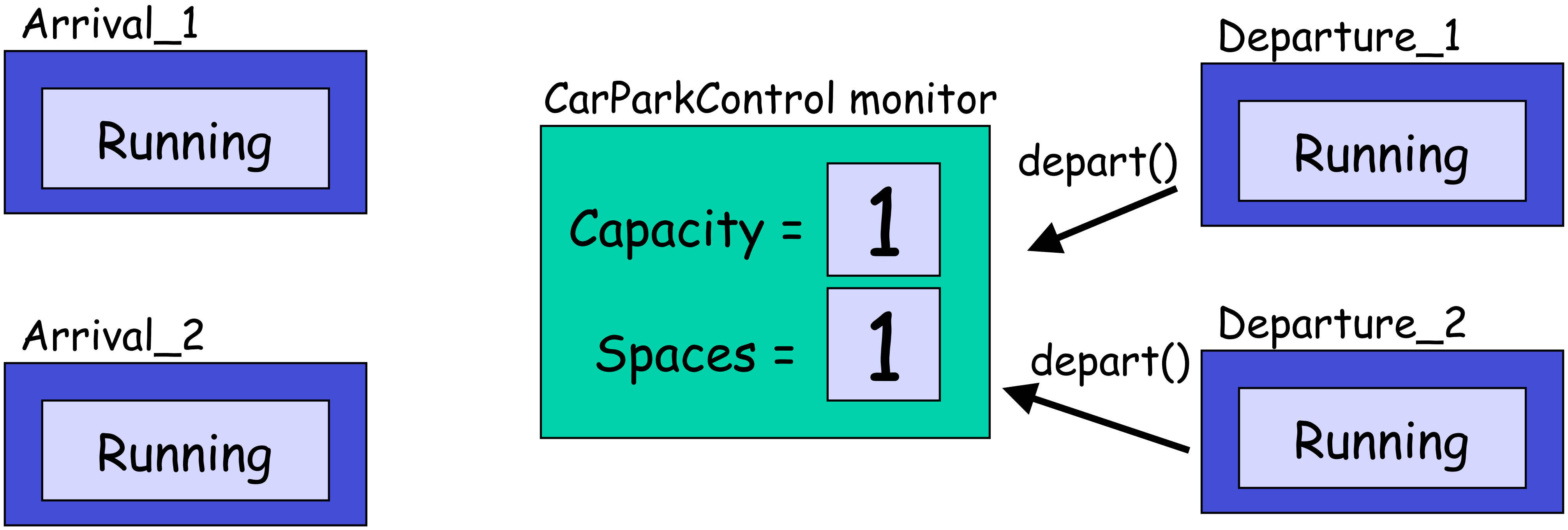
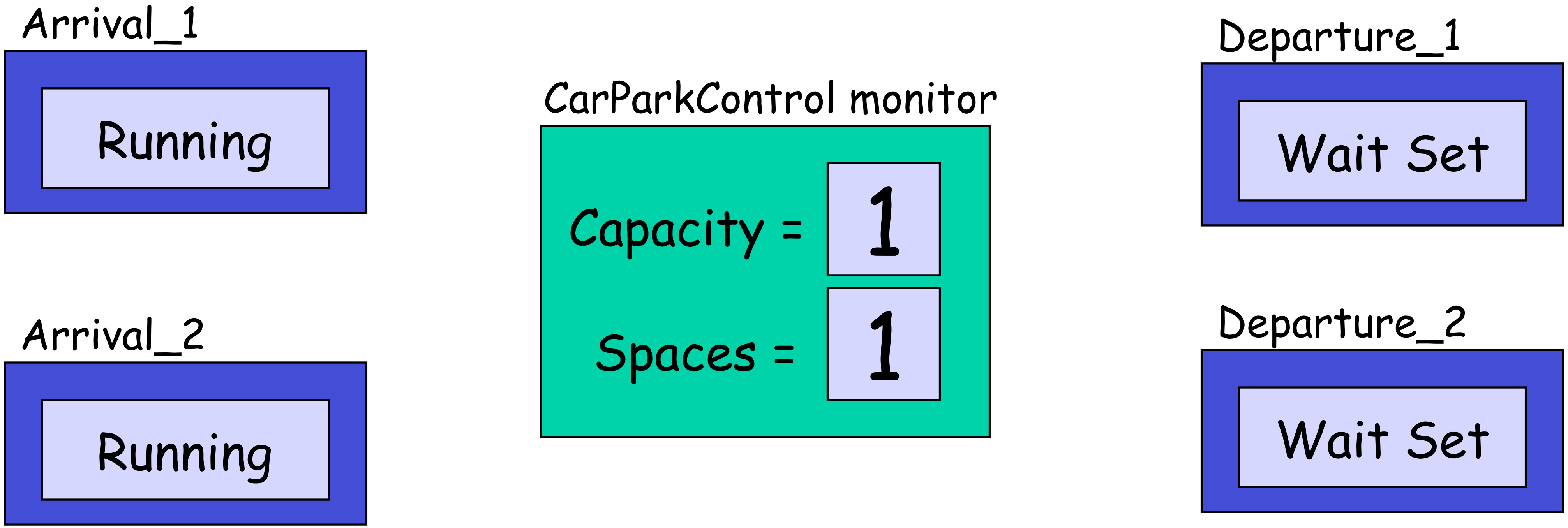
CarParkControl - condition synchronization

```
class CarParkControl {
  protected int spaces;
  protected int capacity;

  CarParkControl(int n)
    {capacity = spaces = n;}

  synchronized void arrive() throws InterruptedException {
    while (spaces==0) wait();
    --spaces;
    notifyAll();
  }

  synchronized void depart() throws InterruptedException {
    while (spaces==capacity) wait();
    ++spaces;
    notifyAll();
  }
}
```

Is it safe to use `notify()` here rather than `notifyAll()`?



The departure threads call depart...



Arrival_1

Running

Arrival_2

Running

CarParkControl monitor

Capacity = 1

Spaces = 1

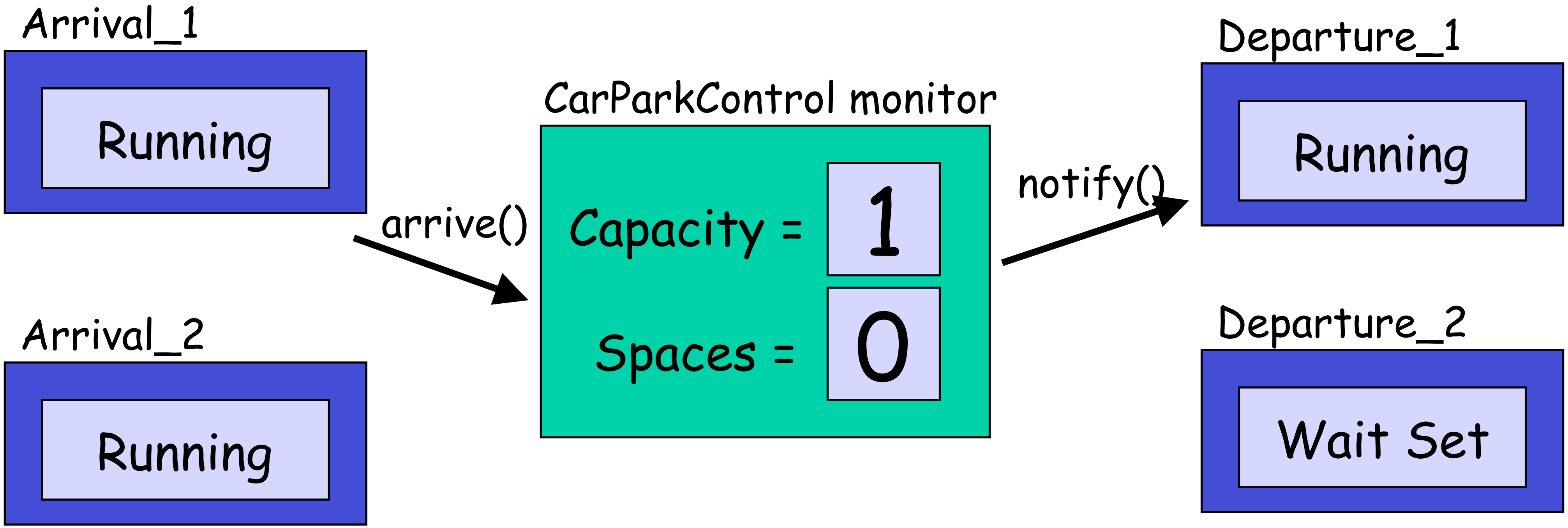
Departure_1

Wait Set

Departure_2

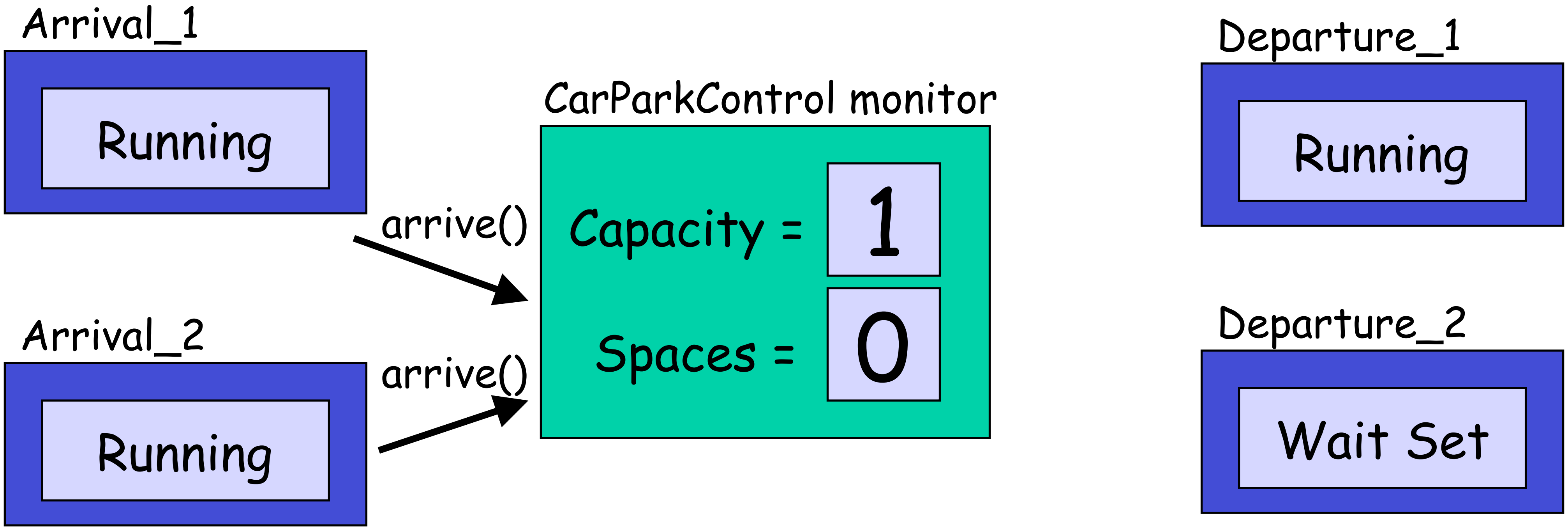
Wait Set

..and because Spaces == Capacity, they block.

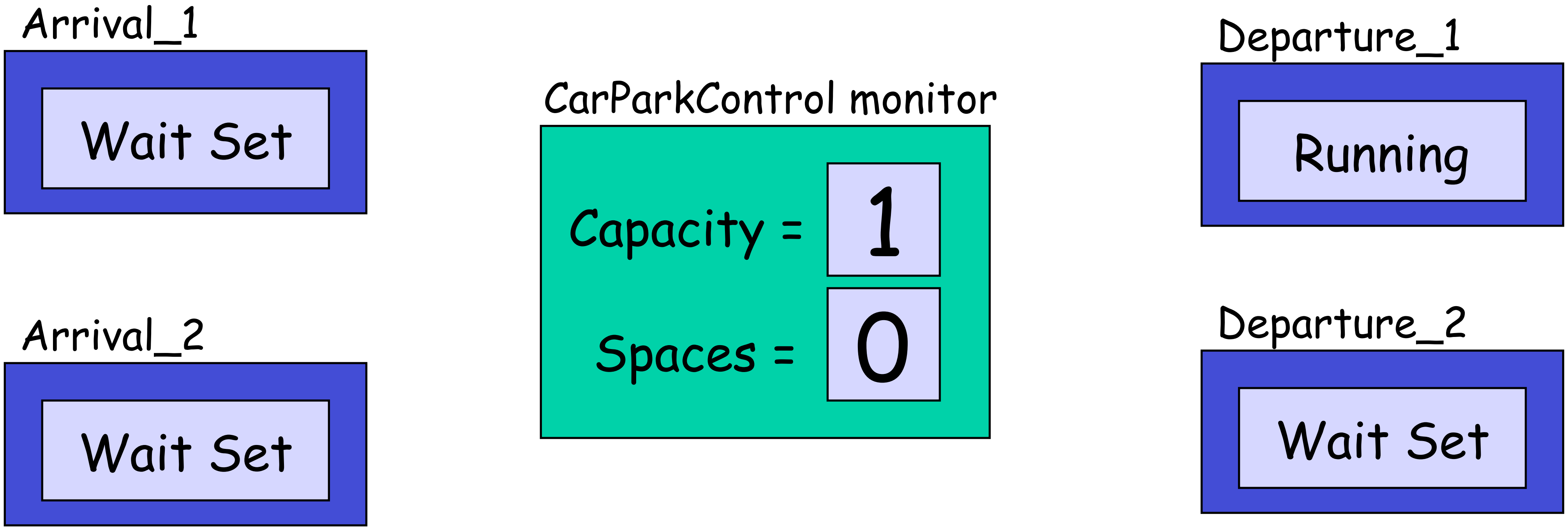


Arrival_1 calls arrive() which sets spaces to 0 and Departure_1 gets notified.

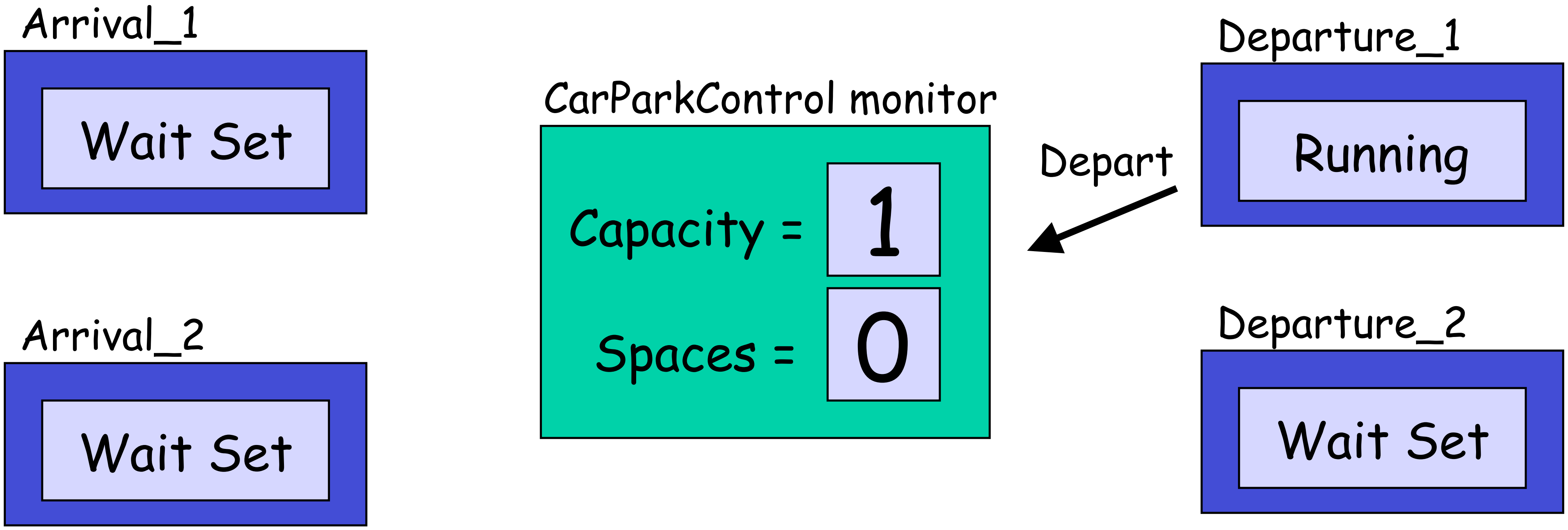
Before Departure_1 gets a chance to acquire the lock to increment spaces...



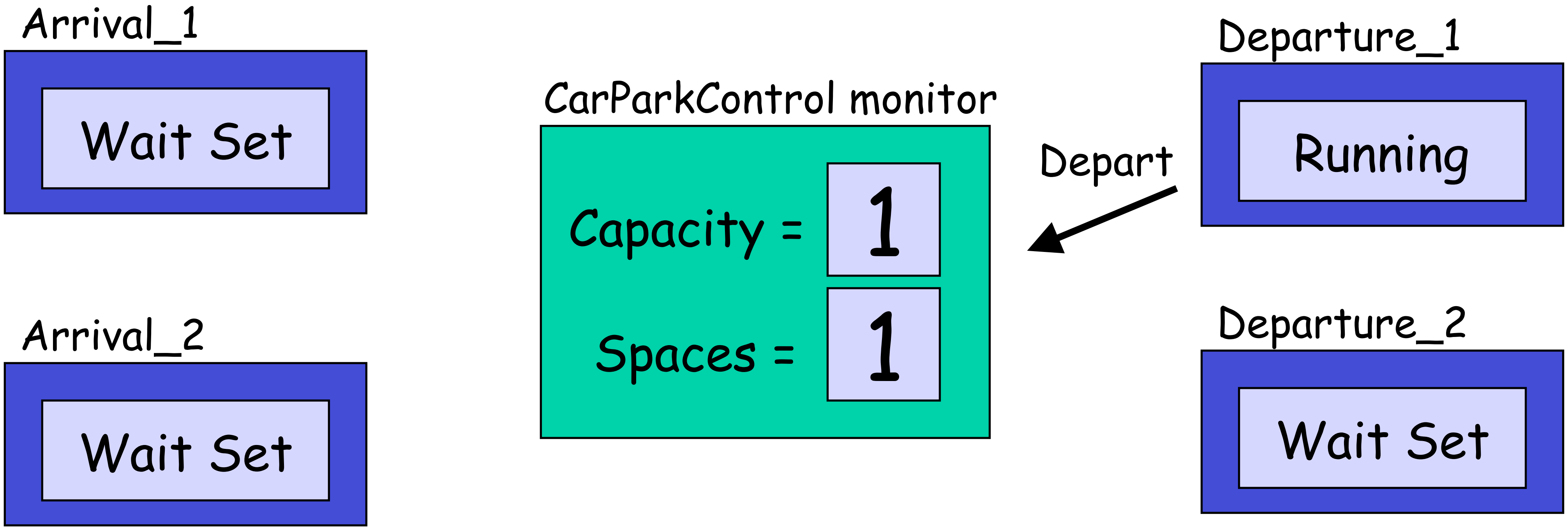
...both arrivals call arrive()...



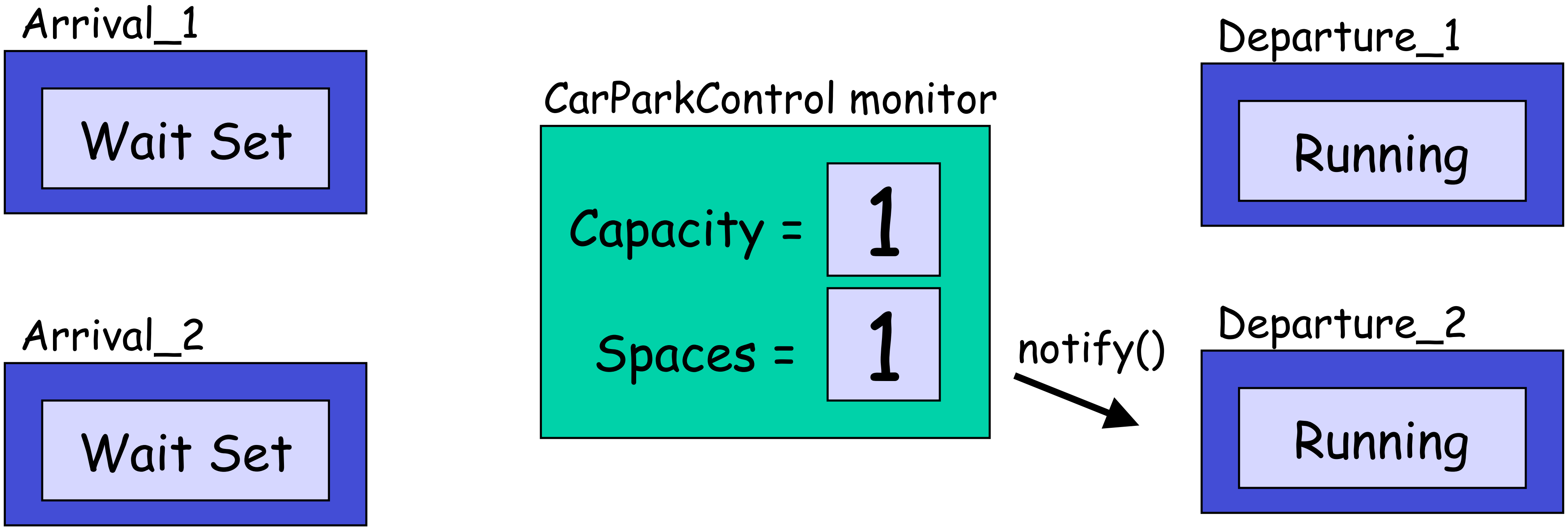
... and block because Spaces==0.



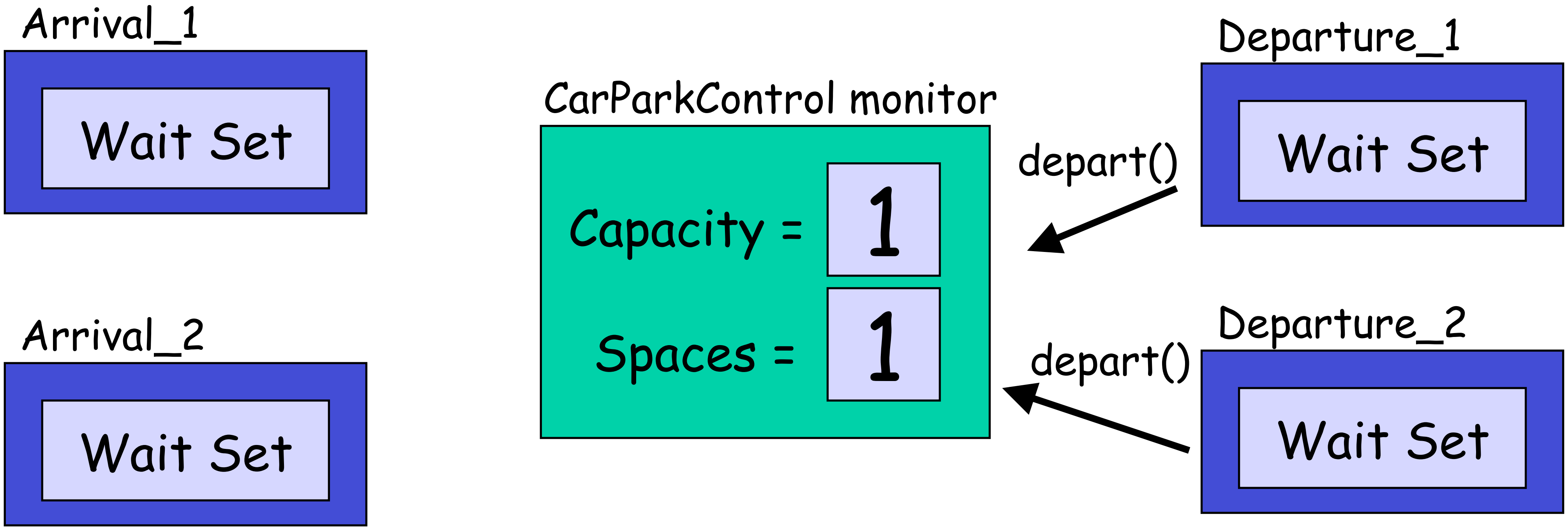
Now Departure_1 gets the lock on the monitor...



... increments spaces...



... and notifies Departure_2.



Now, both departures call depart() and block because Spaces==Capacity

DEADLOCK!