

# Requirements, Assumptions and justifications for the assumptions and/or Specifications



## Requirements

- The module should be able to simulate Multilevel Queue (MLQ) CPU Scheduling.
- The module should be able to allocate CPU time to processes based on their priority levels.
- The module should prevent CPU starvation for lower-priority processes due to continuously occupied higher-priority queues.

- The module should ensure fairness in the allocation of CPU time to processes.
- The module should be implemented using programming languages, data structures, and algorithms that are suited for this purpose.
- The module should include a user interface that enables users to interact with the simulation and view the results.



### Assumptions and Justifications

- Assumption: The module assumes that the priority of a process is fixed and does not change during its execution.
  - Justification: This assumption simplifies the implementation and avoids the complexity of handling priority changes dynamically.
- 
- Assumption: The module assumes that the arrival time of processes is known in advance.
  - Justification: Knowing the arrival time of processes allows for better scheduling decisions and ensures fairness in CPU allocation.
- 
- Assumption: The module assumes that the CPU time required by each process is known.
  - Justification: Having knowledge of CPU time requirements helps in making efficient scheduling decisions and avoids unnecessary context switches.

### Specifications

- The module should provide functions to add processes to the appropriate priority queues.
- The module should implement a scheduling algorithm that allocates CPU time based on priority.

- The module should include mechanisms to prevent CPU starvation, such as aging or time-based priority adjustments.
- The module should handle context switching efficiently to minimize overhead.
- The module should provide statistics and metrics to evaluate the performance of the MLQ CPU scheduling algorithm, such as average waiting time and turnaround time.
- The module should be implemented in a programming language that supports the required functionality and data structures for efficient simulation of MLQ CPU scheduling.

