Lab 05 Report: To Do List

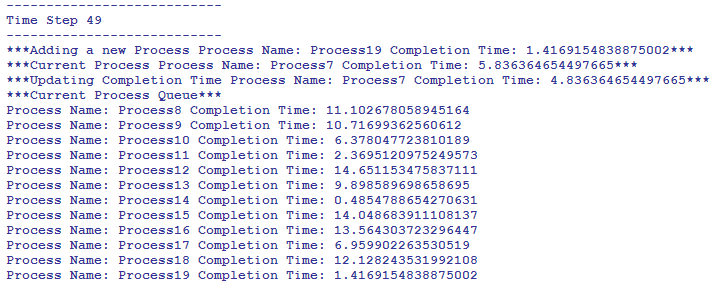
Problem

Write a program which simulates first come first serve scheduling user a queue, and make it work with a driver. This was done by creating a class Process, a class LinkedListQueue with an internal class ListNode, and a class called ProcessScheduler. All of these would work together to store the Processes in a linkedlist, and use a queue to manipulate there positions.

Proposed Solution

1. Create a class Process
   1. Attributes: String name, double completionTime
   2. Constructors: Default and Parameterized
   3. Accessors and Mutators for all attributes.
   4. Other methods: toString
2. Create a class LinkedListQueue
   1. Generic Type
   2. Internal class ListNode
      1. Attributes: data of type T, link of type ListNode
      2. Constructors: Default, parameterized
   3. Attributes: head of type ListNode which is first element of queue, tail of type ListNode that is last element of the queue
   4. Other methods: Enqueue which returns nothing and adds T data at end of queue, Dequeue which pops first element off and removes it, Peek which returns first element, and print which displays entire queue.
3. Create a class ProcessScheduler
   1. Attributes: Processes which is a linkedlist of processes, currentProcess which is currently running process
   2. Methods: getCurrentProcess which returns currentProcess, addProcess which adds at end of queue, runNextProcess which runs next in queue, cancelCurrentProcess which cancels current and replaces with head, printProcessQueue which prints all of the element from the process queue.

Tests and Results



Problems Encountered

The completion time by the driver kept making it currentCompletion time – 1, so in the setter I had to make sure if the value the driver sent was negative then make it 0, else just set it to what the driver sends in.

Conclusions and Discussion

This lab was very helpful in learning processes, and queues. The processes were stored in a queue in a linkedList and was manipulated by the use of queue’s methods with parameters changed in processScheduler.

Additional Questions

1. Describe how a queue is structured.

A queue is structured by being stored in a linked list which works like a linked list where the head points to the first element and the tail is the last element. The methods of queue such as enqueue to add a process in this case at the end of the queue and dequeue to remove the first element of the queue as long as it is not null. They are stored the way they are added to the queue, and must be dequeued to be removed or ran in this case of use of ProcessScheduler class.

1. What are the differences between a queue and a stack?

Stacks are last in first out, while queues are first in first out.