Example OS Exam Question

1a A student studying anthropology and computer science has embarked on a research project to see if African baboons have an inherent understanding for synchronisation. He locates a deep canyon and fastens a rope going East-West across it as a bridge, so the baboons can cross hand-over-hand. Several baboons can cross at the same time, provided that they are all going in the same direction. If east-bound and west-bound baboons ever get onto the rope at the same time the baboons will get stuck in the middle. In order to simulate their intended behaviour, the student writes a simulation of Baboon Processes with synchronisation provided using semaphores to prevent baboons getting stuck on the rope. Assume the rope is strong enough to hold any number of baboons.

You will require 4 access procedures:

Earrive – East-bound baboon arriving to cross bridge
Edepart – East-bound baboon leaving bridge after crossing
Warrive – West-bound baboon arriving to cross bridge
Wdepart – West-bound baboon leaving bridge after crossing.

```
West-bound baboon:  Warrive ();          East-bound baboon: Earrive();
                      cross;                                  cross;
                    Wdepart ();                             Edepart();
```

i) Identify the *variables* and *semaphores* needed to control the baboons crossing the canyon and indicate their initial values.

ii) Give outlines of the above four access procedures indicating how they use the semaphores for synchronisation.

iii) Discuss the fairness of your solution.

b i) Give an example showing why *First Come First Served* is not an appropriate scheduling scheme for interactive users on a multi-access computer system.

ii) Using your example from i) above, explain why *Round-robin* is a better scheduling scheme for interactive users.

Part a i) is 10%, a ii) 60%, a iii) 10% and part b is 20% of the marks.

Example OS Exam Question Solution

1ai) Integer: E, W = 0 { Counts of east and west-bound baboons initially =0}
Semaphore: Emutex = 1
Wmutex =1
Bridgefree = 1

ii) East-bound

```
Procedure Earrive()
Begin
        down (Emutex)
        E := E+1
        if E = 1 then down (Bridgefree)
        //first east-bound baboon checks if bridge is free and if not, it  blocks.
         //Any following east-bound baboons will be held on Emutex
        up (Emutex)
end

Procedure Edepart()
        down (Emutex)
        E := E-1
        if E = 0 then up (Bridgefree)
        //If last east-bound baboon then indicate bridge is free
        up (Emutex)
end
```

West-bound

```
Procedure Warrive ()
Begin
        down (Wmutex)
        W := W+1
        if W = 1 then down (bridgefree)
        //first westbound baboon checks if bridge is free and if so blocks.
         //Any following west-bound baboons will be held on Wmutex
        up (Wmutex)
end

Procedure Wdepart()
Begin
        down (Wmutex)
        W := W-1
        if W = 0 then up (bridgefree)
        //If last west-bound baboon then indicate bridge is free
        down (Wmutex)
end
```

iii) This solution does not give fairness i.e. there can be starvation of either east-bound or west-bound due to a solid stream of baboons in one direction.

Example OS Exam Question Solution

1b i)

The assumption is that with FCFS, once a process is initiated it runs to completion. Such a scheme would prevent the system from guaranteeing good interactive response times. For example, suppose a large batch process enters a uniprocessor system. While that process executes, no other processes can execute. A user that attempts to load a web page or read an email must wait until the batch process completes before the system will respond to those requests.

ii)

Round-robin is a preemptive scheme that makes use of the interrupt clock. Long processes cannot delay shorter ones, because the shorter ones are assured of getting the processor periodically. Interactive users will thus receive the processor frequently enough to maintain good response times. In our example, the large batch job will be interrupted to service the processes that try to load a web page or read an email.