1. (a) A process *p* is executing on a CPU. Explain all the possible state changes that can happen to *p* and the cause(s) behind each of them. [6]

(b) What is *context switching* with respect to scheduling? [2]

(c) Explain whether context switching happens when

(i) a process *blocks* while waiting to access a resource. [2]
(ii) a process *spinlocks* while waiting for a resource. [2]

(d) A variant of the producer-consumer problem is as follows: A producer process reads or writes to a database while a consumer process only reads from the database. Two or more consumers may access shared data while a producer needs to have exclusive access to the shared data. There are multiple producers and consumers in the system. All processes share the following data structure:

semaphore update;

All consumer processes share the following data structures:

semaphore mutex;
int num-of-consumers;

The semaphore *update* acts as a mutual-exclusion semaphore for producers, whereas the semaphore *mutex* is used to ensure mutual exclusion when the variable *num-of-consumers* is being updated. The variable *num-of-consumers* keeps track of the number of consumers currently reading the shared data. The initial values of the different data structures are: (1) *update* is set to 1, (2) *mutex* is set to 1 and (3) *num-of-consumers* is set to 0.

(i) Write pseudocode to illustrate the structure of the producer process [3]

(ii) Write pseudocode to illustrate the structure of the consumer process, making clear any assumption(s) you make. [7]

(iii) Explain a possible weakness of your solution. [3]