Christopher Nguyen
033495382

Akeem Meade
033069931

Design:
Our implementation uses a linked list where each node contains a task with name, priority, and burst time. Each scheduler implements three functions: add() inserts tasks into the list using malloc() and strdup(), pickNextTask() selects which task runs next, and schedule() executes all tasks. For FCFS, add() appends tasks to the end by traversing to the last node, pickNextTask() returns the head task, and schedule() loops through executing each task for its full burst time before freeing memory. Priority scheduling searches the entire list in pickNextTask() to find the highest priority task. Round-Robin executes tasks for 10ms (QUANTUM), then re-queues incomplete tasks to the end. We added a tail pointer for FCFS and Round-Robin to make appending O(1) instead of traversing the list each time. Memory is managed with malloc() for allocation and free() for cleanup when tasks complete.

Individual Contributions:
Christopher
- Wrote primary functionality of code
- Debugged
- Wrote report

Akeem
- Debugged
- Wrote code
- Recorded code compiling

YouTube of code
https://youtu.be/7j0DpTA7808