

Task

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. Represent the bear and honeybees as processes and develop code ***in the Await Language*** that simulates their actions. Use semaphores for synchronization.

sleeping



or



eating



N



H

“Concurrency perspective”

- When is the **mutual exclusion** (critical section) needed?
 - A bee is filling the pot or the bear is eating
- When is the **synchronization** needed?
 - Bees wait for earlier bee to fill the pot
 - Each bee may wait before filling the pot
 - Bees wake up the bear to eat
 - Last bee (the Hth) wakes up the bear after filling the pot
 - The Bear lets all bees to resume filling the pot
 - The Bear allows it after emptying the pot
- When is the **communication** needed?
 - Must know when the pot is full?
 - Number of portion in pot now?