

Question 1

A linked list implementing a Queue ADT can be used to implement order of processes. A Dynamic Array can be used as well. And A Doubly Linked list implementing Queue ADT can be used as well.

Question 2

A Method to Queue a process for execution after it finished using its timeslot. Another Method to get the first process in the Queue waiting with all the processes with waiting state.

Question 3

Queue()

Dynamic Array $O(1)$

Linked List $O(n)$

Doubly Linked List $O(1)$

DeQueue()

Dynamic Array $O(n)$

Linked List $O(1)$

Doubly Linked List $O(1)$

Hence Doubly linked list is efficient to implement in theory.

Question 4

We can simulate it by having a loop for which 3 out of 4 cases are calls to handle the processes in the queue and 1 out of 4 is case is to function which generates new processes. We can do this by generating random number ranging from 1 to 4

Question 5

The correct way to simulate this is creating random ints in a random range signifying milliseconds representing time required by the process 25% of times and subtracting the predefined quantum from these integers and queueing them again 75% of the times.

Question 6

If Everything earlier mentioned is done correctly in the code, it has to be pretty accurate and not be missing anything.