**PA2: CPU Scheduling Algorithms Documentation**

**Author: Darasimi Ogunbinu-Peters**

**Course: CPTS 360**

**Assignment: PA2 - CPU Scheduling Algorithms**

**1. Overview**

The goal of this assignment was to implement three CPU scheduling algorithms:

* **First-Come First-Serve (FCFS)**
* **Round Robin (RR)** with a quantum of 2 cycles
* **Shortest Job First (SJF)** (non-preemptive)

Each scheduling algorithm is intended to manage process execution, CPU bursts, I/O bursts, and produce summary metrics such as CPU utilization, I/O utilization, throughput, average turnaround time, and average waiting time.

**2. Approach**

**Data Structures**

Each process is represented by a structure containing:

* **A**: Arrival time of the process.
* **B**: Upper bound of CPU burst times.
* **C**: Total CPU time required.
* **M**: Multiplier for I/O burst calculation (I/O burst = M \* B).
* **ProcessID**: A unique identifier for each process.

**Scheduling Algorithms**

1. **First-Come First-Serve (FCFS)**:
   * Processes are executed in the order of arrival.
   * Each process runs to completion or I/O burst without interruption.
2. **Round Robin (RR)**:
   * Each process gets a fixed time quantum of 2 cycles. If the process does not complete within this quantum, it moves to the back of the queue.
   * The RR algorithm keeps track of remaining CPU bursts and rotates processes until all are complete.
3. **Shortest Job First (SJF)**:
   * This non-preemptive scheduling algorithm selects the process with the shortest remaining CPU time.
   * Tie-breaking rules apply: processes with the same CPU burst length are chosen by arrival time, and if these are the same, by their order of appearance in the input.

**Summary Data Calculation**

Each algorithm calculates:

* **Finishing Time**: Total time required for all processes to complete.
* **CPU Utilization**: Proportion of time the CPU was actively processing.
* **I/O Utilization**: Proportion of time spent on I/O operations.
* **Throughput**: Number of processes completed per hundred cycles.
* **Average Turnaround Time**: Average time from each process’s arrival to its completion.
* **Average Waiting Time**: Average time each process spends waiting before execution.

**3. Testing and Results**

The program was tested with three processes of varied arrival times and CPU burst lengths. Below are the command-line outputs for each scheduling algorithm, including detailed per-process data and summary metrics.

**Command Line Output:**

**First-Come First-Serve (FCFS)**:

First-Come First-Serve Scheduling

Summary for FCFS:

Process 1: Arrival=0, CPU=5, Finishing=5, Turnaround=5, Waiting=0

Process 2: Arrival=1, CPU=4, Finishing=9, Turnaround=8, Waiting=4

Process 3: Arrival=2, CPU=3, Finishing=12, Turnaround=10, Waiting=7

Total Time: 12

Average Turnaround Time: 7.67

Average Waiting Time: 3.67

**Round Robin (RR)**:

Round Robin Scheduling

Summary for RR:

Process 1: Arrival=0, CPU=5, Finishing=12, Turnaround=12, Waiting=7

Process 2: Arrival=1, CPU=4, Finishing=10, Turnaround=9, Waiting=5

Process 3: Arrival=2, CPU=3, Finishing=11, Turnaround=9, Waiting=6

Total Time: 12

Average Turnaround Time: 10.00

Average Waiting Time: 6.00

**Shortest Job First (SJF)**:

Shortest Job First Scheduling

Summary for SJF:

Process 1: Arrival=0, CPU=5, Finishing=5, Turnaround=5, Waiting=0

Process 2: Arrival=1, CPU=4, Finishing=12, Turnaround=11, Waiting=7

Process 3: Arrival=2, CPU=3, Finishing=8, Turnaround=6, Waiting=3

Total Time: 12

Average Turnaround Time: 7.33

Average Waiting Time: 3.33

**4. Results and Analysis**

The results confirmed the following:

* **FCFS**: Each process ran in the order of arrival, with no preemption. Processes waited for the previous ones to finish, as expected.
* **RR**: Each process received a 2-cycle time quantum. Processes were preempted if incomplete within this time and rotated back into the queue, resulting in longer average waiting times.
* **SJF**: Processes with shorter CPU bursts finished first. When bursts were equal, the process with the earlier arrival time ran, following the assignment’s tie-breaking rules.

**5. Challenges and Adjustments**

* **Implementing Accurate Tie-Breaking Rules for SJF**: Adjustments were made to ensure that SJF prioritized processes by CPU burst length, followed by arrival time.
* **Tracking Waiting and Turnaround Times in RR**: Implementing accurate tracking of waiting times, especially when processes were preempted, required additional adjustments to ensure correct metrics.
* **Memory Management in RR**: Adjustments were made to dynamically allocate memory for process tracking to meet compiler requirements.

**6. Conclusion**

This assignment implemented and tested three CPU scheduling algorithms—FCFS, RR, and SJF. Each algorithm demonstrated the expected behavior, with correct summary metrics displayed in the output. Adjustments were made based on grading feedback, ensuring that the program meets the assignment requirements.

A computer screen shot of a blue screen

Description automatically generated