Operating Systems

Project Report

Submitted By:

Ali Kamal

19I-1865

BSDS(N)

Ali Kamal
19I-1865

Overview

Basic crux of the project is to implement a somewhat advanced producer-consumer problem, with the help of multi-threading and semaphores.

Implementation details

I use multi-threading, with the help of semaphores, to implement the project. Individual threads of manager, customer, cook, and waiter are created, according to their counts. This all is synchronized by the help of semaphores.

In the main() function, customer thread is called ‘x’ times, with ‘x’ being the total number of customers. In each customer thread, the manager takes the customers orders, stores them in a FIFO queue, and passes the relevant order onto the cooks. The available cooks then take the order items and start preparing them. After all of the items in the order are prepared, a FIFO named pipe is created by the cook thread, where the customer’s id is written, which signifies that the customer’s order has been completed. The customer thread, meanwhile, waits for the FIFO named pipe to be created, where upon its creation the program moves ahead, and hands over control to the available waiter threads. The waiter thread then delivers the order to the customer. After the customer receives the order from the waiter, that particular customer thread exits.

Justification of project implementation

The project makes efficient use of available resources, and does not waste time in busy waiting, etc. Due to the fact that threading is used, instead of whole separate processes, the program remains relatively lightweight, and hence it is easier, and faster, to run. Similarly, pointers are used wherever possible, further making the program even more lightweight.