

1 Hungry Birds

(one producer - multiple consumers) (20 points)

Given are n baby birds and one parent bird. The baby birds eat out of a common dish that initially contains W worms. Each baby bird repeatedly takes a worm, eats it, sleeps for a while, takes another worm, and so on. If the dish is empty, the baby bird who discovers the empty dish chirps real loud to awaken the parent bird. The parent bird flies off and gathers W more worms, puts them in the dish, and then waits for the dish to be empty again. This pattern repeats forever. Develop and implement a multithreaded program to simulate the actions of the birds. Represent the birds as concurrent threads (i.e. array of "babyBird" threads and a "parentBird" thread), and the dish as a critical shared resource that can be accessed by at most one bird at a time. Use only semaphores for synchronization. Your program should print a trace of interesting simulation events. Is your solution fair? Explain when presenting homework.

2 The Bear and Honeybees Problem

(multiple producers - one consumer) (20 points)

Given are n honeybees and a hungry bear. They share a pot of honey. The pot is initially empty; its capacity is H portions of honey. The bear sleeps until the pot is full, then eats all the honey and goes back to sleep. Each bee repeatedly gathers one portion of honey and puts it in the pot; the bee who fills the pot awakens the bear. Develop and implement a multithreaded program to simulate the actions of the bear and honeybees. Represent the bear and honeybees as concurrent threads (i.e. a "bear" thread and an array of "honeybee" threads), and the pot as a critical shared resource that can be accessed by at most one thread at a time. Use only semaphores for synchronization. Your program should print a trace of interesting simulation events. Is your solution fair (w.r.t. honeybees)? Explain when presenting homework.

3 The One-Lane Bridge Problem

(Similar to the Readers/Writers problem) (20 points + 20 points for extra work)

Cars coming from the north and the south arrive at a one-lane bridge. Cars heading in the same direction can cross the bridge at the same time, but cars heading in opposite direction cannot.

(a), For 20 points

Develop and implement a multithreaded program that simulates cars and the bridge using only semaphores for synchronization. Represent the cars as threads. Each car should be a thread that crosses the bridge trips times, first in one direction, then in the other direction, and so on. Assume that a car thread with an odd number first crosses the bridge in a northbound direction, and a car with an even number first crosses the bridge in a southbound direction. Have the cars sleep for a random amount of time between crossings of the bridge and have them sleep for a smaller random amount of time to simulate the time it takes to cross the bridge. Do not worry about fairness. Your program should print a trace of interesting simulation events.

(b), For extra 20 points

Modify your solution to ensure either of the following two options.

- Modify your solution to ensure fairness, i.e. to ensure that any car that is waiting to cross the bridge eventually gets to do so. You may want to solve the problem differently. See the Readers/Writers problem discussed in Lecture 10: Semaphores.
- Modify your solution to ensure that cars cannot bypass each other on the bridge, i.e. cars leave the bridge in the same order as they have entered the bridge.

To get 40 points, you may develop and implement only one program (b) that covers (a).