

ESIR SPP – TD4

Exercise 1:

You are designing a data structure for efficient dictionary lookup in a multithreaded application. The design uses a hash table that consists of an array of pointers each corresponding to a hash bin. The array has 1001 elements, and a hash function takes an item to be searched and computes an entry between 0 and 1000. The pointer at the computed entry is either null, in which case the item is not found, or it points to a doubly linked list of items that you would search sequentially to see if any of them matches the item you are searching for. There are three functions defined on the hash table: Insertion (if an item is not there already), Lookup (to see if an item is there), and deletion (to remove an item from the table). Considering the need for synchronization, would you:

1. Use a lock over the entire table?
2. Use a lock over each hash bin?
3. Use a lock over each hash bin and a lock over each element in the doubly linked list?

Justify your answer.

Exercise 2:

You have been hired to coordinate people trying to cross a river. There is only a single boat, capable of holding at most three people. The boat will sink if more than three people board it at a time. Each person is modelled as a separate thread, executing the function below:

```
Person(int location)
// location is either 0 or 1;
// 0 = left bank, 1 = right bank of the river
{
    ArriveAtBoat(location);
    BoardBoatAndCrossRiver(location);
    GetOffOfBoat(location);
}
```

Synchronization is to be done using monitors and condition variables in the two procedures `ArriveAtBoat` and `GetOffOfBoat`. Provide the code for `ArriveAtBoat` and `GetOffOfBoat`. The `BoardBoatAndCrossRiver()` procedure is not of interest in this problem since it has no role in synchronization. `ArriveAtBoat` must not return until it safe for the person to cross the river in the given direction (it must guarantee that the boat will not sink, and that no one will step off the pier into the river when the boat is on the opposite bank). `GetOffOfBoat` is called to indicate that the caller has finished crossing the river; it can take steps to let other people cross the river

Exercise 3:

- 1) Propose an implementation for a synchronization barrier using a monitor.
- 2) Do the same using one lock and one semaphore. (Or one binary semaphore, and one normal semaphore.)