OPERATING SYSTEMS

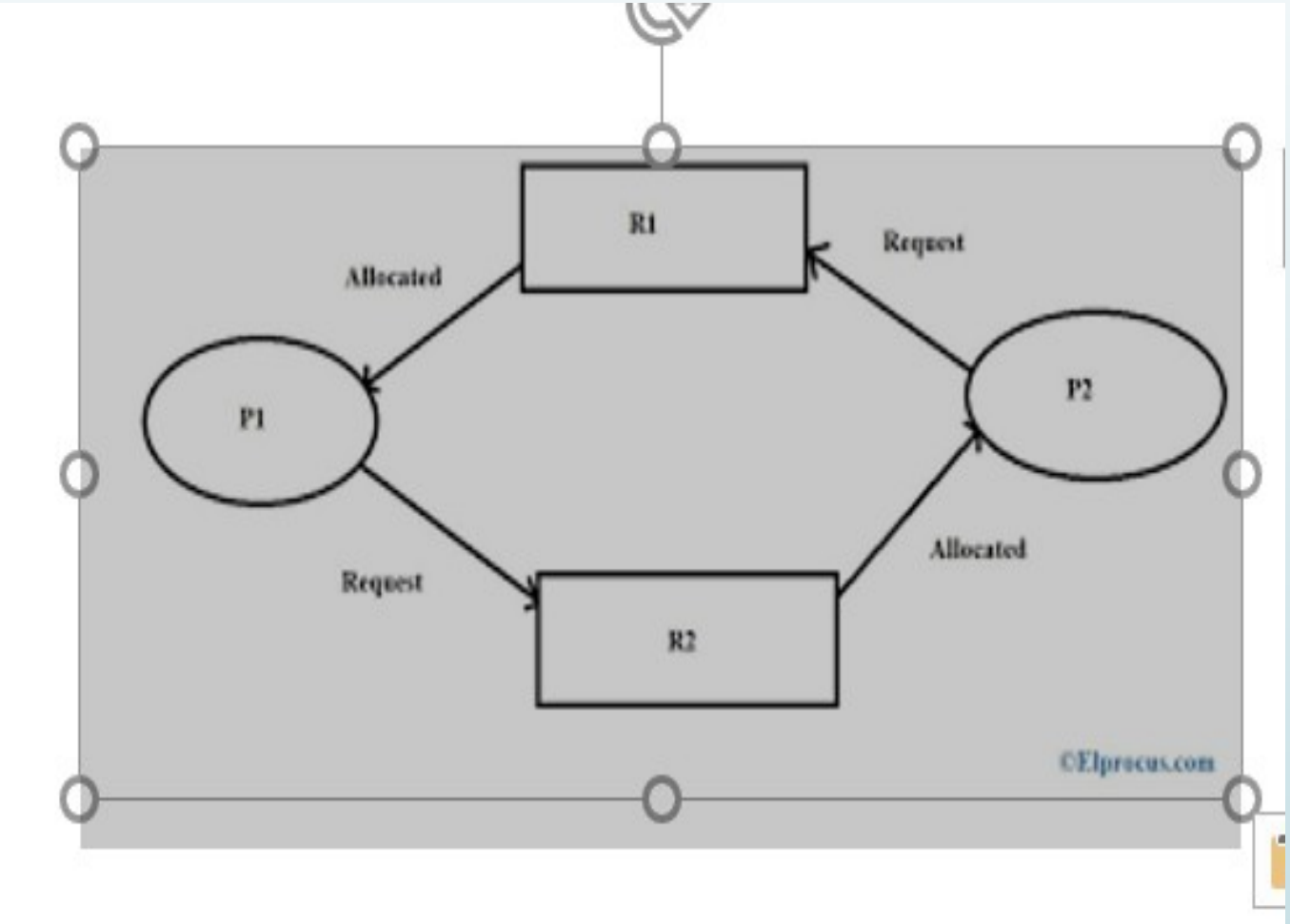
Module3_Part6

Textbook : Operating Systems Concepts by Silberschatz

Deadlocks

- For many applications a process needs exclusive access to not one resource, but several.
- For eg: There are two process each want to record a scanned document on a CD.
- Process A request permission to use the scanner and granted it. Process B programmed differently and requests CD recorder first and it also granted
- Now A asks for CD recorder, but request is denied until B releases it
- Instead of releasing the CD recorder B asks for scanner
- At this point both processes are blocked and remaining so forever.
- This situation is called **Deadlock**

Deadlock


R1
Request
Allocated
P1
P2
Request
Allocated
R2
©Elprocus.com

Deadlock

Deadlock definition

- A set of processes is in a deadlocked state when every process in the set is waiting for an event that can be caused only by another process in the set.

- The events with which we are mainly concerned here are resource acquisition and release.
- The resources may be either physical resources (for example, printers, tape drives, memory space, and CPU cycles) or logical resources (for example, files, semaphores, and monitors).

Resources

- A system consists of a finite number of resources to be distributed among a number of competing processes.
- The resources are partitioned into several types, each consisting of some number of identical instances.
- Memory space, CPU cycles, files, and I/0 devices (such as printers and DVD drives) are examples of resource types.
- A process must request a resource before using it and must release the resource after using it. A process may request as many resources as it requires to carry out its designated task.
- The number of resources requested may not exceed the total number of resources available in the system.
- In other words, a process cannot request three printers if the system has only two.

Resources

Process utilization of resource in following sequence only

- Request. The process requests the resource. If the request cannot be granted immediately (for example, if the resource is being used by another process), then the requesting process must wait until it can acquire the resource.
- Use. The process can operate on the resource (for example, if the resource is a printer, the process can print on the printer).
- Release. The process releases the resource.
- The request and release of resources are system calls,

Resources

- A system table records whether each resource is free or allocated;
- for each resource that is allocated, the table also records the process to which it is allocated.
- If a process requests a resource that is currently allocated to another process, it can be added to a queue of processes waiting for this resource.

Necessary conditions for a deadlock

features that characterize deadlocks.

- A deadlock situation can arise if the following four conditions hold simultaneously in a system:

1.Mutual exclusion. Each resource is exactly assigned to one process or is available. that is, only one process at a time can use the resource. If another process requests that resource, the requesting process must be delayed until the resource has been released.

2. Hold and wait. A process must be holding at least one resource and waiting to acquire additional resources that are currently being held by other processes.

ie a process holding a resource can request for some other resource without releasing it.

Necessary conditions for a deadlock

3. **No preemption.** Resources cannot be preempted; that is, a resource can be released only voluntarily by the process holding it, after that process has completed its task.

4. **Circular wait**. A set { *P0* , Pl, ... , Pn } of waiting processes must exist such that *Po* is waiting for a resource held by P1, P1 is waiting for a resource held by P2, ... , Pn-1 is

   waiting for a resource held by *Pn* and Pn is waiting for a resource held by *Po.*

   There should be a circular chain of processes and resources.

all four conditions must hold for a deadlock to occur.