

ID1217 CONCURRENT PROGRAMMING

ASSIGNMENT -3

Semaphores

SUBMITTED BY-
KISHORE BAKTHA

PROBLEM 1-Hungry Birds Problem

In this problem, I first have two main classes- ParentBird and the BabyBird, both of them extending Thread class. Semaphores are used for synchronization. First the user enters the number of parent Birds and the number of Worms(W). Each baby bird starts consuming the worms and then goes to sleep for fixed amount of time. We then have a dish which is a shared resource. The Babybird can get from the dish and the ParentBird can put into the dish.

First each babybird tries to acquire the lock by using the acquire method of semaphore. Initially, the value is one ,so any bird can enter. Then no other babybird can enter until the lock is released. When the dish is empty, the babybird releases the lock of the parentBird. The parentbird puts fixed amount of worms and the process repeats again.

I think the solution is **Fair** because each bird sleeps the same amount of time before accessing the dish. Also the bird who finds the dish empty will be serviced first after the dish is refilled.

Output-

Enter number of baby birds

5

Enter W(number of worms)

5

Baby Bird- 0consumed-5

Baby Bird- 1consumed-4

Baby Bird- 2consumed-3

Baby Bird- 3consumed-2

Baby Bird- 4consumed-1

Baby Bird- 3wakes up Parent bird

Parent Bird-0 produced- 5

Baby Bird- 3consumed-5

Baby Bird- 1consumed-4

Baby Bird- 2consumed-3

Baby Bird- 0consumed-2

Baby Bird- 4consumed-1

Baby Bird- 1wakes up Parent bird

Parent Bird-0 produced- 5

Baby Bird- 1consumed-5

Baby Bird- 3consumed-4

Baby Bird- 2consumed-3

Baby Bird- 4consumed-2

Baby Bird- 0consumed-1

Baby Bird- 0wakes up Parent bird

Parent Bird-0 produced- 5

Baby Bird- 0consumed-5

Baby Bird- 1consumed-4

Baby Bird- 4consumed-3
Baby Bird- 2consumed-2
Baby Bird- 3consumed-1
Baby Bird- 0wakes up Parent bird
Parent Bird-0 produced- 5
Baby Bird- 0consumed-5
Baby Bird- 1consumed-4
Baby Bird- 4consumed-3
Baby Bird- 2consumed-2
Baby Bird- 3consumed-1
Baby Bird- 1wakes up Parent bird

PROBLEM 2- Bear and HoneyBees problem

In this problem, we have two classes- Bees to represent the bees and the Bear to represent the bear. Both of them extend Thread class. First the user enters the number of bees and the capacity of the pot. I have a count variable to monitor the capacity of the pot filled. Semaphores are used for synchronization in accessing the class Pot which represents the shared resource.

First the bees try to acquire the lock to put a certain amount of honey to the dish. Then the bee is made to sleep for a fixed amount of time so that the next bee can put honey. Once the pot is filled, the bee wakes up the beer who consumes the honey and goes back to sleep again until he is awoken again by the bee.

The solution is fair because each bee gets equal chance to put honey to the pot as all the bees sleep for fixed amount of time. Also, the bee who fills the dish wakes up the beer.

Output-

Enter the number of bees
5
Enter the capacity of pot(H)
5
Bee-0 contributes1
Bee-1 contributes2
Bee-2 contributes3
Bee-4 contributes4
Bee-3 contributes5
Bee-3 awakens beer
Bear consumes honey
Bee-3 contributes1
Bee-0 contributes2
Bee-1 contributes3
Bee-4 contributes4
Bee-2 contributes5
Bee-2 awakens beer

Bear consumes honey
Bee-2 contributes1
Bee-3 contributes2
Bee-1 contributes3
Bee-0 contributes4
Bee-2 contributes5
Bee-2 awakens beer
Bear consumes honey