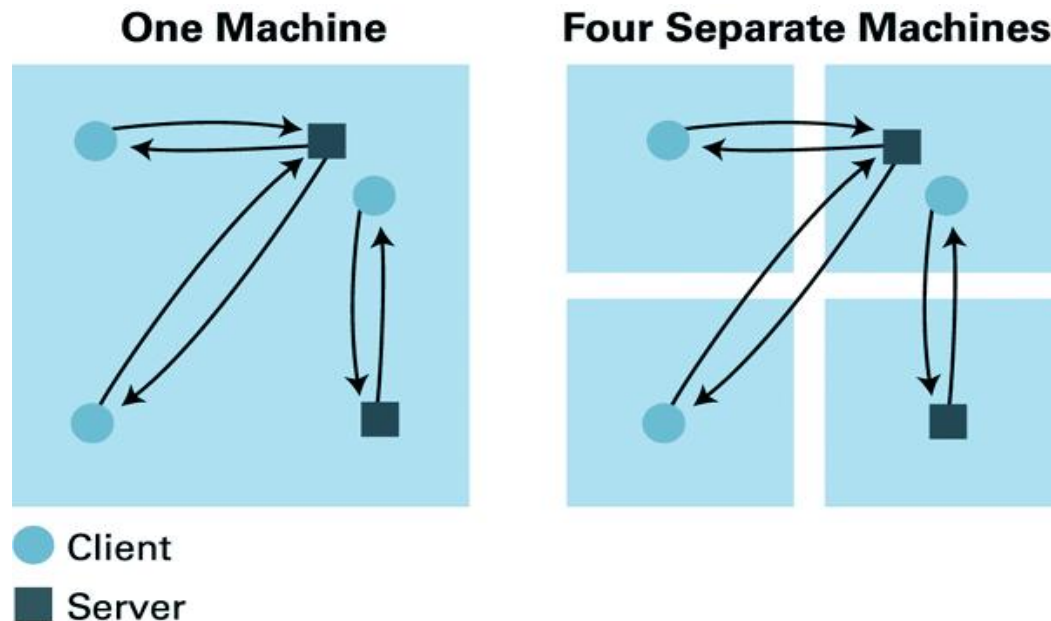
## 3.3: Interprocess Communication

- Various units within an OS also execute as processes
- To coordinate their activities, these processes must communicate with each other
  - Interprocess communication
- One form of Interprocess communication:
  - *client/server model* (also used in computer networks)
  - example: file-server providing access to files on request

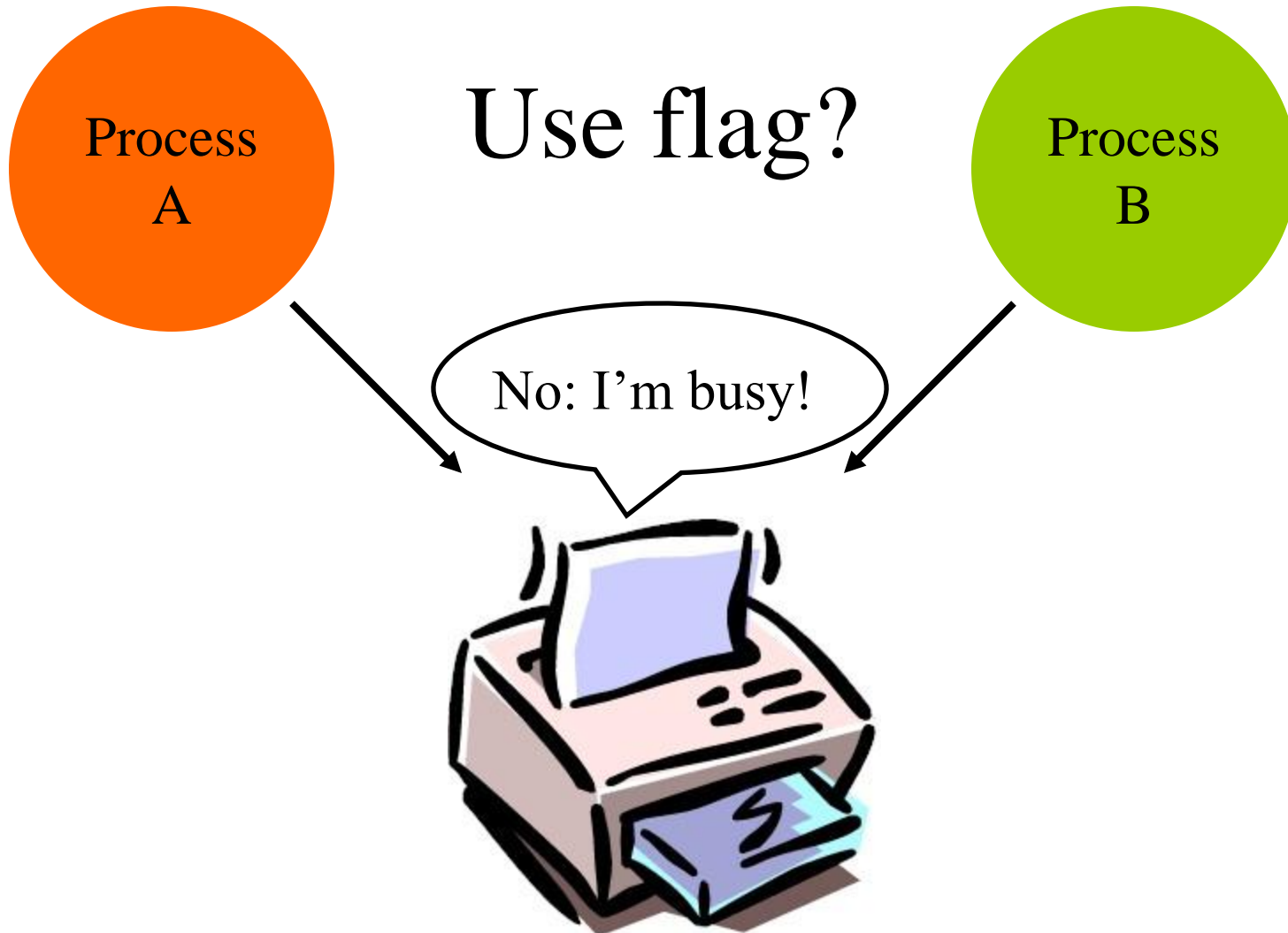


# 3.3: The Client/Server Model

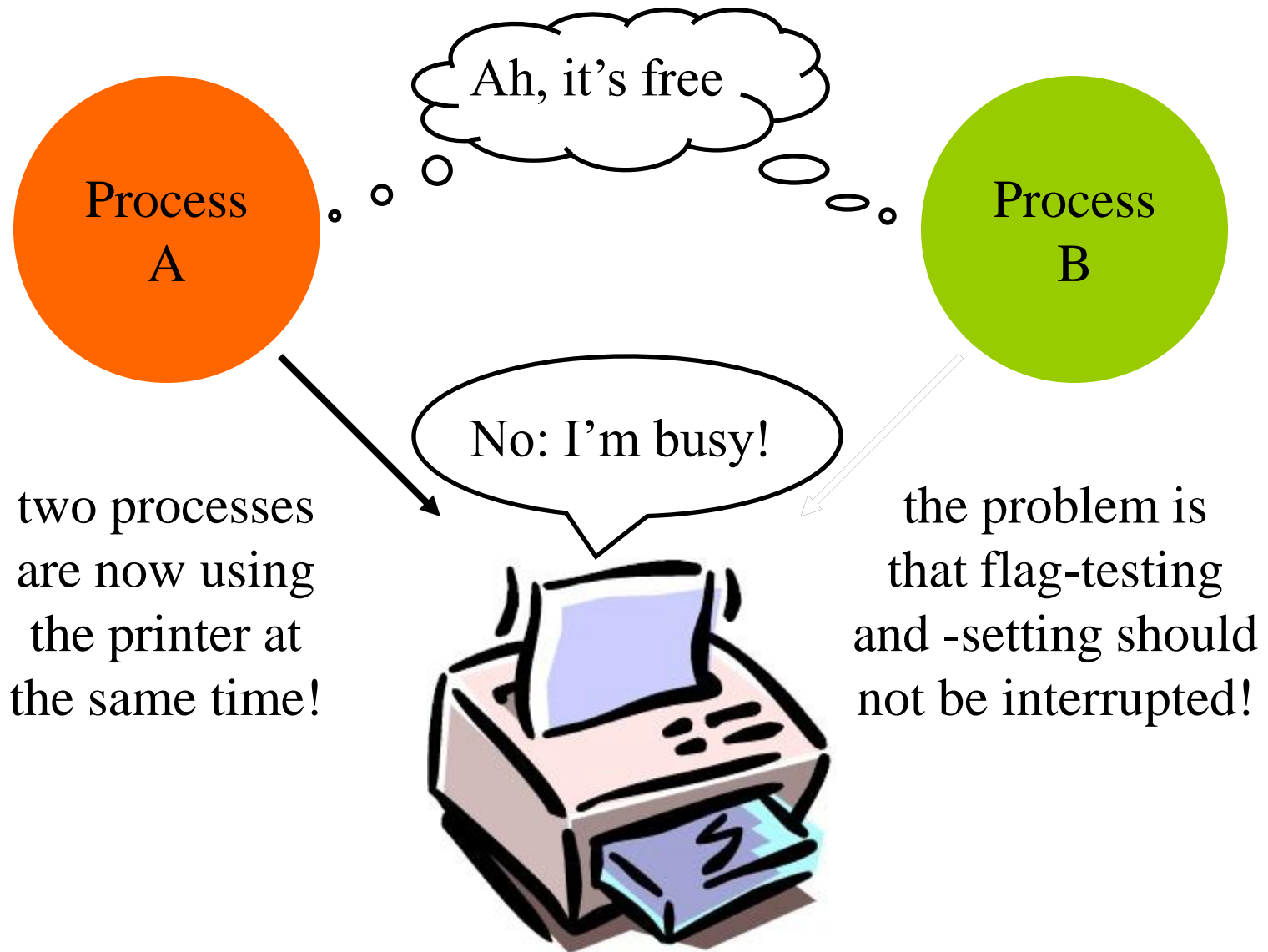
- Client and Servers are softwares within a machine or among different machines
- The role of the server is the same whether the client resides on the same machine or on a distant machine!
- Difference only in the communication software, not in the clients and servers
- CORBA(Common Object Request Broker Architecture): standard for network wide communication between softwares units known as objects (such as clients and servers)



# 3.4: Handling Competition Among Processes



## 3.4: Problems...!

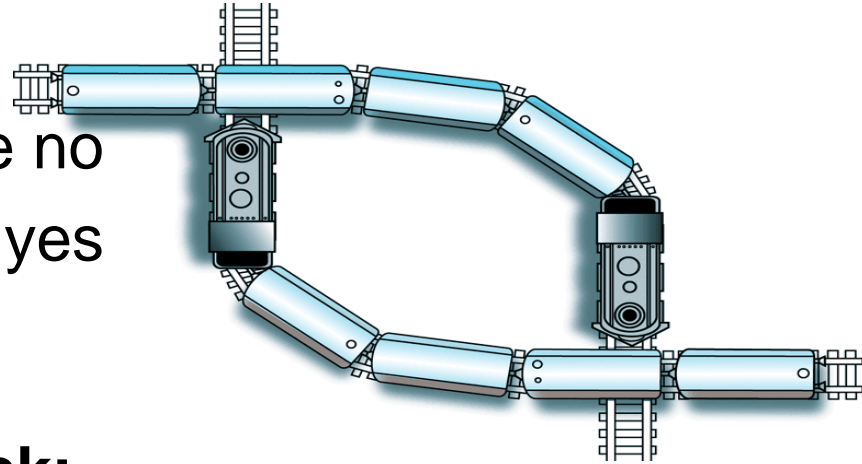


## 3.4: Solutions...

- One possibility is to use *interrupt enable* and *interrupt disable* instructions
  - disadvantage: process may remove the possibility of being interrupted altogether
- Other approach is to use single *test-and-set* instruction
  - always completed before an interrupt can be handled
  - flag implemented this way is a.k.a.: *semaphore*
  - Critical Region
    - A sequence of instructions executed by only one process at a time
    - Mutual Exclusion: A requirement

## 3.4: Another problem: Deadlock

- Two or more processes are blocked because each is waiting for access to resources allocated to another
  - task 1: printer yes, disk drive no
  - task 2: printer no, disk drive yes



### Three must conditions for deadlock:

- 1) There is competition for non shareable resources.
- 2) The resources are requested on partial basis; process returning back for more resources.
- 3) Once a resource has been allocated, it cannot be forcibly retrieved.

# Solutions:

- **Deadlock detection and correction**

- by forcibly retrieving some allocated resources
- Example: Process Kill

- **Deadlock avoidance**

- spooling (make the resource appear as if it can be shared by multiple processes at same time)
- Example: Printer Spooling

- **Other Problems**

- File access: read and write



# **Chapter 3 - Operating Systems: Conclusions**

- Operating System - 'glue' between hardware and applications
- Manages and controls multiple applications running at same time
- May also service multiple users at same time
- Multi-tasking / time-sharing based on processes
- Difficulties arise due to competition among multiple processes and deadlocks