

**Issue date :** June 27 2017

**Due date :** July 11 2017

**Instructor :** Kerly Titus

---

**Question 1. Introduction.**

1. (i) Explain the reason why a system call must be implemented in the kernel of the operating system (i.e. in system mode).  
(ii) Some early CPUs (e.g. INTEL 8086) did not implement a mode bit so the operating system and the user applications execute in the same mode of operation. Explain, if possible, how the system resources could be protected in such architecture.
2. (i) Explain how a multitasking OS protects the CPU from faulty running programs?  
(ii) Explain a problem that could occur if that CPU protection mechanism is not implemented.
3. (i) Throughput is defined as the amount of work done per unit time. “The degree of multiprogramming in a computer system has direct influence on the throughput of the system”. Explain with an example.  
(ii) “Multitasking (i.e. time sharing) is a type of multiprogramming that is useful for interactive systems”. Explain.

**Question 2. Process management.**

4. (i) What is the main difference between user-level threads and kernel-supported threads?  
(ii) For a client application in a client-server system would you use user-level threads or kernel-supported threads? Explain.  
(iii) For a server application in a client-server system would you use user-level threads or kernel-supported threads? Explain.
5. (i) Explain each step that is performed by the dispatcher during a context switch between two processes.  
(ii) Explain what would be different (if any) if the context switch occurs between two threads of the same controlling process.
6. Suppose that an operating system distinguishes the time that a process or thread is running in user mode and the time that code is executed in system mode on behalf of that process or thread. Modify the following process state diagram to reflect the user and system running modes.

