ACTIVITIES UNIT 4

ACTIVITY 1.- Justify whether the following sentences are True or False.

	Coffman conditions...	**T/F**
1	... are 4: mutual exclusion, progress, bounded waiting and no-preemption.	
2	… they are only four and include “hold and wait” and “determinism”.	
3	... they are sufficient conditions; if they are all hold simultaneously it is sure that there is a deadlock.	
4	... they are necessary conditions. You just need to avoid one of them so a deadlock cannot be generated.	
5	… they are sufficient and necessary. If all of them are hold, then there is a deadlock. If there is a deadlock, then all of them are hold.	
6	... they are the base for designing deadlock prevention strategies.	
7	One of the conditions implies that "whenever a resource is assigned to one thread, other threads cannot use this resource".	
8	One of the conditions implies that "an assigned resource can only be released by its owner".	

ACTIVITY 2.- In a company dedicated to editing and layout of documents there are some printers, multifunction devices and plotters, with the following features:

• 2 mono laser printers, which handle A4 paper.

• 3 multifunction (printer + scanner) inkjet (black / colour) devices that handle A4 paper.

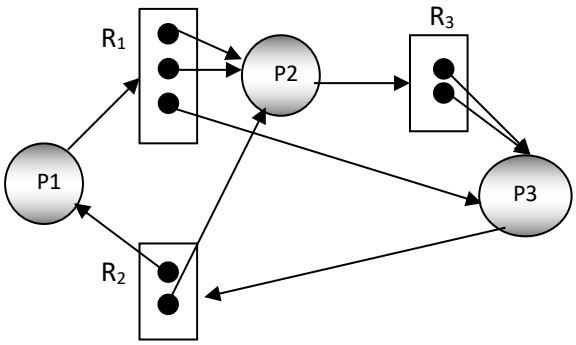
• 1 monochrome plotting device capable of using paper documents of A2 format. This device will not be used to print text documents and / or graphics that can be handled by the other devices.

Requests that applications can make are of the following types:

- Print documents in monochrome A4 paper format.
- Print colour documents in A4 format.
- Scan colour documents in A4 format.
- Plot plans / maps in A2.

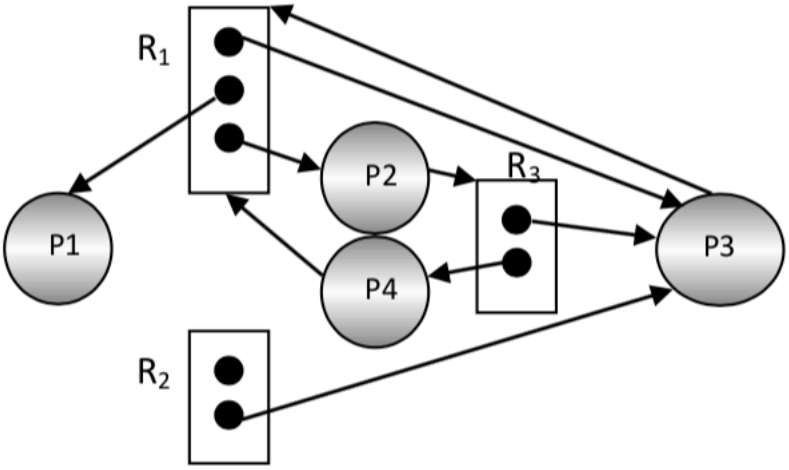
Discuss which resources will be considered and how many instances of each of them there will be when modelling this system using a RAG (Resource Allocation Graph).

ACTIVITY 3. Given the following state of a system and assuming that processes will not make any other request and that they do not release their resources until their completion:



a) Indicate whether there is a deadlock. If so, indicate which processes are in deadlock. If there is not any deadlock, give at least one safe sequence of termination.

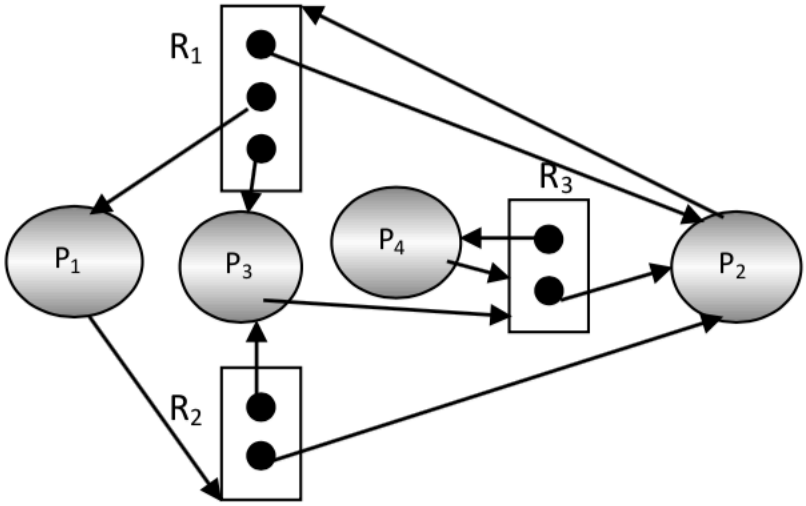
ACTIVITY 4.- Given the following status of a system and assuming that processes P1 and P4 still have one request each from an R2 instance (the order in which they make such requests is unknown) and that after this no process makes any more requests and does not release resources until its completion:



Please, indicate whether there is a deadlock. If so, tell which processes are deadlocked. If there is not any deadlock, then give a termination sequence (i.e. safe sequence of termination).

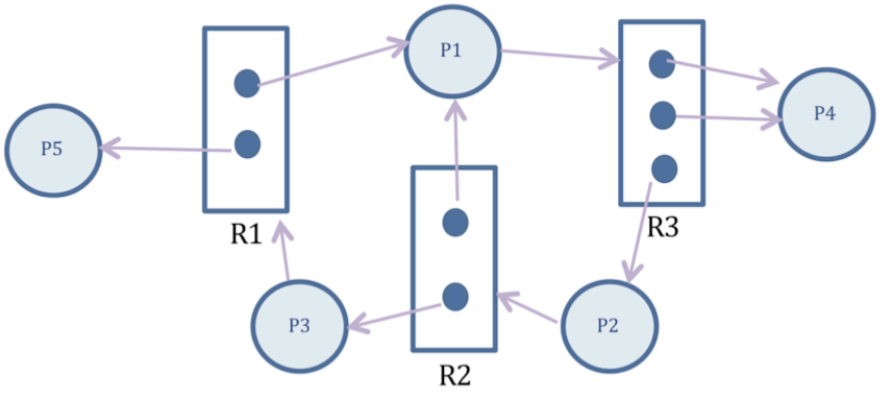
Note: The P4 process will make the request to the R2 instance once it has obtained the previously requested R1 instance (as shown in the RAG).

ACTIVITY 5.- Given the following state of a system and assuming that processes will not make any more requests during the rest of its execution and will not release resources until its completion:



Indicate whether there is a deadlock. If there is, say which processes are in the deadlock and if not, give a termination sequence.

ACTIVITY 6.- Given the following resource allocation graph, indicate whether the following sentences are true (T) or false (F).



1. There is more than one cycle.	
2. We can find a safe sequence.	
3. Process P1 is part of at least one deadlock.	
4. This is a wrong graph, because we have two double instance resources, but there is a resource with 3 instances.	
5. Process P4 cannot have two instances of the same resource assigned at the same time.	

ACTIVITY 7.- In a system there are 5 processes in execution: P0, P1, P2, P3 and P4 that make use of the following resources: R0, R1, R2, R3, R4 and RC. The number of instances of each resource is displayed in the following table:

Resource	R0	R1	R2	R3	R4	RC
Number of instances	1	1	1	1	1	5

The executing profile of a process Pi is different for even and odd processes and it is displayed in the following table:

Profile of even processes

```
while (TRUE) {
      Request(RC);
      Request(Ri);
      Request(R((i+1) % 5)));
      UseOfResources();
      Release(Ri);
      Release(R((i+1) % 5));
      Release(RC);
      RemainingSection();
};
```

Profile of odd processes

```
while (TRUE){
      Request(RC);
      Request(R((i+1) % 5)));
      Request(Ri);
      UseOfResources();
      Release(Ri);
      Release(R((i+1) % 5));
      Release(RC);
      RemainingSection();
};
```

Note: Each request of resources asks for a single instance of the resource and blocks the requesting process if the resource is not available.

a) Can there be a deadlock in this system? Justify your answer. If so, describe a possible scene, showing traces to justify it.
b) If RC=4, but all threads have the same profile (without distinguish between even/odd), can there be a deadlock?

ACTIVITY 8.- Indicate whether the following sentences are true (T) or false (F).

1. Deadlock situations can be avoided if the system monitors resource requests, denying those that create a risk of deadlock.	
2. Deadlock situations can be prevented by allocating resources so that a circular wait is never generated.	
3. In many operating systems, such as Unix or Windows, resource allocation graphs are used to avoid deadlocks.	
4. Coffman's conditions allow you to design systems that meet all of them, so you can be sure that no deadlock will occur.	
5. Coffman conditions are used to design deadlock detection algorithms.	

ACTIVITY 9.- Given the following monitor (that assumes Lampson/Redell variant):

```
monitor ResourceController {
   private int availableItems;
   private condition c;
   private int k, retrying,toBeNotified;

   public ResourceController(int N) {
      availableItems = N;
      k = retrying = toBeNotified = 0;
   }

   entry void request(int p) {
      if (p < availableItems) {
         availableItems -= p;
      } else {
         retrying++;
         retry(p);
      }
   }

   entry void release(int p) {
      availableItems += p;
      toBeNotified = retrying;
      c.notify();
   }

   private void retry(int p) {
      c.wait();
      toBeNotified--;
      if (toBeNotified>0)
         c.notify();
      k++;
      if (p < availableItems) {
         retrying--;
         availableItems -= p;
      } else {
         retry(p);
      }
   }
}
```

A) Indicate which will be the final value of attributes `k` and `availableItems`, assuming that in the constructor method we have employed N=15 and that threads H1, H2 and H3 do the following sequential invocations:

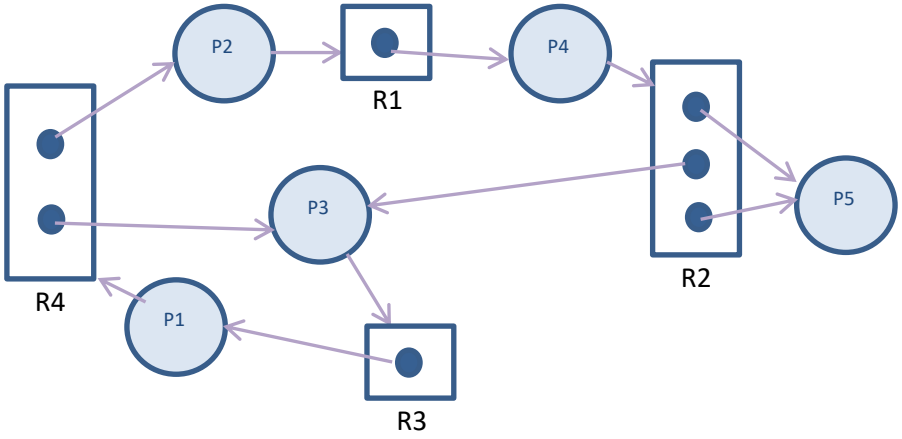
H1 invokes request(11)
H2 invokes request(10)
H3 invokes request(8)
H1 invokes release(3)
H1 invokes release(2)

B) Indicate whether the previous monitor for managing resources can provoke deadlocks. Take as a reference the following sequence of invocations (and an initial value of N=15 at its constructor):

H1 invokes request(8)
H2 invokes request(6)
H3 invokes request(8)
H1 invokes request(6)
H2 invokes request(4)

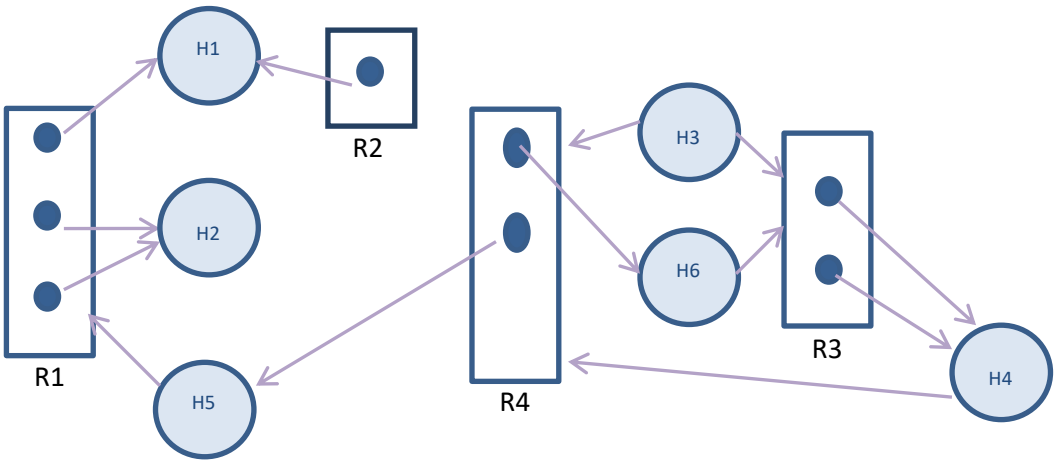
If there is a deadlock, please give the set of threads that are blocked. If not, justify your answer using Coffman's conditions.

ACTIVITY 10. Given the following resource allocation graph:



1. Since there is a directed cycle in which there are resources with a single instance, we can state that there is a deadlock.	
2. We can find the following safe sequence: < P5, P3, P1, P2, P4>.	
3. In this RAG, there is only one safe sequence.	
4. There is a deadlock between P1 and P3.	

ACTIVITY 11. Given the following resource allocation graph, indicate whether the following sentences are true (T) or false (F).



1. In this graph there is a directed cycle and since resource R2 has only one instance, we can state that there is a deadlock.	
2. In this graph there is a safe sequence given by: H1, H2, H5, H4, H3 and H6.	
3. In this graph there is a safe sequence given by: H2, H5, H4, H6, H3 and H1	
4. If H1 and H2 processes request each an instance of resource R4, there will be a deadlock.	

ACTIVITY 12.- The following describes a number of solutions to the deadlock problem. Indicate which technique is and sort all them from best to worst solution. If two solutions correspond to the same technique, assign them the same number of ranking.

SOLUTIONS	TECHNIQUE	RANKING
A RAG is used to determine whether to accept each request.		
If a deadlock situation is detected when monitoring the system, then some of the activities involved are aborted.		
Deadlock situations are not monitored nor simulated, even if they are possible.		
A total order among resources is established and these resources must be requested in this order.		
Using a RAG, if a new request provokes a risk of deadlock, then this request is denied.		
Given the design of the problem, it is impossible that a deadlock appears		
Processes request, at the beginning of their execution, all resources that they will need.		