

Analysis

Data Structure

For this project I decided to use a Sorted Linked List. The sorted list I called "TaskList" was able to sort by Cycle and Priority. When the data is first read and stored into a "Task" object the objects are initially inserted into a "TaskList" and sorted by cycle, or integer of when they're supposed to be executed. This enabled the "CPU" object to sync removing tasks with with the count being incremented in the "CPU" class. In the "CPU" class I also created a second "TaskList" inserting the tasks I removed from the initial list. However this list is sorted by priority, meaning the task with the highest priority (lowest number) will always be executed before the tasks with less priority. Because I was using a *Sorted* Linked List the average case complexity were,

Insertion: **$O(n)$**

Deletion: **$O(1)$**

Challenges

The biggest challenge for me this project was creating a way to make sure all the tasks were being organized and prioritized properly to be executed. The Sorted Linked List ended up being a great way to make sure only the first task was being executed and if necessary replacing it with a higher priority task.