OPERATING SYSTEMS
MINIPROJECT 3
2021114009

Priority Based Scheduler in xv-6.

The task described is to implement a modified priority-based scheduler in the xv-6 operating system, utilizing both static and dynamic priorities to manage process scheduling. Here's an analysis based on the formula and description provided:

Static Priority (SP)

- SP ranges between 0 to 100, with lower values indicating higher priority.
- Default SP is set to 50.
- This is a fixed priority assigned to a process that does not change during its execution unless explicitly modified by a system call.

Dynamic Priority (DP)

- DP is calculated based on the Static Priority and the Recent Behaviour Index (RBI).
- DP is constrained to be at most 100, ensuring that a process cannot exceed the maximum allowable priority.
- Dynamic Priority is used for scheduling decisions.

Recent Behaviour Index (RBI)

- RBI is designed to reflect the recent behavior of a process, with a default value of 25.
- It takes into account the running time (RTime), sleeping time (STime), and waiting time (WTime) since the process was last scheduled.
- The formula for RBI gives more weight to the running time and penalizes for waiting and sleeping, promoting processes that are ready to run but not over-prioritizing those that may be consuming too much CPU time.

Analysis

- The RBI formula ensures that processes that have recently been running are given a higher dynamic priority, potentially improving responsiveness for interactive processes.
- Sleeping or waiting processes have their dynamic priority reduced, which may help in avoiding the starvation of CPU-bound processes by I/O-bound processes.

- However, if a process has excessive waiting time, the negative impact on the RBI could lead to starvation, especially if it waits longer due to external factors beyond its control.

Concurrency:

Waiting Time: The waiting time for each customer is the duration between their arrival and the moment they start receiving their order. It does not include the coffee preparation time. To calculate the average waiting time, we sum up all individual waiting times and divide by the number of customers who received their coffee.

To determine how this compares to the scenario with infinite baristas, we can assume that in that case, each customer would start receiving their order after 1 sec as mentioned in the question after arrival (waiting time = 1). The difference would be the average waiting time calculated from our simulation.

Coffee Wastage: This is determined by the number of customers who leave without their order. Each of these instances represents one wasted coffee. We can count the number of occurrences where a customer leaves without their order to get the total coffee wastage.

2 2 3

Espresso 3

Cappuccino 10

1 Cappuccino 0 15

2 Espresso 3 6

3 Espresso 3 5

Customer 1 arrives at 0 second(s)

Customer 1 orders a Cappuccino

Barista 1 begins preparing the order of customer 1 at 1 second(s)

Customer 2 arrives at 3 second(s)

Customer 2 orders a Espresso

Customer 3 arrives at 3 second(s)

Customer 3 orders a Espresso

Barista 2 begins preparing the order of customer 2 at 4 second(s)

Barista 2 completes the order of customer 2 at 7 second(s)

Customer 2 leaves with their order at 7 second(s)

Barista 2 begins preparing the order of customer 3 at 8 second(s)

Customer 3 leaves without their order at 9 second(s)

Barista 1 completes the order of customer 1 at 11 second(s)

Customer 1 leaves with their order at 11 second(s)

Barista 2 completes the order of customer 3 at 11 second(s)

Avg waiting time : 2.333333

1 coffee wasted

Q2 Concurrency:

Q1: Minimizing Incomplete Orders

Incomplete orders are primarily caused by ingredient shortages or machines stopping work before an order can be completed. To minimize these:

- Dynamic Order Acceptance:
  Implement a system that checks the availability of ingredients and machine time before accepting an order. This system should instantly calculate if an order can be completed based on current stock and machine schedules.
- Reject orders immediately if they cannot be fulfilled, rather than accepting and then failing to complete them.
- Order Prioritization:
  Prioritize orders that can be completed quickly or with readily available ingredients. This could involve a scoring system where simpler orders or orders with abundant ingredients are moved to the front of the queue.
- Real-Time Ingredient Tracking:
  Keep a real-time count of ingredient stocks. As orders are placed, immediately deduct the required ingredients from the inventory.
  This approach ensures the parlor doesn't accept new orders that it cannot fulfill due to ingredient shortages.
- Machine Scheduling Optimization:
  Develop an algorithm to optimize machine schedules, ensuring that they are utilized efficiently and minimize downtime between orders.
  Consider the estimated completion time for each order and match it with the machine's available working time.
- Giving Priority to orders that are already accepted:
  Giving Priority to an already accepted order will reduce the cases of incomplete order when the shop closes.

Q2: Ingredient Replenishment

Assuming that ingredients can be replenished:

- Real-Time Inventory Monitoring:
  Implement a system that monitors ingredient levels in real-time and automatically alerts the management or places orders with suppliers when levels fall below a certain threshold.
- Dynamic Order Adjustment:
  If a popular ingredient is running low and a replenishment has been initiated, temporarily adjust the menu or suggest alternative flavors/toppings to customers.
- Inform customers about potential wait times due to ingredient replenishment and offer them the choice to wait or change their order.
- Supplier Integration:
  Integrate the inventory system with suppliers for faster and more efficient replenishment.

Automate the ordering process based on predicted demand and historical sales data to ensure a steady supply of ingredients.

Q3: Unserviced Orders

To avoid or minimize the number of unserved orders or customers waiting until the parlor closes:

- Advance Order Notifications:
  Inform customers about the expected wait time and the possibility of not being serviced due to machine unavailability.This allows customers to make an informed decision about whether to wait or visit later.

- Queue Management System:
  Implement a digital queuing system that provides customers with a real-time estimate of when their order will be ready.
  This system could also alert customers when their turn is approaching, reducing the time they spend physically waiting in the parlor.

- Machine Utilization Efficiency:
  Optimize the scheduling and use of ice cream machines to reduce idle times and increase throughput.
  Consider having overlapping shifts for machines or flexible working hours based on real-time demand.

- Customer Engagement Strategies:
  Engage waiting customers with interactive activities, tastings, or educational sessions about ice cream making.This approach not only enhances the customer experience but also makes the waiting time seem shorter.

By implementing these strategies, Sunny’s Ice Cream Parlor can improve its operational efficiency, minimize incomplete and unserviced orders, and enhance customer satisfaction.