Project 4 - CPU scheduler

CECS 326 - Operating Systems

Due Date:	April 26, 2023
Contributors:	@029136612 Twan Tran
	@025295541 Bharath Varma Kakarlapudi
Instructor:	Hailu Xu

1 Lab Summary

This project involves implementing several CPU scheduling algorithms, including first-come-first-served (FCFS), priority scheduling, and round-robin (RR) scheduling. The tasks to be scheduled are assigned priorities and CPU bursts, and the scheduler will schedule them based on the selected algorithm. The implementation will be completed in C and the instructions are provided in the zip file.

1.1 Objective

The goal of this lab is to implement CPU scheduling algorithms which are first-come-first-served (FCFS), priority scheduling, round-robin (RR) scheduling and to demonstrate how to they would run given tasks.

1.2 Design

The schedule of tasks has a specific format, and there are different strategies for organizing the list of tasks. This C implementation uses a linked list and a Makefile to determine the specific scheduling algorithm to be used.

2 Implementation

2.1 schedule-fcfs.c (Twan)

To implement the CPU scheduling algorithms for the given set of tasks, the first step is to read in the schedule of tasks from a file in the specified format ([task name] [priority] [CPU burst]). Once the schedule of tasks is read, create a list or queue to hold the tasks. To implement the first-come, first-served (FCFS) scheduling algorithm, sort the list of tasks in the order they were added. Then, execute each task in order until all tasks have been completed. Output the results of each scheduling algorithm, including the order in which the tasks were executed, the total time taken to execute all tasks, and any other relevant metrics.

2.2 schedule-priority.c (Bharath)

To implement the priority scheduling algorithm, the first step is to read in the schedule of tasks from a file in the specified format ([task name] [priority] [CPU burst]). Then, a list will be created to hold the tasks. This could be a list ordered by priority, or a separate queue for each unique priority. Next, the priority scheduling algorithm should be implemented by sorting the list of tasks by priority (higher number = higher priority) with the highest priority tasks at the beginning of the list. Finally, each task should be executed in order of priority until all tasks have been completed.Output the results of each scheduling algorithm, including the order in which the tasks were executed, the total time taken to execute all tasks, and any other relevant metrics.

2.3 schedule-rr.c (Twan)

To implement the round-robin (RR) scheduling algorithm, the first step is to read in the schedule of tasks from a file in the specified format ([task name] [priority] [CPU burst]). Next, create a list to hold the tasks. This could be an unordered list, a list ordered by priority, or a separate queue for each unique priority. Then, add each task to the end of the list for the RR algorithm. Set the time quantum to 10 milliseconds and execute each task for the time quantum. After the task has executed for the time quantum, add it back to the end of the list if it is not finished. Repeat these steps until all tasks have been completed.

3 Result

3.1 Demonstration (Twan)

Link: https://youtu.be/ZVcfKeshz-4

3.2 Output

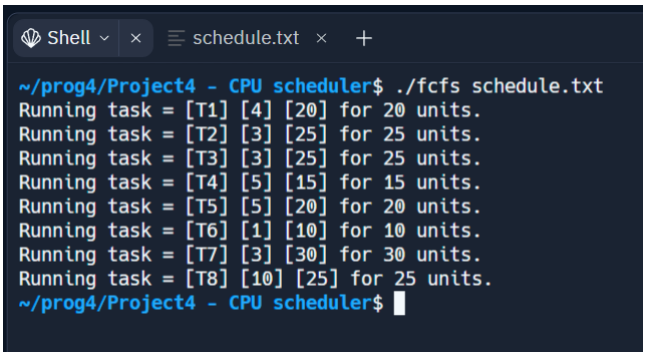

Figure 1: This is the output of the FCFS scheduling.

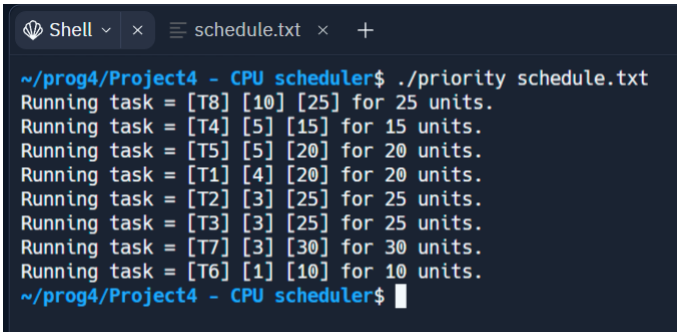

Figure 2: This is the output of the Priority scheduling.

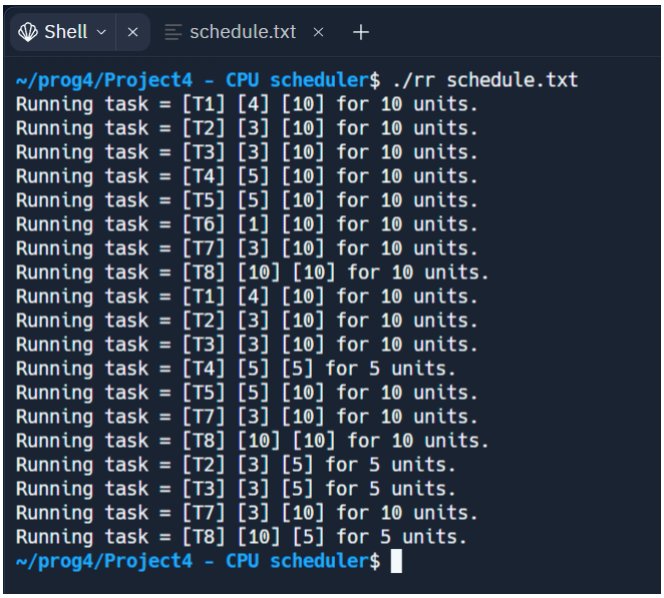

Figure 3: This is the output of the RR scheduling.